Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 162 (2025)

Succinct Zero-knowledge Proofs from One-way Functions: The Blackbox Way

Eden Florentz – Konopnicki* Ron D. Rothblum†

February 21, 2026

Abstract

Zero-knowledge proofs allow to encode a computation so that it can be verified without revealing any additional information beyond its correctness. In this work we focus on proofs that are *statistically sound* meaning that even an unbounded prover cannot make the verifier accept a false statement, except with negligible probability, and *computationally zero-knowledge*. The seminal result of Goldreich, Micali and Wigderson (CRYPTO 1986) shows that, assuming the existence of a one-way function, such zero-knowledge proofs exist for all languages in NP.

Some of the early protocols, such as that of GMW, have a large polynomial overhead in communication compared to the original NP witness. A line of works has shown that in many cases this communication overhead can be avoided. Most recently, Athamnah *et al.* (TCC 2024) constructed zero-knowledge proofs for all bounded-depth NP relations, where the communication complexity is only larger by an additive factor than the original NP witness. The main caveat of their result is that the protocol makes a non-blackbox use of the one-way function.

In this work we show that such succinct zero-knowledge proofs exist for the same class of NP relations, where the protocol makes only a *blackbox* use of a one-way function. Our protocol achieves a negligible soundness error, in contrast to recent works which can achieve, at best, an inverse polynomial error.

*Technion, Taub Faculty of Computer Science. Email: eden.konop@gmail.com
†Succinct. Email: rothblum@gmail.com

ISSN 1433-8092

Contents

1 Introduction

Zero-knowledge proofs, introduced in the pioneering work of Goldwasser, Micali, and Rackoff [GMR89], are a fundamental tool in cryptography. These proofs enable a prover to convince a verifier of the validity of a given computational statement without revealing any additional information. There exist several different notions of zero-knowledge proofs. In this work, we focus on constructing *computational zero-knowledge proofs* with *statistical soundness*, meaning that soundness holds against any *computationally unbounded* malicious prover, while the zero-knowledge property holds against a computationally bounded (potentially malicious) verifier.

In their seminal work, Goldreich, Micali, and Wigderson [GMW86] constructed a zero-knowledge proof for the graph 3-coloring problem. As this problem is NP complete, their result implied that *all* NP languages can be proved in zero-knowledge. Their result relies on the existence of a one-way function (an assumption that is also known to be necessary [OW93, HN24]).

The GMW protocol has a relatively large polynomial overhead in communication. This is due to a combination of two facts: first, the NP witness needs to be reduced to a 3-coloring of a related graph, and second, the 3-coloring protocol itself has (at least) a quadratic overhead. An exciting line of work [IKOS09, KR08, GKR15, GGI+15, NR22, HVW23, AFR24] has asked whether this overhead is inherent, or can a zero-knowledge proof be as short as the underlying NP witness?

The state-of-the-art is a recent result by Athamnah *et al.* [AFR24, Theorems 1 and 2] who constructed zero-knowledge proofs with only *additive communication overhead* (i.e., $(1 + o(1)) \cdot |w|$ bits) for any NP relation that can be verified in low-depth or low-space, while relying only on a one-way function. Here, and throughout this work we mean, by default that the soundness error is negligible.[1] Gentry *et al.* [GGI+15] construct such a protocol for *all* NP relations, but based on the existence of a fully homomorphic encryption (FHE) scheme — a significantly stronger assumption. As it is known (under standard complexity assumptions) that statistically sound zero-knowledge proofs are highly unlikely to be shorter than the witness size [GH98, GVW02], we refer to zero-knowledge proofs with such additive communication overhead as *succinct zero-knowledge proofs*.

The protocols underlying [AFR24, GGI+15] make a non blackbox use of the underlying cryptographic primitive (the one-way function in [AFR24] and the FHE in [GGI+15]). This means that the prover and verifier algorithms depend explicitly on the *code* of the underlying primitives, rather than merely invoking them as a blackbox sub-routine or oracle. Such a non blackbox use introduces significant overhead and is therefore highly undesirable in practice. Thus, a natural question posed by Hazay *et al.* [HVW23] is whether there exist succinct zero-knowledge proofs that make a *black-box* use of the underlying primitives.

Hazay *et al.* [HVW23] gave an initial positive answer to this question by constructing such succinct *blackbox* zero-knowledge proofs, but only achieve a constant soundness error. This falls short of the desired goal of negligible soundness error, which is crucial for most applications (and is often a highly non-trivial goal in settings in which repetition does not work (see, e.g., [BKP19] and references therein). Athamnah *et al.* [AFR24, Theorem 5] slightly improved their result and reduced the soundness error to 1/poly (for the case of low-depth NP relations, and assuming only a one-way function). Still, these results fall short of the desired requirement of a negligible soundness error. We continue the line of work of [HVW23, AFR24] and ask:

Do there exist succinct zero-knowledge proofs for NP (with a negligible soundness

[1] Note that in the setting of short proofs, standard soundness amplification by repetition is too expensive as it increases the proof-size multiplicatively.

error) that only make a black-box use of a one-way function?

1.1 Results

Our main result is a positive answer to this question for the case of NP relations that are verifiable by bounded-depth circuits.

Theorem 1.1. *Assume that one-way functions exist and let $\delta > 0$ be a parameter. 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 that $n \leq \mathrm{poly}(m)$.*

Then R has a zero-knowledge proof with perfect completeness, and soundness error δ in which the verifier, prover and simulator all only make a black-box use of the one-way function. The communication complexity is

$$m + m^{2/3} \cdot \mathrm{poly}\big(D, \log(S), \log(m), \lambda, \log(1/\delta)\big)$$

where λ is the security parameter. The prover and verifier run in polynomial time, the protocol is private-coin and has $\mathrm{poly}(D, \log(S), \log(1/\delta), \lambda)$ *rounds.*

Theorem 1.1 also implies a zero-knowledge proof for *general* NP relations with communication that is only additively larger than the *size* of the verification circuit and only make a black-box use of the one-way function. This follows from the fact that any NP relation can be converted into a new logarithmic-depth relation in which the witness includes the values assigned to all gates in the evaluation of the verification circuit (via the Cook–Levin theorem). Hence, the witness size is proportional to the circuit size.

Corollary 1.2. *Assume the existence of one-way functions. Let R be an* NP *relation that can be verified by a circuit of size S using unbounded fan-in* AND, OR, *and* XOR *gates. Then R has a zero-knowledge proof with soundness error δ, in which the verifier, prover and simulator all only make a black-box use of the one-way function. The communication complexity is*

$$S + S^{2/3} \cdot \mathrm{poly}\big(\log(S), \lambda, \log(1/\delta)\big)$$

where λ is the security parameter. The prover and verifier run in polynomial time, the protocol is private-coin and has $\mathrm{poly}(\log(S), \log(1/\delta), \lambda)$ *rounds.*

Remark 1.3. *As in prior work constructing zero-knowledge protocols with black-box use of the underlying primitive and negligible soundness [IKOS09], the protocol is of the private-coin type. Constructing a protocol with these properties in the public-coin model is an interesting open problem.*

Remark 1.4. *Our protocol has round complexity of* $\mathrm{poly}(D, \log S, \log(1/\delta), \lambda)$*, in contrast to the prior work of [HVW23] that has a constant number of rounds, but only has a constant soundness error.*

The $\mathrm{poly}(D, \log S)$ *round complexity in our construction arises from the reduction to the GKR interactive protocol [GKR15]. This could potentially be reduced to a constant number of rounds by using a constant-round protocol, such as [RRR21] (for bounded-depth NP relations).*

The $\mathrm{poly}(\lambda)$ *factor come from the use of a statistically hiding commitment scheme. Recall that such schemes, that are constructed from one-way functions use* $\mathrm{poly}(\lambda)$ *rounds [HR07] (and this seems inherent [HHRS21]). However, under the stronger assumption of a* non-interactive *statistically hiding commitment scheme (which can be obtained, for example, from collision-resistant hash functions), the overhead is reduced to* $\mathrm{poly}(\log(1/\delta))$ *rounds.*

A key component in our protocol is a new construction of a *statistically binding polynomial commitment scheme (PCS)* (see Definition 3.1). Such a scheme allows a sender to commit to a large polynomial P so that later it can prove correctness of evaluations queries of the form "$P(x) = y$". The computationally binding analog of this notion is central in the construction of succinct argument-systems, but interestingly the statistical counterpart has not been previously studied. Our main technical contribution is a succinct PCS for multilinear polynomials, over any sufficiently large field, in which the communication is only additively larger than the description size of the polynomial. While multilinear polynomials are typically defined over inputs that are a power of two, to allow encoding witnesses of arbitrary length, we extend the definition by interpreting the string as a multilinear polynomial after padding with zeroes up to the next power of two (although importantly, the communication should be proportional to the unpadded length).

Theorem 1.5. *Assume the existence of a one-way function. Then, for every $\delta > 0$ there exists a private coin polynomial commitment scheme for strings of length m over a finite field $\mathbb{F}$ s.t $|\mathbb{F}| = \tilde{\Omega}(m^{2/3}/\delta)$, with binding error δ and communication complexity*

$$m + m^{2/3} \cdot \mathrm{poly}(\lambda, \log(1/\delta))$$

field elements for the commitment and

$$k \cdot m^{2/3} \cdot \mathrm{poly}(\lambda, \log(1/\delta))$$

field elements for the opening, where k is the number of points to be opened and λ is a computational security parameter.

1.2 Techniques

Recall that our goal is to construct a zero-knowledge proof with two key properties: first, it should be succinct (meaning that the communication is only additively longer than the witness) and second, all algorithms should use the underlying one-way function as a blackbox. Following [IKOS09, HVW23, AFR24] we will do so in two steps. First, we construct a *distributed zero-knowledge protocol* in which there is a single prover and a set of verifiers who can communicate both with the prover and among themselves. For the *zero-knowledge* property, we require that any subset of t of the verifiers learn nothing beyond the validity of the statement (i.e., their views can be simulated). We then show how to compile this distributed protocol into a *monolithic* (i.e., one with a single verifier) by having the prover commit to the communication transcript and allow the verifier to open a subset of the views.

GKR-based Distributed Zero-Knowledge Protocol. For the distributed-verifier zero-knowledge protocol, we rely on the doubly-efficient GKR protocol for bounded-depth computations [GKR15] to prove that $w \in C_x$, where $C_x = C(x, w)$ is the circuit C with x hardcoded.

Recall that the GKR protocol is an interactive proof that processes the circuit layer-by-layer, using the sumcheck protocol to transition from layer to layer. We make the protocol distributed between the k verifiers by having the prover, in each step of the protocol, send an additive secret sharing of the relevant GKR messages to the verifier. At the end of the protocol the k-verifiers run an off-the-shelf MPC protocol to check that the GKR verifier would have accepted.

There is one hiccup with the above strategy. At the end of the protocol the GKR verifier needs to access the witness. Specifically, it needs to compute the multilinear extension of the witness at

a given point. Unfortunately, this computation requires an arithmetic circuit of size $O(2^d)$, and incorporating it into an generically into an MPC protocol would incur an overhead of $\Omega(w)$, which we cannot afford.

Instead, intuitively, we would like for the prover to first commit to the witness w (before the GKR interaction starts) and then, at the end, to provide a "zero-knowledge" evaluation proof. This is exactly what a *polynomial commitment scheme* (PCS) enables. Indeed PCS's have emerged as a fundamental component in the construction of *computationally-sound* proof-systems. Unfortunately, all existing constructions of PCS's are only *computationally binding* — in particular, the computationally unbounded cheating prover that we consider could easily violate the binding property of such a commitment. Somewhat surprisingly, the notion of a statistical binding PCS has not been considered in the literature.

Thus, we introduce and construct a statistically binding multilinear PCS. With this primitive in hand, our protocol is as follows:

1. The prover commits to the witness using a succinct PCS.
2. The prover and k-verifiers run the GKR protocol. The prover messages are secret-shared between the k verifiers.
3. The verifiers run an MPC protocol to check that the interactive part of the GKR protocol is valid (this part does not rely on the witness).
4. The prover gives an evaluation proof for the multilinear evaluation of the witness needed to conclude GKR.

As our main technical contribution we construct a *succinct PCS*. Namely, a statisically binding PCS in which the commitment and evaluation proofs are only larger than the witness by a small additive factor. We describe the construction in Section 1.2.1 below but first we simply assume that such a succinct PCS exists and focus on compiling the protocol into one with a single monolithic verifier.

Compilation to a Monolithic Verifier. To convert the distributed protocol to the standard monolithic verifier setting, the prover commits to all interactive messages and sends these commitments to the verifier. The prover then simulates the MPC in its head, committing to all parties' views. The verifier randomly chooses a subset of the parties to open and checks for consistency.

A problem that we run into is that when compiling the distributed zero-knowledge protocol into a monolithic one, the prover provides the opening for some threshold t of the k players. However, in a usual MPC it suffices for just one of the players to cheat and so the soundness error is at best $1/k$.

To reduce this error one could just repeat the entire protocol, but doing so loses the strong notion of succinctness we target. Therefore we repeat only the distributed GKR protocol λ times. Crucially, we do so without recommitting to w each time. Rather, the prover commits to w once and then provides evaluation proofs for the λ desired locations. This yields a negligible soundness while keeping communication overhead low. The repetitions are performed in parallel, so the number of protocol rounds remains unchanged. The main problem with this approach is that the simulator cannot guess the verifier's choices, since there are exponentially many possibilities. To address this, we prove that the protocol is zero knowledge against a semi-malicious verifier[2] and then, using

[2]Recall that such a verifier follows the honest verifier strategy, except that it can choose its random coins maliciously, in advance, see, e.g., [AG21] and Definition 2.23 below.

standard techniques [GK96, BMO90, Ros04, PRS02]), compile it into a private-coin zero-knowledge protocol (see Lemma 2.24). A different approach to this problem, suggested in [IKOS09, Section 4], is to rely on a *robust* MPC protocol.[3] This approach is also limited to offering only honest-verifier zero-knowledge, or achieves malicious-verifier zero-knowledge but similarly requires using the use of private-coins.

Evaluation Point Leakage? The protocol above requires opening some λ points of the multilinear extension of the witness w. These points fully depend on w and so may reveal information about it. To address this, we pad w with a small amount of randomness and commit to this padded version. We show that padding with only λ field elements makes the λ provided evaluations reveal no information about w, for most sets of evaluation points. Moreover, we show that the "bad" evaluation points (for which random padding is insufficient) is easy to recognize and so the prover can refuse to reveal the evaluation points in this rare case. We believe this fact may be of independent interest and useful in other contexts.

1.2.1 Succinct PCS

Thus, to enable our GKR based approach we need to construct a *succinct* polynomial commitment scheme (PCS). This scheme allows us to commit to a polynomial while later opening selected evaluations, keeping the rest hidden. We construct a *succinct* PCS in which the commitment is only larger than the polynomial's description by an additive factor, and with a sub-linear evaluation proof. Interestingly, the construction is heavily inspired by Ligero [AHIV23], a computationally-sound proof-system, which, as pointed out by Golovnev *et al.* [GLS+23], can also be interpreted as a computationally sound PCS.

Let $\mathbb{F}$ be a finite field and let $f : \{0,1\}^d \to \mathbb{F}$ be a description of a multilinear polynomial $\hat{f} : \mathbb{F}^d \to \mathbb{F}$ (which extends f) to which we want to commit. We first rearrange f as an $m_r \times m_c$ matrix, where $m_r \cdot m_c = 2^d$, and the specific choices of these parameters will be determined later.

We pad the matrix with random values as follows. First, we add q additional random columns, where q is an additional parameter to be determined below, and then also add a single random row. Note that this increases the size of the matrix to $(m_c + q) \cdot (m_r + 1) = 2^d + q \cdot m_r + m_c + q$ which, by setting $q = o(m_c)$, we can afford.

The rows of the matrix are then each encoded using the Reed-Solomon code. As we will be committing to all of A, we need to be careful here — for example, we cannot afford to use a code with rate $1/2$ as it would double our communication complexity. Rather, we use a Reed-Solomon code with rate $1 - \varepsilon$. Thus, ignoring the random padding which is small, the size of A is $m_r \cdot \frac{m_c}{1-\varepsilon} \approx (1 + \varepsilon) \cdot 2^d$.

Denote the encoded matrix by A. The prover then commits to the *columns* of A. Here we use a standard cryptographic commitment scheme, that has an *additive* overhead in the cryptographic security parameter λ. Such a commitment follows easily from the existence of a PRG G (with a sufficiently large stretch): to commit to a string α, just output $\alpha \oplus G(s)$ plus a standard commitment to s, where s is a random seed of the PRG. Thus, the length of a commitment to the columns of A is:

[3] In our context, since the MPC function accepts if the prover lies in just one of the inputs, we would require in addition to the robust MPC, to utilize a *verifiable* secret sharing (VSS) scheme to distribute the messages (instead of the standard additive secret sharing). One can then use the MPC to safely reconstruct, ensuring that no small set of players can change the output by modifying their input.

$$\text{num-cols} \times \text{col-commitment-length} \approx (m_c + q) \cdot \frac{(m_r + \lambda)}{1 - \varepsilon} \approx (1 + \varepsilon) \cdot \big(2^d + \lambda \cdot m_c + q \cdot m_r + \lambda \cdot q\big).$$

Thus, the commitment to f is just a commitment to the columns of A. Note that since the column commitment is statistically binding, the overall commitment binds the sender to a particular matrix A. In a proper commitment the rows of A should all be Reed-Solomon codewords. Before describing the evaluation proof, lets consider a dishonest commitment in which at least one of the rows of A is *far* from being a valid codeword. Following [AHIV23], we detect this by having the verifier select a random linear combination of the rows of A that the prover should send back to it.

Using the so-called "proximity gaps" lemma, shown in [AHIV23] (or actually a variant by Diamond and Posen [DP23]), if one of the rows of A is far from a valid codeword, then so is the random linear combination. Thus, a cheating prover must send a *false* combination. This can then be detected by having the verifier sample sufficiently many columns of A, which the prover opens. Since the Reed-Solomon code has distance ε, we need to sample λ/ε columns to get a $2^{-\lambda}$ error. As each column has size m_r the overall communication here is $\lambda \cdot m_r/\varepsilon$.

Thus the overall communication (for both commitment and evaluation) is approximately:

$$(1 + \varepsilon) \cdot 2^d + O\big(\lambda \cdot m_c + (q + \lambda/\varepsilon) \cdot m_r + \lambda \cdot q\big).$$

Beyond the 2^d factor, the main additive factors are (1) $\varepsilon \cdot 2^d$, (2) $\lambda \cdot m_c$ and (3) $(q + \lambda/\varepsilon) \cdot m_r$. Setting $m_r = 2^{d/3}, m_c = 2^{2d/3}$, $\varepsilon = 2^{-d/3}$ and $q = O(2^{d/3})$ balances between these and we obtain overall communication $2^d + 2^{2d/3} \cdot \text{poly}(\lambda, \log(1/\delta))$.

In terms of hiding, the additional row makes the result of the random linear combination be truly random and in particular, not reveal anything about the message. Similarly, the λ random columns that were added, makes the λ revealed columns also entirely random (here we use the secret-sharing property of the Reed-Solomon code).

Thus, the prover is forced to commit to a matrix $\tilde{A}$ whose rows are close to Reed-Solomon codewords, Let $f : \{0,1\}^d \to \mathbb{F}$ denote the function obtained by correcting the rows of A to the nearby codewords and viewing the result matrix as a truth table of a function (as in the commitment process). We show that A acts as a commitment to the function f.

Let us now consider an opening proof and suppose that the prover claims that $\hat{f}(z) = v$, where $z \in \mathbb{F}^d$ and $v \in \mathbb{F}$, even though this is *not* the case. We interpret z as $z = (z_1, z_2) \in \mathbb{F}^{d/3} \times \mathbb{F}^{2d/3}$. In the opening protocol, the prover then commits to the multilinear polynomial $g(\cdot) = \hat{f}(\cdot, z_2)$. This commitment is itself via a PCS. One approach is to construct it by recursion. However, we observe that this commitment is only for a message of size $2^{d/3}$ and so we can afford to use a simpler approach — specifically, we construct a "basic" PCS, which has a small super-linear overhead that we can afford here as the input is sublinear (see Section 4 for details on this basic scheme).

The verifier uses the basic PCS to check that $\hat{g}(z_2) = v$. Note that in an honest execution this should be true since:

$$\hat{g}(z_2) = \sum_b eq(z_2, b) \cdot g(b) = \sum_b eq(z_2, b) \cdot f(z_1, b) = \hat{f}(z_1, z_2)$$

(where eq is the multilinear equality polynomial, see Section 2.2). In contrast, if the prover is cheating, it must send a *wrong* $\tilde{g} \neq g$.

To catch this, the verifier samples a random evaluation point $r \in \mathbb{F}^{2d/3}$ and asks the prover to send $combo(\cdot) = \hat{f}(\cdot, r)$. (The astute reader may notice that this message is similar to the random linear combination of the rows that we used above, and indeed we can use the same message for both tests).

If the prover sends the correct *combo* that corresponds to A, the verifier can detect an inconsistency by checking that $\widehat{combo}(z_1) = \hat{g}(r)$. Note that in an honest execution this should be true since:

$$\begin{aligned}
\widehat{combo}(z_1) &= \sum_{b_1} eq(z_1, b_1) \cdot combo(b_1) \\
&= \sum_{b_1} eq(z_1, b_1) \cdot \hat{f}(b_1, r) \\
&= \sum_{b_1, b_2} eq(z_1, b_1) \cdot eq(r, b_2) \cdot f(b_1, b_2) \\
&= \sum_{b_2} eq(r, b_2) \cdot f(z_1, b_2) \\
&= \sum_{b_2} eq(r, b_2) \cdot g(b_2) \\
&= \hat{g}(r).
\end{aligned}$$

Thus, a cheating prover must send the wrong message *combo* (relative to f). In this case, as above, the verifier can detect this by sampling $\lambda \cdot 2^{d/3}$ random columns of A and asking the prover to decommit to them.

Handling Padding. The above suffices for a succinct PCS to a function $f : \{0,1\}^d \to \mathbb{F}$. In our application however, we need to commit to a witness w whose length may not be a power of two. Typically, one would just pad w to the next power of two, which at most doubles its length. However, since we aim for a truly additive overhead, we cannot afford to do so.

Instead, when rearranging f as a matrix, we consider a version that pads the rows and columns with zeros. Importantly, we cannot afford to encode this padded version, but are still able to make a variant of the above approach go through, see Section 5 for details.

Can we Generically Compile a Succinct Zero-knowledge IOP? Loosely speaking, our proof can be viewed as using a zero-knowledge interactive oracle proof (IOP) that is inspired by Ligero [AHIV23] and then compiling it into a statistically binding zero-knowledge proof using commitments. Given that, one might wonder if one could just generically compile any succinct zero-knowledge IOP, in particular the one constructed by Ron-Zewi and Weiss [RW24], which has appealing parameters — the communication is only additively larger than the witness, and the query complexity is a *constant*.

To compile such an IOP, what we need is a succinct statistically binding commitment with a local opening (aka a vector commitment). As usual, in our context by succinct we mean that the length of the commitment is only additively larger than the message. One way to construct a statistically binding commitment with local opening is to commit separately to each symbol — but this is not succinct as the overall commitment grows multiplicatively with the security parameter.

An alternative approach, implicit in a protocol of Nassar and Rothblum [NR22] is to use a pseudorandom function (PRF) to mask the IOP oracles. The prover can then reveal the desired PRF values using an additional zero-knowledge proof. The problem with this approach is that it seems necessary for this additional zero-knowledge proof to make a non-blackbox use of the PRF.

Thus, our approach is different. At a high-level we utilize the fact that the Ligero based IOP can be thought of as an IOP with sub-linear length over a large alphabet. Using this perspective we can afford to separately commit to each of the symbols (i.e., follow the naive approach above), while retaining succinctness.

1.3 Open Questions

The main open question in this line of research is whether one can construct a succinct zero-knowledge proof for *all* NP relations, based solely on one-way functions (recall that such a result can be obtained assuming FHE [GGI+15]). This question remains unknown even in the non black box setting. As a matter of fact, such a result is not known even if one allows a $\mathrm{poly}(m)$ overhead in communication, where poly is a fixed polynomial that is independent of the circuit size (and recall that m is the witness size).

While our protocol achieves communication roughly $(1+o(1))\cdot m$, the $o(1)$ factor is roughly $m^{-1/3}$ which seems sub-optimal (in particular it is worse than what is achievable in the non-blackbox setting [AFR24]). A natural question therefore is whether it is possible to construct an even more succinct zero-knowledge proof (and PCS), with only *poly-logarimmic additive overhead* in communication (i.e., communication $m + \mathrm{poly}(\log m, \lambda, \log(1/\delta))$).

Our protocol makes use of private coins. The protocol of [IKOS09], which also achieves negligible soundness and uses the underlying primitive in a black-box manner, is likewise private-coin. As far as we are aware it is entirely open to simultaneously achieve black-box use of the underlying primitive, negligible soundness, and a public-coin protocol with communication complexity bounded by even a fixed polynomial in the witness size (let alone with constant or additive overhead).

Finally, we believe the notion of a statistically binding PCS may be of independent interest. Due to the important role that their computationally binding counterparts play, it seems possible that additional applications might be found.

1.4 Organization

We start with preliminaries in Section 2. In Section 3 we define our notion of a statistically binding PCS. In Section 4 we construct our basic PCS scheme and then in Section 5 we construct the succinct PCS. Finally, in Section 6 we give our succinct zero-knowledge protocol.

2 Preliminaries

We start with basic notions and facts from probability theory.

Definition 2.1. *Let $\mathcal{D}, \mathcal{D}'$ be two distributions over a finite domain, their* statistical distance *is defined by:*

$$SD(\mathcal{D}, \mathcal{D}') = \frac{1}{2} \sum_{y} \left| \mathcal{D}(y) - \mathcal{D}'(y) \right|.$$

Proposition 2.2. *Let S be a (non-empty) finite set, and let $S' \subseteq S$. Then, the statistical distance between the uniform distributions over S and over S' is $1 - \frac{|S'|}{|S|}$.*

Proof. Let $\mathcal{D}$ be the uniform distribution over S, and let $\mathcal{D}'$ be the uniform distribution over S'. Using Definition 2.1,

$$\begin{aligned} SD(\mathcal{D}, \mathcal{D}') &= \frac{1}{2}\left(\sum_{y \in S \setminus S'} \left|\mathcal{D}(y) - \mathcal{D}'(y)\right| + \sum_{y \in S'} \left|\mathcal{D}(y) - \mathcal{D}'(y)\right|\right) \\ &= \frac{1}{2}\left(\frac{|S \setminus S'|}{|S|} + |S'| \cdot \left|\frac{1}{|S|} - \frac{1}{|S'|}\right|\right) \\ &= 1 - \frac{|S'|}{|S|}. \end{aligned}$$

□

In this work we will be considering a notion that we refer to as *k-wise identical distributions* [BIVW16] — namely, distributions whose projects to any k points is identically distributed (this relaxes the notion of k-wise independence that requires the distribution to be uniform).

Definition 2.3 (k-wise identical)**.** *Let X and Y be distributions over strings of length n. We say that X and Y are* k-wise identical *if for any set $S \subseteq [n]$ of k indices it holds that X_S and Y_S are identically distributed.*

2.1 Computational Indistinguishably

Next, we define computational indistinguishibility.

Definition 2.4. *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:*

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

Fact 2.5 (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 Multilinear Extension

In this work we extensively use the multilinear extension of functions, defined as follows.

Definition 2.6 (Multilinear Extension)**.** *Let $f : \{0,1\}^d \to \mathbb{F}$ be a function. The* multilinear extension *$\hat{f} : \mathbb{F}^d \to \mathbb{F}$ is the unique function defined as:*

$$\hat{f}(z) = \sum_{b \in \{0,1\}^d} eq(b, z) \cdot f(b)$$

where $eq(b, z)$ is the multilinear basis function given by:

$$eq(b,z) = \prod_{j=1}^{d} \Big((1-z_j)\cdot(1-b_j) + z_j \cdot b_j \Big).$$

Proposition 2.7 ([VSBW13]). *Given $f : \{0,1\}^d \to \mathbb{F}$ and a point $x \in \mathbb{F}^d$, the multilinear extension $\hat{f}(x)$ can be computed by an arithmetic circuit of size $O(2^d)$.*

2.3 Commitment Schemes

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 2.8 (Commitment scheme)**.** *A* commitment scheme in the CRS model*, with δ-binding error, 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,*
$$\Pr_{crs \leftarrow Gen(1^\lambda), (com,dec) \leftarrow Com(crs,m)} \Big[Ver(crs, com, m, dec) = 1 \Big] = 1.$$
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: $\{Com_c(crs, m_1)\}_{\lambda\in\mathbb{N}} \overset{c}{\approx} \{Com_c(1^\lambda, crs, m_2)\}_{\lambda\in\mathbb{N}}$.*
3. ***Binding**: For every $\lambda \in \mathbb{N}$, with probability at least $1-\delta$ over the common reference string, any commitment com^* has at most one value m that can be accepted by Ver, i.e,*
$$\Pr_{crs \leftarrow Gen(1^\lambda)} \left[\exists m_1, m_2, dec_1, dec_2 \in \{0,1\}^* : \begin{array}{c} m_0 \neq m_1, \\ Ver(crs, com^*, m_1, dec_1) = 1, \\ Ver(crs, com^*, m_2, dec_2) = 1 \end{array} \right]$$
is at most $2^{-\lambda}$.

Fact 2.9. *Let com be a commitment scheme, let D be an efficiently samplable distribution, and let A be a PPT algorithm that outputs strings of length λ. Then, the distributions $(D, com(A(D)))$ and $(D, com(0^\lambda))$ are computationally indistinguishable.*

Fact 2.10. *Let com be a commitment scheme and let (X, Y) and (X', Y') be efficiently samplable joint distributions, such that $X \overset{c}{\approx} X'$ and Y and Y' are supported over strings of length λ. Then, the distributions $(X, com(Y))$ and $(X', com(Y'))$ are computationally indistinguishable.*

We use the construction of a non-interactive commitment in the CRS model, due to [Nao91] (building also on [HILL99]).

Theorem 2.11 ([HILL99, Nao91]). *Assuming the existence of one-way functions, for any $\delta > 0$ there exists a commitment scheme in the CRS model with binding error δ such that the size of a commitment for a message of length ℓ, as well as the verification complexity, are* $\mathrm{poly}(\ell, \lambda) \cdot \log(1/\delta)$.

The above commitment scheme can be extended to have a length that is only additively larger than the message, as follows.

Proposition 2.12. *Assuming the existence of one-way functions, for any $\delta > 0$ there exists a commitment scheme in the CRS model with binding error δ such that the size of a commitment for a message of length ℓ, as well as the verification complexity, are $\ell + \mathrm{poly}(\lambda) \cdot \log(1/\delta)$.*

Proof Sketch. The existence of a one-way function implies the existence of a pseudorandom generator G (via [HILL99]) and a non-interactive commitment scheme *com* (in the CRS model) via Theorem 2.11.

Now, to commit to a message m, generate a seed s and send $(com(s), G(s) \oplus m)$. Hiding follows from the hiding property of the commitment scheme and the pseudorandomness of G. Binding follows from the binding of the commitment. □

Theorem 2.13 (Statistically Hiding Commitment from One-Way Functions [HR07]). *Assuming the existence of one-way functions, for any security parameter λ and message length $\ell \in \mathbb{N}$, there exists a commitment scheme with the following properties:*

1. ***Statistically Hiding:*** *For any two distinct messages $m_0, m_1 \in \{0,1\}^\ell$, the statistical distance between the distributions of commitments to m_0 and m_1 is bounded by $2^{-\lambda}$.*
2. ***Computationally Binding:*** *For every probabilistic polynomial-time adversary A and every security parameter λ,*

$$\Pr_{crs \leftarrow Gen(1^\lambda)} \left[(com, m_1, m_2, dec_1, dec_2) \leftarrow A(crs) : \begin{array}{l} m_1 \neq m_2, \\ Ver(crs, com, m_1, dec_1) = 1, \\ Ver(crs, com, m_2, dec_2) = 1 \end{array} \right]$$

is at most $2^{-\lambda}$.

The size of a commitment for a message of length ℓ, as well as the verification complexity, are $\ell \cdot \mathrm{poly}(\lambda)$. The commitment phase requires $\ell \cdot \mathrm{poly}(\lambda)$ rounds of interaction, while the decommitment phase is non-interactive.

2.4 Error Correcting Codes

The *Hamming distance* between two strings $s, s' \in \mathbb{F}^n$ is the number of entries on which they differ:

$$\Delta(s, s') = |\{i \in [n] \colon s_i \neq s'_i\}|.$$

The *relative Hamming distance* is equal to the Hamming distance divided by n.

Definition 2.14 (Linear Code). *A* linear code *over an alphabet $\mathbb{F}$ is an injective linear function $C : \mathbb{F}^k \to \mathbb{F}^{\ell(k)}$.*

Definition 2.15 (Minimum Relative Distance)**.** *The* minimum relative distance *of a code* $C : \mathbb{F}^k \to \mathbb{F}^{\ell(k)}$*, denoted by* d *is:*

$$\min_{m \neq m'} \frac{\Delta(C(m), C(m'))}{\ell(k)}$$

Theorem 2.16 (Rephrasing of [DP23, Theorem 3.1])**.** *Fix an arbitrary linear code* $C : \mathbb{F}^k \to \mathbb{F}^n$*, with relative distance* d *and a proximity parameter* $e \in \left\{0, \ldots, \left\lfloor \frac{n \cdot d - 1}{3} \right\rfloor \right\}$*. Suppose that the function* $u : \{0,1\}^{\log(m)} \to \mathbb{F}^n$ *satisfies:*

$$\Pr_{r \in \mathbb{F}^{\log m}} \left[\Delta\left(\hat{u}(r), C\right) \leq e \right] > 2 \cdot \log(m) \cdot \frac{e+1}{|\mathbb{F}|},$$

where $\hat{u}$ *is the multilinear extension of* u*. Then* $\Delta\Big(\big(u(i)\big)_{i=0}^{m-1}, C^m\Big) \leq e$*, where* C^m *is the* m*-fold interleaving of the code* C*.*

Definition 2.17 (Reed-Solomon [RS60] code)**.** *Let* $\mathbb{F}$ *be a finite field, and let* $k, n \in \mathbb{N}$ *with* $k \leq n \leq |\mathbb{F}|$*. Fix distinct evaluation points* $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$*. The* Reed Solomon code $\mathrm{RS}_{n,k}$ *is the linear code defined by*

$$\mathrm{RS}_{n,k}(m) = \left\{ \big(p(\alpha_1), \ldots, p(\alpha_n)\big) \,\middle|\, p(X) = \sum_{i=0}^{k-1} m_i \cdot X^i \right\}.$$

Equivalently, each message $m \in \mathbb{F}^k$ *is identified with the coefficient vector of a polynomial of degree less than* k*, and its encoding is the evaluation of this polynomial on the points* $\alpha_1, \ldots, \alpha_n$*.*

We also use "zero-knowledge codes" [ISVW13, DGR20, BCL22, RW24] These are randomized codes in which a small number of symbols reveal nothing about the message.

Definition 2.18 (Zero-knowledge Code)**.** *A* λ-zero knowledge code *is a linear code* $C : \mathbb{F}^k \to \mathbb{F}^{\ell(k)}$ *with the following property: for every two messages* $m_0, m_1 \in \mathbb{F}^k$*, the distributions*

$$\left(C(m_0; r)\right)_{r \leftarrow \mathbb{F}^\lambda} \quad \textit{and} \quad \left(C(m_1; r)\right)_{r \leftarrow \mathbb{F}^\lambda}$$

are λ*-wise identical (see Definition 2.3).*

Lemma 2.19. *The Reed Solomon code is a zero knowledge code.*

Proof. Let $\alpha_1, \ldots, \alpha_n$ denote the evaluation points of the RS code. Given a message $m = (m_0, \ldots, m_{k-1}) \in \mathbb{F}^k$, sample $r = (r_0, \ldots, r_{\lambda-1}) \leftarrow \mathbb{F}^\lambda$ uniformly, and define

$$\mathrm{RS}_{n,k}(m; r) = \big(p(\alpha_1), \ldots, p(\alpha_n)\big),$$

where $p(x) = r_0 + r_1 x + \cdots + r_{\lambda-1} x^{\lambda-1} + m_0 x^\lambda + \cdots + m_{k-1} x^{k+\lambda-1}$.

This can also be written as,

$$\mathrm{RS}_{n,k}(m; r) = \underbrace{\begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{\lambda-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{\lambda-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \cdots & \alpha_n^{\lambda-1} \end{pmatrix}}_{V_{n,\lambda}} \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{\lambda-1} \end{pmatrix} + \underbrace{\begin{pmatrix} \alpha_1^\lambda & \alpha_1^{\lambda+1} & \cdots & \alpha_1^{k+\lambda-1} \\ \alpha_2^\lambda & \alpha_2^{\lambda+1} & \cdots & \alpha_2^{k+\lambda-1} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_n^\lambda & \alpha_n^{\lambda+1} & \cdots & \alpha_n^{k+\lambda-1} \end{pmatrix}}_{V_{n,k}^{(\lambda)}} \begin{pmatrix} m_0 \\ m_1 \\ \vdots \\ m_{k-1} \end{pmatrix}$$

which is equal to $= V_{n,\lambda} \cdot r + V_{n,k}^{(\lambda)} \cdot m$, where $V_{n,\lambda}$ and $V_{n,k}^{(\lambda)}$ together form the Vandermonde matrix of dimension $n \times (\lambda + k)$. Now, fix any subset $S \subseteq [n]$ with $|S| = \lambda$.

Since any λ rows of $V_{n,\lambda}$ form a Vandermonde matrix, by a standard property of Vandermonde matrices, the submatrix $V_{S,\lambda}$, (where A_S is the restriction of A to the coordinates in S if A is a vector, and to the rows in S if A is a matrix), has full rank λ.

For a given message $m \in \mathbb{F}^k$ the projection of $RS_{n,k}(m;r)$ to coordinates in S is equal to $V_{S,\lambda} \cdot r + V_{S,k}^{(\lambda)} \cdot m$. Since $r \leftarrow \mathbb{F}^\lambda$ is uniform, the distribution $V_{S,\lambda} \cdot r$ is uniform over $\mathbb{F}^\lambda$.

Consequently, for any two messages $m, m' \in \mathbb{F}^k$, the projected distributions

$$(RS_{n,k}(m;r)_S)_{r \sim \mathbb{F}^\lambda} \quad \text{and} \quad (RS_{n,k}(m';r)_S)_{r \sim \mathbb{F}^\lambda}$$

are identically distributed.

□

2.5 Interactive Proofs

We first recall the notion of an interactive proof [GMR89].

Definition 2.20 (Interactive Proof)**.** *A pair of interactive machines (P, V) is called an* interactive proof system *for a language ℓ with soundness error $\varepsilon = \varepsilon(\lambda) > 0$, if V is a probabilistic polynomial-time machine, and the following conditions hold for every security parameter $\lambda \in \mathbb{N}$:*

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

We say that the interactive proof is public-coin *if all of the verifier's messages consist of random coin tosses (and the verifier does not toss additional coins). Otherwise, we say that the proof is* private-coin*. 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*, the honest prover is given access to* an NP witness as an auxiliary input*.*

We remark that all interactive 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* (see Definition 2.6). We state their result in a way that will be convenient for our construction.

Theorem 2.21 (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 three phase public-coin interactive proof $(P, V_{interactive}, V_{post}, V_{eval})$ with the following properties*

1. *In the interactive phase $(P, V_{interactive})$, P gets as input (C,x) and $V_{interactive}$ gets only the parameters S and D. The prover P runs in time $\mathrm{poly}(S)$, and the verifier $V_{interactive}$ 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* $z \in \mathbb{F}^{\log(n)} \alpha \in \mathbb{F}$ *and* $\langle C \rangle \in \{0,1\}^{\text{poly}(D,\log(S),\log(|\mathbb{F}|))}$ *in time* $\text{poly}(S)$.

3. V_{post} *gets as input* $\left(\text{transcript}, \langle C \rangle, z\right)$ *and either accept or rejects.* V_{post} *performs a test on* $(transcript, \langle C \rangle, z)$ *and runs in time* $\text{poly}(D, \log(S), \log(|\mathbb{F}|)$

4. V_{eval} *gets as input* (α, z) *and just checks the claim* $\hat{x}(z) = \alpha$. V_{eval} *runs in time* $\text{poly}(D, \log(S), \log(|\mathbb{F}|)$.

The interactive protocol obtained by first running the interactive phase, then having the verifier derive $\langle C \rangle, \hat{x}(z)$ *and running* V_{post}*, and finally deriving* α *from the transcript and running* V_{eval} *on* (x, z, α) *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 three phases as above. However, such a separation follows easily using the fact that the GKR protocol is *holographic*, meaning that the verifier only needs to make a single query to the low degree extension of the input prior to the interaction, and subsequently runs in $\text{poly}(D, \log(S), \log(|\mathbb{F}|))$ time.

2.6 Zero-Knowledge Proofs

Next, we define (computationally) zero-knowledge proofs [GMR89].

Definition 2.22 (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*
- $\left\{Sim\left(x, 1^\lambda\right)\right\}_{\lambda \in \mathbb{N}}$,

where $View_{\hat{V}}^{P(w)}\left(x, 1^\lambda\right)$ *is the content of the random tape of* $\hat{V}$ *and the messages sent and received by* $\hat{V}$ *in an interaction with* P *on common input* $(x, 1^\lambda)$.

For simplicity we use here the standalone definition but remark that our results can be extended to the setting of *auxiliarly input* zero-knowledge.

Definition 2.23. *[Zero-Knowledge with Respect to a Semi-Malicious Verifier] Zero-knowledge with respect to a semi-malicious verifier is similar to standard zero-knowledge, except that the zero-knowledge property is required to hold only against a verifier who, in advance, fixes an arbitrary string* r *(of the appropriate length), and then operate according to the* honest verifier strategy *using the string* r *as its random string. The simulator receives* r *as an auxiliary input. We denote such an honest verifier with fixed* r *by* $\hat{V}_r$.

Lemma 2.24 (From Semi-Malicious ZKP to Fully-Malicious)**.** *Assume the existence of one-way functions. Suppose the language* L *has a public-coin zero-knowledge proof with respect to a semi-malicious verifier with soundness error* δ*, communication complexity* c*, round complexity* r*, and where the verifier uses* t *bits of randomness. Then, for every parameter* $\delta' > 0$ *the language* L *has a*

private-coin zero-knowledge proof, with respect to a malicious verifier, with soundness error $\delta + \delta'$, communication complexity $c + \text{poly}(\log(1/\delta'), \lambda, t)$, and round complexity $r + \text{poly}(\log(1/\delta')) \cdot t \cdot \lambda$, where λ denotes the (computational) security parameter).

Proof Sketch. The proof is based on the standard technique (see [GK96, BMO90, Ros04, PRS02]) of adding an initial step to the protocol in which the verifier commits in advance to its random coins using a *statistically* hiding commitment scheme.[4] This step can be used by the simulator to learn the verifier's choices. Given these random choices, we can employ the semi-malicious simulator. During the emulation of the semi-malicious protocol, the verifier is also required to prove consistency of its randomness by gradually opening the commitment in every round. The binding of the commitment ensures that its messages are consistent with the initial commitment. □

2.7 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.

Definition 2.25 (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 2.26 (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 2.27 ([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* $\text{poly}(S, n)$.

3 Polynomial Commitment Schemes

Definition 3.1 (Polynomial Commitment Scheme)**.** *A PCS with δ binding error is a tuple (C, P, V), where C is a probabilistic polynomial time algorithm and (P, V) are a pair of probabilistic polynomial time interactive machines with the following syntax:*

[4] Recall that the existence of such a commitment scheme follows from the existence of a one-way function, see Theorem 2.13.

- *For any function* $f : \{0,1\}^d \to \mathbb{F} : (com, dec) \leftarrow C(1^\lambda, f)$
- P *and* V *interaction:*
  - *The common input :* $\big(com, \vec{z} \in (\mathbb{F}^d)^k, \bar{\alpha} \in \mathbb{F}^k\big)$.
  - P *gets* dec *as an additional input.*
  - *At the end of the interaction, the verifier outputs either 0 or 1.*

  We denote by $(V, P_{dec})(1^\lambda, com, \vec{z}, \bar{\alpha})$ *the output of* V *when interacting with* P *given the input as above, where* λ *is the security parameter.*

The scheme must satisfy the following requirements:

- ***Correctness:*** *For any function* $f : \{0,1\}^d \to \mathbb{F}$*, any set of* k *distinct inputs* $\vec{z} = (z_1, \ldots, z_k) \in (\mathbb{F}^d)^k$*, and any security parameter* λ*,*

$$\Pr_{(com,dec)\leftarrow C(1^\lambda,f)}\Big[(P_{dec}, V)(1^\lambda, com, \vec{z}, \hat{f}(\vec{z})) = 1\Big] = 1,$$

  where $f(\vec{z}) = (\hat{f}(z_1), \ldots f(z_k)) \in \mathbb{F}^k$*, and* $\hat{f}$ *is the multilinear extension of* f*.*

- ***Binding:*** *For any commitment* com^**, there exist a multilinear polynomial* $f_{com^*} : \mathbb{F}^d \to \mathbb{F}$ *such that the following holds. For every prover* P^* *any security parameter* λ*, and any set of* k *input and output values:* $(z_1, \ldots, z_k) \in (\mathbb{F}^d)^k, (\alpha_1, \ldots, \alpha_k) \in \mathbb{F}^k$*, such that* $\exists i, f_{com^*}(z_i) \neq \alpha_i$ *it holds that,*

$$\Pr\Big[(P^*, V)(1^\lambda, com^*, z_1, \ldots, z_k, \alpha_1, \ldots, \alpha_k) = 1\Big] < \delta.$$

- ***Commitment Hiding****: For every distribution* F *over* d*-variate functions from* $\{0,1\}^d$ *to* $\mathbb{F}$ *it holds that* $C(F)$ *is a commitment in the sense of Definition 2.8.*
- ***Evaluation Hiding****: For a given* $f : \{0,1\}^d \to \mathbb{F}$*, denote by* $(C, P, V^*)(f)_k$ *the view of the full execution consists of:*

$$\Big(com, z_1, f(z_1), \ldots z_k, f(z_k), (V^*, P_{dec})(1^\lambda, com, z_1, \ldots, z_k, f(z_1), \ldots, f(z_k))\Big),$$

  generated as follows:
  - V^* *selects* k *points* $z_1, \ldots, z_k \in \mathbb{F}^d$*.*
  - $(dec, com) \leftarrow C(f)$*.*
  - P *receives* com, dec *and the points chosen by* V^**, it then provides provides the corresponding evaluations* $f(z_1), \ldots, f(z_k)$*.*
  - V^* *and* P *then run the* $(V^*, P_{dec})(1^\lambda, com, \vec{z}, f(\vec{z}))$ *protocol .*

  For every probabilistic polynomial-time machine V^* *and for every pair of functions* $f_1, f_2 : \{0,1\}^d \to F$ *and points* $z_1, \ldots, z_d \in \mathbb{F}^d$ *on which* $\hat{f}_1$ *and* $\hat{f}_2$ *agree, it holds that*

$$(C, P, V^*)(f_1)_k \approx_c (C, P, V^*)(f_2)_k.$$

Definition 3.2 (Polynomial Commitment Scheme with Respect to a Semi-Malicious Verifier)**.** *A polynomial commitment scheme with respect to a semi-malicious verifier is similar to a polynomial commitment scheme, except that the evaluation hiding property is required to hold only against a verifier who, in advance, fixes an arbitrary string r (of the appropriate length), and then operate according to the* honest *verifier strategy using the string r as its random string. We denote such an honest verifier with fixed r by $\hat{V}_r$.*

Remark 3.3 (On Padded Polynomials)**.** *In this work our focus is on communication complexity and we care even about very small multiplicative factors. In particular, we will sometimes want to commit to multilinear polynomials representing data of length that is not a power of 2. The straightforward solution of padding to the next power of 2 does not work for us, as we would like to avoid a potential doubling of the communication.*

Accordingly, we extend the definition of the polynomial commitment scheme to strings of arbitrary length m (not necessarily a power of two) by interpreting the string as a multilinear polynomial after padding with zeroes up to the next power of two. However, we analyze the communication complexity with respect to original length m (rather than $2^{\lceil \log m \rceil}$).

Lemma 3.4. *Let $z_1, \ldots, z_k \in \mathbb{F}^d$ and let F_1, F_2 be distributions over d-variate multilinear functions. Assume that $(F_1(z_1), \ldots, F_1(z_k)) \equiv (F_2(z_1), \ldots, F_2(z_k))$, that is, the distributions of F_1 and F_2 restricted to these points are identically distributed.*

Let (C, P, V) be a polynomial commitment scheme. Then, for every probabilistic polynomial-time verifier V^*,

$$(C, P, V^*)(F_1) \approx_c (C, P, V^*)(F_2).$$

Proof. Let A be a probabilistic polynomial-time adversary.

We bound the distinguishing advantage $p = \big|\Pr[A((C, P, V^*)(F_1)) = 1] - \Pr[A((C, P, V^*)(F_2)) = 1]\big|$ as:

$$\begin{aligned}
p = \Bigg| \sum_{r \in \mathbb{F}^k} \Big(& \Pr_{f_1 \leftarrow F_1}[f_1(\vec{z}) = r] \cdot \Pr[A((C, P, V^*)(f_1)) = 1 \mid f_1(\vec{z}) = r] \\
& - \Pr_{f_2 \leftarrow F_2}[f_2(\vec{z}) = r] \cdot \Pr[A((C, P, V^*)(f_2)) = 1 \mid f_2(\vec{z}) = r] \Big) \Bigg| \\
\leq \sum_{r \in \mathbb{F}^k} & \Pr[f_1(\vec{z})] \cdot \Big| \Pr[A((C, P, V^*)(f_1)) = 1 \mid f_1(\vec{z}) = r] \qquad (1) \\
& - \Pr[A((C, P, V^*)(f_2)) = 1 \mid f_2(\vec{z}) = r] \Big|, \qquad (2)
\end{aligned}$$

where the last step follows from the assumption that F_1 and F_2 are distributed identically over z and the triangle inequality.

Fix any $r \in \mathbb{F}^k$. Consider any $f_1 \leftarrow F_1$ and $f_2 \leftarrow F_2$ such that $f_1(\vec{z}) = f_2(\vec{z}) = r$. By the evaluation-hiding property of the commitment scheme, for every probabilistic polynomial-time verifier V^*,

$$(C, P, V^*)(f_1)_k \approx_c (C, P, V^*)(f_2)_k.$$

Consequently,

$$\big|\Pr[A((C, P, V^*)(f_1)) = 1 \mid f_1(\vec{z}) = r] - \Pr[A((C, P, V^*)(f_2)) = 1 \mid f_2(\vec{z}) = r]\big|$$

is negligible for every $r \in \mathbb{F}^k$.

Combined with Eq. (1) this implies that:

$$p = \left| \Pr[A((C, P, V^*)(F_1)) = 1] - \Pr[A((C, P, V^*)(F_2)) = 1] \right|$$

is negligible, as required. □

4 Basic Polynomial Commitment

In this section, we present a basic polynomial commitment scheme. While the scheme is not optimized for communication efficiency, it is conceptually simple and will also serve as a building block for the *succinct* polynomial commitment that we construct in Section 5. Thus, the main result that we prove in this section is the following lemma.

Lemma 4.1. *Assume the existence of a one-way function. Then, for every $\delta = \delta(d) \in (0, 1)$ there exists a polynomial commitment scheme for d-variate multilinears over a finite field $\mathbb{F}$, with binding error δ and communication complexity $2^d \cdot \log(\frac{1}{\delta}) \cdot \text{poly}(\lambda) \cdot \log(|\mathbb{F}|)$ for the commitment and $O\big((2^d \cdot \text{poly}(\lambda) + k) \cdot \log(\frac{1}{\delta}) \cdot \log(|\mathbb{F}|)\big)$ for the opening, where k is the number of points to be opened, λ is the security parameter and the commitment and opening protocols have $O(\log(1/\delta))$ round complexity.*

Furthermore, a semi-malicious polynomial commitment scheme with similar parameters exists except that it has a constant-round complexity.

The rest of this section is devoted to proving Lemma 4.1.

4.1 Proof of Lemma 4.1

We first describe a protocol with constant binding error (specifically 0.6), and then extend it via repetition to reduce the binding error to δ which then implies the lemma (see Remark 4.3 for details).

By Theorem 2.11, the existence of a one-way function implies the existence of a standard bit-commitment *Com* in the CRS model, with binding error 0.1. We assume that a fixed CRS is chosen that is indeed statistically binding and the protocol below is relative to this fixed CRS. The protocol (with binding error 0.6), which establishes Lemma 4.1 (up to the aforementioned repetitions) is described in Fig. 1. We proceed to the analysis.

Correctness. Let $f : \{0, 1\}^d \to \mathbb{F}$ be a function, $z_1, \ldots, z_k \in \mathbb{F}^d$ be a sequence of points and $\alpha_1, \ldots, \alpha_k \in \mathbb{F}$ s.t. $\alpha_i = \hat{f}(z_i)$, for every $i \in [k]$.

If the prover is honest, the commitment and decommitment pass the verifier's check in Step 4a of the evaluation protocol. Since f_0, f_1 are an additive secret sharing of f, it holds that $\hat{f}_0(z_i) + \hat{f}_1(z_i) = \hat{f}(z_i) = \alpha_i$, for every $i \in [k]$, and so the check in Step 4b passes. Since P sends the correct evaluations of f_0 and f_1 in Step 1, the check in Step 4c also passes and so the verifier accepts.

Binding. We first note that with all but 0.1 probabililty, the commitment scheme is perfectly binding. Let us assume that this is the case.

Let com^* be a (possibly maliciously generated) polynomial commitment. Recall that com^* should consist of commitments to two functions $f_0^*, f_1^* : \{0, 1\}^d \to \mathbb{F}$. By the binding property of the commitment scheme (see Definition 2.8) there exists at most one such f_0^* and f_1^* that can pass

Basic Polynomial Commitment Scheme

$C(f, 1^\lambda)$, where $f : \{0,1\}^d \to \mathbb{F}$:

1. Generate an additive secret sharing $f_0, f_1 : \{0,1\}^d \to \mathbb{F}$ of f (that is, $f_0 : \{0,1\}^d \to \mathbb{F}$ is chosen uniformly at random and $f_1 = f - f_0$).
2. For $b \in \{0,1\}$, generate:
$$(com_b, dec_b) \leftarrow Com(f_b).$$
3. Set $com = (com_0, com_1)$ and $dec = (dec_0, f_0, dec_1, f_1)$.

Basic Polynomial Evaluation Protocol

Common Input: commitment $com = \big(com_0, com_1\big)$, points $z_1, \ldots, z_k \in \mathbb{F}^d$, evaluation claims $\alpha_1, \ldots, \alpha_k \in \mathbb{F}$ and security parameter 1^λ.
Prover's Additional Input: decommitment $dec = \big(dec_0, f_0, dec_1, f_1\big)$.

1. For every $\sigma \in \{0,1\}$ and $i \in [k]$, the prover P computes the evaluations $\beta_i^{(\sigma)} = \hat{f}_\sigma(z_i)$ and sends them to V.
2. V randomly chooses $b \leftarrow \{0,1\}$ and sends it to P.
3. P sends (f_b, dec_b) to V.
4. V checks:
   (a) The decommitment dec_b is a valid decommitment to f_b.
   (b) For every $i \in [k]$, it holds that $\beta_i^{(0)} + \beta_i^{(1)} = \alpha_i$.
   (c) For every $i \in [k]$, it holds that $\hat{f}_b(z_i) = \beta_i^b$.

   If all checks pass then V accepts, otherwise it rejects.

Figure 1: Basic Protocol

the verifier's decommitment check. We assume wlog that f_0^* and f_1^* indeed exist since otherwise the verifier will reject with probability at least $\frac{1}{2}$ in Step 3. We define $f_{com^*} = f_0^* + f_1^*$.

Let $z_1, \ldots, z_k \in (\mathbb{F}^d)^k$ and $\alpha_1, ..., \alpha_k \in \mathbb{F}^k$ such that there exists $j^* \in [k]$ such that $\hat{f}_{com^*}(z_{j^*}) \neq \alpha_{j^*}$, and let P^* be a malicious prover strategy. For every $\sigma \in \{0,1\}$ and $i \in [k]$, let $\beta_i^{(\sigma)}$ be the claimed evaluations sent by P^* in Step 1.

If P^* sends values $\beta_j^{(0)}$, $\beta_j^{(1)}$ such that $\beta_j^{(0)} + \beta_j^{(1)} \neq \alpha_j$, then the verifier V rejects in Step 4b and so we may assume that $\beta_j^{(0)} + \beta_j^{(1)} = \alpha_j$, for all $j \in [k]$.

Since $f_0^*(z_{j^*}) + f_1^*(z_{j^*}) = f_{\mathsf{com}^*}(z_{j^*}) \neq \alpha_{j^*}$, the prover must be dishonest in at least one of the values: either $\beta_{j^*}^{(0)} \neq f_0^*(z_{j^*})$ or $\beta_{j^*}^{(1)} \neq f_1^*(z_{j^*})$. Therefore, with probability at leasts $1/2$, it holds that V chooses the index b (in Step 2) on which the prover lied. Then, either P^* sends an invalid decommitment and V rejects in Step 4a, or P^* successfully decommits and V rejects in Step 4c due to an incorrect evaluation.

Overall, the verifier accepts here with probability at most $\frac{1}{2}$. Accounting also for the probability that the commitment is not binding, we get a binding error of 0.6.

Commitment Hiding. The commitment hiding property follows immediately from the fact that the polynomial commitment $C(F)$ just consists of a pair of standard commitments.

Evaluation Hiding. Let V^* be a probabilistic polynomial-time machine, and let $f, g : \{0,1\}^d \to \mathbb{F}$ be a pair of functions. Let $z_1, \ldots, z_k \in \mathbb{F}^d$ be points on which $\hat{f}$ and $\hat{g}$ agree. Then it holds that

$$(C, P, V^*)(f)_k \approx_c (C, P, V^*)(g)_k.$$

where we recall that (see Definition 3.1)

$$\begin{aligned}\Big(C, P, V^*\Big)(f)_k &= \Big(\vec{z}, C(f), \vec{z}, f(\vec{z}), f_0(\vec{z}), f_1(\vec{z}), b, dec_{f_b}\Big) \\ &= \Big(\vec{z}, (com_{f_0}, com_{f_1}), f(\vec{z}), f_0(\vec{z}), f_1(\vec{z}), b, dec_{f_b}\Big)\end{aligned}$$

and where,

- $\vec{z} = (z_1, \ldots, z_k)$ are the k points chosen in advance by V^* (on which f and g agree).
- f_0, f_1 are a random secret sharing of f as in Step 1.
- b is the selection bit chosen by V^* (which may depend both on the commitment and the evaluation sent by P in Step 1).
- com_{f_0} and com_{f_1} are the (standard) commitments to the secret sharing components f_0 and f_1, respectively, as defined in Step 1, and dec_{f_0} and dec_{f_1} are the corresponding decommitments.

The proof loosely follows the structure of the zero knowledge proof for the 3-coloring protocol in [Gol01, Section 4.4.2.3].

Assume, toward a contradiction, that $(C, P, V^*)(f)$ and $(C, P, V^*)(g)$ are distinguishable. Let $B_1, B_2 \in \{0,1\}$ be the bit chosen by V^* in Step 2, with respect to the interaction over $C(f)$ and $C(g)$, respectively. Note that these random variables may depend on the commitment and previous messages during the interaction. For every fixed $b \in \{0,1\}$, let:

- $\mu_b(f)$ denote the output of the full interaction given f as input, conditioned on V^*'s query being b; and
- $p_b(f)$ denote the probability that V^* selects b during the interaction with P given f as input.

Proposition 4.2. *For every $b \in \{0,1\}$ it holds that $\left|p_b(f) - p_b(g)\right|$ is negligible.*

Proof. Assume towards a contradiction that there exists b such that $\left|p_b(f) - p_b(g)\right|$ is non-negligible.

Claim 4.2.1. *It holds that*

$$\left(\vec{z}, (com_{f_0}, com_{f_1}), f(\vec{z}), f_0(\vec{z}), f_1(\vec{z})\right) \overset{c}{\approx} \left(\vec{z}, (com_{g_0}, com_{g_1}), g(\vec{z}), g_0(\vec{z}), g_1(\vec{z})\right)$$

where f_0 and f_1 (resp. g_0 and g_1) form an additive secret sharing of f (resp. g).

Proof. From our assumption,

$$\left(\vec{z}, f(\vec{z})\right) \equiv \left(\vec{z}, g(\vec{z})\right).$$

Moreover, since f_0, f_1 and g_0, g_1 are random secret sharings of f and g, respectively, we have,

$$\left(\vec{z}, f(\vec{z}), f_0(\vec{z}), f_1(\vec{z})\right) \equiv \left(\vec{z}, g(\vec{z}), g_0(\vec{z}), g_1(\vec{z})\right).$$

From the hiding property of the commitment, we conclude that,

$$\left(\vec{z}, (com_{f_0}, com_{f_1}), f(\vec{z}), f_0(\vec{z}), f_1(\vec{z})\right) \overset{c}{\approx} \left(\vec{z}, (com_{g_0}, com_{g_1}), g(\vec{z}), g_0(\vec{z}), g_1(\vec{z})\right).$$

□

Now, we could construct a (non-uniform) distinguisher between the distributions

$$\left(\vec{z}, (com_{f_0}, com_{f_1}), f(\vec{z}), f_0(\vec{z}), f_1(\vec{z})\right) \quad \text{and} \quad \left(\vec{z}, (com_{g_0}, com_{g_1}), g(\vec{z}), g_0(\vec{z}), g_1(\vec{z})\right).$$

The distinguisher (which has b hardcoded) simply runs V^* on its input and outputs 1 if V^* selects b, and 0 otherwise. This contradicts Claim 4.2.1, completing the proof of Proposition 4.2. □

Recall that we assumed that the ensembles $(C, P, V^*)(f)_k$ and $(C, P, V^*)(g)_k$ are distinguishable. Next, we show that there exists a fixed $b \in \{0,1\}$ such that a distinguisher can distinguish $(C, P, V^*)(f)_k$ and $(C, P, V^*)(g)_k$ even conditioned on this choice, despite the fact that $p_b(f) \overset{c}{\approx} p_b(g)$.

Using the distinguisher between $(C, P, V^*)(f)_k$ and $(C, P, V^*)(g)_k$ and an averaging argument, we obtain that there exists a probabilistic polynomial-time algorithm A, a polynomial $q(\cdot)$, and an infinite sequence of integers d such that, for each d in the sequence, and f, g being functions from $\{0,1\}^d$ to $\mathbb{F}$ that are identical over z, there exist a choice b such that

$$\left|p_b(f) \cdot \Pr[A(\mu_b(f)) = 1] - p_b(g) \cdot \Pr[A(\mu_b(g)) = 1]\right| \geq \frac{1}{q(\lambda)}.$$

This further implies, the following:

1. $p_b(f) > \frac{1}{2 \cdot q(\lambda)}$.
2. $\left|p_b(f) - p_b(g)\right| < \frac{1}{8 \cdot q(\lambda)^2}$.

3. $\left|\Pr[A(\mu_b(f))=1]-\Pr[A(\mu_b(g))=1]\right| > \frac{1}{2\cdot q(\lambda)}$.

To see this, note that Item 2 follows immediately from Proposition 4.2. Given Item 2 we conclude that $\left|p_b(f)\cdot\Pr[A(\mu_b(f))]-p_b(f)\cdot\Pr[A(\mu_b(g))]\right| \geq \frac{1}{2\cdot q(\lambda)}$ and Items 1 and 3 follow.

We now show that this contradicts the following claim:

Claim 4.2.2. *Let $b\in\{0,1\}$ be a fixed query from V^*. Denote*

$$H_0 = \left(\vec{z},(com_{f_0},com_{f_1}),f(\vec{z}),f_0(\vec{z}),f_1(\vec{z}),b,f_b,dec_{f_b}\right),$$
$$H_1 = \left(\vec{z},(com_{g_0},com_{g_1}),g(\vec{z}),g_0(\vec{z}),g_1(\vec{z}),b,g_b,dec_{g_b}\right),$$

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

Proof. We show that the two distributions are indistinguishable in a step-by-step manner, as follows:

1. From the assumption we have that:
$$(\vec{z},f(\vec{z}))\equiv(\vec{z},g(\vec{z})).$$
2. Since f_0,f_1 and g_0,g_1 are random secret sharings of f and g, respectively, together with the previous item, we obtain:
$$(\vec{z},f(\vec{z}),f_0(\vec{z}),f_1(\vec{z}),b)\equiv(\vec{z},g(\vec{z}),g_0(\vec{z}),g_1(\vec{z}),b).$$
3. Because each share f_b and g_b is individually uniform, except where constrained by the shared evaluations ($f(\vec{z})$ or $g(\vec{z})$, respectively), it follows that:
$$(\vec{z},f(\vec{z}),f_0(\vec{z}),f_1(\vec{z}),b,f_b)\overset{c}{\approx}(\vec{z},g(\vec{z}),g_0(\vec{z}),g_1(\vec{z}),b,g_b).$$
4. Finally, by the commitment hiding property of the commitment scheme and Fact 2.5, we conclude: $H_0\overset{c}{\approx}H_1$.

□

Now define A' as the algorithm that runs the full interaction and checks whether b was selected by V^*. If so, it runs A; otherwise, it outputs 0. Then we have:

$$\begin{aligned}
\left|\Pr[A'(H_0)]-\Pr[A'(H_1)]\right| &= \left|p_b(f)\cdot\Pr[A(\mu_b(f))]-p_b(g)\cdot\Pr[A(\mu_b(g))]\right| \\
&\geq p_b(f)\cdot\left|\Pr[A(\mu_b(g))]-\Pr[A(\mu_b(f))]\right| \\
&\quad -\Pr[A(\mu_b(g))]\cdot\left|p_b(g)-p_b(f)\right| \\
&\geq p_b(f)\cdot\left|\Pr[A(\mu_b(g))]-\Pr[A(\mu_b(f))]\right| \\
&\quad -\left|p_b(g)-p_b(f)\right| \\
&> \frac{1}{2\cdot q(\lambda)}\cdot\frac{1}{2\cdot q(\lambda)}-\frac{1}{8\cdot q(\lambda)^2} \\
&= \frac{1}{8\cdot q(\lambda)^2},
\end{aligned}$$

which contradicts Claim 4.2.2. Therefore, we reach a contradiction and conclude that

$$(C,P,V^*)(f)_k\overset{c}{\approx}(C,P,V^*)(g)_k.$$

Communication Complexity: The prover first commits to two d-variate multilinears, sending a message of length $2^d \cdot \text{poly}(\lambda) \cdot \log(|\mathbb{F}|)$. The verifier then sends k values, and the prover responds by sending $2k$ evaluations, requiring a message of length $2k \cdot \log(|\mathbb{F}|)$. Finally, the prover decommits to one of the functions as requested by the verifier, requiring a message of length $2^d \cdot \text{poly}(\lambda) \cdot \log(|\mathbb{F}|)$.

Overall, taking into consideration also the repetitions, the total communication complexity is:

$$\text{Commitment: } 2^d \cdot \text{poly}(\lambda) \cdot \log(|\mathbb{F}|), \text{ Opening: } O\big((2^d \cdot \text{poly}(\lambda) + k) \cdot \log(|\mathbb{F}|)\big).$$

Remark 4.3 (Amplification)**.** *As shown above, the protocol described in Fig. 1 has a binding error of* 0.6*. To reduce this error to $\delta > 0$, we repeat each of the commitment and evaluations phases $O(\log(1/\delta))$ times.*

For the binding property, since for each repetition the probability that P can convince V of a wrong opening is at most 0.6*, and the secret sharings are independent, we obtain the desired binding bound.*

As for the evaluation-hiding property, by repeating the protocol sequentially, we obtain a round complexity of $O(\log(1/\delta))$ against a malicious verifier.

In the setting of a semi-malicious verifier (whose queries are chosen independently of the commitment) we can repeat the protocol in parallel rather than sequentially. In this case, the evaluation-hiding property continues to hold by a standard hybrid argument. Hence, we obtain constant round complexity against a semi-malicious verifier.

5 Succinct Polynomial Commitment

In this section, we prove Theorem 1.5 by constructing a *succinct* polynomial commitment scheme for strings of length m. The commitment communication complexity is only slightly more than m field elements and the evaluation proof is sublinear in m.

To prove Theorem 1.5 we construct a PCS that uses a base PCS as a blackbox. Theorem 1.5 then follows immediately from the following lemma by combining it with the base PCS given in Lemma 4.1 and applying similar techniques to Lemma 2.24 for achieving a private-coin PCS.

Lemma 5.1. *Assume there exist a polynomial commitment for d-variate multilinears over a finite field $\mathbb{F}$, such that $|\mathbb{F}| = \Omega\Big(\frac{1}{\delta} \cdot \big(m^{2/3} + \log(1/\delta) \cdot \log(m)\big)\Big)$, with binding error $\frac{\delta}{5}$ and communication complexity $C_{com}(2^d), C_{open}(2^d, k)$ where k is the number of openings.*

Then, there exist a polynomial commitment scheme with respect to a semi-malicious verifier for any string of length m with binding error δ and communication complexity at most

$$m + m^{2/3} \cdot \text{poly}(\lambda, \log(1/\delta))$$

field elements for the commitment and

$$O(m^{2/3} \cdot \log(1/\delta)) + k \cdot \big(C_{com}(m^{1/3}) + C_{open}(m^{1/3}, 2)\big) + \text{poly}(\lambda)$$

field elements for the opening, and a constant-round complexity, where k is the number of points to be opened and λ is a security parameter.

5.1 Proof of Lemma 5.1

Let $(C_{base}, P_{base}, V_{base})$ be a polynomial commitment scheme and let Com be a standard bit-commitment scheme with an additive overhead[5] (see Proposition 2.12). We set both commitments to have a binding error $\frac{\delta}{5}$. The succinct polynomial commitment scheme and the opening protocol that establish Lemma 5.1 are presented in Fig. 2 and Fig. 3, respectively. We proceed to the analysis.

Correctness. Let $s \in \{0,1\}^m$ be a string, let $z_1, \ldots, z_k \in \mathbb{F}^d$ be points. For every $i \in [k]$, let $\alpha_i = \hat{s}(z_i)$, where $\hat{s}$ denotes the multilinear extension of s (see Definition 2.6), padded with zeros to have length that is a power of 2.

If the prover is honest, then the commitment and decommitment are valid, and the test in Step 7 of the evaluation protocol passes.

Let $V \in \mathbb{F}^{2^{1+d_r} \times 2^{1+d_c}}$ be the matrix obtained by arranging the values $v(i,j)$ (see Step 1) in a matrix in the natural way. Thus, V has the form

$$V = \begin{pmatrix} S & 0 & R & 0 \\ 0 & 0 & 0 & 0 \\ (R')^T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

where S, R, and R' are defined in Step 2 of Fig. 2 and the zeros represent all-zero matrices of the appropriate sizes.

The matrix A (from Step 2 of Fig. 2) can be obtained from V (by removing the padding blocks) and vice versa. These adjustments yield the equality in Step 8, details follow.

From the definition of h and Definition 2.6, in an honest execution we have

$$\begin{aligned} h(b_2, x) &= \hat{v}(\rho, b_2, x) \\ &= \sum_{i \in \{0,1\}^{1+d_r}} \sum_{j \in \{0,1\}^{1+d_c}} eq(i,\rho) \cdot eq(j,x) \cdot v(i,j) \\ &= \sum_i eq(i,\rho) \cdot v(i,x). \end{aligned}$$

Equivalently, in vector notation, h defined over $\{0,1\}^{1+dc}$ can be written as,

$$h = eq_\rho^T \cdot V,$$

where $eq_\rho \in \mathbb{F}^{2^{1+d_r}}$ denotes the vector $eq_\rho(b) = eq(\rho, b)$ for every $b \in \{0,1\}^{1+d_r}$ (where we view b simultaneously as a bit vector and an integer in the natural way). In Step 8 only the nonzero columns of V are encoded, i.e. the (inherently) zero coordinates of h are omitted. Thus

$$E_h = G_E \left(eq_\rho^T \cdot \begin{pmatrix} S & R \\ 0 & 0 \\ (R')^T & 0 \\ 0 & 0 \end{pmatrix} \right)^T = G_E \left(\begin{pmatrix} S^T & 0 & R' & 0 \\ R^T & 0 & 0 & 0 \end{pmatrix} \cdot eq_\rho \right).$$

[5] Our assumption that a basic PCS exists implies also the existence of a standard commitment scheme which in turn implies the existence of a one-way function. A commitment with additive overhead can then be obtained using Proposition 2.12.

Succinct Polynomial Commitment Scheme

Ingredients:

- A λ-zero knowledge code (see Definition 2.18) $E : \mathbb{F}^k \to \mathbb{F}^{\ell(k)}$ with blocklength $\ell(k)$.
- A standard bit-commitment scheme Com with additive overhead (as in Proposition 2.12).

Commitment: $C(s, 1^\lambda)$, where $s \in \{0,1\}^m$:

1. Let $m_r, m_c \in [m]$ such that $m_r \cdot m_c = m$.[a] View the input s as a matrix $S \in \mathbb{F}^{m_r \times m_c}$ in the natural way. That is, by setting
$$S[i,j] = s_{(i-1)\cdot m_c + j}$$
for all $i \in [m_r]$ and $j \in [m_c]$.
2. (Random Padding:) Sample uniformly at random a matrix $R \in \mathbb{F}^{m_r \times q}$ and a vector $R' \in \mathbb{F}^{m_c}$. Construct the extended matrix $A \in \mathbb{F}^{(m_r+1)\times(m_c+q)}$ as $A = \begin{bmatrix} S & R \\ R'^T & \mathbf{0} \end{bmatrix}$.
3. (Encode Rows:) Let $G_E \in \mathbb{F}^{\ell(m_c+q)\times(m_c+q)}$ denote the generator matrix of the code E, for messages of length $m_c + q$. Compute
$$\tilde{A} = A \cdot (G_E)^T.$$
(In other words, use E to encode the rows of A.)
4. (Commit to Columns:) For each $j \in [\ell(m_c + q)]$, compute
$$(com_j, dec_j) \leftarrow Com\left(\tilde{A}^{(j)}\right),$$
where $\tilde{A}^{(j)}$ denotes the j-th column of $\tilde{A}$.
5. Output:
$$com = \left(com_1, \ldots, com_{\ell(m_c+q)}\right), \quad dec = \left(\tilde{A}^{(1)}, dec_1, \ldots, \tilde{A}^{(\ell(m_c+q))}, dec_{\ell(m_c+q)}\right).$$

[a] We assume such a factorization exists; otherwise we can pad s with at most $\sqrt{m}$ zeroes.

Figure 2: Succinct PCS

Succinct Polynomial Evaluation Protocol

Common Input: commitment $com = \left(com_1, \ldots, com_{\ell(m_c+q)}\right)$, points $z_1, \ldots, z_k \in \mathbb{F}^d$, evaluations $\alpha_1, \ldots, \alpha_k \in \mathbb{F}$ and security parameter 1^λ.
Prover's Additional Input: decommitment $dec = \left(\tilde{A}^{(1)}, dec_1, \ldots, \tilde{A}^{\ell(m_c+q)}, dec_{\ell(m_c+q)}\right)$.

1. Let $d_r = \lceil \log(m_r) \rceil$ and $d_c = \lceil \log(m_c) \rceil$. We extend S, R, R' into functions as follows:
   - Let $\bar{S} : \{0,1\}^{d_r} \times \{0,1\}^{d_c} \to \mathbb{F}$ such that $\bar{S}(i,j) = S_{i,j}$ if i, j (viewed as integers) are within the index range of S, and $\bar{S}(i,j) = 0$ otherwise.
   - Let $\bar{R} : \{0,1\}^{d_r} \times \{0,1\}^{d_c} \to \mathbb{F}$ (resp. $\bar{R}' : \{0,1\}^{d_r} \times \{0,1\}^{d_c} \to \mathbb{F}$) as $R_{i,j}$ (resp. $(R'^T)_{i,j}$) , if (i,j) is within the index range of R (resp. R'^T) and 0 otherwise.
   - Define $v : \{0,1\}^{1+d_r+1+d_c} \to \mathbb{F}$ as
   $$v(b_1, i, b_2, j) = (1-b_1) \cdot (1-b_2) \cdot \bar{S}(i,j) + (1-b_1) \cdot b_2 \cdot \bar{R}(i,j) + b_1 \cdot (1-b_2) \cdot \bar{R}'(i,j).$$
   - For every $i \in [k]$, define the function $g_i : \{0,1\}^{1+d_r} \to \mathbb{F}$ as $g_i(b_1, x) = \hat{v}(b_1, x, 0, z_i^{(r)})$, where $z_i^{(r)} \in \mathbb{F}^{d_c}$ are the last d_c bits of z_i and $\hat{v}$ is the multilinear extension of v.
2. For every $i \in [k]$, the prover P computes $(c_{base}^{(i)}, d_{base}^{(i)}) \leftarrow C_{base}(g_i)$ and sends $c_{base}^{(i)}$ to V.
3. V chooses a random evaluation point $\rho \leftarrow \mathbb{F}^{1+d_r} \backslash \{0\} \times \mathbb{F}^{d_r}$, and sends it to P.
4. P computes and sends $h : \{0,1\}^{1+d_c} \to \mathbb{F}$ defined as $h(b_2, x) = \hat{v}(\rho, b_2, x)$ to V.[a]
5. V chooses $j_1, \ldots j_q \in [\ell(m_c + q)]$ at random and sends them to P.
6. For every $t \in [q]$, the prover P sends $(\tilde{A}^{(j_t)}, dec_{j_t})$ to V.
7. For every $t \in [q]$, the verifier V checks that dec_{j_t} is a valid decommitment for $\tilde{A}^{(j_t)}$.
8. Denote by $E_h = E\big((h(i))_{i \in \{1,\ldots,m_c, 2^{d_c}+1,\ldots,d_c+q\}}\big)$ i.e., the encoding of the non-zero padded entries of h, where i denotes the binary encoding of the corresponding integer.
   The verifier then checks that
   $$\widehat{E}_h(j_t) = \sum_{i \in [m_r]} eq(i, \rho) \cdot \tilde{A}_i^{(j_t)} + eq(2^{d_r}, \rho) \cdot \tilde{A}_{m_r+1}^{(j_t)}.$$
   where the integer argument of eq (see Definition 2.6) is understood as its binary encoding and $\tilde{A}_i^{(j_t)}$is the i-th entry of $\tilde{A}^{(j_t)}$.
9. For every $i \in [k]$, the verifier V and prover P emulate the base PCS evaluation protocol (P_{base}, V_{base}) with respect to: (1) the commitment $c_{base}^{(i)}$, (2) the points $0z_i^{(\ell)}$ and ρ, (3) the claimed evaluations α_i and $h(0, z_i^{(r)})$, which serve as common input, and with $d_{base}^{(i)}$ as the prover's auxiliary input. If all tests pass (Step 8 and Step 9) then V accepts. Otherwise it rejects.

[a] Here and throughout the analysis, we use $\hat{v}$ on three inputs instead of four; this is merely a different parsing, and $\rho \in \mathbb{F}^{1+d_r}$ corresponds to the size of two inputs.

Figure 3: Succinct Polynomial Evaluation Protocol

By Definition 2.6 we have

$$\widehat{E}_h(j_t) = eq_{j_t}^T \cdot E_h,$$

where $eq_{j_t} \in \mathbb{F}^{\ell(m_c+q)}$ denotes the vector $eq_{j_t}(b) = eq(j_t, b)$ for every $b \in \{0,1\}^{\log(\ell(m_c+q))}$. Now since, some of the columns of the resulting matrix $G_E \cdot \begin{pmatrix} S^T & 0 & R' & 0 \\ (R)^T & 0 & 0 & 0 \end{pmatrix}$ are inherently zero (due to the zero columns), we obtain:

$$\widehat{E}_h(j_t) = eq_{j_t}^T \cdot G_E \left(\begin{pmatrix} S^T & R' \\ (R)^T & 0 \end{pmatrix} \cdot \bar{eq}_\rho \right),$$

where $\bar{eq}_\rho \in \mathbb{F}^{m_r+1}$ denotes the entries of eq_ρ after omitting the zeros. That is,

$$\bar{eq}_\rho = \begin{pmatrix} eq(0,\rho) \\ \vdots \\ eq(m_r-1,\rho) \\ eq(2^{d_r},\rho). \end{pmatrix}$$

Since $A^T = \begin{pmatrix} S^T & R' \\ R^T & 0 \end{pmatrix}$ and $j_t \in \{0,1\}^{1+d_c}$ it follows from the definition of $eq(\cdot,\cdot)$ that,

$$\begin{aligned} \widehat{E}_h(j_t) &= eq_{j_t}^T \cdot G_E \cdot A^T \cdot \bar{eq}_\rho \\ &= (\tilde{A}^{(j_t)})^T \cdot \bar{eq}_\rho \\ &= \sum_{i \in [m_r]} eq(i,\rho) \cdot \tilde{A}_i^{(j_t)} + eq(2^{d_r},\rho) \cdot \tilde{A}_{m_r+1}^{(j_t)}, \end{aligned}$$

where $\tilde{A}^{(j_t)}$ is the j_t-th column of $\tilde{A} = A \cdot (G_E)^T$ as defined in Step 4 of Fig. 2. Hence, the check in Step 8 passes.

For Step 9, we have:

$$\begin{aligned} \hat{g}_i(0, z_i^{(\ell)}) &= \hat{v}(0, z_i^{(\ell)}, 0, z_i^{(r)}) \\ &= \sum_{x \in \{0,1\}^{1+d_r+1+d_c}} eq(b, 0, z_i^{(\ell)}, 0, z_i^{(r)}) \cdot v(x) \\ &= \sum_{\substack{b_1 \in \{0,1\}, x_1 \in \{0,1\}^{d_r} \\ b_2 \in \{0,1\}, x_2 \in \{0,1\}^{d_c}}} eq(b_1, 0) \cdot eq(x_1, z_i^{(\ell)}) \cdot eq(b_2, 0) \cdot eq(x_2, z_i^{(r)}) \cdot v(b_1, x_1, b_2, x_2) \\ &= \sum_{x_1 \in \{0,1\}^{d_r}} \sum_{x_2 \in \{0,1\}^{d_c}} eq(x_1, z_i^{(\ell)}) \cdot eq(x_2, z_i^{(r)}) \cdot v(0, x_1, 0, x_2) \\ &= \sum_{x_1 \in \{0,1\}^{d_r}} \sum_{x_2 \in \{0,1\}^{d_c}} eq(x_1, z_i^{(\ell)}) \cdot eq(x_2, z_i^{(r)}) \cdot \bar{S}(x_1, x_2) \\ &\overset{(1)}{=} \sum_{x_1 \in [m_r]} \sum_{x_2 \in [m_c]} eq(x_1, z_i^{(\ell)}) \cdot eq(x_2, z_i^{(r)}) \cdot s_{(x_1-1)m_c + x_2} \\ &= \hat{s}(z_i) = \alpha_i, \end{aligned}$$

where (1) follows from the definition of $\bar{S}$ (Step 1 in Fig. 3) and S (Step 1 in Fig. 2). Also, by definition, $\hat{h}(0, z_i^{(r)}) = \hat{v}(\rho, 0, z_i^{(r)}) = \hat{g}_i(\rho)$. Hence, by the completeness of the base polynomial commitment protocol, the check in Step 9 passes, and the verifier accepts.

Binding. Note that in Step 3 of the evaluation proof protocol, the verifier samples ρ from the set $\mathbb{F}^{1+d_r} \setminus 0\mathbb{F}^{d_r}$. In the binding proof, however, we consider a variant of the protocol in which the verifier instead samples ρ uniformly from $\mathbb{F}^{1+d_r}$. This is sufficient since, by Proposition 2.2, the statistical distance between the two distributions is $\frac{1}{|\mathbb{F}|}$. Therefore, any adversary that breaks binding with probability ϵ in the real protocol would also succeed in the modified protocol with probability at least $\epsilon - \frac{1}{|\mathbb{F}|}$.

Let com^* be a polynomial commitment generated by a possibly malicious prover. Recall that com^* is supposed to consist of a set of valid commitments $com^* = (com_1, \ldots, com_{\ell(m_c+q)})$. By the binding property of the commitment scheme[6] (see Definition 2.8) there exists at most one value m_i that can pass the verifier's decommitment check for com_i. Let $m_i \in \mathbb{F}^{m_r+1}$ be the unique message in case it exists, and set $m_i = \mathbf{0}$ otherwise (i.e., if no such mssage exists).

Let $e = \frac{\ell(m_c+q)\cdot d}{4}$ (where d is the relative distance of the code) be a proximity parameter. Define $B : \{0,1\}^{1+d_r} \to \mathbb{F}^{\ell(m_c+q)}$ by,

$$B(j) = \begin{cases} \big(m_{1,j+1}, \ldots, m_{\ell(m_c+q),j+1}\big), & j \in \{0, \ldots m_r - 1\}, \\ \big(m_{1,m_r+1}, \ldots, m_{\ell(m_c+q),m_r+1}\big), & j = 2^{d_r}, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

where $m_{i,j}$ denotes the j-th entry of m_i, and j is interpreted as a binary string when used as input to B, and as an integer when used as an index.

If the Hamming distance $\Delta\left(\big(B(j)\big)_{j=0}^{2^{d_r+1}-1}, E^{2^{d_r+1}}\right) > e$, by Theorem 2.16, we obtain:

$$\Pr_{\rho \in \mathbb{F}^{d_r+1}}\left[\Delta\left(\hat{B}(\rho), E\right) \leq e\right] \leq 2 \cdot (d_r + 1) \cdot \frac{e+1}{|\mathbb{F}|}.$$

Assume that the event in the above equation does not hold — that is, $\Delta(\hat{B}(\rho), E) > e$. Let h^* be the function sent by P^* in Step 4. Then there are more than e columns of $\hat{B}(\rho)$ that disagree with $E(h^*)$. The probability that the verifier does not select any of these columns when opening q columns in Step 5, and therefore fails to detect the error and reject, is at most

$$\binom{\ell(m_c+q)-e}{q} \Big/ \binom{\ell(m_c+q)}{q} \leq \left(1 - \frac{e}{\ell(m_c+q)}\right)^q.$$

Overall we get that in case $\Delta\left(\big(B(j)\big)_{j=0}^{2^{d_r+1}-1}, E^{2^{d_r+1}}\right) > e$ the verifier rejects with all but $2 \cdot (d_r + 1) \cdot \frac{e+1}{|\mathbb{F}|} + (1 - \frac{e}{\ell(m_c+q)})^q$ probability.

Thus, we may assume that $\Delta\left(\big(B(j)\big)_{j=0}^{2^{d_r+1}-1}, E^{2^{d_r+1}}\right) \leq e$. For every $j \in [2^{d_r+1}]$ define $s_j \in \mathbb{F}^{\ell(m_c+q)}$ to be the message whose codeword under E is closest to $B(j)$, i.e., $s_j = \arg\min_S \Delta\big(E(S), B(j)\big)$.

Define $v : \mathbb{F}^{1+d_r+1+d_c} \to \mathbb{F}$,

$$v(i, b_2, x) = \begin{cases} s_{i,x} & \text{if } x < m_c \\ s_{i,x-2^{d_c}} & \text{if } 2^{d_c} \leq x < 2^{d_c+q} \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

[6] Since we are using a commitment scheme in the CRS model, the commitment is binding with probability at least $1 - 2^{-\lambda}$ over the choice of the common reference string.

where $s_{i,x}$ denotes the x-th coordinate of s_i. We treat i and x as binary strings when given as inputs to v, and as integers when used as indices.

Finally, define $s \in \mathbb{F}^{m_r \cdot m_c} = \mathbb{F}^m$ by

$$s_{m_c \cdot (x_1 - 1) + x_2} = v(0, x_1, 0, x_2),$$

where $x_1 \in [m_r]$ and $x_2 \in [m_c]$ are interpreted as binary strings when used as inputs to v, and s_i denoted the i-th value of s.

Let $z_1, \ldots, z_k \in (\mathbb{F}^d)^k$ and $\alpha_1, ..., \alpha_k \in \mathbb{F}^k$ such that there exists $j^* \in [k]$ such that $\hat{s}(z_{j^*}) \neq \alpha_{j^*}$ (where $\hat{s}$ is defined over s padded with zeroes), and let P^* be a malicious prover strategy. We assume wlog that P^* is deterministic. Denote by g_i^* and h^* the functions committed to and sent by P^* in Steps 2 and 4, respectively.

Consider the following cheating strategies:

1. P^* sends in Step 2 a commitment to $g_{j^*}^*(b, x) = \hat{v}(b, x, 0, z_{j^*}^{(r)})$ (when v is as defined in Eq. (3)). In this case,
$$\hat{g}_{j^*}^*(0, z_{j^*}^{(\ell)}) \neq \alpha_{j^*},$$
since
$$\hat{g}_{j^*}^*(0, z_{j^*}^{(\ell)}) = \hat{v}(0, z_{j^*}^{(\ell)}, 0, z_{j^*}^{(r)}) = \hat{s}(z_{j^*}) \neq \alpha_{j^*}.$$
By the binding property of the base polynomial commitment, the verifier rejects in Step 9 with probability at least $1 - \frac{\delta}{5}$.

2. P^* sends in Step 2 a commitment to $g_{j^*}^*(b, x) \neq \hat{v}(b, x, 0, z_{j^*}^{(r)})$ and in Step 4 it sends $h^*(b_2, x) = \hat{v}(\rho, b_2, x)$, where ρ is chosen by V in Step 3:
   - By the Schwartz-Zippel lemma, since $g_{j^*}^*(x) \neq v(b, x, 0, z_{j^*}^{(r)})$,
   $$\Pr\left[g_{j^*}^*(\rho) = v(\rho, 0, z_{j^*}^{(r)})\right] \leq \frac{d_r + 1}{|\mathbb{F}|}.$$
   - Otherwise, if $g_{j^*}^*(\rho) \neq v(\rho, 0, z_{j^*}^{(r)})$, then from the binding property of the base polynomial commitment, with probability at least $1 - \frac{\delta}{5}$ the verifier rejects in Step 9.

   Overall, in this case V rejects with probability $1 - \frac{d_r+1}{|\mathbb{F}|} - \frac{\delta}{5}$.

3. P^* sends in Step 2 a commitment to $g_{j^*}^*(b, x) \neq \hat{v}(b, x, 0, z_{j^*}^{(r)})$ and in Step 4 it sends $h^*(b_2, x) \neq v(\rho, b_2, x)$. Since $v \neq h$, the relative distance between their encoding is at least d. However, since v is defined as the closest message consistent with the commitment, and the commitment is at most e far from any encoding, the encoding of v is at most e far from the commitment. Therefore, the relative distance between the encoding of h and the commitment is at least $d - \frac{e}{\ell(m_c+q)}$. Thus, the probability that the verifier accepts in Step 8 is exactly the probability that none of the q checks reveal a column on which the two encodings disagree, $\left(1 - \left(d - \frac{e}{\ell(m_c+q)}\right)\right)^q$.

Overall we get that V accepts with probability at most

$$\frac{2\delta}{5} + \left(1 - \frac{e}{\ell(m_c + q)}\right)^q + \left(1 - \left(d - \frac{e}{\ell(m_c+q)}\right)\right)^q + O\left(\frac{d_r \cdot e}{|\mathbb{F}|}\right).$$

Commitment Hiding. Recall that $C(F)$ consist of a set of standard commitments. Commitment hiding now follows from the hiding property of the commitment (together with a standard hybrid argument).

Evaluation Hiding. Recall that V is the honest verifier strategy. Let V_r denote V's strategy when it uses the fixed string r as its randomness.

Let $s_1, s_2 : \{0,1\}^d \to \mathbb{F}$ be a pair of strings, and let $z_1, \ldots, z_k \in \mathbb{F}^d$ be points on which $\hat{s}_1$ and $\hat{s}_2$ agree. We prove that

$$(C, P, V_r)(s_1)_k \approx_c (C, P, V_r)(s_2)_k.$$

Recall that (see Definition 3.1) $\Big(C, P, V_r\Big)(s)$ is defined as

$$\Big(\vec{z}, C(s), \hat{s}(\vec{z}), \{com_{base}(g_i)\}_{i\in[k]}, \rho, h, \textsf{check-h}, \{\textsf{Interact}\big(g_i, \big(0z_i^{(\ell)}, \rho\big), \big(\hat{s}(z_i), \hat{h}(0z_i^{(r)})\big)\big)\}_{i\in[k]}\Big)$$

where,

- $\vec{z} = (z_1, \ldots, z_k)$ are the k points chosen in advance by V_r, on which s_1 and s_2 agree.
- $C(s)$ is the polynomial commitment as defined in Fig. 2.
- $\textsf{check-h} = (j_t)_{t\in[q]}, \{\tilde{A}^{(j_t)}, dec_{j_t}\}_{t\in[q]}$, where $(j_t)_{t\in[q]}$ denotes the q openings chosen by V_r in Step 5, and $\{\tilde{A}^{(j_t)}, dec_{j_t}\}_{t\in[q]}$ are the data and decommitments sent by P in Step 6.
- $\textsf{Interact}\big(g_i, \big(0z_i^{(\ell)}, \rho\big), \big(\hat{s}(z_i), \hat{h}(0z_i^{(r)})\big)\big)$ is the base polynomial commitment scheme evaluation proof (P_{base}, V_{base}) relative to the commitment to g_i sent in Step 2, the points $(0z_i^{(\ell)}, \rho)$ and the claimed evaluations $(\hat{s}(z_i), \hat{h}(0z_i^{(r)}))$.

We will show that for every fixed choice r of the verifier, the distributions are computationally indistinguishable. Since this choice is made in advance, this suffices to prove the claim for every such V_r.

Claim 5.1.1. *Let $\vec{z}, \rho, \mathsf{j}$ be a fixed choice of V_r, denote:*

$$H_0 = \Big(\vec{z}, C(s^{(1)}), \hat{s}^{(1)}(\vec{z}), \{com_{base}(g_i^{(1)})\}_{i\in[k]}, \rho, h^{(1)}, \textit{check-h}, \quad \{\textsf{Interact}\big(g_i^{(1)}, \big(0z_i^{\ell}, \rho\big), \big(\hat{s}^{(1)}(z_i), \hat{h}^{(1)}(0z_i^{(r)})\big)\}_{i\in[k]}\Big)$$
$$H_1 = \Big(\vec{z}, C(s^{(2)}), \hat{s}^{(2)}(\vec{z}), \{com_{base}(g_i^{(2)})\}_{i\in[k]}, \rho, h^{(2)}, \textit{check-h}, \quad \{\textsf{Interact}\big(g_i^{(2)}, \big(0z_i^{\ell}, \rho\big), \big(\hat{s}^{(2)}(z_i), \hat{h}^{(2)}(0z_i^{(r)})\big)\}_{i\in[k]}\Big)$$

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

Proof. We first recall,

- $C(s)$ is the polynomial commitment as defined in Fig. 2 consisting of a set of standard commitments.
- $\textsf{check-h} = \mathsf{j}, \{\tilde{A}^{(j)}, dec_j\}_{j\in\mathsf{j}}$, where $\{\tilde{A}^{(j)}, dec_j\}_{j\in\mathsf{j}}$ are the data and the decommitment sent by P in Step 6.
- $\textsf{Interact}\big(g_i, \big(0z_i^{\ell}, \rho\big), \big(\hat{s}(z_i), \hat{h}(0z_i^{(r)})\big)\big)$ is the base PCS evaluation proof (P_{base}, V_{base}) relative to the commitment to g_i the points $(0z_i^{(\ell)}, \rho)$ and the claimed evaluation $(\hat{s}(z_i), \hat{h}(0z_i^{(r)}))$.

Recall

$$
\begin{aligned}
h^{(b)}(x) &= \hat{v}(\rho, x) \\
&= \sum_{i\in\{0,1\}^{1+d_r}} \sum_{j\in\{0,1\}^{1+d_c}} eq(\rho, i)\cdot eq(x, j)\cdot v(i, j) \\
&= \sum_{i\in\{0,1\}^{d_r}} \sum_{j\in\{0,1\}^{1+d_c}} eq(\rho, 0i)\cdot eq(x, j)\cdot v(0i, j) + \sum_{i\in\{0,1\}^{d_r}} \sum_{j\in\{0,1\}^{1+d_c}} eq(\rho, 1i)\cdot eq(x, j)\cdot v(1i, j) \\
&\overset{(1)}{=} \sum_{i\in\{0,1\}^{d_r}} \sum_{j\in\{0,1\}^{1+d_c}} eq(\rho, 0i)\cdot eq(x, j)\cdot v(0i, j) + \sum_{i\in\{0,1\}^{d_r}} \sum_{j\in\{0,1\}^{1+d_c}} eq(\rho, 1i)\cdot eq(x, j)\cdot \bar{R}'(i, j) \\
&\overset{(2)}{=} \sum_{i\in\{0,1\}^{d_r}} \sum_{j\in\{0,1\}^{1+d_c}} eq(\rho, 0i)\cdot eq(x, j)\cdot v(0i, j) + \sum_{j\in[m_c]} eq(\rho, 10^{d_r})\cdot eq(x, j)\cdot \bar{R}'(0^{d_r}, j)
\end{aligned}
$$

where (1) follows from the definition of v (see Step 1) and (2) from the definition of $\bar{R}'$.

Now since we chose $\rho \in \mathbb{F}^{1+d_r} \setminus 0\mathbb{F}^{d_r}$, we have $eq(\rho, 10^{d_r}) \neq 0$, and since $\bar{R}'(0^{d_r}, j) = R'_j$ is uniformly random for $j \in [m_c]$ (see Step 2 in Fig. 2), it follows that h is uniformly random. From the definition of $g^{(b)}$ for $b \in \{0, 1\}$ (Step 2) since $s^{(1)}$ and $s^{(2)}$ are identical over z we have that $g^{(1)}$ and $g^{(2)}$ are also identical over $0z_i^{(\ell)}$. Now, since h is distributed uniformly (as explained above) from Lemma 3.4 it follows that

$$
\Big(\vec{z}, \hat{s}^{(1)}(\vec{z}), \{com_{base}(g_i^{(1)})\}_{i\in[k]}, \rho, \hat{h}^{(1)}(0z_i^{(r)}), \{\mathsf{Interact}\big(g_i^{(1)}, \big(0z_i^{(\ell)}, \rho\big), \big(\hat{s}^{(1)}(z_i), \hat{h}^{(1)}(0z_i^{(r)})\big)\}_{i\in[k]}\Big)
$$

$$
\overset{c}{\approx}
$$

$$
\Big(\vec{z}, \hat{s}^{(2)}(\vec{z}), \{com_{base}(g_i^{(2)})\}_{i\in[k]}, \rho, \hat{h}^{(2)}(z^{(r)}), \{\mathsf{Interact}\big(g_i^{(2)}, \big(0z_i^{(\ell)}, \rho\big), \big(\hat{s}^{(2)}(z_i), \hat{h}^{(2)}(0z_i^{(r)})\big)\}_{i\in[k]}\Big)
$$

From the definition of zero knowledge code(see Definition 2.18) we obtain,

$$
\Big(\vec{z}, \hat{s}^{(1)}(\vec{z}), \{com_{base}(g_i^{(1)})\}_{i\in[k]}, \rho, h^{(1)}, \mathsf{j}, \{\tilde{A}_1^{(j)}\}_{j\in\mathsf{j}}, \{\mathsf{Interact}\big(g_i^{(1)}, \big(0z_i^{(\ell)}, \rho\big), \big(\hat{s}^{(1)}(z_i), \hat{h}^{(1)}(0z_i^{(r)})\big)\}_{i\in[k]}\Big)
$$

$$
\overset{c}{\approx}
$$

$$
\Big(\vec{z}, \hat{s}^{(2)}(\vec{z}), \{com_{base}(g_i^{(2)})\}_{i\in[k]}, \rho, h^{(2)}, \mathsf{j}, \{\tilde{A}_2^{(j)}\}_{j\in\mathsf{j}}, \{\mathsf{Interact}\big(g_i^{(2)}, \big(0z_i^{(\ell)}, \rho\big), \big(\hat{s}^{(2)}(z_i), \hat{h}^{(2)}(0z_i^{(r)})\big)\}_{i\in[k]}\Big),
$$

where $\tilde{A}_i^{(j)}$ is the j-th column of the matrix $\tilde{A}_i$, as defined in Step 4 of Fig. 2, given $s^{(i)}$.

Finally, by combining the above and applying Fact 2.5 to perform the commitment and decommitment on the columns in order to generate $\mathsf{check\text{-}h}$, and Fact 2.9 to add the commitments for the remaining columns, we obtain exactly the distributions H_0 and H_1, thus establishing $H_0 \overset{c}{\approx} H_1$. □

We conclude that

$$
(C, P, V_r)(s_1) \overset{c}{\approx} (C, P, V_r)(s_2).
$$

Communication Complexity. The commitment consists of $\ell(m_c+q))$ commitments for messages of length m_r+1. Thus, the overall size of the commitment is: $(m_r+\text{poly}(\lambda))\cdot(\ell(m_c+q)$.

For the opening protocol, the prover P first sends for every $i\in[k]$ a base PCS commitment for a multilinear polynomial on (d_r+1)-variables, which has length $C_{com}(2^{d_r+1})$. Then, V chooses ρ and P sends the description of h, which has length $2^{d_c+1}\cdot\log(|\mathbb{F}|)$. The prover P then opens q columns by sending the decommitments of total size $q\cdot(m_r+\text{poly}(\lambda))$. For the last step we apply k openings of length $C_{open}(2^{d_r+1},2)$. Overall we get:

- Commitment: $\big(m_r+\text{poly}(\lambda)\cdot\log(1/\delta)\big)\cdot\ell(m_c+q)$.
- Opening: $2^{d_c+1}+m_r\cdot q+k\cdot\big(C_{com}(2^{d_r+1})+C_{open}(2d_r+1,2)\big)+\text{poly}(\lambda)$.

Remark 5.2. *By choosing $m_c=m^{2/3}$ and $m_r=m^{1/3}$ and utilizing the linear code of Reed-Solomon (Definition 2.17) with $\ell(k)=(1+m^{-1/3})k$ and fixing $q=O(\frac{1+m^{-1/3}}{m^{-1/3}})\log(\frac{1}{\delta})$ we obtain Lemma 5.1*

6 Succinct Zero-Knowledge Proofs

In this section we prove Theorem 1.1 by constructing a succinct zero-knowledge proof for bounded-depth NP relations.

We first consider semi-malicious zero-knowledge for arithmetic circuits over a sufficiently large finite field. Then, we use Lemma 2.24 to compile it into a *malicious* zero knowledge proof for arithmetic circuits. Theorem 1.1, which considers Boolean circuits, follows easily by packing bits of the witness and emulating the Boolean circuit using an arithmetic one.

Lemma 6.1. *Assume that one-way functions exist and let $\delta>0$ be a parameter. Let R be an* NP *relation with input size n and witness size m, that is computable by a (non-uniform) arithmetic circuit family C of size $S=S(n)$ and depth $D=D(n)$ over a finite field $\mathbb{F}$ s.t. $|\mathbb{F}|=\Omega\Big(\frac{1}{\delta}\cdot D\log(S)\cdot\log(m)\cdot\log(1/\delta)\Big)$, and assume that $n\le\text{poly}(m)$.*

Then R has a zero-knowledge proof with perfect completeness, and soundness error δ in which the verifier, prover and simulator all only make a black-box use of the one-way function. The communication complexity is

$$m+m^{2/3}\cdot\text{poly}\big(\lambda,\log(1/\delta)\big)+\text{poly}\big(\log(1/\delta),\lambda,D,\log(S),\log(|\mathbb{F}|)\big)$$

field elements where λ is the security parameter. The prover and verifier run in polynomial time, the protocol is private-coin and the number of rounds is $\text{poly}(D,\log(S),\log(1/\delta),\lambda,\log(\mathbb{F}))$.

Lemma 6.1 follows immediately from the next lemma combined with the succinct PCS from Theorem 1.5 and the semi-malicious to fully-malicious transformation of Lemma 2.24.

Lemma 6.2. *Assume that one-way functions exist, and assume there exists a polynomial commitment with respect to a semi-malicious verifier for strings of length m with binding δ' and communication complexity $C_{com}(m),C_{open}(m,k)$ for k openings.*

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 $n\le\text{poly}(m)$. Then the relation R has a zero-knowledge proof with respect to a semi-malicious verifier, with perfect*

completeness, and soundness error $\frac{3\delta}{4} + O\left(\frac{D\log(S)+\log(m+\log(1/\delta))\cdot\log(1/\delta)}{\mathbb{F}}\right)$*, in which the verifier, prover and simulator all only make a black-box use of the one-way function. The communication complexity is*

$$C_{com}(m+\log(1/\delta)) + C_{open}(m+\log(1/\delta), \log(1/\delta)) + \log(1/\delta)\cdot \mathrm{poly}(\lambda, D, \log(S), \log(|\mathbb{F}|)).$$

where λ *is the security parameter and* $\mathbb{F}$ *a finite field. The prover and verifier run in polynomial time and the number of rounds is* $\mathrm{poly}(D, \log(S))$.

Section Organization. In Section 6.1 we show that a short random padding of a string suffices to make its multilinear evaluations at a set of points look random. Using this fact, in Section 6.2 we prove Lemma 6.2. Finally, in Section 6.3 we show how to derive Theorem 1.1 from Lemma 6.1.

6.1 Random Padding of Multilinear Extensions

Let $d, \lambda \in \mathbb{N}$ be parameters. We use $y_i \in \{0,1\}^d$ to denote the i-th binary vector in lexicographic order.

For a sequence of points $\vec{z} = (z_1, \ldots, z_\lambda) \in (\mathbb{F}^d)^\lambda$, let $M_{\vec{z}} \in \mathbb{F}^{\lambda\times\lambda}$ be a matrix defined as:

$$M_{\vec{z}}[i,j] = eq(y_j, z_i),$$

for all $i, j \in [\lambda]$.

Definition 6.3. *We say that* $\vec{z}$ *is* good *if* $M_{\vec{z}}$ *has full rank.*

Lemma 6.4. *With probability* $1 - \frac{\lambda\cdot d}{\mathbb{F}}$ *over* $\vec{z} \in (\mathbb{F}^d)^\lambda$*, it holds that* $M_{\vec{z}}$ *has full rank.*

Proof. Each entry of $M_{\vec{z}}$ is a polynomial of total degree at most d (since the (i,j)-th entry is $eq(y_j, z_i)$). Thus, that the determinant of $M_{\vec{z}}$ can be expressed as the polynomial:

$$\det(M_{\vec{z}}) = \sum_{\sigma\in S_\lambda} \mathrm{sign}(\sigma)\cdot\prod_{i=1}^{\lambda} eq(y_{\sigma(i)}, z_i).$$

Each term is a product of λ polynomials, each with total degree at most d, so the total degree of the polynomial $\det(M(\vec{z}))$ is at most $d\lambda$.

Next, we show that this polynomial is not identically zero. Indeed, taking $z_j = y_j$ for every $j \in [\lambda]$, the matrix M becomes the identity matrix, whose determinant is equal to 1. Hence, $\det(M(\vec{z}))$ is not identically zero.

Since $\det(M(\vec{z}))$ is a non-zero multivariate polynomial of total degree at most $\lambda\cdot d$, and each z_i is chosen uniformly from $\mathbb{F}^d$, by the Schwartz-Zippel lemma we have:

$$\Pr_{\vec{z}\in(\mathbb{F}^d)^\lambda}[\det(M_{\vec{z}}) = 0] \le \frac{\lambda\cdot d}{|\mathbb{F}|},$$

and the lemma follows □

Lemma 6.5. *Let* $x \in \{0,1\}^m$*, and define* $x' = r\|x$*, where* $x' \in \{0,1\}^{2^d}$ *and* $r \in \mathbb{F}^\lambda$ *is chosen uniformly at random. Let* f *be the multilinear extension of* x' *and let* $\vec{z} \in (\mathbb{F}^d)^\lambda$ *be a good sequence of points (as per Definition 6.3). Then* $f(z) = (f(z_1), \ldots, f(z_\lambda))$ *is distributed identically to the uniform distribution over* $\mathbb{F}^\lambda$.

Proof. Let $\vec{z} \in (\mathbb{F}^d)^\lambda$ be the good set of points and recall that this means that $M_{\vec{z}}$ has full rank.

We now observe that for each $j \in [\lambda]$,

$$f(z_j) = \sum_{i=1}^{2^d} eq(y_i, z_j) \cdot x'_i = \sum_{i=\lambda+1}^{\lambda+m} eq(y_i, z_j) \cdot x_{i-\lambda} + \sum_{i=1}^{\lambda} eq(y_i, z_j) \cdot r_i = \sum_{i=\lambda+1}^{\lambda+m} eq(y_i, z_j) \cdot x_{i-\lambda} + \langle M_{\vec{z}}^{(j)}, r \rangle, \tag{4}$$

where $M_{\vec{z}}^{(j)}$ denotes the j-th row of $M_{\vec{z}}$. Let $E_{\vec{z}} \in \mathbb{F}^{\lambda \times d}$ be a matrix whose j-th row is equal to $\big(eq(y_{\lambda+1}, z_j), \ldots eq(y_d, z_j)\big)$. Thus, we can use Eq. (4) to write $f(\vec{z})$ as:

$$f(\vec{z}) = E_{\vec{z}} \cdot x + M_{\vec{z}} \cdot r.$$

Since $M(\vec{z})$ has full rank, we have that $M(\vec{z}) \cdot r$ is uniformly distributed over $\mathbb{F}^\lambda$ and therefore, so is $E_{\vec{z}} \cdot x + M_{\vec{z}} \cdot r$.

□

6.2 Proof of Lemma 6.2

Let R be an NP-relation, let $(C_{pcs}, P_{pcs}, V_{pcs})$ be a polynomial commitment scheme and Com be a standard bit-commitment scheme with additive overhead (see Proposition 2.12).[7] We set both commitments to have a binding error of $\frac{\delta}{4}$.

The interactive protocol for R that establishes Lemma 6.2 is presented in Fig. 4.

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

6.2.1 Completeness

Let $(x, w) \in R$. If (P, V) follow the protocol, then for each iteration $t \in [\ell]$ the GKR interactive phase is executed correctly, and the input to each player j is

$$\big(coins_V^{(t)}, b_j^{(t)}, z_t, \hat{\omega}(z_t), \langle C \rangle\big),$$

where z_t can be easily derived from $coins_V^{(t)}$.

The MPC protocol performs two checks:

- V_{post} accepts: The input to V_{post} in Step 3(b)ii is

$$\big(coins_V^{(t)}, \bigoplus_{j \in [k]} b_j^{(t)}\big),$$

  where, by construction, $\bigoplus_{j\in[k]} b_j^{(t)} = \big(m_i^{(t)}\big)_{i\in[r]}$. Hence, the input is the transcript of a valid interaction between P_{GKR} and $V_{interactive}$. Therefore, V_{post}'s input in each iteration of Step 3 corresponds to a valid run of the GKR protocol, and V_{post} accepts.

- $\hat{\omega}(z_t)$ is derived correctly: $\hat{\omega}(z_t)$ can be correctly obtained from $\bigoplus_{j\in[k]} b_j^{(t)}$, i.e., from the transcript of the interactive phase. In a valid GKR execution, the prover outputs $\hat{\omega}(z_t)$, which is then derived from the transcript.

[7] For sake of readability we omit the CRS from the notation.

Succinct Zero Knowledge Protocol

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

1. P samples a uniformly random string $r \in \{0,1\}^\ell$, where $\ell \in \mathbb{N}$ is a parameter to be determined later. Denote by $\omega = r \| w$ the concatenation of r and w.
2. P computes $(c_{pcs}, d_{pcs}) \leftarrow C_{pcs}(\omega)$ and sends c_{pcs} to V.
3. In parallel, for $t = 1, \ldots, \ell$:
   (a) P and V emulate the interactive phase of the GKR protocol (see $(P, V_{interactive})$ in Theorem 2.21) on input (C_x, ω) (where C_x denotes the circuit that computes the relation R with x hardcoded, extended to accept ℓ additional input bits, which do not affect the output) as follows: in every round $i \in [r]$, the prover does not send the message $m_i^{(t)}$ directly, but rather generates an additive secret sharing of the message s.t $m_{i,1}^{(t)} \oplus \cdots \oplus m_{i,k}^{(t)} = m_i^{(t)}$, and sends to V commitments to $m_{i,1}^{(t)}, \ldots, m_{i,k}^{(t)}$. We denote the coins sent from V to P in the t-th repetition of this stage as $coins_V^{(t)}$.
   (b) i. P derives $z_t \in \mathbb{F}^d$, where $d = \log(|\omega|)$, and $\langle C \rangle$ (by Theorem 2.21 it can do so from Step 3a).
   ii. P executes ("in its head") the following k-party MPC protocol:
   - Player j input: $(coins_V^{(t)}, b_j^{(t)}, z_t, \hat{\omega}(z_t))$, where $b_j^{(t)} = (m_{i,j}^{(t)})_{i \in [r]}$.
   - Functionality: (1) verify that $\hat{\omega}(z_t)$ can be correctly derived from $\bigoplus_{j \in [k]} b_j^{(t)}$ and (2) that $V_{post}\Big(\big(coins_V^{(t)}, \bigoplus_{j \in [k]} b_j^{(t)}\big), \langle C \rangle, z_t\Big)$ accepts.

   iii. P sends commitments to the views of the k parties in the MPC protocol.
   (c) V randomly chooses a party $q_t \in [k]$ and sends it to P.
4. If $\vec{z} = (z_1, \ldots, z_\ell)$ is not good (see Definition 6.3) V accepts and halts.
5. For every $t \in [\ell]$ and $j \in [k] \backslash \{q_t\}$, the prover P decommits to everything related to j in iteration t, namely $(m_{i,j}^{(t)})_{i \in [r]}$, and the view of party j, in the t-th iteration.
6. V checks that for every $t \in [\ell]$: (1) all inputs of the parties are correct, (2) all their views are consistent (3) all parties properly followed the specification of the MPC protocol, and (4) all of the parties accepted. If any test fails then V rejects.
7. Finally, V and P emulate the PCS evaluation protocol (P_{pcs}, V_{pcs}) with respect to the commitment c_{pcs}, the points $(z_t)_{t \in [\ell]}$ and the claimed evaluations $\hat{\omega}(z_t)$ (that V received from the MPC). If V_{pcs} accepts then V accepts. Otherwise it rejects.

Figure 4: Succinct Zero-Knowledge Proof for R

Thus, both checks pass, and by the perfect completeness of the MPC protocol, the players accept. Since P follows the protocol, it can open all commitments correctly, so all checks in Step 6 pass. Additionally, in the (unlikely event) that z is not good, in Step 4 the verifier also accepts.

Finally, by the perfect completeness of the polynomial commitment scheme, the verifier accepts the openings.

6.2.2 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 first note that by Lemma 6.5, the probability that $\vec{z}$ is not good is at most $\frac{O(\log(m+\ell))\ell}{|\mathbb{F}|}$. Thus, we may assume that $\vec{z}$ is good (and in particular the verifier does not accept in Step 4).

Our commitment is defined with respect to a CRS and is binding with probability $1 - \frac{\delta}{4}$. We assume that it is indeed binding.

Consider the following possible behaviors of P^*:

1. It cheats in all the rounds of Step 3 (distributed GKR and MPC). This happens either when the behavior of one of the parties in the MPC protocol transcript, as defined by the commitment, does not follow the protocol specification. Or a pair of views is inconsistent, i.e., messages sent by one party are not received correctly by the other parties.

2. It produces an invalid decommitment in Step 5.

3. It runs in at least one round of Step 3 (the MPC) correctly and sends a valid decommitment to all the required views and messages, and then attempts to cheat in the evaluation proof interactive protocol of the PCS.

In the first case, if one of the parties in the MPC protocol deviates from the specification in a given iteration, the verifier V selects that party and rejects with probability $\frac{1}{k}$. Otherwise, if a pair of views is inconsistent, the inconsistent pair of parties is selected with probability $\frac{2}{k}$. Therefore, the overall probability that P^* misbehaves in all ℓ iterations of Step 3 while V does not reject is at least $\left(1 - \frac{1}{k}\right)^{\ell}$, since the verifier's choices in different iterations are independent.

In the second case, the verifier will reject when checking the decommitments in Step 6.

In the third case, assume that all commitments can be opened in exactly one way, and that P^* simulates the MPC protocol correctly in at least one iteration t^* on the inputs derived from the opening of the commitments, as defined in the protocol. Since $x \notin L_R$, for any w^* it holds that $(x, w^*) \notin R$, and therefore the circuit C_x does not accept w^*. Let w^* be the function that P^* committed to in Step 2, and let α_{t^*} be the value derived from the transcript of the interactive phase of the GKR protocol in the t^*-th iteration, as defined in Theorem 2.21 (in a correct run, $\alpha_{t^*} = \hat{w}^*(z_{t^*})$).

We distinguish two cases:

(i) $\alpha_{t^*} = \hat{w}^*(z_{t^*})$. By Theorem 2.21, this implies that V_{eval} accepts. For a false statement, this happens with probability at most $O\left(\frac{D \log S}{|\mathbb{F}|}\right)$.

(ii) $\alpha_t \neq \hat{w}^*(z_t^*)$. In this case, acceptance can only occur if the polynomial commitment opens inconsistently, by the binding property of the PCS, this happens with probability at most $\frac{\delta}{4}$.

Overall, the probability that V accepts is at most $\frac{\delta}{2} + (1 - \frac{1}{k})^{\ell} + O\left(\frac{D \log(S)}{|\mathbb{F}|}\right) + \frac{O(\log(m+\ell))\ell}{|\mathbb{F}|}$.

6.2.3 Zero-knowledge with respect to a semi-malicious verifier

Let V_r be the honest polynomial-time verifier with a fix string r as defined in Definition 2.23. We construct a simulator $S_r(x)$ (with respect to the fixed choice r for the honest verifier) in Fig. 5. We proceed to the analysis.

Proposition 6.6. *The ensembles $S_r(x)$ and $\{View_{V_r}^{P(w)}(x,\lambda)\}_{x\in L}$ are computationally indistinguishable.*

Proof. For $x \in L$, both $S_r(x)$ and $\{View_{V_r}^{P(w)}(x,\lambda)\}_{x\in L}$ are sequences of one of the following two forms, depending on z (see Step 4 of Fig. 4 and Step 3 of Fig. 5).

$$\Big(x, ref, C_{pcs}(\omega), \{\mathsf{distributed}_{\mathsf{GKR_t}}, q_t, \mathsf{open\text{-}post}_{GKR_t}\}_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))\Big),$$

$$\Big(x, ref, C_{pcs}(\omega), \{\mathsf{distributed}_{\mathsf{GKR_t}}, q_t\}_{t\in[\ell]}\Big).$$

- $\mathsf{distributed}_{\mathsf{GKR}t}$ is the t-th iteration of Step 3 :

$$\mathsf{distributed}_{\mathsf{GKR}t} = \Big(\mathsf{tr}_{\mathsf{GKR_t}}, \{com(view_i^{(t)})\}_{i\in[k]}\Big).$$

- $\mathsf{open\text{-}post}_{GKR_t}$ is the opening of the commitment for $T_{q_t} = [k]\backslash\{q_t\}$ in Step 5:

$$\mathsf{open\text{-}post}_{GKR_t} = \Big(\{dec_m^{(t)}(i), dec_v^{(t)}(i)\}_{i\in T_{q_t}}\Big).$$

- $\vec{z} = (z_1, \ldots, z_\ell) \in (\mathbb{F}^d)^\ell$ are the values from the distributed GKR execution that can be derived from $coins_{V_r} = (coins_{V_r}^{(1)}, \ldots, coins_{V_r}^{(\ell)})$ (see Item 2 in Theorem 2.21).

- $\hat{\omega}(\vec{z}) = (\hat{\omega}(z_1), \ldots, \hat{\omega}(z_\ell)) \in \mathbb{F}^\ell$ are the evaluations of $\hat{\omega}$ on these values, computed and checked in the MPC in Step 3(b)ii.

- $(P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))$ denotes the opening protocol of the polynomial commitment. Here, V_{pcs} is run by V_r, and P_{pcs} is run by either P or S.

We prove the following two claims:

Claim 6.6.1. *Let $coins_{V_r}$ and $(q_1, \ldots, q_\ell) \in [k]^\ell$ be a fixed randomness choice of the verifier s.t $\vec{z} = (z_1, \ldots, z_\ell) \in (\mathbb{F}^d)^\ell$ (uniquely derived from $coins_{V_r}$) is good (see Definition 6.3). Then, $H_0 \overset{c}{\approx} H_1$, where:*

$$H_0 = \Big(x, ref, C_{pcs}(\omega), \{\mathsf{distributed}_{\mathsf{GKR}}, q_t, \mathsf{open\text{-}post}_{GKR_t}\}_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{(\omega)}(\vec{z})\Big);$$
$$H_1 = \Big(x, ref, C_{pcs}(w^*), \{\mathsf{distributed}_{\mathsf{GKR}}, q_t, \mathsf{open\text{-}post}_{GKR_t}\}_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{(w^*)}(\vec{z})\Big).$$

Proof. We first show that the messages sent in the clear, together with the decommitted message, are computationally indistinguishable in the two cases: when V_r interacts with P, and when it interacts with the simulator. We then add the commitments to the remaining messages and conclude the proof. For simplicity of notation, we use T_t to denote the selected set, namely $T_t = \{1, \ldots, k\} \setminus q_t$.

The Simulator for V_r

Input: main input $x \in L$, $r = (coins_{V_r}, q_1^* \ldots, q_\ell^*)$ and security parameter 1^λ.

1. S computes $(c_{pcs}, d_{pcs}) \leftarrow C_{pcs}(w^*)$, where $w^* \in \{0,1\}^{m+\ell}$ is random and sends c_{pcs} to V_r.
2. In parallel, for $t = 1, \ldots \ell$:
   (a) S emulates with V_r the interactive phase of the GKR protocol (see $(P, V_{interactive})$ in Theorem 2.21) on input (C_x, w^*) as follows: In every round $i \in [r]$, the simulator S randomly chooses $\tilde{m}_{i,1}^{(t)}, \ldots, \tilde{m}_{i,k}^{(t)}$ and sends to V_r commitments to them. We denote the coins sent from V_r to S during this stage by $coins_{V_r}^{(t)}$, and the entire interaction (the commitments to all shares as well as the verifier's coins) by $\widetilde{\mathsf{tr}_{\mathsf{GKR}}}^{(t)}$.
   (b) i. S derives $z_t \in \mathbb{F}^d$ and the circuit $\langle C \rangle$ (see Item 2) of Theorem 2.21).
   ii. S runs the MPC simulator S_{MPC} (see Definition 2.26) for all parties except q_t^*, wrt the functionality described in Step 3(b)ii of the protocol. We denote S_{MPC}'s output for party $q \in [k] \setminus \{q_t^*\}$ by $\widetilde{view}_q$.
   iii. S sets the view of the remaining party q_t^* to a default value $\widetilde{view}_{q_t^*} = 0^{|view|}$, and sends to V_r the commitments to $(\widetilde{view}_i)_{i\in[k]}$. Denote these commitments by $com(\widetilde{view}_i)_{i\in[k]}$.
   (c) V_r responds with the party q_t^*.
3. If $\vec{z} = (z_1, \ldots, z_\ell)$ is not good (see Definition 6.3) S halts and outputs: $\left(x, C_{pcs}(\hat{w}^*), \left(\widetilde{\mathsf{tr}_{\mathsf{GKR}}}, \{com(\widetilde{view}_i)\}_{i\in[k]}, T_{q_t}\right)_{t\in[\ell]}\right)$.
4. S sends, for every $t \in [\ell]$, the decommitments $\{\widetilde{dec}_m^{(t)}(i), \widetilde{dec}_v^{(t)}(i)\}_{i\in[k]\setminus\{q_t\}}$, where $\widetilde{dec}_m^{(t)}(i)$ and $\widetilde{dec}_v^{(t)}(i)$ are the decommitments to the message $\tilde{m}_i^{(t)}$ and to party i's view, sent in the t-th iteration, respectively.
5. S and V_r run the evaluation protocol (P_{pcs}, V_{pcs}) for the polynomial commitment c_{pcs} at the points $\{z_t\}_{t\in[\ell]}$.
6. S outputs

$$\left(x, c_{pcs}, \left(\widetilde{\mathsf{tr}_{\mathsf{GKR}}}, \{com(\widetilde{view}_i)\}_{i\in[k]}, q_t\right)_{t\in[\ell]}, \left(\{\widetilde{dec}_m(i), \widetilde{dec}_v(i)\}_{i\in[k]\setminus\{q_t\}}\right)_{t\in[\ell]}, \chi\right),$$

where $\chi = (P_{pcs}, V_{pcs})(c_{pcs}, \vec{z}, w^*(\vec{z}))$ is the interactive opening protocol for the polynomial commitment, $\vec{z}$ is computed from $coins_{V_r}$, and $w^*(\vec{z})$ is given from the MPC protocol in Step 2(b)ii.

Figure 5: The Simulator

Thus, first denote,

$$A_0 = \Big(C_{pcs}(\hat{w}^*), \big((\tilde{m}_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}, (\widetilde{view}_i^{(t)})_{i\in[T_t]}\big)_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{w}^*(\vec{z}))\Big)$$
$$A_3 = \Big((C_{pcs}(\hat{\omega}), \big((m_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}, (view_i^{(t)})_{i\in[T_t]}\big)_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))\Big),$$

where recall that:

- m_i is the share for party i of the GKR message in the real interaction (Fig. 4, Step 3a), and $\tilde{m}_i$ is the corresponding share of a random message in the simulation (Fig. 5, Step 2a). $\big((\tilde{m}_i^{(t)})_{i\in T}, coins_{V_r}^{(t)}\big)$ and $\big((m_i^{(t)})_{i\in T}, coins_{V_r}^{(t)}\big)$ are the transcripts of the interaction phase of the GKR protocol in the t-st step (without the commitment) in the simulator and in the real interaction, respectively.
- $\widetilde{view}_i^{(t)}$ consists of the input for party i followed by the output of S_{MPC} for party i.
- $view_i^{(t)}$ the input for party i followed by the view of that party in the MPC protocol (all in the real interaction).

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

$$A_0 := \Big(C_{pcs}(\hat{w}^*), \big((\tilde{m}_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}, (\widetilde{view}_i^{(t)})_{i\in[T_t]}\big)_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{w}^*(\vec{z}))\Big)$$
$$= \Big(C_{pcs}(\hat{w}^*), \big((\tilde{m}_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}, (\widetilde{input}_i^{(t)})_{i\in T_t}, S_{MPC}((\widetilde{input}_i^{(t)})_{i\in T_t})\big)_{t\in[\ell]},](P_{pcs}, V_{pcs})(\vec{z}, \hat{w}^*(\vec{z}))\Big)$$

where $\widetilde{input}_i^{(t)} = \big(\tilde{m}_i{}^{(t)}, z_t, \hat{w}^*(z_t)\big)$, is the input to party i in iteration t derived by the simulator.

$$A_1 := \Big(C_{pcs}(\hat{\omega}), \big((\tilde{m}_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}, (\overline{input}_i^{(t)})_{i\in T_t}, S_{MPC}((\overline{input}_i^{(t)})_{i\in T_t})\big)_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))\Big)$$

where $(\overline{input_k}) = \Big((\tilde{m}_i^{(t)}, \vec{z}, \hat{\omega}(\vec{z})\Big)$

$$A_2 := \Big(C_{pcs}(\hat{\omega}), \big((m_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}, (input_i^{(t)})_{i\in T_t}, S_{MPC}((input_i^{(t)})_{i\in T_t})\big)_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))\Big)$$

where $input_i^{(t)} = \big(m_i{}^{(t)}, z_t, \hat{w}^*(z_t)\big)$, is the input to party i in iteration t derived by the prover.

$$A_3 := \Big(C_{pcs}(\hat{\omega}), \big((m_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}, (view_i^{(t)})_{i\in[T_t]}\big)_{t\in[\ell]}, \big(P_{pcs}, V_{pcs}\big)(\vec{z}, \hat{\omega}(\vec{z}))\Big)$$

Note that A_0 is the output of the simulator. In A_1, we change the commitment and all related components (namely, the input to the simulator and the PCS) from w^* to ω. A_2 is defined similarly to A_1, but with m_i instead of $\tilde{m}_i$. Finally, A_3 is the view in a real interaction with P that is with a real execution of the MPC instead of the simulator.

$A_0 \overset{c}{\approx} A_1$: Since z is good and from the proof of Lemma 6.5, it holds that $\hat{\omega}(\vec{z})$ is uniformly distributed. Moreover, since $\hat{w}^*$ is a random multilinear low-degree polynomial, $\hat{w}^*(\vec{z})$ is also uniformly distributed. Thus from Lemma 3.4,

$$\Big(C_{pcs}(\hat{w}^*), \vec{z}, \hat{w}^*(\vec{z}), (P_{pcs}, V_{pcs})(\vec{z}, \hat{w}^*(\vec{z}))\Big) \overset{c}{\approx} \Big(C_{pcs}(\hat{\omega}), \vec{z}, \hat{\omega}(\vec{z}), (P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))\Big).$$

Since $coins_{V_r}$ is fixed , and $(\tilde{m}_i^{(t)})_{i\in[T_t]}$ is a secret sharing, and the restriction of an additive secret sharing to any set of $k-1$ shares is uniformly random, it follows that:

$$\Big(C_{pcs}(\hat{w}^*), \big((\tilde{m}_i^{(t)})_{i\in T}, coins_{V_r}^{(t)}\big)_{t\in[\ell]}, \vec{z}, \hat{w}^*(\vec{z}), (P_{pcs}, V_{pcs})(\vec{z}, \hat{w}^*(\vec{z}))\Big)$$

$$\overset{c}{\approx}$$

$$\Big(C_{pcs}(\hat{\omega}), \big((\tilde{m}_i^{(t)})_{i\in T_t}, coins_{V_r}^{(t)}\big)_{t\in[\ell]}, \vec{z}, \hat{\omega}(\vec{z}), (P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))\Big)$$

From Fact 2.5 (applying S_{MPC}) we conclude that $A_0 \overset{c}{\approx} A_1$.

$A_1 \equiv A_2$**:** Recall $(m_i)_{i\in[k]}$ is the distribution of the additive secret sharing of the GKR messages as in Fig. 4, Step 3a, whereas $(\tilde{m}_i)_{i\in[k]}$ is the distribution of a secret sharing of a random message in Fig. 5, Step 2a.

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

Thus, since the rest of the distributions are simply computed in the same manner for both distributions similarly to the previous case, we conclude $A_1 \equiv A_2$.

$A_2 \equiv A_3$ By the $(k-1)$-privacy of the MPC protocol (see Definition 2.26), it holds that $\{(view_i^{(t)})\}_{i\in[T_t]}$ is distributed identically to $S_{MPC}((input_i^{(t)})_{i\in T_t})$. Thus, since the input and the rest of the hybrids are identical in the two cases, we have that $A_2 \equiv A_3$.

Thus, we conclude that $A_0 \overset{c}{\approx} A_3$. Denote:

$$C_0 := \Big(C_{pcs}(\hat{w}^*), \big((com(\tilde{m}_i^{(t)}))_{i\in T_t}, coins_{V_r}^{(t)}, \big\{com(\widetilde{view}_i^{(t)})\big\}_{i\in[T_t]}, \overline{dec}^{(t)}\big)_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{w}^*(\vec{z}))\Big)$$
$$C_1 := \Big(C_{pcs}(\hat{\omega}), \big((com(m_i^{(t)}))_{i\in T_t}, coins_{V_r}^{(t)}, \big\{com(view_i^{(t)})\big\}_{i\in[T_t]}, \overline{dec^{(t)}}\big)_{t\in[\ell]}, (P_{pcs}, V_{pcs})(\vec{z}, \hat{\omega}(\vec{z}))\Big),$$

where $\overline{dec}^{(t)} = \{dec_m^{(t)}(i), dec_v^{(t)}(i)\}_{i\in T_t}$ and $dec_m^{(t)}(i)$, $dec_v^{(t)}(i)$ are the decommitments to $\tilde{m}_i^{(t)}$ and party i's view, in the t-st repetition respectively.

C_0 and C_1 are computed from A_0 and A_3 by the same procedure – committing to $(\tilde{m}_i)_{i\in T}$, and $(\widetilde{view}_i)_{i\in T}$, or to $(m_i)_{i\in T}$) and $(view_i)_{i\in T}$. Thus, by Fact 2.5, we conclude that $C_0 \overset{c}{\approx} C_1$.

Finally, from the hiding of the commitment and Fact 2.9 we can add the commitment to the remaining parties in each repetition $t \in [\ell]$ and the claim follows. □

Claim 6.6.2. *Let $(coins_{V_r}, (q_1, \ldots, q_\ell)$ be a fixed randomness choice of the verifier s.t $\vec{z} = (z_1, \ldots, z_\ell)$ (uniquely derived from $coins_{V_r}$), is not good (see Definition 6.3) denote:*

$$H_0 = \Big(x, ref, C_{pcs}(\omega), \{\mathsf{distributed}_{\mathsf{GKR}}, Tq_t\}_{t\in[\ell]}\Big)$$
$$H_1 = \Big(x, ref, C_{pcs}(w^*), \{\mathsf{distributed}_{\mathsf{GKR}}, Tq_t\}_{t\in[\ell]}\Big)$$

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

Proof. The argument follows the proof of Claim 6.6.1, with the final part of the distribution omitted. Hence, the claim concerning $\vec{z}$ is unnecessary here. □

From these two propositions, we conclude that for every fixed choice of the verifier's randomness r, the views of the verifier and the simulator (which receives r) are computationally indistinguishable. Since r is distributed identically in the real execution and in the simulation, this completes the proof. □

6.2.4 Communication Complexity

The prover first commits to w padded with a uniformly random string of length ℓ, sending a message of length $C_{com}(m + \ell)$.

Next, the prover and verifier execute the interactive phase ℓ times. This phase has communication complexity $\text{poly}(D, \log(S), \log(|\mathbb{F}|))$, and since the protocol is secret-shared among k parties and then committed, the communication per round is $k \cdot \text{poly}(\lambda, D, \log(S), \log(|\mathbb{F}|))$. Afterwards, P sends commitments to the views of all k parties in the MPC. The input for each party consists of $\left(coins_V^{(t)}, b_j^{(t)}, z_t, \hat{\omega}(z_t), \langle C \rangle\right)$ (where $b_j = (m_{i,j})_{i \in [r]}$, the messages from Step 3a), each of size $\text{poly}(D, \log(S), \log(|\mathbb{F}|))$. The circuit computed by the MPC has size $\text{poly}(D, \log(S), \log(|\mathbb{F}|))$, as derived from the complexity of V_{post} in Theorem 2.21.

Since each party runs in time polynomial in the input size and the circuit size, the resulting view has size at most $\text{poly}(k, D, \log(S), \log(|\mathbb{F}|))$. The corresponding commitment and decommitments, each of size $\text{poly}(\lambda, D, \log(S), \log(|\mathbb{F}|))$. Finally, P and V execute (P_{pcs}, V_{pcs}) on ℓ points, contributing $C_{open}(m + \ell, \ell)$ communication.

Overall, the total communication complexity is

$$C_{com}(m + \ell) + C_{open}(m + \ell, \ell) + \ell \cdot \text{poly}(\lambda, k, D, \log(S), \log(|\mathbb{F}|)).$$

By setting $\ell = \log(4/\delta)$ and $k = 3$ we achieve Lemma 6.2.

6.3 Using Lemma 6.1 to derive Theorem 1.1

Recall that we are given a Boolean circuit $C : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$. Let $\mathbb{F}$ be finite field, $n' = n/\log(|\mathbb{F}|)$ and $m' = m/\log(|\mathbb{F}|)$. Consider the arithmetic circuit $C' : \mathbb{F}^{n'} \times \mathbb{F}^{m'}$ that operates as follows:

1. Compute from the input $x \in \mathbb{F}^{n'}$ and witness $w \in \mathbb{F}^{m'}$ a redundant representation $x' \in \mathbb{F}^n$ and $w' \in \mathbb{F}^m$ by unpacking the individual bits of each field element into $\log(\mathbb{F})$ field elements.
2. Output $C'(x', w')$, where the Boolean operations are replaced with the corresponding field operations.

Applying Lemma 6.1 to C' we obtain Theorem 1.5.

Acknowledgments

We thank an anonymous reviewer for finding a bug in a previous version of this work that claimed a public-coin version of our main result.

Eden Florentz – Konopnicki is 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.

References

[AFR24] Noor Athamnah, Eden Florentz-Konopnicki, and Ron D. Rothblum. Rate-1 zero-knowledge proofs from one-way functions. In Elette Boyle and Mohammad Mahmoody, editors, *Theory of Cryptography - 22nd International Conference, TCC 2024, Milan, Italy, December 2-6, 2024, Proceedings, Part I*, volume 15364 of *Lecture Notes in Computer Science*, pages 319–350. Springer, 2024. 3, 5, 10

[AG21] Benny Applebaum and Eyal Golombek. On the randomness complexity of interactive proofs and statistical zero-knowledge proofs. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography, ITC 2021, Virtual Conference, July 23-26, 2021*, volume 199 of *LIPIcs*, pages 4:1–4:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 6

[AHIV23] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: lightweight sublinear arguments without a trusted setup. *Des. Codes Cryptogr.*, 91(11):3379–3424, 2023. 7, 8, 9

[BCL22] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge iops with linear-time prover and polylogarithmic-time verifier. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 275–304. Springer, 2022. 14

[BIVW16] Andrej Bogdanov, Yuval Ishai, Emanuele Viola, and Christopher Williamson. Bounded indistinguishability and the complexity of recovering secrets. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 593–618. Springer, 2016. 11

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. *SIAM J. Comput.*, 52(on):STOC19–156–STOC19–199, 2019. 3

[BMO90] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. The (true) complexity of statistical zero knowledge. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 494–502. ACM, 1990. 7, 17

[DGR20] Scott E. Decatur, Oded Goldreich, and Dana Ron. A probabilistic error-correcting scheme that provides partial secrecy. In Oded Goldreich, editor, *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 1–8. Springer, 2020. 14

[DP23] Benjamin E Diamond and Jim Posen. Proximity testing with logarithmic randomness. *Cryptology ePrint Archive*, 2023. 8, 14

[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. 3, 10

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

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *J. Cryptol.*, 9(3):167–190, 1996. 7, 17

[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. 3, 4, 5, 15, 16

[GLS+23] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic snarks for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II*, volume 14082 of *Lecture Notes in Computer Science*, pages 193–226. Springer, 2023. 7

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. 3, 15, 16

[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. 3

[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. 17

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

[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

[HHRS21] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - tight lower bounds on the round and communication complexities of statistically hiding commitments. *CoRR*, abs/2105.01417, 2021. 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. 13

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

[HR07] Iftach Haitner and Omer Reingold. Statistically-hiding commitment from any one-way function. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 1–10. ACM, 2007. 4, 13

[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. 3, 4, 5

[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. 3, 4, 5, 7, 10, 17

[ISVW13] Yuval Ishai, Amit Sahai, Michael Viderman, and Mor Weiss. Zero knowledge ltcs and their applications. In Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, volume 8096 of *Lecture Notes in Computer Science*, pages 607–622. Springer, 2013. 14

[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. 3

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptol.*, 4(2):151–158, 1991. 13

[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. 3, 10

[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. 3

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *43rd Symposium on Foundations of Computer Science, FOCS 2002, Vancouver, BC, Canada, November 16-19, 2002, Proceedings*, pages 366–375. IEEE Computer Society, 2002. 7, 17

[Ros04] Alon Rosen. A note on constant-round zero-knowledge proofs for NP. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004,*

Cambridge, MA, USA, February 19-21, 2004, Proceedings, volume 2951 of *Lecture Notes in Computer Science*, pages 191–202. Springer, 2004. 7, 17

[RRR21] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM J. Comput.*, 50(3), 2021. 4

[RS60] Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960. 14

[RW24] Noga Ron-Zewi and Mor Weiss. Zero-knowledge IOPs approaching witness length. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part X*, volume 14929 of *Lecture Notes in Computer Science*, pages 105–137. Springer, 2024. 9, 14

[VSBW13] Victor Vu, Srinath Setty, Andrew J Blumberg, and Michael Walfish. A hybrid architecture for interactive verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 223–237. IEEE, 2013. 12

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il