Provably-Secure NIZKs From Multi-Round Oracle Proofs

Chaya Ganesh* and Mor Weiss†

June 4, 2026

Abstract

We study the security of the Fiat-Shamir transformation when applied to *multi-round oracle proofs* in the standard model, showing that Interactive Oracle Proofs (IOPs) can be transformed via Fiat-Shamir into Non-interactive Zero-knowledge (NIZK) arguments.

We obtain our result in two steps. First, we show that instantiating Kilian's transformation with an IOP satisfying a mild Zero-Knowledge (ZK) property called semi-Honest-Verifier ZK (semi-HVZK), and a commitment scheme that is extractable and equivocal, yields a ZK argument. We then obtain provably-secure NIZK arguments from our interactive ZK arguments by instantiating Fiat-Shamir with a Correlation-Intractable Hash (CIH) function. The second step involves carefully defining an appropriate "bad relation" for which CIHs are known to exist. Existence of CIH for our bad relation requires the underlying semi-HVZK IOP to have an additional "doom checkability" property which we introduce. Roughly, doom checkability guarantees that one can efficiently check whether a non-empty partial IOP transcript is "doomed" in the sense that it will lead the verifier to reject. We then show that IOPs based on the "MPC in the Head" paradigm are doom checkable.

Our results show that the Fiat-Shamir paradigm can be applied to semi-HVZK *IOPs* to yield *provably-secure* NIZKs. This extends similar results known for *single-round* IOPs (i.e., Probabilistically Checkable Proofs) and interactive proofs (IPs), and establishes the soundness of this method. Along the way, we also prove composition theorems for semi-HVZK IOPs, which might be of independent interest, and use them to improve the soundness error of linear-time IOPs and interactive arguments from the literature.

*Indian Institute of Science, India, chaya@iisc.ac.in.
†Bar-Ilan University, Israel, mor.weiss@biu.ac.il.

Contents

1 Introduction **3**
- 1.1 Our Results . . . 5

2 Technical Overview **7**
- 2.1 ZK Arguments from Semi-HVZK-IOPs and EE Commitments . . . 7
- 2.2 Eliminating Interaction in Kilian's Transformation, Without Random Oracles . . . 9
- 2.3 This Work: CIH for the "Bad" Relation of Multi-Round IOPs . . . 10

3 Related Works **14**

4 Preliminaries **17**
- 4.1 Commitment Schemes . . . 18
- 4.2 (ZK) Argument Systems . . . 19
- 4.3 Interactive Oracle Proofs (IOPs) and Honest-Verifier Zero-Knowledge (HVZK) IOPs . . . 21
- 4.4 Correlation Intractability . . . 24

5 Parallel Repetition of semi-HVZK-IOP **26**
- 5.1 Applications: Sounder Linear-Time ZK IOPs and Arguments . . . 34

6 CZK Arguments from HVZK-IOPs and EEs **35**
- 6.1 Analysis . . . 37
  - 6.1.1 Completeness and Parameter Analysis . . . 37
  - 6.1.2 Soundness . . . 38
  - 6.1.3 Zero Knowledge . . . 41
- 6.2 Achieving Adaptive Soundness . . . 44

7 NIZK from CZK Arguments and CI Hash **46**
- 7.1 A BAD Relation and a CIH for BAD . . . 46
- 7.2 NIZK Construction . . . 54

A Parallel Repetition of semi-HVZK IOP with Implicit Inputs **68**
- A.1 (Semi-HVZK) IOPs with Implicit Inputs . . . 69
- A.2 The Composition Theorem . . . 70

B A Base IOP for Our Constructions **71**
- B.1 MPC Preliminaries . . . 71
- B.2 An IOP From MPC in the Head . . . 72
- B.3 Instantiating the semi-HVZK-IOP . . . 78

C Alternative Bad Relation Definitions **79**

1 Introduction

An interactive argument allows an efficient prover $\mathcal{P}$, that is given an input $x \in \mathcal{L}$ (for some NP language $\mathcal{L}$) and a corresponding witness w, to convince an efficient verifier $\mathcal{V}$ that $x \in \mathcal{L}$, while guaranteeing computational soundness, namely, that any efficient prover $\mathcal{P}^*$ convinces the verifier of a false claim only with small probability.[1]

Interactive Arguments from IOPs. A leading paradigm for designing interactive arguments [Kil92, BCS16] is by combining commitments/hashes with *Interactive Oracle Proofs (IOPs)* [BCS16, RRR16]. In an IOP, the prover $\mathcal{P}_{\mathsf{IOP}}$ and verifier $\mathcal{V}_{\mathsf{IOP}}$ interact as in a standard interactive proof/argument, but throughout the interaction the verifier messages are simply random strings,[2] and the verifier does not read the prover messages. Instead, the verifier has oracle access to prover messages, and once the interaction ends, she reads few proof bits, and either accepts or rejects. Moreover, soundness holds information-theoretically: any (even computationally unbounded) prover $\mathcal{P}^*_{\mathsf{IOP}}$ convinces the verifier of a false claim only with small probability. At a high level, the interactive argument $(\mathcal{P}, \mathcal{V})$ operates by emulating $\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}}$, but instead of sending $\mathcal{P}_{\mathsf{IOP}}$'s messages directly, $\mathcal{P}$ commits to them (via a commitment or a hash). When the IOP interaction ends, $\mathcal{V}$ sends $\mathcal{V}_{\mathsf{IOP}}$'s queries to $\mathcal{P}$, who then opens the commitments/hashes, and $\mathcal{V}$ can then determine whether $\mathcal{V}_{\mathsf{IOP}}$ would have accepted. If the underlying commitment/hash is sufficiently shrinking, the resultant argument system is also *succinct*, namely the communication complexity is $\mathsf{poly}(\kappa, \log |x|)$, where κ is a security parameter.[3]

This paradigm was first introduced by Kilian [Kil92], who constructed succinct interactive arguments by combining a Merkle-Hash Tree with *Probabilistically Checkable Proofs (PCPs)* [ALM+92, AS92], a special case of IOPs in which the interaction between $\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}}$ consists of a single message from the prover to the verifier. A formal security analysis for this construction was given in [BG08] for a specific class of PCP constructions, and more recently in [CDGS22]. The paradigm was extended to the IOP setting in [BCS16], with a formal security proof (in the interactive setting with succinct commitments) given recently in [CDGS22]. Replacing the underlying PCP with an IOP is motivated by provable efficiency advantages of IOPs over their PCP counterparts. For example, while there are IOPs whose length is a fixed polynomial in the witness length [KR08], such PCPs do not exist [FS11] unless $\mathsf{NP} \not\subseteq \mathsf{NP/poly}$; IOPs achieve better tradeoffs between the proof length and verifier query complexity — namely, the number of queries she makes to the proof — compared to the shortest known PCPs (constant-query IOPs for circuit SAT with linear length [BCG+17], or even approaching witness length [RR20], compared to sublinear-query and linear-length PCPs for circuit SAT [BKK+16]); and there exist IOPs that operate natively on a wider range of (useful) NP complete languages (such as R1CS), avoiding expensive Karp reductions.

Non-Interactive Arguments in the ROM. IOP/PCP-based interactive arguments can also be made *Non-Interactive* (NI) in the *Random Oracle Model (ROM)* [Mic00, BCS16]. That is, the entire interaction can be reduced to a single message from the prover to the verifier, assuming both parties have oracle access to a truly random function. More specifically, the prover $\mathcal{P}_{\mathsf{NI}}$ in the non-interactive argument emulates the interactive argument prover $\mathcal{P}$, and applies the random oracle to the partial transcript of the interaction so far ($\mathcal{P}$'s messages and the verifier challenges) to derive the next random challenge of the interactive argument verifier $\mathcal{V}$. In this way, $\mathcal{P}_{\mathsf{NI}}$ can generate the entire transcript between $\mathcal{P}, \mathcal{V}$, which he then sends to the verifier $\mathcal{V}_{\mathsf{NI}}$ of the non-interactive argument.

[1] In contrast, in *proofs*, soundness holds against *computationally unbounded* provers.

[2] Such verifiers are called "public-coin". Throughout the paper, we consider only public-coin IOPs.

[3] In the following, we use the term "Kilian's transformation" to refer to interactive arguments constructed by combining a PCP/IOP with a (not necessarily succinct) commitment scheme.

This construction is provably secure *in the ROM* [BCS16], assuming the underlying IOP satisfies an enhanced soundness notion called "round by round" soundness. Non-interactive arguments in the *plain model* can be obtained via a variant of the Fiat-Shamir (FS) transformation [FS86] by heuristically instantiating the random oracle with a hash function, but this is known to be insecure in general [BGI+01, GK03, BDG+13, KRS25].

Provably-Secure NI Arguments in the Plain Model. An exciting recent line of works shows that in certain cases, the Fiat-Shamir paradigm *is* provably secure [KRR17a, CCRR18, HL18, CCH+19, PS19, BKM20, CPV20, CKU20, JJ21, JKKZ21, CJJ21, CRT24] (see Section 3 for a detailed description of these results). That is, the random oracle can be instantiated with a special kind of hash function, called *Correlation-Interactable Hash (CIH)*, and the resultant non-interactive argument is *provably secure*, assuming both parties have access to a Common Reference String (CRS). The underlying proof systems used in these aforementioned works are PCPs, and certain types of interactive proofs such as trapdoor Sigma protocols or "round-by-round" sound interactive proofs [CCH+19]. These results culminated in the construction of non-interactive arguments that are *provably secure in the plain model*, based on a variety of *standard assumptions* (e.g., the Learning With Errors assumption). We note that these NI arguments are non-succinct, since they employ non-succinct commitments. Such non-succinct NI arguments are motivated by known limitations on *succinct* NI arguments (SNARGs), which either require non-falsifiable cryptographic assumptions, rely on sub-exponentially-hard indistinguishability obfuscation, or only achieve non-adaptive soundness (see, e.g., [BISW18, WW25] and references therein).

(Non-Interactive) Arguments with Zero-Knowledge. It is often useful for the argument system to have a *Zero Knowledge (ZK)* guarantee which, informally, means the verifier learns nothing except that $x \in \mathcal{L}$. (For example, this is useful for obtaining anonymity in blockchain and crypto-currency applications [BCG+14, AH18].) More formally, an interactive argument has ZK if for every (possibly malicious) efficient verifier $\mathcal{V}^*$, there exists an efficient simulator Sim such that for every $x \in \mathcal{L}$, the simulator, given only x, can simulate the entire *view* of $\mathcal{V}^*$ – consisting of her random coin tosses, and the messages she received from the prover – and the simulated and real views are computationally close. In the non-interactive setting, the simulator on input x generates *both the CRS and the (single) prover message*, and its output should be computationally close to the real-world distribution of the CRS and prover message. Such arguments are called *Non-Interactive ZK (NIZK) arguments.*

It is known that (non-interactive) arguments constructed from PCPs have ZK, so long as the PCP satisfies a mild ZK guarantee called *semi Honest-Verifier ZK (semi-HVZK)*. Semi-HVZK guarantees the existence of an efficient simulator $\mathsf{Sim}_{\mathsf{PCP}}$ such that for any $x \in \mathcal{L}$, and any random string r of the *honest* PCP verifier, $\mathsf{Sim}_{\mathsf{PCP}}$ on input x, r *perfectly* simulates the view of the *honest* PCP-verifier, when she uses randomness r. Kilian's transformation provably achieves ZK interactive arguments when using a semi-HVZK-PCP [IMS12], and NIZK arguments in the ROM when using semi-HVZK-IOPs [BCS16],[4] and is generally believed to extend also to ZK (interactive) argument constructions in the *plain model* from semi-HVZK *IOPs*, but to the best of our knowledge, was never formally proven. The motivation for using an *IOP* (instead of a *PCP*) as the underlying proof system in the NIZK construction is the improved efficiency one could potentially achieve by exploiting the provable efficiency advantages of IOPs over PCPs discussed on page 3 (i.e., shorter proofs, avoiding expensive Karp reductions, etc.).

Provably-Secure NIZK Arguments in the Plain Model. The recent breakthrough on provably-secure instantiations of the Fiat-Shamir paradigm in the plain model led to constructions of NIZK

[4]The semi-HVZK property naturally extends to the IOP setting, by having the simulator generate the answers to the honest verifier's queries to *all* prover oracle messages.

arguments in the plain model based on various assumptions; see Section 3. These are obtained by combining semi-HVZK-PCPs or trapdoor Sigma protocols with appropriate CIHs. The extensive research on (semi-HVZK) IOPs in recent years, as well as their various advantages over their PCP counterparts, naturally motivates the following questions:

Can the Fiat-Shamir Paradigm be applied to semi-HVZK IOPs *in a provable way?*
Does this transformation give provably-secure NIZK arguments?

Affirmatively answering these questions boils down to designing CIH functions that are "compatible" with (semi-HVZK) IOPs. We note that the existence of such CIHs does not follow immediately from the existence of CIHs for classes of public-coin multi-round proofs [CCH+19] (e.g., the GKR protocol [GKR08]), since these results apply to *standard* interactive proofs, in which prover messages are read in full.

1.1 Our Results

We show that applying Fiat-Shamir to semi-HVZK *IOPs* and CIHs yields provably-secure NIZK arguments in the plain model. While this follows the popular belief that the transformation remains sound when replacing PCPs with IOPs, formally proving this is much more intricate than it initially appears.

Our generic NIZK construction has two ingredients: the first is an interactive ZK argument system based on semi-HVZK IOPs and commitments, and the second is a CIH which is "compatible" with this ZK argument system. We now give more details on each of these ingredients, then combine them together to obtain NIZKs.

ZK Arguments from Semi-HVZK-IOPs via Kilian's Transformation. We show that instantiating Kilian's transformation with semi-HVZK *IOPs* gives ZK arguments in the plain model. More specifically, we employ commitments that are both *extractable* and *equivocal*. Such commitments have two indistinguishable modes: a standard mode and an equivocation mode. The standard mode is *extractable* in the sense that the CRS is generated together with a trapdoor τ, and if a commitment c can be successfully opened to a value b, then b can be *efficiently* extracted from c given τ (except with some small probability, which is called the extraction error). In the equivocation mode, the CRS is generated together with an "equivocal" commitment c^* and two openings σ_0, σ_1 which can be used to successfully open c^* to *both 0 or 1*. (See Section 2.1 for a discussion of how these properties are used in the security proof, and a comparison with constructions using non-equivocal commitment schemes.) We call such commitments *Extractable Equivocal (EE)* commitments, and prove the following (see Theorem 6.1 for the formal statement):

Theorem 1.1 (ZK Arguments from Semi-HVZK-IOPs and EEs – Informal)**.** *The system obtained by applying Kilian's transformation to a semi-HVZK-IOP and an EE commitment scheme is a ZK argument. Moreover, the ZK simulation is black-box and straight-line. Furthermore, if the underlying IOP and EE schemes have soundness error ϵ_{IOP} and extraction error ϵ_{EE}, respectively, then the argument system has soundness error $\epsilon_{\mathsf{IOP}} + \epsilon_{\mathsf{EE}}$.*

Theorem 1.1 shows that Kilian's transformation remains secure when applied to (semi-HVZK) IOPs. While this follows the general belief, to the best of our knowledge it is the first formal analysis of Kilian's transformation when applied to IOPs with ZK guarantees in the plain model. A formal analysis is particularly important since proving ZK in this setting is subtle (see Section 2.1).

By appropriately instantiating the IOP and EE schemes, Theorem 1.1 implies the following (see Corollary 6.2 for the formal statement):

Corollary 1.2 (ZK Arguments from PKE – Informal)**.** *Assume that public-key encryption exists. Then any NP language has a 3-round ZK argument with setup, with negligible soundness error.*

Correlation-Intractable Hash (CIH) for IOPs. The second ingredient in provable instantiations of Fiat-Shamir in the plain model is a CIH "compatible" with the underlying interactive argument. Roughly, a CIH for a relation $\mathcal{R}_{\mathsf{BAD}}$ is a hash function family $\mathcal{H} = (\mathcal{H}_\kappa)_\kappa$ with the additional guarantee that it is infeasible for an efficient adversary, given a function key $\mathcal{H}_\kappa$, to find an x such that $(x, \mathcal{H}_\kappa(x)) \in \mathcal{R}_{\mathsf{BAD}}$. Holmgren, Lombardi and Rothblum [HLR21] show that under the Learning With Errors (LWE) assumption, an efficiently-computable CIH exists for any relation $\mathcal{R}_{\mathsf{BAD}}$ in which (1) the set of witnesses $\{y : (x, y) \in \mathcal{R}_{\mathsf{BAD}}\}$ for any given instance x is a Cartesian product $\mathcal{R}_1 \times \ldots \times \mathcal{R}_k$ (for some k), (2) each $\mathcal{R}_i$ is sparse, and (3) given $i \in [k], x$ and y, one can efficiently check whether $y \in \mathcal{R}_i$ (see Section 4.4 for the formal definitions and theorem statement).

Recall that the CIH will be used to instantiate the random oracle, which is used to derive verifier randomness from the partial transcript of the interaction so far (consisting of previous verifier challenges and previous messages of the argument prover $\mathcal{P}$). Roughly, for the CI property to suffice for soundness, the relation $\mathcal{R}_{\mathsf{BAD}}$ for which the CIH is designed should consist of pairs $((x, \mathcal{T}_i), r_i)$ where $x \notin \mathcal{L}$, $\mathcal{T}_i$ is a partial transcript up to round i, and r_i is a verifier challenge which is "bad" in the sense that $\mathcal{T}_i$ would ultimately cause the verifier to reject (with high probability), but $\mathcal{T}_i | r_i$ would ultimately cause the verifier to accept (with high probability). Thus, the bad relation should combine both aspects of the EE commitment (since $\mathcal{T}_i$ contains prover messages, which are commitments), and the IOP scheme (since the verifier's final decision is determined by examining the decommitted IOP bits).

Unfortunately, as explained in Section 2.3 below, the "bad" relation $\mathcal{R}_{\mathsf{BAD}}$ induced by a semi-HVZK-IOP will not necessarily have the product structure required to apply the results of [HLR21]. We solve this issue by using a t-wise parallel-repetition version $(\mathcal{P}^{\otimes t}, \mathcal{V}^{\otimes t})$ of the semi-HVZK-IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. Notice that simply repeating the verification procedure of the IOP does not generally preserve semi-HVZK. Instead, $(\mathcal{P}^{\otimes t}, \mathcal{V}^{\otimes t})$ run t *full* parallel executions of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. We prove that this indeed preserves semi-HVZK (see Theorem 5.4 for the formal statement which also handles additional properties such as Round-by-Round soundness):

Theorem 1.3 (Semi-HVZK is Preserved in Parallel Repetition of IOPs – Informal)**.** *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be a semi-HVZK-IOP for an NP language $\mathcal{L}$ with soundness error ϵ. Then for any $t \in \mathbb{N}$, its parallel repetition $(\mathcal{P}^{\otimes t}, \mathcal{V}^{\otimes t})$ is a semi-HVZK-IOP for $\mathcal{L}$ with soundness error ϵ^t.*

Theorem 1.3 shows that soundness of semi-HVZK-IOPs can be amplified via parallel repetition without violating semi-HVZK. This is known to hold for the special case of PCPs, but not for the general case (even though prior works generally assume that this holds). Using Theorem 1.3, we improve the soundness error of the linear-time semi-HVZK-IOPs and ZK arguments of [BCL22]; see Section 5.1. Since IOPs are often constructed by combining several sub-IOPs, composition theorems for ZK-IOPs are extremely important, with only few formal proofs in the literature (e.g., in [BCL22], and more recently in [VW26, CFW26], see Section 3). We therefore expect Theorem 1.3 to find applications elsewhere, and show in Appendix A that it extends also to the more general setting in which part of the input is given to the verifier *as an oracle* (aka the "implicit input" setting, see Appendix A).

Going back to the task of designing a bad relation for the ZK argument, we show that by using a parallel-repetition semi-HVZK-IOP and an EE commitment, we can define a bad relation $\mathcal{R}_{\mathsf{BAD}}$ for which we prove that a CIH exists using similar techniques to [HLR21]. As described in Section 2.3 below, this requires carefully defining the relation to guarantee a "correct" interplay between the different iterations in the parallel repetition, and requires some additional properties from the base

semi-HVZK-IOP used in the parallel repetition (these properties are satisfied by, e.g., IOPs constructed via the "MPC in the Head" paradigm [IKOS07]).[5] The description of $\mathcal{R}_{\mathsf{BAD}}$ combines aspects of "bad" relations defined for PCPs [CJJ21] and interactive proofs [CCH+19]. We then prove the following (see Theorem 7.4 for the formal statement):

Theorem 1.4 (CIH for IOPs – Informal)**.** *Let $(\mathcal{P}, \mathcal{V})$ be a ZK argument obtained from a parallel-repetition semi-HVZK-IOP and an EE commitment scheme. Then, under the LWE assumption, there exists an efficiently-computable hash family* h *which is correlation intractable for the bad relation $\mathcal{R}_{\mathsf{BAD}}$ for $(\mathcal{P}, \mathcal{V})$.*

NIZKs from IOPs and CIH. Combining our ZK arguments of Theorem 1.1 (when the underlying IOP is a parallel-repetition IOP) with the CIH of Theorem 1.4, we obtain NIZK arguments in the plain model via the Fiat-Shamir paradigm applied to *multi-round* IOPs. Specifically, we show (see Theorem 7.6 and Corollary 7.7 for the formal statement):

Theorem 1.5 (NIZK from Semi-HVZK-IOPs and CIH – Informal)**.** *Let $(\mathcal{P}_{\mathsf{CZK}}, \mathcal{V}_{\mathsf{CZK}})$ be an l-round interactive ZK argument constructed from a parallel-repetition semi-HVZK-IOP and an EE commitment with extraction error ϵ_{EE}, and let* Hash *be a CIH for the bad relation $\mathcal{R}_{\mathsf{BAD}}$ of $(\mathcal{P}_{\mathsf{CZK}}, \mathcal{V}_{\mathsf{CZK}})$ with CI error δ. Then the system obtained by instantiating the Fiat-Shamir paradigm with $(\mathcal{P}_{\mathsf{CZK}}, \mathcal{V}_{\mathsf{CZK}})$ and* Hash *is a NIZK argument with soundness error $l \cdot \ell \cdot \epsilon_{\mathsf{EE}} + \delta$, where ℓ denotes the length of $\mathcal{P}_{\mathsf{CZK}}$'s messages.*

Theorem 1.5 has the potential of yielding more efficient NIZK arguments by, e.g., applying it to IOPs designed natively for languages of interest such as R1CS. More generally, it offers flexibility in using *IOPs* for NIZK design, and can therefore benefit from future progress in ever-improving IOP constructions. We demonstrate this flexibility by applying it to a class of IOPs based on MPC in the Head (see Sections 2.3 and 7.2, and Appendix B).

2 Technical Overview

2.1 ZK Arguments from Semi-HVZK-IOPs and EE Commitments

Our ZK arguments are obtained by applying Kilian's transformation [Kil92] — which was originally presented for (non-ZK) PCPs — to semi-HVZK-IOPs and EE commitments. We therefore first recall the base transformation. While Kilian's transformation is often employed to get *succinct* arguments (of sublinear size in the witness length) using, e.g., Merkle-Hash Trees (MHTs), we describe the non-succinct variant since we focus on non succinct NIZK arguments.

Kilian's Transformation. Kilian [Kil92] transforms a PCP system $(\mathcal{P}_{\mathsf{PCP}}, \mathcal{V}_{\mathsf{PCP}})$ into an interactive argument system $(\mathcal{P}, \mathcal{V})$ in the Common Reference String (CRS) model using a hash/commitment.[6] The prover $\mathcal{P}$ emulate $\mathcal{P}_{\mathsf{PCP}}$ to generate a PCP proof Π, generates a commitment c to Π, and sends c to $\mathcal{V}$. $\mathcal{V}$ then sends randomness r for $\mathcal{V}_{\mathsf{PCP}}$, that determines a set $\mathcal{Q}$ of queries to Π. $\mathcal{P}$ then opens c at locations $\mathcal{Q}$ to $\{\Pi(j) : j \in \mathcal{Q}\}$ (where $\Pi(j)$ denotes the jth bit of Π), and $\mathcal{V}$ emulates $\mathcal{V}_{\mathsf{PCP}}$ with these answers to determine her output. Soundness of the argument system can be reduced to PCP soundness using an *extraction* property of the commitment. Roughly, extraction guarantees that if c can be successfully opened to some value b at j, then b can be efficiently extracted from c,

[5] The main property is a "doom-checkability" property, which is reminiscent of properties considered in [CLW18, HLR21]. The IOP should also have round-by-round soundness, as well as an extractability property which extends a similar property for PCPs used in [CJJ21]. Further details are given in Section 2.3.

[6] Kilian's original construction is in the plain model, where the verifier $\mathcal{V}$ chooses the CRS (the description of a hash function). However, we chose to present (even the interactive) argument in this way, since our final goal is to obtain *non-interactive* arguments, in which the verifier doesn't send messages.

given an appropriate trapdoor sampled together with the CRS. Thus, the argument system exploits both extraction of the commitment scheme, as well as a local opening property, allowing the prover to decommit a subset of committed bits. In non-succinct arguments, both properties can be easily achieved by committing to Π bit-by-bit using an extractable bit commitment. (We note that the CRS in this case will consist of multiple CRSs for the bit commitment scheme, one for every bit of the committed string.)

Obtaining ZK in Kilian's Transformation. The argument system described above can be enhanced to also achieve ZK, by replacing the last opening round with an interactive ZK argument in which $\mathcal{P}$ proves that *had* he opened the commitments at $\mathcal{Q}$, $\mathcal{V}$ *would* have accepted. The latter is an NP statement, so generic interactive ZK arguments can be employed, but these use the underlying commitment scheme in a *non-black-box* way. Ishai et al. [IMS12] show that if the underlying PCP has a mild ZK property known as semi-HVZK, then the resultant argument system will have *full-fledged* ZK, while using the underlying commitment scheme *as a black-box*. Recall that semi-HVZK guarantees the existence of an efficient simulator $\mathsf{Sim}_{\mathsf{PCP}}$ that, for $x \in \mathcal{L}$ and random r for $\mathcal{V}_{\mathsf{PCP}}$, simulates $\mathcal{V}_{\mathsf{PCP}}$'s view on input x when using randomness r, up to a negligible statistical distance. The natural strategy towards reducing ZK of the argument system to semi-HVZK of the PCP is the following. The argument simulator Sim generates a commitment c, and sends c to the verifier to obtain a randomness string r^*. Then, Sim emulates $\mathsf{Sim}_{\mathsf{PCP}}$ on x, r^* to obtain simulated PCP bits $\{\Pi^*(j) \ : \ j \in \mathcal{Q}\}$, where $\mathcal{Q}$ are the queries determined by r^*. Finally, for every $j \in \mathcal{Q}$, Sim opens c at location j to $\Pi^*(j)$. The only question is how to generate c before knowing r^* (which in turn determines $\mathcal{Q}$). In [IMS12], Sim *guesses* r^*, and commits to the string that is $\Pi^*(j)$ at location $j \in \mathcal{Q}$ and 0 everywhere else. If Sim guessed incorrectly, it rewinds the verifier. Obtaining efficient verification then requires assuming that $|r| = O(\log |x|)$. This restriction was later removed in [IMSX15] by either assuming the underlying PCP has *full fledged ZK* (i.e., one can simulate the view of any *malicious* verifier that does not make "too many" queries), or relaxing the resultant argument system to only have ZK against *honest* verifiers. We note that [IMS12, IMSX15] focus on constructing *succinct* ZK arguments and their ZK analysis therefore relies on rewinding (see Section 3 for further details).

In contrast to [IMS12, IMSX15], we do not aim for succinctness. We can therefore hope to obtain *straight-line simulation*. Specifically, we observe that the natural straight-line simulation strategy described in the previous paragraph *does* work when the underlying commitment is enhanced to satisfy a stronger secrecy guarantee called *equivocation*. Informally, an equivocal commitment scheme has an additional indistinguishable *equivocation mode* in which the CRS is generated together with an equivocal commitment c^* and two openings σ^0, σ^1, such that for any $b \in \{0,1\}$, σ_b successfully opens c^* to b. Instantiating Kilian's transformation with a semi-HVZK-PCP and an *Extractable and Equivocal (EE)* bit commitment scheme yields ZK arguments with *straight-line* simulation, and without having to restrict the randomness complexity of the underlying PCP, or enhance it to have full-fledged ZK. This will follow as a special case of Theorem 1.1. To see why equivocation helps, notice that the simulator can now operate as follows. For every index j, Sim will generate a CRS together with an equivocal commitment c_j and openings σ_j^0, σ_j^1. It will send $(\mathsf{c}_j)_j$ to $\mathcal{V}^*$ to obtain r, emulate $\mathsf{Sim}_{\mathsf{PCP}}$ on x, r to obtain simulated PCP bits $\{\Pi^*(j) \ : \ j \in \mathcal{Q}\}$, then open the commitments to these values using the appropriate equivocated openings.

This Work: Kilian's Transformation in the IOP Setting. Kilian's transformation naturally extends to the IOP setting using an underlying l-round IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. The argument prover $\mathcal{P}$ and verifier $\mathcal{V}$ now emulate the interaction between $\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}}$ round by round. That is, in each round i, $\mathcal{P}$ emulates $\mathcal{P}_{\mathsf{IOP}}$ to obtain the i-th round oracle Π^i, then sends a commitment c^i to Π^i, and $\mathcal{V}$ sends a random challenge r_i for $\mathcal{V}_{\mathsf{IOP}}$. Once the interaction ends, $\mathcal{P}$ uses $r_1, \ldots, r_l$ to determine the queries $\mathcal{Q}$ which $\mathcal{V}_{\mathsf{IOP}}$ makes to her oracles, and opens $\mathsf{c}^1, \ldots, \mathsf{c}^l$ at the locations in $\mathcal{Q}$. This transformation from (non-

ZK) IOPs to (non-ZK) arguments was described, using succinct commitments, in [BCS16, CDGS22]. (More specifically, Ben-Sasson et al. [BCS16] give an analysis for the non-interactive version in the ROM based on MHTs, and Chiesa et al. [CDGS22] analyze the interactive version in the plain model, based on any vector commitment.)

It is generally believed that if the underlying IOP has semi-HVZK, then applying Kilian's transformation to it will result in ZK arguments. However, to the best of our knowledge, this has not been formally proven, and as is evident from the description above, some subtleties arise in the construction. We prove that instantiating Kilian's transformation with a semi-HVZK-IOP and EE commitments gives a ZK argument. The ZK simulation is black-box and straight-line, and the soundness analysis is tight (namely, the soundness error of the argument system is the sum of the soundness error of the IOP and the extraction error of the EE). We note that since our argument system is not succinct, and can therefore rely on fully extractable commitments, the soundness analysis is fairly simple. This should be contrasted with the soundness analysis for *succinct* interactive arguments [CDGS22] which is much more involved (and this seems to be inherent to using commitments that are only partially extractable, which is in turn inherent to obtaining succinctness). This proves Theorem 1.1. Corollary 1.2 follows by instantiating the generic transformation with (a parallel-repetition of) the 2-round semi-HVZK-IOPs of [BCGV16], and the EE commitments of [BWW24, Construction 4.2] based on public-key encryption.

We use our ZK arguments to construct NIZK arguments. We explain this transformation next.

2.2 Eliminating Interaction in Kilian's Transformation, Without Random Oracles

As discussed in Section 1, the interactive arguments obtained via Kilian's transformation can be made Non-Interactive (NI) in the ROM. More Specifically, to generate his (single) message in the NI argument system, the prover $\mathcal{P}_{\mathsf{NI}}$ emulates, round by round, the full interaction between the prover $\mathcal{P}$ and verifier $\mathcal{V}$ of the interactive argument. In every round i, $\mathcal{V}$'s random challenge r_i is derived as $r_i := \mathsf{RO}(\mathcal{T}_i, \mathsf{c}^i)$, where c^i is $\mathcal{P}$'s round-i message, and $\mathcal{T}_i$ is the partial transcript of the execution so far, consisting of x, the prover messages $\mathsf{c}^1, \ldots, \mathsf{c}^{i-1}$ and prior verifier challenges $r_1, \ldots, r_{i-1}$.

Historically, interaction in Kilian's interactive arguments was eliminated either using the ROM as described above, or by heuristically instantiating the random oracle with a hash function (which is known to be insecure in general [BGI+01, GK03, BDG+13, KRS25]). However, a sequence of recent works have *provably* instantiated the random oracle with a specific form of hash functions, called *Correlation Interactable Hash (CIH)* [KRR17a, CCRR18, HL18, CCH+19, PS19, HLR21, BKM20, CPV20, CKU20, JJ21], resulting in *provably secure* NI (ZK) arguments in the plain model under a variety of standard assumptions. Recall from Section 1.1 that a hash family $\mathcal{H} = \{\mathcal{H}_\kappa\}_\kappa$ is CI for a relation $\mathcal{R}_{\mathsf{BAD}}$ if for every PPT adversary $\mathcal{A}$,

$$\Pr_{\mathsf{h} \leftarrow \mathcal{H}_\kappa}[(x, \mathsf{h}(x)) \in \mathcal{R}_{\mathsf{BAD}} \ : \ x \leftarrow \mathcal{A}(1^\kappa, \mathsf{h})] = \mathsf{negl}(\kappa).$$

Obtaining NI arguments in this way therefore reduces to constructing a CIH for an appropriate definition of $\mathcal{R}_{\mathsf{BAD}}$, which will enable reducing soundness of the NI argument to soundness of the underlying IOP/PCP. To demonstrate this, we consider first the special case in which the underlying IOP is *a PCP*.

Bad Relations for PCPs, and their CIHs. Consider the special case in which the interactive argument $(\mathcal{P}, \mathcal{V})$ for an NP language $\mathcal{L}$ is constructed from a PCP system $(\mathcal{P}_{\mathsf{PCP}}, \mathcal{V}_{\mathsf{PCP}})$, and then made non interactive (e.g., in the ROM). Clearly, soundness of the NI argument should be reduced to the soundness of the PCP, but there is an inherent discrepancy between the nature of soundness in both cases. To see why, notice that while in the PCP system the verifier randomness r is *random and*

independent of the proof oracle Π, in the NI argument, r *is correlated* with Π via the random oracle RO. In particular, even though RO is uniformly random, still the NI argument prover $\mathcal{P}_{\mathsf{NI}}$ can "try" multiple different proof strings Π^*: for each Π^*, $\mathcal{P}_{\mathsf{NI}}$ can generate the commitment c^* and verifier randomness $r := \mathsf{RO}(x, \mathsf{c}^*)$ and check whether $\mathcal{V}$ would have accepted when using randomness r. Therefore, the soundness reduction is related to a set of "bad" verifier challenges:

$$\mathcal{R}_{\mathsf{PCP}} := \left\{ ((x, \mathsf{c}), r) \ : \ \nexists w \text{ s.t. } \mathcal{R}_{\mathcal{L}}(x, w) = 1 \ \wedge \ \mathcal{V}_{\mathsf{PCP}}(x; r) \text{ accepts given } \widetilde{\Pi} \right\}$$

where $\mathcal{R}_{\mathcal{L}}$ is the NP relation corresponding to $\mathcal{L}$, and $\widetilde{\Pi}$ is the PCP proof extracted from c ($\widetilde{\Pi}$ is well defined, since the commitment scheme is fully extractable). Thus, when instantiating the random oracle, we need to insure that it would be infeasible for an efficient prover to find (x, c) and randomness r such that $((x, \mathsf{c}), r) \in \mathcal{R}_{\mathsf{PCP}}$. This is indeed the case if the random oracle is replaced with a CIH for $\mathcal{R}_{\mathsf{PCP}}$. Constructing NI arguments in this way then boils down to constructing a CIH for the bad relation $\mathcal{R}_{\mathsf{PCP}}$ of the PCP.

Several recent works (e.g., [HLR21, CJJ21]) construct such CIHs using a result of [HLR21], who design a CIH for any relation $\mathcal{R}$ satisfying the following 3 properties. First, $\mathcal{R}$ is a *product relation*, i.e., $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}^t$ for some t, and for every x, the set $\mathcal{R}^x := \{w : (x, w) \in \mathcal{R}\}$ is a Cartesian product $\mathcal{R}^x = \mathcal{R}^x_1 \times \ldots \times \mathcal{R}^x_t$ for $\mathcal{R}^x_1, \ldots, \mathcal{R}^x_t \subseteq \mathcal{W}$. Second, $\mathcal{R}$ is *sparse* in the sense that for every $k \in [t]$, $|\mathcal{R}^x_k| / |\mathcal{W}|$ is small. Third, $\mathcal{R}$ is *efficient product verifiable*, namely there exists a circuit C of polynomial size such that for any $x, k \in [t]$ and w_k, $w_k \in \mathcal{R}^x_k$ if and only if $C(x, k, w_k) = 1$ (see Section 4.4 for the formal definitions). For the bad relation $\mathcal{R}_{\mathsf{PCP}}$ of a PCP to be a product relation, prior works used a *parallel repetition* version of the PCP, in which t random and independent executions of the PCP verification procedure are executed in parallel *with the same proof oracle*. This, however, does not preserve semi-HVZK, since making repeated queries to the same proof oracle deviates from the honest PCP verifier's strategy. A natural approach is to have the prover sample t *random and independent PCP proofs*, and have the PCP verifier use a different proof in each of the parallel repetitions. The fact that this parallel repetition preserves semi-HVZK follows as a special case of our composition theorems for IOPs (Section 5).

The definition of the bad PCP relation $\mathcal{R}_{\mathsf{PCP}}$ crucially relies on the fact that the interaction consists of a single round (a single prover oracle message, followed by the verifier picking her random challenge and querying the oracle). However, for interactive arguments constructed from *multi-round* IOPs, the bad relation should capture the multi-round nature of the interaction. We turn our attention to this next.

2.3 This Work: CIH for the "Bad" Relation of Multi-Round IOPs

To securely apply the Fiat-Shamir paradigm to *multi-round* proofs (possibly with $\omega(1)$ rounds), the underlying proof system should satisfy an enhanced soundness notion called "Round By Round" (RBR) soundness [CCH+19]. Roughly, RBR soundness classifies partial transcripts in an interactive proof system as either "*doomed*" or "*not doomed*". The set of doomed transcripts satisfies two properties: (1) for $x \notin \mathcal{L}$, the partial transcript before the interaction began (which consists solely of x), is doomed; and (2) if a full transcript for x is doomed, then it will be rejected (with probability 1). RBR soundness requires that for every round i, any $x \notin \mathcal{L}$, and any doomed partial transcript $\mathcal{T}^i$ for x at round i, the following holds: for every round-$(i+1)$ prover message M, the probability, over the choice of a random verifier challenges r^{i+1}, that $\mathcal{T}^i \mid M \mid r^{i+1}$ is *not* doomed, is small. (See Definition 4.16 in Section 4.3 for the formal definition.)

Using RBR soundness, Canetti et al. [CCH+19] define the bad relation for *Interactive Proofs (IPs)* (i.e., in which prover messages are read in full) as the set of pairs of the form $((\mathcal{T}, M), r)$ where $\mathcal{T}$ is

a doomed partial transcript, M is a prover message, and r is a verifier challenge such that $\mathcal{T} \mid M \mid r$ is *not doomed*:

$$\mathcal{R}_{\mathsf{IP}} := \left\{ ((\mathcal{T}, M); r) \ : \ \begin{array}{c} \mathcal{T} \text{ is a doomed partial transcript} \\ M \text{ is a prover message, } r \text{ is a verifier challenge} \\ \mathcal{T} \mid M \mid r \text{ is not doomed} \end{array} \right\}$$

Naturally, a bad relation $\mathcal{R}_{\mathsf{BAD}}$ for interactive arguments obtained from interactive *oracle* proofs would combine aspects of both the bad relation $\mathcal{R}_{\mathsf{PCP}}$ of PCPs and the bad relation $\mathcal{R}_{\mathsf{IP}}$ of IPs. Indeed, to reduce NIZK-soundness to the RBR soundness of the interactive argument, we need to consider partial transcripts (as in $\mathcal{R}_{\mathsf{IP}}$), but we then also need to extract the underlying IOP proofs determined by the commitments (as in $\mathcal{R}_{\mathsf{PCP}}$). For this, we define the doomed set and RBR soundness for the *underlying IOP* $(\mathcal{P}_B, \mathcal{V}_B)$ (instead of the interactive argument from Section 2.1). The bad relations of PCPs and IPs can then be naturally combined as follows:

$$\mathcal{R}^{\geq 1}_{\mathsf{BAD}} = \left\{ ((\mathcal{T}, \mathsf{c}^{i+1}); r^{i+1}) \ : \ \begin{array}{c} \mathcal{T} = (x, \{\mathsf{c}^j, r^j\})_{j \in [i]} \text{ is a round-}i \text{ interactive argument partial transcript} \\ \{\Pi^j\}_{j \in [i+1]} \text{ are the prover oracle messages extracted from } \{\mathsf{c}^j\}_{j \in [i+1]} \\ (x, \Pi^1, r^1, \ldots, \Pi^i, r^i) \text{ is a doomed IOP partial transcript} \\ (x, \Pi^1, r^1, \ldots, \Pi^i, r^i, \Pi^{i+1}, r^{i+1}) \text{ is a non-doomed IOP partial transcript} \end{array} \right\}.$$

Unfortunately, for an arbitrary semi-HVZK-IOP, this bad relation will not necessarily be a product relation, which (as discussed in Section 2.2 above) is needed to construct a CIH for it via the [HLR21] techniques. We therefore replace the semi-HVZK-IOP $(\mathcal{P}_B, \mathcal{V}_B)$ with its t-fold parallel repetition $(\mathcal{P}^{\otimes t}, \mathcal{V}^{\otimes t})$, in which case the resultant bad relation described above will indeed be a product relation. (A similar approach was taken in [CJJ21], who use CIH for non-ZK *PCPs* to design Batch Arguments.) Notice that by Theorem 1.3, this parallel-repetition is still a semi-HVZK-IOP. We note each round-j IOP message Π^j of $\mathcal{P}^{\otimes t}$ has the form $\Pi^j = (\Pi^j_1, \ldots, \Pi^j_t)$, where every Π^j_k is a round-j message in the k'th repetition of the base IOP $(\mathcal{P}_B, \mathcal{V}_B)$.

Formalizing this intuition entails the following steps: (1) formally defining the doomed set $\mathcal{D}^{\otimes t}$ of $(\mathcal{P}^{\otimes t}, \mathcal{V}^{\otimes t})$, (2) proving that $\mathcal{R}^{\geq 1}_{\mathsf{BAD}}$ is a sparse product relation, and (3) proving that $\mathcal{R}^{\geq 1}_{\mathsf{BAD}}$ has efficient product verifiability. We now discuss each of these steps.

Step (1): defining the doomed set $\mathcal{D}^{\otimes t}$. Although the RBR soundness of the base-IOP is tied to a specific doomed set $\mathcal{D}$, there are several ways to define $\mathcal{D}^{\otimes t}$ (this issue does not arise when using parallel-repetition *PCPs* as in [CJJ21], since the bad relation for PCPs does not use doomed sets). We define (Definition 5.2 in Section 5) a round-i partial transcript $\mathcal{T}^{\otimes}$ as doomed if the number of round-i partial transcripts of the base-IOP passes some threshold, where the threshold is *dynamic* and depends on the round number i. More formally, a round-i partial transcript $\mathcal{T}^{\otimes} = (\mathcal{T}_1, \ldots, \mathcal{T}_t)$ is doomed (where $\mathcal{T}_k$ is a partial transcript for the k'th parallel repetition) if and only if $|\{k : \mathcal{T}_k \in \mathcal{D}\}| \geq \mathsf{t}_i$ for some threshold t_i. This dynamic-threshold doomed set is similar to the doomed state function for parallel repetition of interactive *proofs* considered in [HLR21, Def. 6.8]. We discuss several alternative doomed-set definitions, and the reasons they fail, in Appendix C.

Step (2): $\mathcal{R}^{\geq 1}_{\mathsf{BAD}}$ is an (approximate) sparse product relation. Recall that $\mathcal{R}^{\geq 1}_{\mathsf{BAD}}$ consists of pairs of the form $((\mathcal{T}, \mathsf{c}), r)$ where $r = (r_1, \ldots, r_t)$ consists of verifier randomness (in a single round) for all repetitions of the base-IOP. Therefore, $\mathcal{R}^{\geq 1}_{\mathsf{BAD}} \subseteq \mathcal{X} \times \mathcal{W}^t$ is a sparse product relation if for every $(\mathcal{T}, \mathsf{c})$, the set $\mathcal{R}^{\mathcal{T}, \mathsf{c}} := \{r : ((\mathcal{T}, \mathsf{c}), r) \in \mathcal{R}^{\geq 1}_{\mathsf{BAD}}\}$ can be decomposed as $\mathcal{R}^{\mathcal{T}, \mathsf{c}} = \mathcal{R}^{\mathcal{T}, \mathsf{c}}_1 \times \ldots \times \mathcal{R}^{\mathcal{T}, \mathsf{c}}_t$ where each $\left|\mathcal{R}^{\mathcal{T}, \mathsf{c}}_k\right| / |\mathcal{W}|$ is small (i.e., sparse). Unfortunately, $\mathcal{R}^{\geq 1}_{\mathsf{BAD}}$ actually is *not* a sparse product relation. To see why, notice that intuitively, to get sparsity, $\mathcal{R}^{\mathcal{T}, \mathsf{c}}_i$ should consist of exactly those random strings r_k which take the transcript in the *k'th repetition* (as determined by $\mathcal{T}$ and c) from

doomed to not doomed. However, these $\mathcal{R}_i^{\mathcal{T},\mathsf{c}}$'s do not give a decomposition of $\mathcal{R}^{\mathcal{T},\mathsf{c}}$. Indeed, the fact that $((\mathcal{T},\mathsf{c}),r) \in \mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ for a round-i partial transcript $\mathcal{T} = (\mathcal{T}_1,\ldots,\mathcal{T}_t)$ and $r = (r_1,\ldots,r_t)$ guarantees that $\mathcal{T} \in \mathcal{D}^{\otimes t}$, which in turn only guarantees that $|\{k : \mathcal{T}_k \in \mathcal{D}\}| \geq \mathsf{t}_i$. In particular, it might be the case that, e.g., $\mathcal{T}_1 \notin \mathcal{D}$ for the transcript $\mathcal{T}_1$ in the first base-IOP execution, in which case we have no guarantee that $r_1 \in \mathcal{R}^{\mathcal{T},\mathsf{c}}$ (since we have no guarantees on how the verifier randomness affects non-doomed transcripts). More generally, once a partial transcript $\mathcal{T}_k$ in the k'th repetition becomes non-doomed (this can happen even if the parallel-repetition transcript $\mathcal{T}$ itself *is* doomed), *all* possible verifier challenges r_k in the k'th repetition might be "bad" in the sense that there exists a Π_k such that $\mathcal{T}_k \mid \Pi_k \mid r_k$ is not doomed. (This is indeed the case in, e.g., IOPs for sumcheck.)

Fortunately, it suffices for $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ to satisfy a relaxed notion of sparse *approximate* product relation [HLR21, Def. 3.2] which, roughly, means $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ still has decomposition sets $\mathcal{R}_1^{\mathcal{T},\mathsf{c}},\ldots,\mathcal{R}_t^{\mathcal{T},\mathsf{c}}$ as above, but we now only require that $(r_1,\ldots,r_t) \in \mathcal{R}^{\mathcal{T},\mathsf{c}} \Leftrightarrow \left|\{k : r_k \in \mathcal{R}_k^{\mathcal{T},\mathsf{c}}\}\right| \geq \alpha t$ for some approximation parameter α. The techniques of [HLR21] imply the existence of CIHs also for *approximate* product relations, so long as the relation is sparse and has efficient product verifiability.

This brings us to the next issue, which is that [HLR21] require sparsity to be very small (compared to the round complexity), which in turn translates to an upper bound on the RBR soundness error. Unfortunately, many IOPs do not have such a small RBR error (e.g., this is the case for standard IOPs based on arithmetization and sumcheck, and for the IOPs obtained via the MPCitH technique from the class of MPC protocols we use in Appendix B). To handle this, we take a 2-step approach: we first show (Theorem 5.7) that parallel-repetition of IOPs with *mild* RBR soundness guarantees gives mild RBR soundness amplification, which results in an IOP with sufficiently small RBR error. We can then use this as the base-IOP and get the desired sparsity. (We note that our main repetition theorem, Theorem 1.3, cannot be used here since to get RBR soundness amplification it requires the base IOP to already have a small RBR soundness error; a similar phenomena was observed in [HLR21] for IPs. See the discussion on Page 31 for further details.)

Step (3): $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ has efficient product verifiability. Recall that $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ has efficient product verifiability if, given k, $\mathcal{T} = (\mathcal{T}_1,\ldots,\mathcal{T}_t)$, c and r_k as above, we can *efficiently* determine whether r_k is a "bad" random string in the k'th repetition (namely, r_k takes the k'th repetition partial transcript $\mathcal{T}_k$ from doomed to not doomed). We show this is implied by a property of the IOP which we call "doom checkability" (Definition 7.2 in Section 7.1). Roughly, an IOP with a corresponding doomed set has doom checkability if one can efficiently determine, for every transcript $\mathcal{T}$,whether it is doomed or not.[7] We note that prior works prove efficient product verifiability by imposing, in different ways, requirements on the underlying proof system. Specifically, [CLW18, Def. 3.6] define "efficiently-searchable relations" for Sigma protocols with unique bad challenges; [HLR21, Sec. 6.2.1] define properties that jointly consider the underlying interactive proof and the bad challenge relation for it, and show efficient product verifiability given a trapdoor; and [CJJ21, Sec. 4.2] require that the PCP be extractable. Our notion of doom checkability is perhaps most related to the extractable PCPs of [CJJ21], since it captures efficient product verifiability via a clean property *of the IOP*, independently of the bad relation and any trapdoors.

One subtle point that arises when considering doom checkability is that such a property is meaningful for NP only if $\mathcal{T}$ is not empty. Indeed, efficiently determining whether the *empty* transcript is doomed or not requires one to efficiently *decide* the language. (To see why, notice that if $\mathcal{T}$ is empty, the doom checking circuit is given only x. From completeness, if $x \in \mathcal{L}$ then the empty transcript isn't doomed, and otherwise from soundness the empty transcript is doomed.) Therefore,

[7] To see why doom checkability implies efficient product verifiability, notice that given $k, \mathcal{T} = (\mathcal{T}_1,\ldots,\mathcal{T}_t), \mathsf{c}$ and r_k, we can first use doom checkability to determine whether $\mathcal{T}_k$ is doomed. Then, we can extract a prover message $\Pi = (\Pi_1,\ldots,\Pi_t)$ from c, and use doom checkability to determine whether $(\mathcal{T}_k \mid \Pi_k \mid r_k)$ is doomed.

for NP-complete languages, we cannot hope to efficiently determine whether the empty transcript is doomed or not (unless P=NP). We therefore define doom checkability to consider only *non-empty* transcripts. Consequently, the relation $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ described above is efficient product verifiable so long as $i \geq 1$, and we add this requirement to the definition of $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ above. However, restricting $i \geq 1$ in $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ introduces a problem in the NIZK soundness proof (Page 56). At a high level, the soundness proof of the NIZK reduces a soundness-violating NIZK prover $\mathcal{P}_{\mathsf{NIZK}}^*$ to a CI-breaking adversary $\mathcal{A}$, but for the above definition of $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ (with $i \geq 1$), the reduction does not rule out the possibility that the prover successfully cheats *in the first round* of the underlying interactive argument/IOP (i.e., moves a doomed empty transcript for $x \notin \mathcal{L}$ to a non-doomed round-1 transcript). To explain why, we take a closer look at the security reduction.

In the reduction, a convincing NIZK argument generated by $\mathcal{P}_{\mathsf{NIZK}}^*$ induces a partial round-i transcript $\mathcal{T}$ for some i, a round-$(i+1)$ message c^{i+1}, and a round-$(i+1)$ challenge r^{i+1} such that the IOP transcript determined by $\mathcal{T}$ is doomed (i.e., in $\mathcal{D}^{\otimes t}$) but the IOP transcript determined by $\mathcal{T}_i \mid \mathsf{c}_{i+1} \mid r_{i+1}$ is not doomed. If $i \geq 1$, $((\mathcal{T}, \mathsf{c}^{i+1}), r^{i+1}) \in \mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ and we are done. But what if $i = 0$? In this case, $\mathcal{T} = x$ and as noted above, doom checkability does not help. To handle the special case that $i = 0$, we require that the underlying IOP have an additional *extraction* property which, roughly, states that there exists an efficient extractor algorithm E such that for every first IOP-prover message Π^1, if $(x, \mathsf{E}(x, \Pi^1)) \notin \mathcal{R}$ then with high probability over the choice of first-round IOP-verifier randomness r, (x, Π_1, r) is doomed.[8] We show in Appendix B that this property is satisfied by IOPs based on the "MPC in the Head" paradigm [IKOS07] (roughly, this is because in such IOPs the first prover message contains an additive secret-sharing of the witness). Using this property, we now define another bad relation $\mathcal{R}_{\mathsf{BAD}}^0$ to handle round-1 verifier challenges:

$$\mathcal{R}_{\mathsf{BAD}}^0 := \left\{ ((x, \mathsf{c}^1); r^1) \ : \ \begin{array}{c} \Pi^1 \text{ is the round-1 prover oracle message extracted from } \mathsf{c}^1, \Pi^1 = (\Pi_1^1, \ldots, \Pi_t^1) \\ (x, \Pi^1 \mid r^1) \text{ is not doomed and } \forall k \in [t], (x, \mathsf{E}(x, \Pi_k^1)) \notin \mathcal{R} \end{array} \right\}.$$

Finally, we define the bad relation to be $\mathcal{R}_{\mathsf{BAD}} := \mathcal{R}_{\mathsf{BAD}}^{\geq 1} \cup \mathcal{R}_{\mathsf{BAD}}^0$. We show it has efficient product verifiability by reduction to the two IOP properties discussed above: doom checkability (Definition 7.2) and extractability (Definition 4.17). In Section 7.1, we show that $\mathcal{R}_{\mathsf{BAD}}$ satisfies all the properties needed to obtain a CIH for $\mathcal{R}_{\mathsf{BAD}}$ via the techniques of [HLR21]. We note that our soundness reduction requires *both* extractability *and* doom checkability. Indeed, extraction is needed to rule-out the case that $\mathcal{P}_{\mathsf{NIZK}}^*$ finds a "bad" round-1 verifier challenge. This guarantees that if $x \notin \mathcal{L}$ then the round-1 partial transcript is still doomed, and if $\mathcal{P}_{\mathsf{NIZK}}^*$ violates NIZK-soundness then the full transcript is not doomed. Thus, between round 2 and the final round, the IOP transcript moved from "doomed to "not doomed". Doom checkability is then used to obtain efficient product verifiability. In Appendix C we discuss alternative definitions of the bad relation, and the reasons they fail.

Handling IOPs with imperfect completeness. The FS+CIH paradigm (both for PCPs in prior works, and for IOPs here) implicitly assumes the PCP/IOP has perfect completeness (i.e., $x \in \mathcal{L}$ is accepted with probability 1). The issue with using PCPs/IOPs with imperfect completeness is in the NIZK completeness proof: since the CIH skews the verifier randomness, it is not clear how to bound the probability that the CIH output will be one that leads the verifier to reject $x \in \mathcal{L}$. Following [CCH+19, Remark 5.2], we resolve this issue by making sure the CIH is also l-wise independent (see Lemma 7.5 in Section 7.1). This increases the flexibility of our "FS+CIH+IOP to NIZK" transformation, since we are no longer restricted to using perfectly-complete IOPs.

How To Use Our Paradigm: an Example Using MPCitH. Our analysis yields provably-secure NIZKs in the plain model from IOPs, so long as the IOP has: (1) semi-HVZK, (2) extractability, and

[8] For the special case in which the IOP is a PCP, this extractability notion is equivalent to the standard extractability property of PCPs [CJJ21], which was used in [CJJ21] to define a bad relation for PCPs in their BARGs construction.

(3) doom checkability. Thus, our results can be interpreted as showing a reduction from designing NIZKs from IOPs, to designing IOPs satisfying these properties, and could potentially yield new NIZK constructions that work directly over useful "native" representations (such as R1CS constraints), by exploiting the known efficiency advantages of IOPs. To demonstrate how to use our paradigm, we construct in Appendix B a class of IOPs satisfying these properties based on the "MPC in the Head" (MPCitH) technique [IKOS07]. Roughly, the MPCitH techniques give semi-HVZK PCPs from perfectly correct/secure MPC protocols. This was used in [GLS22] to design a PCP which they then plug into the FS+CIH paradigm to get short NIZKs for NP. When the MPCitH paradigm is applied to MPC protocols with *imperfect* correctness, it yields a semi-HVZK *IOP* (instead of a PCP). We prove this class of IOPs satisfies the 3 properties discussed above. We also discuss limitations to generalizing our results to other classes of MPC protocols (e.g., statistically secure). We then instantiate the generic construction with the MPC of [BI05] to obtain a scaled-down version of the NIZKs of [GLS22] with shorter proofs (of size $\tilde{O}(|w|)$ vs. $\tilde{O}(|C|)$, where w, C are the NP witness and NP verification circuit of the relation, respectively) for a smaller class of languages ($\mathsf{AC}^0[\oplus]$ vs. all of NP) assuming LWE. While better NIZKs (specifically, with rate 1) from LWE are known [ACG+24, BDS25], we view our NIZK instantiation as indicating the potential of extending the paradigm from PCPs to IOPs. See Corollary 7.8 and the following discussion for further details, and a more comprehensive and concise comparison with prior works.

3 Related Works

Most relevant to our results are works that construct NIZK arguments in the plain model from MPCitH, works that instantiate Kilian's transformation with ZK PCPs/IOPs, and works proving composition theorems for IOPs with ZK. We discuss these works in the next 3 paragraphs, comparing our results to prior works. To give a broader context, we then give a more through survey of other related works.

Provably-Secure NIZKs from MPC in the Head. Ghosal, Lou, and Sahai [GLS22] recently used MPC in the Head (MPCitH) to design semi-HVZK PCPs which they then use to construct NIZKs via the "Fiat-Shamir + CIH" paradigm. Despite using the same paradigms (MPCitH, Fiat-Shamir and CIHs), our semi-HVZK IOPs from MPCitH (Appendix B) differ significantly from [GLS22]. First, [GLS22] use perfectly-robust protocols, whereas we use statistically-correct protocols. Thus, they obtain a PCP, whereas we obtain an IOP. We note that proving doom-checkability (more generally, efficient product verifiability) for a PCP is significantly easier than for IOPs. Second, they employ a large number of parties in the MPC to obtain a bad relation that is a product relation of sufficient arity. Such a strategy would not work for us, since we need to keep the number of parties small to obtain the desired efficiency advantages in our NIZKs of Corollary 7.8. (We note that the product relation of [GLS22] has better arity compared to ours'.) Finally, the resultant NIZKs achieve different tradeoffs between the parameters, underlying assumptions, and supported relation class, as discussed in the previous paragraph. See Remark B.14 for a more detailed comparison.

Composition Theorems for ZK-IOPs. Composition theorems for ZK-IOPs have been considered before in [VW26, CFW26]. Our composition theorems differ from these prior works in several respects. Compared to [CFW26], we consider *parallel* composition of *multiple* (possibly more than 2) *semi* HVZK-IOPs, and show RBR amplification. In contrast, [CFW26] focus on the *sequential* com-

position of *2 Interactive Oracle Reductions (IORs)*[9] with HVZK (and not semi-HVZK), and do not show soundness amplification. Jumping ahead, we note that the resultant proof system obtained by [CFW26]'s composition theorem has HVZK, whereas the stronger *semi* HVZK property (which allows the verifier's randomness to be skewed) is crucial for CIH-based constructions, in which the verifier's randomness is sampled using a CIH and is therefore *inherently* skewed. We further note that the HVZK of the underlying IORs is in fact insufficient for the composition theorem of [CFW26], and they need to require an enhanced ZK guarantee from the first IOR; see Section 5 for a more detailed comparison. [VW26] prove that *full-fledged* ZK (against query-bounded verifiers) is preserved under composition for public-coin IOPs, so long as ZK holds with a straight-line and black-box simulator. They consider concurrent composition of multiple IOPs, and as such capture parallel composition as a special case. However, their results do not immediately imply ZK preservation when the underlying IOPs satisfy only the milder *(semi)-HVZK* property. Additionally, they do not consider the effect of composition on other system properties (i.e., completeness, soundness, RBR soundness, and knowledge extraction). See Section 5 for further details.

Black-Box (Interactive) ZK Arguments from (semi) HVZK-PCPs and IOPs. Our constructions instantiate Kilian's transformation with semi-HVZK *IOPs*. As discussed in Section 2.1, several prior works instantiated the transformation with *PCPs* with ZK guarantees [IMS12, IMSX15]. Using a ZK PCP/IOP results in *black-box* ZK arguments, but the goals and ZK analysis in these works are quite different, as we now discuss. In terms of goals, [IMS12, IMSX15] aim to obtain *succinct* ZK arguments (whose proof length is sublinear in the length of the NP witness), whereas we focus on (non succinct) NIZKs as the final goal, and can therefore employ non-succinct commitments. In particular, such commitments can be equivocal (while still guaranteeing binding and even full extraction). We show that using equivocation, the natural straight-line simulation strategy works. Specifically, the argument simulator first generate bogus commitments, uses them to obtain the verifier's purported randomness r, then uses equivocation to open the commitments to the "right" values (i.e., the simulated IOP bits generated by the IOP simulator). In contrast, since the arguments in [IMS12, IMSX15] are succinct, they prove ZK using a rewinding-based strategy. More specifically, [IMS12] use semi-HVZK PCPs in which the verifier's randomness has logarithmic size. This allows the argument simulator to *guess* this randomness, generate a commitment + opening for this guess, then rewind in case it guessed incorrectly. Ishai et al. [IMSX15, Sec. 6.1] later removed the restriction on the randomness size in the *truly HVZK* setting (i.e., when the verifier's randomness is sampled uniformly at random). They present a *straight-line* simulator that samples r on its own. Unfortunately, this strategy does not extend to the *semi* HVZK setting we need in this work. Indeed, in the semi-HVZK setting, generating the correct distribution on r requires interacting with the argument verifier (who chooses r and can therefore skew its distribution), for which the simulator first needs to generate commitments to the proof. Then, to continue the simulation, it seems that the simulator would either need to rewind the verifier, or rely on some equivocation property of the commitment scheme. [IMSX15, Sec. 6.2] provide an analysis for PCPs with *full-fledged ZK*, where the verifier first commits to her queries, the simulator then commits to a bogus proof, and the verifier then opens her commitments. If the verifier successfully opened the commitments, the simulator can rewind and provide honestly-generated commitments to the "correct" proof bits. It is possible that the analysis can be relaxed to work for PCPs that only have semi-HVZK, however this is not discussed in [IMSX15]. Finally, we stress that the main point of difference between [IMS12, IMSX15] and the current work is that the former is for PCPs whereas we analyze the construction for IOPs.

[9] Informally, IORs are a generalization of IOPs in which the task of verifying membership in a relation $\mathcal{R}_1$ is reduced to verifying membership in a relation $\mathcal{R}_2$.

CIH Constructions and Provably-Secure Fiat-Shamir in the Plain Model. Canetti, Chen, and Reyzin [CCR16] construct CIH for sparse relations with apriori bounded polynomial complexity, assuming sub-exponentially secure indistinguishability obfuscation, puncturable pseudorandom functions, and input-hiding obfuscation for evasive circuits. A recent line of works showed that the Fiat-Shamir (FS) transform is *provably secure* when instantiated with certain types of proof systems, together with an appropriate CIH. Kalai, Rothblum and Rothblum [KRR17b] constructed CIH for constant-round interactive proofs under strong assumptions (specifically, the existence of sub-exponentially secure indistinguishability obfuscation, the existence of exponentially secure input-hiding obfuscation for the class of multi-bit point functions, and the existence of sub-exponentially secure one-way functions). Using their CIH, they show that applying FS to any constant-round interactive proof is secure. The underlying assumption needed for the CIH was improved by [CCRR18], who constructed CIH using KDM-security assumptions on secret-key encryption schemes (they also gave parameter settings where ElGamal [Gam85] encryption and Regev's encryption [Reg05] plausibly satisfy the assumption). While [CCRR18] did not assume obfuscation, their assumption still involves a computationally unbounded adversary. Holmgren and Lombardy [HL18] construct CIH for sparse relations, assuming sufficiently-hard "one way product functions".[10] Combining such functions with indistinguishability obfuscation, they obtain CIH for, roughly, constant-round public coin interactive proofs in which the prover can determine on which verifier messages he can cheat (such proof systems capture PCPs as a special case, via "commit-challenge-response" systems). Canetti, Lombardy and Wichs [CLW18] design CIHs for trapdoor Sigma protocols from circular-secure Fully-Homomorphic Encryption (FHE). Canetti et al. [CCH+19] design CIH for the GKR protocol [GKR08] (which has $\omega(1)$ rounds), assuming optimal security of the LWE-based FHE schemes from the literature (such as [BV11, Bra12, BGV12, GSW13, BV14]) against polynomial-size key-recovery attacks, and prove that applying FS to [GKR08] together with this hash, results in a secure (succinct) NI argument for log-space uniform NC. Peikert and Shiehian [PS19] construct a CIHs from plain LWE. The works of [CCH+19, PS19] apply to protocols in which the "bad" verifier challenge (that allows the prover to cheat) is not unique. Holmgren, Lombardi and Rothblum [HLR21] extend the class of protocols for which CIHs exist, by constructing CIHs for protocols in which there are polynomially many such bad challenges. Additional CIH constructions were given in [BKM20, CKU20, JJ21, CRT24], as discussed in the next paragraph.

Other NIZK Arguments in the Plain Model From CIH. Several of the aforementioned works constructed *provably secure* NIZK arguments in the CRS model by combining proof systems with mild ZK guarantees, and appropriate CIH. Canetti et al. [CCRR18] constructed NIZK arguments by combining their CIH with a 4-message computational HVZK proof system obtained using the FLS trick [FLS99]. This yields NIZK arguments for all NP under the same assumption as their CIH construction. Canetti et al. [CCH+19] construct NIZK arguments for NP from HVZK-PCPs[11] (alternatively, trapdoor Sigma protocols), PKE and their CIHs. They then instantiate their generic transformation using an HVZK-PCP based on (a parallel repetition of) [GMW91], and Regev's PKE scheme [Reg05], to obtain NIZK arguments for NP from search LWE. Peikert and Shiehian [PS19] instantiate the NIZK construction of [CCH+19] with their new CIH, to get NIZK arguments for NP from plain LWE. Couteau, Katsumata, and Ursu [CKU20] follow a similar approach to [CCH+19] towards designing a CIH, but their CIH are constructed for Sigma protocols for more limited lan-

[10] Intuitively, this is a tuple of functions $(f_1, \ldots, f_k)$ such that it is hard to find a pre-image of $(f_1(x_1), \ldots, f_k(x_k))$ for uniformly random $x_1, \ldots, x_k$.

[11] Their construction is based on a proof system that they call "commit-challenge-response" which, in an oracle model, can be thought of as a PCP.

guages (thus yielding NIZKs only for these limited languages), which are then bootstrapped to give NIZKs for NP. In this way, they get NIZK arguments assuming Computational Diffie-Hellman (CDH) and a KDM-type assumption on ElGamal. Holmgren and Lombardy [HL18] obtain NIZK arguments by applying their CIH for interactive proofs to an HVZK-PCP.[12] Canetti, Lombary and Wichs [CLW18] obtain NIZK arguments for NP assuming circular-secure FHE by combining their CIH with the Sigma protocol for graph Hamiltonicity of [FLS99]. Brakerskey, Koppula, and Mour [BKM20] construct NIZK arguments from Learning Parity with Noise (LPN), and trapdoor hash functions (which can be based on several standard assumptions, including DDH, QR, DCR, and LWE). Their NIZK arguments are constructed by considering a strengthening of CIH to relations given in *probabilistic* representation (i.e., which have a circuit that approximates them). They then exploit the fact that such probabilistic representations can be much simpler than exact representations, to design CIH for these relations from the aforementioned assumptions and combine them with Sigma-protocols in the FS transform to get NIZK. Several other papers constructed NIZK arguments with various properties by combining CIH and trapdoor Sigma protocols. Jain and Jin [JJ21] construct NIZK arguments for NP based on the sub-exponential hardness of Decisional Diffie-Hellman (DDH) against polynomial time adversaries. To obtain their NIZKs, they construct CIH for constant-depth threshold circuits, assuming sub-exponential DDH, then apply the NIZK to a trapdoor Sigma protocol. Holmgren et al. [HLR21] show that their CIH are applicable to a wider class of HVZK-PCPs[13] compared to [CCH+19, PS19]. This allows them to apply the "CIH + HVZK-PCP to NIZK" paradigm to a wider class of HVZK-PCPs. Chen, Rothblum and Tell [CRT24] show a transformation from HVZK-PCP to CZK arguments, which is similar to the transformation of Theorem 1.1, but *restricted to the PCP setting*. They then construct NIZK arguments for NP by combining a variant of CIH (which they call "list-CIH") with PCPs that have perfect *full-fledged* ZK.

(Non) Interactive Arguments from IOPs and Commitments. As discussed in Section 1, interactive argument constructions from IOPs and commitments/hashes (and non-interactive in the ROM) appear in several prior works. These works all follow the same general blueprint of our CZK arguments of Section 6, by having the argument prover commit/hash the oracle messages, then open the locations which the IOP verifier queries. More specifically, Kilian's [Kil92] construction is obtained by combining PCPs with Merkle-Hash Trees (MHTs). Ben-Sasson, Chiesa and Spooner [BCS16] extend this transformation by combining IOPs with MHTs, and additionally show that the argument is computational ZK if the underlying IOP has semi-HVZK. Their construction is actually described and analyzed as a (succinct) NIZK argument in the ROM. The analysis for the interactive case, but *without ZK*, and with *any* Vector Commitment (VC) scheme as the underlying commitment scheme, was given recently in [CDGS22]. (Chiesa et al. [CDGS22] also give the tightest soundness analysis to-date of Kilian's construction, and when applied to any VC.) We note that all these constructions achieve *succinct* arguments since they rely on VCs. The succinctness property significantly complicates the soundness analysis, since the underlying commitment scheme isn't fully extractable. In this work, we combine IOPs with *extractable and equivocal* commitments, obtaining provably-secure non-succinct computational ZK arguments with a conceptually simple security analysis.

4 Preliminaries

Notations. For a bit string $v \in \{0,1\}^n$ and an index $i \in [n]$, we use $v(i)$ to denote the i'th bit of v. More generally, for $\mathcal{T} \subseteq [n]$, we use $v(\mathcal{I})$ to denote the restriction of v to the indices in $\mathcal{I}$, i.e.,

[12] Their results are also described as using "commit-challenge-response" protocols.

[13] They call these "commit and open" protocols.

$v(\mathcal{I}) := \{v(i) \ : \ i \in \mathcal{I}\}$. For $i \in \mathbb{N}$, we use $[i]$ to denote the set $\{1, \ldots, i\}$.

Relations. An NP relation $\mathcal{R} = \mathcal{R}(x, w)$ is associated with a family of verification circuits $C = \{C_n\}_{n\in\mathbb{N}}$, where C_n is the verification circuit for instances x of length $|x| = n$. We sometimes abuse notation and use C to denote both the circuit family and the circuit C_n for length n (where n is clear from the context).

The LWE Assumption. Our constructions rely on the learning with errors problem [Reg03]:

Definition 4.1 (LWE)**.** *Let κ be a security parameter. The* Learning With Errors (LWE) assumption *with parameters $n = n(\kappa), m = m(\kappa), q = q(\kappa)$ and an error probability χ over $\mathbb{Z}_q$, states that $\left\{(A, \vec{b})\right\}_\kappa \approx \{(A, \vec{r})\}_\kappa$, where $A \leftarrow \mathbb{Z}_q^{n\times m}, \vec{r} \leftarrow \mathbb{Z}_q^m$, and $\vec{b^T} = \vec{s}^T A + \vec{e}^T \ (mod\ q)$ for $\vec{s} \leftarrow \mathbb{Z}_q^n$ and $\vec{e} \leftarrow \chi^m$. (ϵ, T)-LWE states that for any adversary running in time $\mathsf{poly}(\kappa, T)$, the advantage in distinguishing between these two distributions is $\leq \epsilon$. The* sub-exponential variant *of the LWE assumption states that for some $\epsilon > 0$, every adversary of size 2^{n^ϵ} obtains advantage $\leq 2^{-n^\epsilon}$ in distinguishing between these two distributions.*

Concentration Bounds.

Theorem 4.2 (Hoeffding's inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be independent random variables taking values in $\{0, 1\}$. Let $X = \sum_i X_i$, and $\mu = n \cdot \mathbb{E}[X]$. Then, for all $\epsilon > 0$,*

$$\Pr\left[\frac{1}{n}X \geq \mu + \epsilon\right] \leq \exp(-2\epsilon^2 n).$$

l-Wise Independent Functions.

Definition 4.3 (l-wise Independent Functions)**.** *Let $l \in \mathbb{N}$, and let $\mathsf{h}' = \{K : \{0,1\}^n \to \{0,1\}^m\}$ be a function family. h' is l-*wise independent *if for all distinct $x_1, \ldots, x_l \in \{0,1\}^n$ and all $y_1, \ldots, y_l \in \{0,1\}^m$, we have:*

$$\Pr_{K\leftarrow\mathsf{h}'}[\forall 1 \leq i \leq l \ y_i = K(x_i)] = 2^{-l\cdot m}$$

4.1 Commitment Schemes

Our construction will employ a bit commitment scheme that is both extractable and equivocal. Formally:

Definition 4.4 (Extractable Equivocal (EE) Bit Commitments)**.** *An* Extractable Equivocal Bit-Commitment Scheme *(*EE*) is a tuple* $(\mathsf{EE.setup}, \mathsf{EE.TDsetup}, \mathsf{EE.Com}, \mathsf{EE.Decom}, \mathsf{EE.Ver}, \mathsf{EE.Ext})$ *such that:*

- $(\mathsf{crs}, \tau_1) \leftarrow \mathsf{EE.setup}(1^\kappa)$. *On input a security parameter κ,* EE.setup *outputs a common random string* crs *and a trapdoor τ_1.*
- $(\mathsf{tdcrs}, \mathsf{c}^*, \sigma_0, \sigma_1) \leftarrow \mathsf{EE.TDsetup}(1^\kappa)$. *On input a security parameter κ,* EE.TDsetup *outputs a trapdoor-CRS* tdcrs *along with a commitment* c^* *and openings σ_0, σ_1.*
- $(\mathsf{c}, \sigma) \leftarrow \mathsf{EE.Com}(\mathsf{crs}, b)$. *On input a CRS* crs *and a bit $b \in \{0, 1\}$,* EE.Com *outputs a commitment* c *and an opening σ.*

- $b' \leftarrow \mathsf{EE.Ver}(\mathsf{crs}, \mathsf{c}, \sigma, b)$. *On input a CRS* crs*, a commitment* c*, an opening* σ *and a bit* b*,* EE.Ver *outputs* 1 *(i.e., accepting* b *as the value committed in* c*) or* 0 *(reject).*
- $b \leftarrow \mathsf{EE.Ext}(\tau_1, \mathsf{c})$. *On input the trapdoor* τ_1*, and a commitment* c*, the extraction algorithm outputs a bit* b*.*

We require that the scheme satisfy the following properties:

- **Correctness.** *For all* $\kappa \in \mathbb{N}$*, all* $\mathsf{crs} \in \mathsf{Supp}(\mathsf{EE.setup}(1^\kappa)) \cup \mathsf{Supp}(\mathsf{EE.TDsetup}(1^\kappa))$*, and all bits* $b \in \{0,1\}$*,*
$$\Pr\left[\mathsf{EE.Ver}(\mathsf{crs}, \mathsf{c}, \sigma, b) \ : \ (\mathsf{c}, \sigma) \leftarrow \mathsf{EE.Com}(\mathsf{crs}, b)\right] = 1.$$
- **Mode indistinguishability.** *For all (non-uniform) PPT* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, the following is negligible in* κ*:*
$$\left| \Pr\left[\mathcal{A}_2(\mathsf{st}, \mathsf{c}, \sigma) = 1 \ : \ \begin{array}{c}(\mathsf{crs}, \tau_1) \leftarrow \mathsf{EE.setup}(1^\kappa) \\ (b, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\kappa, \mathsf{crs}) \\ (\mathsf{c}, \sigma) \leftarrow \mathsf{EE.Com}(\mathsf{crs}, b)\end{array}\right] - \Pr\left[\mathcal{A}_2(\mathsf{st}, \mathsf{c}^*, \sigma_b) = 1 \ : \ \begin{array}{c}(\mathsf{tdcrs}, \mathsf{c}^*, \sigma_0, \sigma_1) \leftarrow \mathsf{EE.TDsetup}(1^\kappa) \\ (b, \mathsf{st}) \leftarrow \mathcal{A}_1(1^\kappa, \mathsf{tdcrs})\end{array}\right] \right|.$$
- ϵ**-Extractability.** *For any* $(\mathsf{crs}, \tau_1) \in \mathsf{Supp}(\mathsf{EE.setup}(1^\kappa))$*, any commitment* c*, any bit* b *and opening* σ*:*
$$\Pr\left[\mathsf{EE.Ver}(\mathsf{crs}, \mathsf{c}, \sigma, b) = 1 \implies \mathsf{EE.Ext}(\tau_1, \mathsf{c}) = b\right] \geq 1 - \epsilon$$

Remark 4.5 (Mode Indistinguishability Implies Indistinguishability Without Decommit)**.** *The mode indistinguishability property of Definition 4.4 gives a strong guarantee: one cannot distinguish honestly-generated CRS, commitment and opening, from a trapdoor-mode CRS generated together with a "bogus" commitment and an equivocated opening (that could be either to 0 or to 1). This property implies, as a special case, that honestly-generated CRS and commitments are computationally indistinguishable from a trapdoor-mode CRS and an equivocal (bogus) commitment.*

A commitment scheme with these properties and a negligible extractability error was constructed in [BWW24, Construction 4.2] where such a scheme is called "One-Time Dual-Mode Bit Commitment with Extraction". The construction is based on a Public-Key Encryption (PKE) scheme, and an equivocal bit commitment scheme (due to Naor [Nao89], constructed from one-way functions, which are implied by PKE). Specifically, we instantiate our EE with the following:

Theorem 4.6 (EE Scheme, [BWW24])**.** *Assume the existence of public-key encryption. Then there exists an EE scheme with* $\mathsf{negl}(\kappa)$*-extractability. Moreover, all algorithms of the EE scheme run in time* $\mathsf{poly}(\kappa)$*, and the length of the commitment and the opening (outputs of* EE.Com*) are both* $\mathsf{poly}(\kappa)$*.*

4.2 (ZK) Argument Systems

Definition 4.7 (Computational Zero-Knowledge Arguments with Setup)**.** *A* Computational Zero-Knowledge (CZK) Argument with setup *for an NP language* $\mathcal{L}$ *is a triple* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *of PPT algorithms satisfying the following properties.*

- **Syntax.**

- $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)$ *is a trusted PPT algorithm that on input a security parameter* κ *outputs a CRS* crs.
- **Protocol Structure.** *The prover* $\mathcal{P}$ *has input* x *with a corresponding witness* w*, the verifier* $\mathcal{V}$ *has input* x*, and both parties are given the CRS* crs*. The execution consists of multiple communication rounds, following which the verifier outputs either 1 (signaling accept) or 0 (signaling reject). We denote* $\mathcal{V}$*'s output in the execution by* $(\mathcal{P}(\mathsf{crs}, x, w), \mathcal{V}(\mathsf{crs}, x))$.

- **Semantics.** *The protocol satisfies the following semantic properties:*
  - **Completeness:** *For every* $(x, w) \in \mathcal{R}_\mathcal{L}$*,*
  $$\Pr_{\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)}[(\mathcal{P}(\mathsf{crs}, x, w), \mathcal{V}(\mathsf{crs}, x)) = 1] \geq 1 - \mathsf{negl}(\kappa)$$
  where the probability is also over the randomness of $\mathcal{P}, \mathcal{V}$.
  - **Soundness:** *If* $x \notin \mathcal{L}$*, then for any (possibly malicious) PPT prover strategy* $\mathcal{P}^*$*, when* $\mathcal{V}$ *interacts with* $\mathcal{P}^*$*,*
  $$\Pr_{\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)}[(\mathcal{P}^*(\mathsf{crs}, x), \mathcal{V}(\mathsf{crs}, x)) = 1] = \mathsf{negl}(\kappa)$$
  where the probability is also over the randomness of $\mathcal{P}^*, \mathcal{V}$.
  - **Computational Zero Knowledge (CZK):** *For every PPT verifier* $\mathcal{V}^*$ *there exists a PPT simulator* Sim *such that for every* $(x, w) \in \mathcal{R}_\mathcal{L}$*,*
  $$\mathsf{Sim}(1^\kappa, x) \approx \{\mathsf{View}_{\mathcal{V}^*}(\mathsf{crs}, x, w)\}_{\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)},$$
  where $\mathsf{View}_{\mathcal{V}^*}(\mathsf{crs}, x, w)$ *denotes* $\mathcal{V}^*$*'s view when interacting with the honest prover* $\mathcal{P}$ *that has input* (crs, x, w).[14]

The argument system of Definition 4.7 has *non-adaptive* soundness. We also consider the stronger variant of *adaptive* soundness, defined next:

Definition 4.8 (Adaptive Soundness — Arguments)**.** *We say that a CZK argument with setup* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *for* L *has* adaptive soundness *if for every (possibly malicious) PPT prover* $\mathcal{P}^*$*, the following holds:*
$$\Pr\left[\begin{matrix}(\mathcal{P}^*(\mathsf{st}, x), \mathcal{V}(\mathsf{crs}, x)) = 1 \\ \wedge\, x \notin L\end{matrix} : \begin{matrix}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) \\ (\mathsf{st}, x) \leftarrow \mathcal{P}^*(\mathsf{crs})\end{matrix}\right] = \mathsf{negl}(\kappa)$$

We follow the NIZK definition of [GOS12, Sec. 2.1].

Definition 4.9 (NIZK)**.** *A* Non-interactive Zero-Knowledge Argument *for an NP language* $\mathcal{L}$ *consists of a tuple* $(\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *of PPT algorithms satisfying the following properties.*

- **Syntax.**
  - $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)$ *is a PPT algorithm that on input a security parameter* κ *outputs a CRS* crs.
  - $\mathcal{P}$ *takes as input the CRS* crs*, and a statement* x *with a corresponding witness* w *such that* $(x, w) \in \mathcal{R}_\mathcal{L}$ *and outputs a proof* π.
  - $\mathcal{V}$ *takes as input the CRS* crs*, input* x*, and proof* π*, and outputs either* 1 *(signaling accept) or* 0 *(signaling reject).*

[14] The view consists of crs, x, $\mathcal{V}^*$'s random coin tosses, and the messages she received from $\mathcal{P}$.

- **Semantics.** *The protocol satisfies the following semantic properties:*
  - **Completeness:** *For every* $(x, w) \in \mathcal{R}_{\mathcal{L}}$,

$$\Pr\left[\mathcal{V}(\mathsf{crs}, x, \pi) = 1 \ : \ \begin{matrix} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) \\ \pi \leftarrow \mathcal{P}(\mathsf{crs}, x, w) \end{matrix}\right] \geq 1 - \mathsf{negl}(\kappa)$$

  - **Soundness:** *For any (possibly malicious) PPT prover strategy* $\mathcal{P}^*$,

$$\Pr\left[\mathcal{V}(\mathsf{crs}, x, \pi) = 1 \wedge x \notin \mathcal{L} \ : \ \begin{matrix} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa) \\ (x, \pi) \leftarrow \mathcal{P}^*(\mathsf{crs}) \end{matrix}\right] = \mathsf{negl}(\kappa)$$

  - **Computational Zero Knowledge (CZK):** *There exists a PPT simulator* Sim *such that for every* $(x, w) \in \mathcal{R}_{\mathcal{L}}$*, the following distributions are* negl*-computationally close:*
    - (crs, π)*, where* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\kappa)$*, and* $\pi \leftarrow \mathcal{P}(\mathsf{crs}, x, w)$.
    - (crs^*, π^*) *which is the output of* $\mathsf{Sim}(1^\kappa, x)$.

Definition 4.10 (NIZK AoK)**.** *A NIZK is an argument of knowledge (AoK) if in addition to the properties in Definition 4.9, the following is satisfied: there exists an additional PPT trapdoor setup algorithm that on input a security parameter* κ *outputs a CRS* crs*, and a trapdoor* td*,* $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{TdSetup}(1^\kappa)$ *such that the following properties are satisfied:*

- **CRS indistinguishability:** *For any PPT prover* $\mathcal{P}^*$,

$$\Pr\left[\mathcal{P}^*(\mathsf{crs}_1) = 1 \ : \ \mathsf{crs}_1 \leftarrow \mathsf{Setup}(1^\kappa)\right] = \Pr\left[\mathcal{P}^*(\mathsf{crs}_2) = 1 \ : \ (\mathsf{crs}_2, \mathsf{td}) \leftarrow \mathsf{TdSetup}(1^\kappa)\right]$$

- **Extraction:** *There exists a PPT extractor* E *such that for any PPT prover strategy* $\mathcal{P}^*$*:*

$$\Pr\left[\mathcal{V}(\mathsf{crs}, x, \pi) = 1 \wedge (x, w^*) \notin \mathcal{R}_{\mathcal{L}} \ : \ \begin{matrix} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{TdSetup}(1^\kappa) \\ (x, \pi) \leftarrow \mathcal{P}^*(\mathsf{crs}) \\ w^* \leftarrow \mathsf{E}(\mathsf{td}, x, \pi) \end{matrix}\right] = \mathsf{negl}(\kappa)$$

4.3 Interactive Oracle Proofs (IOPs) and Honest-Verifier Zero-Knowledge (HVZK) IOPs

In this section we define IOPs, focusing on IOPs for NP languages. This captures languages in P as a special case (by omitting the NP witness), and can also be generalized to languages beyond NP. We only consider *public-coin* IOPs, in which the verifier's messages are random, and the verifier does not have any private randomness. Throughout this work, we consider languages in which the input is given *in full* to the verifier, since this suffices for our results on CZK arguments and NIZKs. The more general case in which part of the input is given *implicitly* to the verifier (as an oracle), has also been considered in the literature. We discuss this setting in Section A.2, where we show that our composition theorems for IOPs extend also to this more general setting.

Definition 4.11 (Interactive Oracle Proof (IOP))**.** *An* l*-round* (public coin) Interactive Oracle Proof (IOP) with soundness error ϵ_{S} and complentess error ϵ_{C} for a language $\mathcal{L}$ *with corresponding NP relation* $\mathcal{R}$ *is a pair* $(\mathcal{P}, \mathcal{V})$ *of PPT algorithms satisfying the following properties.*

- **Syntax.**
  - ***Inputs:*** *Prover* $\mathcal{P}$ *receives as input* x *with a corresponding witness* w*. Verifier* $\mathcal{V}$ *has input* x*.*

– **Protocol Structure.** *The protocol consists of a* Communication Phase *followed by a* Query Phase*. In the* Communication phase*, $\mathcal{P}$ and $\mathcal{V}$ interact for l rounds, where $\mathcal{V}$'s messages are uniformly random strings $r_1, \ldots, r_l$, and $\mathcal{P}$'s messages are* oracle *messages. In the* Query phase*, $\mathcal{V}$ makes (non-adaptive) queries to $\mathcal{P}$'s oracle messages. These queries are determined solely by the input x and $\mathcal{V}$'s randomness string $r := (r_1, \ldots, r_l)$. $\mathcal{V}$ then deterministically decides whether to accept or reject based on the answers to these queries, the input x, and the randomness strings $r_1, \ldots, r_l$.*

- **Semantics.** *The protocol satisfies the following semantic properties:*
  - **Completeness:** *If $(x, w) \in \mathcal{R}$, then when $\mathcal{V}$ interacts with $\mathcal{P}$, she accepts with probability at least $1 - \epsilon_{\mathsf{C}}$.*
  - **Soundness:** *If $x \notin \mathcal{L}$, then for any (possibly unbounded) prover strategy $\mathcal{P}^*$, when $\mathcal{V}$ interacts with $\mathcal{P}^*$, she accepts with probability at most ϵ_{S}.*

It will sometimes be beneficial to have the soundness and completeness error depend on a statistical security parameter λ, in which case both parties take 1^λ as input, and $\epsilon_{\mathsf{C}}, \epsilon_{\mathsf{S}}$ are functions of λ.

A Probabilistically Checkable Proof (PCP) is a special case of an IOP in which the communication phase has only one round.

Notation 4.12 (Proof length)**.** *We say that an IOP system $(\mathcal{P}, \mathcal{V})$ has* proof length ℓ *if every oracle message sent by $\mathcal{P}$ has proof length ℓ.*[15]

In this work, we consider IOPs with a weak zero-knowledge guarantee known as *semi Honest-Verifier Zero-Knowledge (semi-HVZK)*. Intuitively, this means that there exists a PPT simulator Sim such that for every possible randomness r of the *honest* verifier $\mathcal{V}$, Sim on input x and r can perfectly simulate the honest verifier's view. Before formally defining this property, we need to set some notation.

Definition 4.13 (Verifier View (IOP))**.** *Let $\mathcal{L}$ be an NP language with a corresponding NP relation $\mathcal{R} = \mathcal{R}(x, w)$. Let $(\mathcal{P}, \mathcal{V})$ be an IOP for $\mathcal{L}$, and let $\mathcal{V}^*$ be a (possibly malicious) verifier. For every $(x, w) \in \mathcal{R}$, the* view $\mathsf{View}_{\mathcal{V}^*}(x, w)$ *of $\mathcal{V}^*$ in the interaction with $\mathcal{P}$ consists of x (and also λ, if the system depends on a statistical security parameter), $\mathcal{V}^*$'s random coins, and the answers to $\mathcal{V}^*$'s queries to $\mathcal{P}$'s oracle messages.*

Definition 4.14 (Semi Honest-Verifier Zero-Knowledge IOP (semi-HVZK-IOP))**.** *Let $\mathcal{L}$ be an NP language with a corresponding NP relation $\mathcal{R} = \mathcal{R}(x, w)$. We say that an IOP $(\mathcal{P}, \mathcal{V})$ for $\mathcal{L}$ has* semi Honest-Verifier Zero-Knowledge (semi-HVZK) *if there exists a PPT simulator* Sim *such that for every $(x, w) \in \mathcal{R}$, and every possible randomness string r for $\mathcal{V}$, the following are identically distributed.*

- *The view $\mathsf{View}_{\mathcal{V}}(x, w; r)$, which is $\mathcal{V}$'s view conditioned on $\mathcal{V}$ using randomness r.*
- *The output $\mathsf{Sim}(r, x)$ of the simulator, when given the verifier's randomness r and the input x.*

We say that $(\mathcal{P}, \mathcal{V})$ has statistical semi-HVZK *if all parties also take as input a statistical security parameter λ (in unary), and there exists a negligible function $\epsilon_{\mathsf{ZK}}(\lambda)$ such that for every x, w, r as above, the real and simulated views are $\epsilon_{\mathsf{ZK}}(\lambda)$-statistically close.*

Next, we define a strong soundness guarantee for IOPs, which is called "round by round soundness". For this, we first need to define the notion of partial transcripts:

[15] We assume without loss of generality that all prover oracle messages have the same length, otherwise they can be augmented to the length of the longest prover message.

Definition 4.15 (Partial Transcripts). *Let $(\mathcal{P}, \mathcal{V})$ be an l-round IOP for a relation $\mathcal{R}$. We let $\mathcal{M}_1, \ldots, \mathcal{M}_l$ denote the sets of all potential prover messages such that in any round $i \in [l]$, $\mathcal{P}$ sends a message from $\mathcal{M}_i$. Similarly, let $\mathcal{C}_1, \ldots, \mathcal{C}_l$ denote the sets of all potential verifier challenges such that in any round $i \in [l]$, $\mathcal{V}$ sends a challenge from $\mathcal{C}_i$ sampled uniformly at random. For $i \in [l]$, an i-round partial transcript is a vector of the form $\mathcal{T} = (\Pi_1, r_1, \ldots, \Pi_i, r_i)$ where $\Pi_j \in \mathcal{M}_j$ and $r_j \in \mathcal{C}_j$ for all $j \in [i]$. We also let a 0-round partial transcript be a vector of the form $(x, \emptyset)$. We write $\mathsf{PartTr}(i)$ to denote the set of all round-i partial transcripts. A* complete transcript *is a transcript of the form $(x, \mathcal{T})$ where $(x, \mathcal{T}) \in \mathsf{PartTr}(l)$. Such a transcript is* accepting *if $\mathcal{V}^{\Pi_1, \ldots, \Pi_l}(x, r_1, \ldots, r_l) = 1$.*

We consider the notion of generalized round-by-round soundness (RBR soundness).

Definition 4.16 (Generalized Round-by-round (RBR) Soundness [BGTZ23]). *An IOP $\Pi = (\mathcal{P}, \mathcal{V})$ for a relation $\mathcal{R}$ has (generalized) Round-By-Round (RBR) soundness* with error ϵ_{RBR} *if there exists a (not necessarily efficiently computable) "*doomed set*" $\mathcal{D}$ of partial transcripts such that the following properties hold:*

1. If $x \notin \mathcal{L}_{\mathcal{R}}$, then $(x, \emptyset) \in \mathcal{D}$, where $\emptyset$ denotes the empty transcript.
2. For any x and any l-round transcript $(x, \mathcal{T}) \in \mathsf{PartTr}(l)$, if $(x, \mathcal{T}) \in \mathcal{D}$ then $\mathcal{V}(x, \mathcal{T}) = 0$.
3. If $(x, \mathcal{T}) \in \mathsf{PartTr}(i-1)$ is an $(i-1)$-round partial transcript for some $i \in [l]$ and $(x, \mathcal{T}) \in \mathcal{D}$, then for all $\Pi \in \mathcal{M}_i$ we have
$$\Pr_{r \leftarrow \mathcal{C}_i}[(x, \mathcal{T}, \Pi, r) \notin \mathcal{D}] \leq \epsilon_{\mathsf{RBR}}.$$

In this case, we say that $(\mathcal{P}, \mathcal{V})$ has RBR soundness with the doomed set $\mathcal{D}$. *Similar to standard soundness, we also allow the RBR soundness error to depend on the statistical security parameter, i.e., $\epsilon_{\mathsf{RBR}} = \epsilon_{\mathsf{RBR}}(\lambda)$.*

For our NIZK construction in Section 7, we will need the IOP to satisfy an extraction property which, roughly, states that if an NP witness cannot be efficiently extracted from the first prover message on $x \in \mathcal{L}$, then with high probability over the verifier's round-1 challenge, the round-1 partial transcript is doomed. This property is satisfied by the IOP construction described in Appendix B.

Definition 4.17 (Knowledge Extractor (IOPs)). *Let $\epsilon_{\mathsf{E}} \in (0, 1)$, and let $(\mathcal{P}, \mathcal{V})$ be an IOP for a relation $\mathcal{R}$ with RBR soundness with a doomed set $\mathcal{D}$. We say that $(\mathcal{P}, \mathcal{V})$ has* knowledge extraction with error ϵ_{E} *if there exists an efficient deterministic extraction algorithm E such the following holds. For every $x, \Pi_1 \in \{0,1\}^*$, if $(x, \mathsf{E}(x, \Pi_1)) \notin \mathcal{R}$ then*
$$\Pr_{r \leftarrow \mathcal{C}_1}[(x, \Pi_1, r) \notin \mathcal{D}] \leq \epsilon_{\mathsf{E}}.$$
where $\mathcal{C}_1$ denotes the sets of all potential round-1 verifier challenges.[16] *We also allow the knowledge soundness error to depend on the statistical security parameter, i.e., $\epsilon_{\mathsf{E}} = \epsilon_{\mathsf{E}}(\lambda)$.*

Remark 4.18 (Connection Between PCP- and IOP-Extractability). *We note that Definition 4.17 captures as a special case the notion of knowledge extraction for* PCPs *considered in [CJJ21, Sec. 4.2]. Indeed, knowledge extraction for PCPs requires that if the PCP verifier accepts the purported proof Π^* with probability $> \epsilon_{\mathsf{E}}$ then the extractor successfully extracts a witness. Notice that for the special case of PCPs in Definition 4.17, the transcript (Π_1, r) is the* full *transcript, and so it is not doomed if and only if the PCP verifier accepts. Therefore, the condition $\Pr[(x, \Pi_1, r) \notin \mathcal{D}] \leq \epsilon_{\mathsf{E}}$ is equivalent to the condition that the PCP verifier accepts with probability $\leq \epsilon_{\mathsf{E}}$. The counter positive of Definition 4.17 guarantees that if $\Pr[(x, \Pi_1, r) \notin \mathcal{D}] > \epsilon_{\mathsf{E}}$ (i.e., the PCP verifier accepted with probability $> \epsilon_{\mathsf{E}}$) then extraction succeeds, as defined in the PCP extraction property of [CJJ21].*

[16] We note that the definition is not trivial, in the sense that E cannot simply output an invalid witness *for every* x, Π_1. Indeed, if $(x, w) \in \mathcal{R}$ and Π_1 is a valid 1-round prover message for (x, w), then it must be the case that $(x, \mathsf{E}(x, w)) \in \mathcal{R}$ (i.e., E successfully extracted a valid witness), otherwise extractability contradicts completeness.

4.4 Correlation Intractability

Definition 4.19 (Correlation Intractable Hash). *Let $\mathcal{R} = \{\mathcal{R}_\kappa\}_{\kappa\in\mathbb{N}}$, where each $\mathcal{R}_\kappa$ is an ensemble of relations. A* Correlation Intractable Hash (CIH) for $\mathcal{R} = \{\mathcal{R}_\kappa\}_{\kappa\in\mathbb{N}}$ *is a function family* $\mathsf{Hash} = (\mathsf{Gen}, \mathsf{Eval})$ *with the following properties:*

- **Syntax.**
  - $K \leftarrow \mathsf{Gen}(1^\kappa)$*:* Gen *is a PPT algorithm that takes as input a security parameter 1^κ, and outputs a function key K.*
  - $y = \mathsf{Eval}(K, x)$*:* Eval *is a polynomial-time algorithm that takes as input a key K, and an input $x \in \{0,1\}^{n(\kappa)}$, and outputs an output $y \in \{0,1\}^\kappa$.*
- **Correlation Intractability.** *For every non-uniform polynomial-size adversary $\mathcal{A} = \{\mathcal{A}_\kappa\}$ there exists a negligible function $\epsilon = \epsilon(\kappa)$ such that for every $R \in \mathcal{R}_\kappa$:*

$$\Pr[(x, \mathsf{Hash.Eval}(K, x)) \in R \ : \ K \leftarrow \mathsf{Hash.Gen}(1^\kappa), x \leftarrow \mathcal{A}_\kappa(K)] \leq \epsilon(\kappa).$$

We will use CIH for *sparse efficiently verifiable approximate*product relations [HLR21] that we now define.

Definition 4.20 (Product Relations [HLR21]). *A relation $\mathcal{R} \subseteq X \times Y^T$ is a* Product Relation *if for any x, the set $\mathcal{R}_x := \{y \ : \ (x, y) \in \mathcal{R}\}$ is the Cartesian product $\mathcal{R}_x = \mathcal{R}_{1,x} \times \mathcal{R}_{2,x} \times \ldots \times \mathcal{R}_{T,x}$ for sets $\mathcal{R}_{1,x}, \mathcal{R}_{2,x}, \ldots, \mathcal{R}_{T,x} \subseteq Y$.*

Definition 4.21 ((Approximate) Product Relations [HLR21]). *A relation $\mathcal{R} \subseteq X \times Y^T$ is an α-*Approximate Product Relation *if for any x, the set $\mathcal{R}_x := \{y \in Y^T \ : \ (x, y) \in \mathcal{R}\}$ consists of exactly those $y \in Y^T$ for which*

$$|\{i \in [T] : y_i \in \mathcal{R}_{i,x}\}| \geq \alpha T$$

for some sets $\mathcal{R}_{1,x}, \mathcal{R}_{2,x}, \ldots, \mathcal{R}_{T,x} \subseteq Y$.

Definition 4.22 (Product Sparsity [HLR21]). *A product (respectively, α-approximate product) relation $\mathcal{R} \subseteq X \times Y^T$ has* sparsity ρ *if for any x, the sets $\mathcal{R}_{1,x}, \ldots, \mathcal{R}_{T,x}$ (as in Definition 4.20) (respectively, Definition 4.21) have size $|\mathcal{R}_{i,x}| \leq \rho \cdot |Y|$.*

Definition 4.23 (Efficient Product Verifiability [HLR21]). *A (α-approximate) relation $\mathcal{R}$ is* efficient product verifiable *if there exists a polynomial-size circuit C such that for any x, the sets $\mathcal{R}_{1,x}, \mathcal{R}_{2,x}, \ldots, \mathcal{R}_{T,x}$ (as in Definitions 4.20 and 4.21) satisfy that for any $i \in [T]$: $y_i \in \mathcal{R}_{i,x}$ if and only if $C(x, y_i, i) = 1$. $\mathcal{R}$ is T-*time product verifiable *if the circuit C has size T.*

We recall the following theorem from [HLR21], which shows that assuming the hardness of Learning With Errors (LWE), efficient product verifiable approximate relations have a CIH.

Theorem 4.24 (CIH for Efficient Product Verifiable Relations, [HLR21], Thm. 6.1). *Let $R \subseteq X \times Y^t$ be a time-T verifiable α-approximate product relation with product sparsity at most $\rho < \alpha$.*

Set $\kappa = t \cdot (\alpha - \rho)^3$. Then there exists a hash family $\mathsf{h} = \{h : X \to Y^t\}$ that is $(T + \mathsf{poly}(\kappa), \epsilon \cdot \mathsf{poly}(\kappa))$-correlation intractable for R assuming (ϵ, T)- LWE. Moreover, h depends only on X, Y, T, t and α.

In [HLR21], Theorem 4.24 is stated for a single fixed relation, and we restate it below for an ensemble of relations. We note that the proof of Theorem 4.24 in [HLR21] is by combining a family of list-recoverable codes in the presence of errors, which they construct, with a family of CIHs. Hence the theorem holds for a relation ensemble as well.

Theorem 4.25 (CIH for Efficient Product Verifiable Relation Ensemble)**.** *Let κ be a security parameter, and let $\mathcal{R} = \{\mathcal{R}_\kappa \subseteq X_\kappa \times Y_\kappa^{t_\kappa}\}_\kappa$ where each $\mathcal{R}_\kappa$ is an ensemble of α-approximate product relations that are time-$T(\kappa)$ product-verifiable with product sparsity at most $\rho < \alpha$. Suppose that: $|Y_\kappa|,\ \log|X_\kappa|,\ T(\kappa),\ t_\kappa$ are all upper bounded by $\kappa^{O(1)}$, and $t_\kappa \geq \frac{\kappa}{(\alpha-\rho)^3}$.*

Then there exists a hash family $\mathsf{h} = \{h_\kappa : X_\kappa \to Y_\kappa^{t_\kappa}\}_\kappa$ that is $(T + \mathsf{poly}(\kappa), \epsilon \cdot \mathsf{poly}(\kappa))$-correlation intractable for $\mathcal{R}$ assuming ϵ-hardness of LWE for adversaries running in time $\mathsf{poly}(T, \kappa)$. Moreover, h_κ depends only on $X_\kappa, Y_\kappa, T, t_\kappa$ and α, and can be evaluated in time $\mathsf{poly}(\log|X_\kappa|, |Y_\kappa|, T, t_\kappa)$.

The above theorems apply only when $|Y| = \kappa^{\omega(1)}$. [HLR21] achieve correlation intractability even for large values of $|Y|$ by sub-sampling a random alphabet $\widetilde{Y} \subseteq Y$, and restricting the relation to this sub-alphabet. We state the sub-sampling lemma from [HLR21] below.

Theorem 4.26 (Sub-sampling, [HLR21], Lemma 3.8)**.** *Let $R \subseteq X \times Y^t$ be an α-approximate product relation with product sparsity ρ. Let $\eta > 0$ and $\lambda \in \mathbb{Z}^+$, and let $\widetilde{Y} \subseteq Y$ be a uniformly random subset of size*

$$q \geq \frac{\log|X| + \log t + \lambda}{\eta^2}.$$

Then, the restricted relation $R_{\widetilde{Y}} := R \cap \left(X \times \widetilde{Y}^t\right)$ is an α-approximate product relation that, with probability at least $1 - 2^{-\lambda}$ over the choice of $\widetilde{Y}$, has product sparsity at most $\rho + \eta$.

[HLR21] conclude that sub-sampling gives a reduction from CI over large alphabets to CI over polynomial-size alphabets. We state this corollary for an ensemble of relations.

Theorem 4.27 (CIH by Subsampling, [HLR21], Corollary 3.9, stated for relation ensembles)**.** *Let $\mathcal{R} = \{\mathcal{R}_\kappa \subseteq X_\kappa \times Y_\kappa^{t_\kappa}\}_\kappa$ where each $\mathcal{R}_\kappa$ is an ensemble of α-approximate product relations with product sparsity ρ, and let $q = q(\kappa)$ be an integer such that $q \geq \frac{\log|X_\kappa| + \log t_\kappa + \kappa}{\eta^2}$ for some $\eta > 0$. Suppose that for each $\tilde{Y}_\kappa \in \binom{Y_\kappa}{q}$, $\mathsf{h}_{\tilde{Y}} = \{h_\kappa : X_\kappa \to \tilde{Y}_\kappa^{t_\kappa}\}_\kappa$ is a correlation intractable hash family for α-approximate product relations with product sparsity $\rho + \eta$. Then, the hash family h, where $h \leftarrow \mathsf{h}$ is sampled by $h \leftarrow \mathsf{h}_{\tilde{Y}}$, for uniformly random $\tilde{Y} \leftarrow \binom{Y}{q}$ is correlation intractable for $\mathcal{R}$.*

We note that Theorem 4.27 was stated in [HLR21] for a single relation R, however it extends to ensemble relations. Indeed, the theorem in [HLR21] is obtained by first applying Theorem 4.26 to R to prove that the sub-sampled relation $R_{\widetilde{Y}}$ is still a sparse approximate product relation, and then applying Theorem 4.25 to $R_{\widetilde{Y}}$. Given an ensemble relation $\mathcal{R}$, we can first apply Theorem 4.26 to obtain the relation $R_{\widetilde{Y}}$ that is sparse with high probability, then apply Theorem 4.25 to the ensemble relation obtained in this way. (A similar analysis appears at the end of the proof of Theorem 4.28 below.)

Holmgren, Lombardi and Rothblum prove [HLR21, Thm. 5.5] that under the LWE assumption, efficiently-verifiable sparse product relations over large alphabets (i.e., when $Y_\kappa^{t_\kappa}$ has super-polynomial size) have CIHs. In the following theorem, we extend their results to *approximate* product relations.

Theorem 4.28 (CIH for Approximate Product Relations over Large Alphabets)**.** *Let $\mathcal{R} = \{\mathcal{R}_\kappa \subseteq X_\kappa \times Y_\kappa^{t_\kappa}\}_\kappa$ where each $\mathcal{R}_\kappa$ is an ensemble of (α-approximate) product relations that is time-$T(\kappa)$ product-verifiable (where $\log|X_\kappa|, T, t = \mathrm{poly}(\kappa)$) with product sparsity $\rho < \alpha$.*

Then, if $\alpha - \rho \geq \kappa^{-O(1)}$, and $t_\kappa \geq \frac{8\kappa}{(\alpha-\rho)^3}$, there exists a hash family $\mathsf{h} = \{\mathsf{h}_\kappa : X_\kappa \to Y_\kappa^{t_\kappa}\}_\kappa$ that is correlation intractable for $\mathcal{R}$, under the LWE assumption. Furthermore, h depends only on $X_\kappa, Y_\kappa, T, t_\kappa, \alpha$ and ρ, and can be evaluated in time $\mathsf{poly}\left(\kappa, t_\kappa, T(\kappa), \log|X_\kappa|\right)$.

Proof: Set $\delta := \alpha - \rho$, and $\eta = \delta/2$. Define

$$q \;=\; \left\lceil \frac{\log |X_\kappa| + \log t_\kappa + \kappa}{\eta^2} \right\rceil .$$

Since $\log |X_\kappa|, t_\kappa = \mathrm{poly}(\kappa)$ and $\eta = \delta/2 \geq \kappa^{-O(1)}$ (by the assumption $\delta \geq \kappa^{-O(1)}$), we have $q \leq \mathrm{poly}(\kappa)$.

For every κ, fix some $\widetilde{Y}_\kappa \in \binom{Y_\kappa}{q}$. Let $\widetilde{\mathcal{R}}$ be the ensemble obtained from $\mathcal{R}$ as follows: $\widetilde{\mathcal{R}} := \{\mathcal{R}_\kappa \cap X_\kappa \times (\widetilde{Y}_\kappa)^{t_\kappa} \;:\; \mathcal{R}_\kappa \in \mathcal{R}\}_\kappa$. Let $\rho' := \rho + \eta = \rho + \delta/2$, and let $\widetilde{\mathcal{R}}' := \{R \in \widetilde{\mathcal{R}} \;:\; R \text{ is } \rho' \text{ sparse}\}$. Then $\widetilde{\mathcal{R}}'$ is an ensemble of α-approximate product relations of sparsity at most ρ'. We now show that we can apply Theorem 4.25 to it, by verifying that the preconditions required by Theorem 4.25 are satisfied:

- *Polynomial alphabet:* $|\widetilde{Y}_\kappa| = q = \mathrm{poly}(\kappa)$. ✓
- *Gap:* $\alpha - \rho' = \alpha - \rho - \delta/2 = \delta - \delta/2 = \delta/2 > 0$. ✓
- *Number of repetitions:* Theorem 4.25 requires

  $$t_\kappa \;\geq\; \frac{\kappa}{(\alpha - \rho')^3} = \frac{\kappa}{(\delta/2)^3}$$

  By our hypothesis $t_\kappa \geq 8\kappa/\delta^3$. ✓
- *Efficient verifiability:* $R_{\widetilde{Y}_\kappa}$ is verifiable in time $T' = T + |\widetilde{Y}_\kappa| \cdot \log |Y_\kappa| = T + \mathrm{poly}(\kappa) = \mathrm{poly}(\kappa)$. ✓

Therefore, by Theorem 4.25 there exists a hash family $\mathsf{h} = \{\mathcal{H}_{\widetilde{Y}_\kappa} : X_\kappa \to \widetilde{Y}_\kappa^{t_\kappa}\}$ that is correlation intractable for $\widetilde{\mathcal{R}}'$, and moreover h is independent of $\widetilde{\mathcal{R}}'$. By Corollary 4.27 (with the parameter $\eta = \delta/2$ and sub-alphabet size q), the hash family

$$\mathcal{H} \;:\; \text{sample } \widetilde{Y}_\kappa \leftarrow \tbinom{Y_\kappa}{q}, \text{ then sample } h \leftarrow \mathcal{H}_{\widetilde{Y}_\kappa}$$

is CI for $\mathcal{R}' := \{\mathcal{R}_\kappa \in \mathcal{R} \;:\; \text{s.t. } \mathcal{R}_\kappa \cap X_\kappa \times (\widetilde{Y}_\kappa)^{t_\kappa} \text{ is } \rho' \text{ sparse}\}$. We show that this implies that $\mathcal{H}$ is also CI for the original $\mathcal{R}$. Indeed, for any $R_\kappa \in \mathcal{R}$, by Theorem 4.26, $\mathcal{R}_\kappa \cap X_\kappa \times (\widetilde{Y}_\kappa)^{t_\kappa}$ is ρ' sparse with probability $\geq 1 - 2^{-\kappa}$ over the choice of a random $\widetilde{Y}_\kappa$. Therefore, except with probability $2^{-\kappa}$, $\mathcal{R}_\kappa \in \mathcal{R}'$, in which case any PPT adversary breaks CI for it with probability at most $\mathsf{negl}(\kappa)$ over the choice of $\mathcal{H}$. Therefore, any PPT breaks CI for $\mathcal{R}_\kappa$ with probability at most $\mathsf{negl}(\kappa) + 2^{-\kappa} = \mathsf{negl}(\kappa)$. ■

5 Parallel Repetition of semi-HVZK-IOP

We describe a parallel-repetition version of an IOP, and show that semi-HVZK is closed under parallel repetition in the IOP setting. This parallel-repetition version will be used in Section 7 to obtain a BAD-challenge relation for the CZK argument system that is a product relation.

Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be the underlying base-IOP. We construct $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$ below which is the parallel repetition version, namely, $\mathcal{P}^\otimes, \mathcal{V}^\otimes$ emulate multiple random and independent executions of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. In this section, we consider semi-HVZK *without* implicit inputs (as defined in Section 4.3), since that suffices for our end goal of constructing a NIZK. However, as composition theorems for semi-HVZK-IOPs may be of independent interest, we extend this result to IOPs with implicit input in Appendix A. We now formally define the parallel repetition.

Construction 5.1 (Parallel Repetition IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$)**.** *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be a semi-HVZK-IOP for a language $\mathcal{L}$, and let $t_\lambda \in \mathbb{N}$. We construct the t_λ-fold parallel repetition $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$, which is an IOP for $\mathcal{L}$, as follows. The prover $\mathcal{P}^{\otimes}$ receives as input x with a corresponding witness w. The verifier $\mathcal{V}^{\otimes}$ receives as input x. (If the underlying system $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ also depends on a statistical security parameter λ then both parties also take 1^λ as input.) $\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes}$ emulate t_λ random and independent copies of $\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}}$ (respectively), with inputs (x, w) and x, respectively. Specifically, in the* Communication phase*, in each round $i \in [l]$, $\mathcal{P}^{\otimes}$ invokes t_λ independent runs of $\mathcal{P}_{\mathsf{IOP}}$ to obtain oracle messages $\Pi^i_1, \ldots, \Pi^i_{t_\lambda}$. It computes $\widehat{\Pi}^i = (\Pi^i_j)_{j \in [t_\lambda]}$ and sends $\widehat{\Pi}^i$ to $\mathcal{V}^{\otimes}$. $\mathcal{V}^{\otimes}$ invokes t_λ independent runs of $\mathcal{V}_{\mathsf{IOP}}$ to obtain and send uniformly random strings $r^i_1, \ldots, r^i_{t_\lambda}$ for each round $i \in [l]$. In the* Query phase*, $\mathcal{V}^{\otimes}$ determines the queries that $\mathcal{V}_{\mathsf{IOP}}$ would make to each of $\mathcal{P}^{\otimes}$'s oracle messages given the input x and randomness string $r_j := (r^1_j, \ldots, r^l_j)$ for each $j \in [t_\lambda]$. $\mathcal{V}^{\otimes}$ accepts if $\mathcal{V}_{\mathsf{IOP}}$ accepts for each $j \in [t_\lambda]$. Otherwise, $\mathcal{V}^{\otimes}$ rejects.*

We now analyze the properties of the IOP system $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ of Construction 5.1. For this, we first need to define a doomed set for the parallel-repetition IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$. As shown in [HLR21, Lemma 6.9], the RBR-soundness of a parallel-repetition IOP amplifies if the doomed set is defined using a dynamic threshold [HLR21, Def. 6.8]. We use a similar definition here, adapted to the IOP setting.

Definition 5.2 (Threshold Doomed Set for Parallel-Repetition IOPs)**.** *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be an l-round public-coin IOP with RBR soundness error ϵ_{RBR} and corresponding doomed set $\mathcal{D}$. Let $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ be the t_λ-fold parallel-repetition of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. For every $i \in [l]$, let $\mathsf{PartTr}_{\mathsf{IOP}}(i), \mathsf{PartTr}^{\otimes}(i)$ denote the set of round-i partial transcripts of the IOPs $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ and $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$, respectively. We define the* threshold doomed set *$\mathcal{D}^{\otimes}$ for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ as follows:*

$$
\begin{aligned}
(x, \mathcal{T}) \in \mathcal{D}^{\otimes} \\
&\Leftrightarrow \mathcal{T} = (\mathcal{T}_1 \ldots, \mathcal{T}_{t_\lambda}) \text{ and } \exists i \in [l] \text{ s.t. } \forall j \in [t_\lambda], \mathcal{T}_j \in \mathsf{PartTr}_{\mathsf{IOP}}(i) \\
&\qquad \text{and } |\{j \in [t_\lambda] \ : \ (x, \mathcal{T}_j) \in \mathcal{D}\}| \geq 1 + \frac{l-i}{l}(t_\lambda - 1)
\end{aligned}
$$

Remark 5.3. *Notice that in Definition 5.2, $(x, \mathcal{T}) \notin \mathcal{D}^{\otimes}$ for $\mathcal{T} \in \mathsf{PartTr}^{\otimes}(i)$ if and only if the number of non-doomed underlying partial transcripts $(x, \mathcal{T}_j)$ is more than $i(t_\kappa - 1)/l$, namely:*

$$
|\{j \in [t_\lambda] \ : \ (x, \mathcal{T}_j) \notin \mathcal{D}\}| > \frac{i(t_\kappa - 1)}{l}.
$$

Notice also that if $\mathcal{T} \in \mathsf{PartTr}(i-1)$, $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$, and $(x, \mathcal{T}|\Pi|r) \notin \mathcal{D}^{\otimes}$ for some Π, r, then in the i'th round at least $(t_\kappa - 1)/l$ underlying partial transcripts of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ moved from doomed to not doomed.

Theorem 5.4 (IOP Composition Theorem)**.** *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be an l-round IOP for $\mathcal{L}$ (Definition 4.11) with soundness error ϵ_{S} and completeness error ϵ_{C}, that has (statistically) semi-HVZK (Definition 4.14). Then the t_λ-fold parallel repetition $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ (Construction 5.1) is a (statistical) semi-HVZK-IOP for $\mathcal{L}$ with soundness error $\epsilon_{\mathsf{S}}^{t_\lambda}$ and completeness error $t_\lambda \cdot \epsilon_{\mathsf{C}}$. Moreover, if $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has RBR soundness error $\epsilon_{\mathsf{RBR}} < \frac{(t_\lambda - 1)}{l t_\lambda}$ (Definition 4.16), then $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has RBR soundness error $\epsilon^{\otimes}_{\mathsf{RBR}} := e^{-2t_\lambda(\frac{t_\lambda - 1}{l \cdot t_\lambda} - \epsilon_{\mathsf{RBR}})^2}$ with respect to the threshold doomed set of Definition 5.2. Furthermore, if $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has knowledge extraction with error ϵ_{E}, then $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has knowledge extraction with error $\exp\left(-2\left(\frac{1}{l} - \epsilon_{\mathsf{E}}\right)^2 t_\lambda\right)$ with respect to the threshold doomed set of Definition 5.2.*

Proof: We analyze each of the properties separately.
Completeness. If $x \in \mathcal{L}$ then the completeness of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ guarantees that each execution of the base-IOP fails with probability at most ϵ_{C}. Since $\mathcal{V}^{\otimes}$ accepts if and only if all executions of the base-IOP accepted, then using the union bound $\mathcal{V}^{\otimes}$ rejects only with probability $t_\lambda \cdot \epsilon_{\mathsf{C}}$.

(Statistical) semi-HVZK. We show that $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has statistical semi-HVZK whenever the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has statistical semi-HVZK. The proof for the case of perfect semi-HVZK follows similarly. Let $\mathsf{Sim}_{\mathsf{IOP}}$ be the simulator whose existence is guaranteed by the semi-HVZK property $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. That is, there exists a negligible function $\epsilon_{\mathsf{ZK}}(\lambda)$ such that for every possible randomness r_{IOP} of the honest verifier $\mathcal{V}_{\mathsf{IOP}}$, and any $x \in \mathcal{L}$, $\mathsf{Sim}_{\mathsf{IOP}}$ on input $1^{\lambda}, r_{\mathsf{IOP}}, x$ can simulate the view of the honest verifier $\mathcal{V}_{\mathsf{IOP}}$ when using randomness r_{IOP}, and the real and simulated views are $\epsilon_{\mathsf{ZK}}(\lambda)$-statistically close. We now describe a simulator Sim for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$.

Sim is given as input $1^{\lambda}, r, x$. Sim parses r as $r = (r_1, \ldots, r_{t_\lambda})$ where $r_j := (r_j^1, \ldots, r_j^l)$ for each $j \in [t_\lambda]$. For each $j \in [t_\lambda]$, Sim invokes $\mathsf{Sim}_{\mathsf{IOP}}$ on input $(1^{\lambda}, r_j, x)$, and outputs $(\mathsf{Sim}_{\mathsf{IOP}}(1^{\lambda}, r_j, x))_{j \in [t_\lambda]}$.

Let $\mathcal{R} = \mathcal{R}(x, w)$ denote the relation corresponding to $\mathcal{L}$. We show that for any $(x, w) \in R$, and any randomness r for $\mathcal{V}^{\otimes}$, the output of Sim is $t_\lambda \cdot \epsilon_{\mathsf{ZK}}(\lambda)$-statistically close to the real-world view of $\mathcal{V}^{\otimes}$, conditioned on randomness r. We show this via a hybrid argument where we replace the views of the underlying IOP from real to simulated one at a time. Specifically, we define hybrids $\mathcal{H}_0, \ldots, \mathcal{H}_{t_\lambda}$ where

$$\mathcal{H}_i = \left(\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}(\lambda, x, w; r_1), \ldots, \mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}(\lambda, x, w; r_i), \mathsf{Sim}_{\mathsf{IOP}}(1^{\lambda}, r_{i+1}, x), \ldots, \mathsf{Sim}_{\mathsf{IOP}}(1^{\lambda}, r_{t_\lambda}, x)\right)$$

Then $\mathcal{H}_{t_\lambda}$ is the view of $\mathcal{V}^{\otimes}$ in the real execution, and $\mathcal{H}_0$ is the output of Sim. For the sake of contradiction, assume that $\mathsf{SD}(\mathcal{H}_0, \mathcal{H}_{t_\lambda}) > t_\lambda \cdot \epsilon_{\mathsf{ZK}}(\lambda)$. Then, there exists a distinguisher $\mathcal{D}$ that obtains advantage $> t_\lambda \cdot \epsilon_{\mathsf{ZK}}(\lambda)$ in distinguishes between $\mathcal{H}_0$ and $\mathcal{H}_{t_\lambda}$. We show that this implies that there exists an $1 \leq i^* \leq t_\lambda$ such that $\mathsf{SD}(\mathcal{H}_{i^*-1}, \mathcal{H}_{i^*}) > \epsilon_{\mathsf{ZK}}(\lambda)$, which contradicts the statistical semi-HVZK property of the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. Indeed, we construct a non-uniform distinguisher that has r, x, w, i^* hard-coded, takes as input a random variable V that is distributed according to either $\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}(\lambda, x, w; r)$ or $\mathsf{Sim}_{\mathsf{IOP}}(1^{\lambda}, r, x)$. The distinguisher parses r as $r = (r_1, \ldots, r_{t_\lambda})$. It honestly generates $i^* - 1$ IOP views $\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}(\lambda, x, w; r_i)$, for $i \in [i^* - 1]$ using r_i as the random string, and (x, w) (by emulating an honest execution of $\mathcal{P}_{\mathsf{IOP}}$ with $\mathcal{V}_{\mathsf{IOP}}$). Then, it invokes $t_\lambda - i^*$ random and independent copies of $\mathsf{Sim}_{\mathsf{IOP}}$ to generate simulated views $\mathsf{Sim}_{\mathsf{IOP}}(1^{\lambda}, r_i, x)$ for $i \in \{i^*, \ldots, t_\lambda\}$. Now, the distinguisher invokes $\mathcal{D}$ on $\left\{\{\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}(\lambda, x, w; r^i)\}_{i \in [i^*-1]}, V, \{\mathsf{Sim}_{\mathsf{IOP}}(1^{\lambda}, r^i, x)\}_{i \in \{i^*, \ldots, t_\lambda\}}\right\}$, and outputs whatever $\mathcal{D}$ outputs.

Soundness. The underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has soundness error ϵ_{S}. Since the verifier $\mathcal{V}$ accepts only if each and every one of the t_λ random and independent invocations of $\mathcal{V}_{\mathsf{IOP}}$ accepts, the soundness error of the resultant scheme $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ is $\epsilon_{\mathsf{S}}^{t_\lambda}$.

Round-by-round Soundness. Assume that the underlying IOP system $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has ϵ_{RBR}-round-by-round soundness with doomed set $\mathcal{D}_{\mathsf{IOP}}$. We show that $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has $\epsilon_{\mathsf{RBR}}^{\otimes}$-round-by-round soundness, for $\epsilon_{\mathsf{RBR}}^{\otimes} := e^{-2t_\lambda\left(\frac{t_\lambda - 1}{l \cdot t_\lambda} - \epsilon_{\mathsf{RBR}}\right)^2}$. Note that any round-$i$ partial transcript $\mathsf{PartTr}^{\otimes}(i)$ of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$, by construction, consists of t_λ partial transcripts of the base-IOP. That is, $\mathsf{PartTr}^{\otimes}(i) = (\mathsf{PartTr}_1(i), \ldots, \mathsf{PartTr}_{t_\lambda}(i))$, where $\mathsf{PartTr}_j(i)$ is a round-i partial transcript of the jth repetition. We show that $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has round-by-round soundness with respect to the threshold doomed set of Definition 5.2, namely, we now show that $\mathcal{D}^{\otimes}$ satisfies the properties of Definition 4.16 with parameter $\epsilon_{\mathsf{RBR}}^{\otimes}$.

- If $x \notin \mathcal{L}_{\mathcal{R}}$, then $(x, \emptyset) \in \mathcal{D}^{\otimes}$:
  Let $\mathcal{T} \in \mathsf{PartTr}(i)$ be a partial transcript of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$. By construction, $\mathcal{T} := (\mathcal{T}_1, \ldots, \mathcal{T}_{t_\lambda})$, where $\mathcal{T}_k$ is the partial transcript induced on the kth repetition of the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. By round-by-round soundness of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$, if $\mathcal{T}_i = \emptyset$ then for each $k \in [t_\lambda]$ we have $(x, \mathcal{T}_i) \in \mathcal{D}_{\mathsf{IOP}}$. Therefore, by definition of $\mathcal{D}^{\otimes}$, $(x, \emptyset) \in \mathcal{D}^{\otimes}$.

- For any x, and any l-round partial transcript, $(x, \mathcal{T}) \in \mathsf{PartTr}(l)$, if $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$ then $\mathcal{V}^{\otimes}(x, \mathcal{T}) = 0$:

Let $\mathcal{T} := (\mathcal{T}_1, \ldots, \mathcal{T}_{t_\lambda})$ be the l-round partial transcript where $\mathcal{T}_k$ is the l-round partial transcript induced on the k'th execution of the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. By the definition of $\mathcal{D}^{\otimes}$, $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$ only if there exists a k such that $(x, \mathcal{T}_k) \in \mathcal{D}_{\mathsf{IOP}}$. (Indeed, when $i = l$ the bound in Definition 5.2 becomes $1 + \frac{l-i}{l}(t_\lambda - 1) = 1$.) By the 2nd property in Definition 4.16, this guarantees that $\mathcal{V}_{\mathsf{IOP}}(x, \mathcal{T}_k) = 0$ (with probability 1). But by construction, $\mathcal{V}^{\otimes}(x, \mathcal{T}) = 1$ only if $\mathcal{V}_{\mathsf{IOP}}(x, \mathcal{T}_k) = 1$ for all $k \in [t_\lambda]$. So $\mathcal{V}^{\otimes}(x, \mathcal{T}) = 0$.

- If $(x, \mathcal{T}) \in \mathsf{PartTr}(i-1)$ is an $(i-1)$-round partial transcript for some $i \in [l]$ and $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$, then for all $\Pi \in \mathcal{M}_i$,

$$\Pr_{r \leftarrow \mathcal{C}}\left[(x, \mathcal{T}, \Pi, r) \notin \mathcal{D}^{\otimes}\right] \leq \epsilon^{\otimes}_{\mathsf{RBR}} \tag{1}$$

where $\mathcal{C} := \mathcal{C}_1 \times \cdots \times \mathcal{C}_{t_\lambda}$ is the randomness space of $\mathcal{V}^{\otimes}$ in round i, and $\mathcal{C}_j$ is the randomness space of $\mathcal{V}_{\mathsf{IOP}}$ in round i in the jth repetition.

Assume for the sake of contradiction that $(x, \mathcal{T}) \in \mathsf{PartTr}(i-1)$, $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$, and there exists a Π such that Eq. (1) doesn't hold. $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$ implies that $|\{j \in [t_\lambda] \ : \ (x, \mathcal{T}_j) \in \mathcal{D}_{\mathsf{IOP}}\}| \geq 1 + \frac{l-i+1}{l}(t_\lambda - 1)$, where $\mathcal{T} = (\mathcal{T}_1, \ldots, \mathcal{T}_{t_\lambda})$. Define the number of doomed repetitions in the $i-1$th round, $\mathsf{d} := |\{j \in [t_\lambda] \ : \ (x, \mathcal{T}_j) \in \mathcal{D}_{\mathsf{IOP}}\}| > 1 + \frac{l-i+1}{l}(t_\lambda - 1) = \frac{t_\lambda(l-(i-1))+(i-1)}{l}$.

For any Π, r, we have $(x, \mathcal{T}, \Pi, r) \notin \mathcal{D}^{\otimes}$ only if $|\{j \in [t_\lambda] \ : \ (x, \mathcal{T}_j, \Pi_j, r_j) \notin \mathcal{D}_{\mathsf{IOP}}\}| > \frac{i(t_\lambda - 1)}{l}$. On the other hand, since $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$, $|\{j \in [t_\lambda] \ : \ (x, \mathcal{T}_j) \notin \mathcal{D}_{\mathsf{IOP}}\}| < \frac{(i-1)(t_\lambda - 1)}{l}$.

Therefore, for $(x, \mathcal{T}, \Pi, r) \notin \mathcal{D}^{\otimes}$, it must be the case that at least $\frac{t_\lambda - 1}{l}$ independent executions out of d executions switched from doomed to not doomed. In particular, by the 3rd property in Definition 4.16, for each of these doomed distributions we have

$$\Pr_{r_{k^*} \leftarrow \mathcal{C}_{k^*}}\left[(x, \mathcal{T}_{k^*}, \Pi_{k^*}, r_{k^*}) \notin \mathcal{D}_{\mathsf{IOP}}\right] \leq \epsilon_{\mathsf{RBR}},$$

where $\Pi = (\Pi_1, \ldots, \Pi_{t_\lambda})$.

Let X_j be the indicator variable for the event that the jth execution moved from doomed to not doomed in round i. We have from above that, $\mathbb{E}[X_j] \leq \epsilon_{\mathsf{RBR}}$, for all $j \in [t_\lambda]$ (if the jth execution is already not doomed, then this probability is 0). Define $Z = \frac{1}{t_\lambda}\sum_{j=1}^{t_\lambda} X_j$. We have $\mathbb{E}[Z] \leq \epsilon_{\mathsf{RBR}}$. Wlog, let the first d repetitions be the doomed transcripts in the $i-1$st partial transcript. Now, we have that

$$\begin{aligned}\Pr\left[\sum_{j=1}^{\mathsf{d}} X_j \geq \frac{t_\lambda - 1}{l}\right] &\leq \Pr\left[\sum_{j=1}^{t_\lambda} X_j \geq \frac{t_\lambda - 1}{l}\right] \\ &= \Pr\left[Z \geq \frac{t_\lambda - 1}{t_\lambda l}\right] \leq \exp\left(-2\left(\frac{t_\lambda - 1}{t_\lambda l} - \epsilon_{\mathsf{RBR}}\right)^2 t_\lambda\right)\end{aligned}$$

where the last step follows from Hoeffding (Theorem 4.2). Therefore, for $\epsilon_{\mathsf{RBR}} < \frac{(t_\lambda - 1)}{l t_\lambda}$, $\Pr_{r \leftarrow \mathcal{C}}\left[(x, \mathcal{T}, \Pi, r) \notin \mathcal{D}^{\otimes}\right] \leq \epsilon^{\otimes}_{\mathsf{RBR}}$.

Knowledge Extraction. Assume the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has knowledge extraction with error ϵ_{E}, and let E denote the extractor whose existence is guaranteed by Definition 4.17. We show that $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has knowledge extraction with an extractor $\mathsf{E}^{\otimes}$ that operates as follows. On input x and a round-1 prover message Π_1, $\mathsf{E}^{\otimes}$ interprets Π_1 as the concatenation $\Pi_1 = (\Pi^1, \ldots, \Pi^{t_\lambda})$ of t_λ round-1 messages of $\mathcal{P}_{\mathsf{IOP}}$. It then emulates E on x with each of the messages $\Pi^k, k \in [t_\lambda]$, to obtain

purported witnesses $w_1, \ldots, w_{t_\lambda}$. For each $w_k, k \in [t_\lambda]$ it checks whether $(x, w_k) \in \mathcal{R}$, and outputs w_k for the minimal k that passes verification (or w_1 if all purported witnesses fail verification). We now show that the extraction error of $\mathsf{E}^{\otimes}$ is $\exp\left(-2\left(\frac{1}{l} - \epsilon_{\mathsf{E}}\right)^2 t_\lambda\right)$. For this, notice that $(x, \mathsf{E}^{\otimes}(x, \Pi_1)) \notin \mathcal{R}$ only if $(x, \mathsf{E}(x, \Pi^i)) \notin \mathcal{R}$ for all $k \in [t_\lambda]$. The knowledge extraction of the base-IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ then guarantees that for every $k \in [t_\lambda]$,

$$\Pr_{r_k \leftarrow \mathcal{C}_1}\left[(x, \Pi^k, r_k) \notin \mathcal{D}_{\mathsf{IOP}}\right] \leq \epsilon_{\mathsf{E}}. \tag{2}$$

We need to show that

$$\Pr_{r \leftarrow \mathcal{C}_1}\left[(x, \Pi_1, r) \notin \mathcal{D}^{\otimes}\right] \leq \exp\left(-2\left(\frac{1}{l} - \epsilon_{\mathsf{E}}\right)^2 t_\lambda\right)$$

By the definition of the threshold doomed set, $(x, \Pi_1, r) \notin \mathcal{D}^{\otimes}$ iff the number of non-doomed underling base-IOP round-1 transcripts is more than

$$t_\lambda - \left(1 + \frac{l-1}{l}(t_\lambda - 1)\right) = (t_\lambda - 1) - \left[(t_\lambda - 1) - \frac{1}{l}(t_\lambda - 1)\right] = \frac{t_\lambda - 1}{l}.$$

We complete the knowledge extraction proof using a similar analysis to the last bullet in the RBR proof. Specifically, let X_k be the indicator of the event that $(x, \Pi^k, r_k) \notin \mathcal{D}_{\mathsf{IOP}}$, then by Eq. (2), $\mathbb{E}[X_k] \leq \epsilon_{\mathsf{E}}$ for all $k \in [t_\lambda]$, and for $Z = \frac{1}{t_\lambda}\sum_{k=1}^{t_\lambda} X_k$, we have $\mathbb{E}[Z] \leq \epsilon_{\mathsf{E}}$. Then:

$$\begin{aligned}\Pr\left[(x, \Pi_1, r) \notin \mathcal{D}^{\otimes}\right] &= \Pr\left[\sum_{k=1}^{t_\lambda} X_k > \frac{t_\lambda - 1}{l}\right] = \Pr\left[\sum_{k=1}^{t_\lambda} X_k \geq \frac{t_\lambda}{l}\right] \\ &= \Pr\left[Z \geq \frac{1}{l}\right] = \Pr\left[Z \geq \epsilon_{\mathsf{E}} + \left(\frac{1}{l} - \epsilon_{\mathsf{E}}\right)\right] \leq \exp\left(-2\left(\frac{1}{l} - \epsilon_{\mathsf{E}}\right)^2 t_\lambda\right)\end{aligned}$$

where the rightmost inequality follows from Hoeffding's bound (Theorem 4.2). ■

Remark 5.5 (Semi-HVZK Preservation Under Sequential Repetition). *Given an IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$, its sequential repetition $(\mathcal{P}^{\oplus}, \mathcal{V}^{\oplus})$ is defined similarly to Construction 5.1, except that $\mathcal{P}^{\oplus}, \mathcal{V}^{\oplus}$ emulate t_λ* sequential *executions of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$, and in the end $\mathcal{V}^{\oplus}$ accepts if and only if $\mathcal{V}_{\mathsf{IOP}}$ accepted in all these executions. semi-HVZK is preserved under such sequential repetition, and the proof follows a similar line to the proof of semi-HVZK preservation under parallel repetition in Theorem 5.4.*

Comparison with Prior Works on ZK Preservation Under IOP Composition. Theorem 5.4 shows that semi-HVZK is preserved under parallel repetition in the IOP setting. Related composition theorems for ZK-IOPs have been considered before in [VW26, CFW26]. In more detail, [VW26] prove that *full-fledged* ZK (against query-bounded verifiers) is preserved under composition for public-coin IOPs, so long as ZK holds with a straight-line and black-box simulator. They consider concurrent composition of multiple IOPs, and as such capture parallel composition as a special case. However, their results do not immediately imply ZK preservation when the underlying IOPs satisfy only the milder *(semi)-HVZK* property. Additionally, they do not consider the affect of composition on other system properties (i.e., completeness, soundness, RBR soundness, and knowledge extraction). Chiesa et al [CFW26] consider HVZK preservation for Interactive Oracle Reductions (IORs), a generalization of IOPs in which the task of verifying membership in a relation $\mathcal{R}_1$ is reduced to

verifying membership in a relation $\mathcal{R}_2$. They prove that the sequential composition of 2 IORs with HVZK has HVZK. More accurately, when sequentially composing $(\mathcal{P}_1, \mathcal{V}_1)$ with $(\mathcal{P}_2, \mathcal{V}_2)$, [CFW26] require that $(\mathcal{P}_1, \mathcal{V}_1)$ satisfy an enhanced HVZK guarantee that, roughly, ZK holds against the verifier obtained by composing the honest verifier $\mathcal{V}_1$ with the simulator for the honest verifier $\mathcal{V}_2$. They also consider a soundness guarantee (which they call RBR knowledge soundness) that combines aspects of knowledge extraction and RBR soundness. They show the composition preserves the RBR knowledge soundness error (there is no amplification in their construction due to the nature of IOR composition). Theorem 5.4 differ from [CFW26] in several respects. Specifically, we consider parallel composition of an IOP with itself any number of times, whereas [CFW26] consider sequential composition of two IORs, not necessarily the same. Moreover, HVZK is insufficient for their composition theorem (as discussed above, $(\mathcal{P}_1, \mathcal{V}_1)$ is required to have an enhanced ZK guarantee), whereas we only need semi-HVZK. Finally, we stress that their results are for the truly HVZK setting, whereas we consider semi-HVZK which allows the verifier's randomness to be skewed. Jumping ahead, this stronger guarantee will be crucial for our NIZK construction, in which the verifier's randomness will be sampled using a CIH and will therefore *inherently* be skewed. Thus, HVZK is insufficient to reduce NIZK ZK to the ZK guarantee of the underlying IOP, and semi-HVZK seems to be needed. Finally, we note that, as observed in [VW26], ZK preservation for IOPs does not follow from known results even in the *perfect ZK* setting, e.g., the results of [KLR06], because such works consider the standard setting in which parties have full access to their incoming message, and assume the underlying protocols have *full security*. Therefore, the results of [KLR06] do not apply to the IOP setting, in which protocols are only guaranteed to have (separately) ZK and soundness, and ZK is based on the fact that the verifier has restricted access to the prover messages.

Parallel-Repetition for 2-Round IOPs. Next, we prove a scaled-down version of Theorem 5.4 for *2-round* IOPs. Roughly, Theorem 5.4 guarantees that if the base IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has a very small RBR soundness error (less than 1 over the number of rounds), then the parallel repetition strongly amplifies the RBR soundness error (it becomes negligible). Theorem 5.7 below shows that for 2-round IOPs, as long as the base IOP has a mild RBR soundness error, the parallel repetition will mildly amplify the RBR soundness error with respect to *a more refined definition of the doomed set of the parallel repetition*. The definition of this doomed set (see Definition 5.6 below) crucially relies on the fact that the base IOP has only 2 rounds. The analysis likewise exploits this fact, which implies that (1) after the first round, more than half of the underlying partial transcripts are still doomed (so the probability that they will become not doomed in the next round is small), and (2) in the second round all of these (many) doomed executions must become not doomed (otherwise the verifier $\mathcal{V}$ would not accept). This allows us to eliminate the requirement on the RBR soundness which was crucial to proving RBR preservation in Theorem 5.4.

In particular, the scaled-down version works also when the base-IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has a *large* RBR soundness error. This will be used in Appendix B to construct a base-IOP for our CZK arguments and NIZK system. Jumping ahead, we note that our CZK arguments and NIZK system will be based on an IOP $(\mathcal{P}, \mathcal{V})$ which will be obtained as the t_λ-parallel repetition of a base IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ which itself will be an $t_\lambda{}'$-fold parallel repetition of an initial IOP $(\mathcal{P}_I, \mathcal{V}_I)$. We will use Theorem 5.7 below to claim that RBR soundness of $(\mathcal{P}_I, \mathcal{V}_I)$ implies RBR soundness of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ (with respect to the doomed set of Definition 5.6), then use Theorem 5.4 to prove this implies RBR soundness of $(\mathcal{P}, \mathcal{V})$. Theorem 5.7 will be imperative for our construction, since $(\mathcal{P}_I, \mathcal{V}_I)$ will have a large RBR soundness error.

We further note that while the resultant IOP $(\mathcal{P}, \mathcal{V})$ is a $(t_\lambda \cdot t_\lambda{}')$-fold parallel repetition of $(\mathcal{P}_I, \mathcal{V}_I)$, the two-phase construction described above leads to a *different doomed set* compared to directly defining $(\mathcal{P}, \mathcal{V})$ as the $(t_\lambda \cdot t_\lambda{}')$-fold parallel repetition of $(\mathcal{P}_I, \mathcal{V}_I)$. Roughly, doomed transcripts in the

direct construction guarantee that sufficiently many underlying transcripts of $(\mathcal{P}_I, \mathcal{V}_I)$ are doomed, whereas the two phase construction gives a more refined guarantee that a transcript of $(\mathcal{P}, \mathcal{V})$ can be partitioned into "chunks", and in sufficiently many of these "chunks", enough transcripts of $(\mathcal{P}_I, \mathcal{V}_I)$ are doomed.

Definition 5.6 (Threshold Doomed Set for Parallel-Repetition of 2-Round IOPs)**.** *Let $\epsilon_{\mathsf{RBR}} \in (0,1)$, and $\delta \in (0, 1-\epsilon_{\mathsf{RBR}})$. Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be an* 2-round *public-coin IOP with RBR soundness error ϵ_{RBR} for a doomed set $\mathcal{D}$. Let $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ be the t_λ-fold parallel-repetition of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. For every $i \in \{0,1,2\}$, let $\mathsf{PartTr}_{\mathsf{IOP}}(i), \mathsf{PartTr}^{\otimes}(i)$ denote the set of round-i partial transcripts of the IOPs $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ and $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$, respectively. We define the* threshold doomed set $\mathcal{D}^{\otimes 2}$ *for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ where $(x, \mathcal{T} = (\mathcal{T}_1 \ldots, \mathcal{T}_{t_\lambda})) \in \mathcal{D}^{\otimes 2}$ if an only if one of the following holds:*

- *$\mathcal{T}_j = \emptyset$ for every $j \in [t_\lambda]$.*
- *$\mathcal{T}_j \in \mathsf{PartTr}_{\mathsf{IOP}}(1)$ for every $j \in [t_\lambda]$, and $|\{j \in [t_\lambda] \ : \ (x, \mathcal{T}_j) \in \mathcal{D}\}| \geq (1 - (\epsilon_{\mathsf{RBR}} + \delta))t_\lambda$.*
- *$\mathcal{T}_j \in \mathsf{PartTr}_{\mathsf{IOP}}(2)$ for every $j \in [t_\lambda]$, and $(x, \mathcal{T}_j) \in \mathcal{D}$ for some $j \in [t_\lambda]$.*

Theorem 5.7 (Composition Theorem for 2-round IOPs)**.** *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be a* 2-round *IOP for $\mathcal{L}$ (Definition 4.11) with soundness error ϵ_{S} and completeness error ϵ_{C} that has (statistical) semi-HVZK (Definition 4.14). Then the t_λ-fold parallel repetition $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ (Construction 5.1) is a (statistical) semi-HVZK-IOP for $\mathcal{L}$ with soundness error $\epsilon_{\mathsf{S}}^{t_\lambda}$ and completeness error $t_\lambda \cdot \epsilon_{\mathsf{C}}$. Moreover, if $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has RBR soundness error ϵ_{RBR} (Definition 4.16), then $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has RBR soundness error $\epsilon_{\mathsf{RBR}}^{\otimes} := \max\{\exp\left(-2\delta^2 t_\lambda\right), \epsilon_{\mathsf{RBR}}^{(1-(\epsilon_{\mathsf{RBR}}+\delta))t_\lambda}\}$ with respect to the threshold doomed set of 2-round IOPs of Definition 5.6. If in addition $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has knowledge extraction with error ϵ_{E}, then $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has knowledge extraction with error $\epsilon_{\mathsf{E}}^{\otimes} := \exp\left(-2\left(\epsilon_{\mathsf{RBR}} + \delta - \epsilon_{\mathsf{E}}\right)^2 t_\lambda\right)$ with respect to the threshold doomed set of 2-round IOPs of Definition 5.6.*

Proof: Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be a 2-round IOP with round-by-round (RBR) soundness error ϵ_{RBR} with respect to a doomed set $\mathcal{D}$. Consider its t_λ-fold parallel repetition $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ (as in Construction 5.1). Then it follows from Theorem 5.4 that $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ has completeness with error $t_\lambda \cdot \epsilon_{\mathsf{C}}$, (statistical) semi-HVZK, and soundness error $\epsilon_{\mathsf{S}}^{t_\lambda}$, since the proof of these properties was independent of the number of rounds and the definition of the doomed set. We now prove RBR soundness and knowledge extraction.

RBR Soundness. We show that the RBR properties of Definition 4.16 are satisfied with respect to the doomed set $\mathcal{D}^{\otimes 2}$ of Definition 5.6. For this, let $\mathcal{M}_1, \mathcal{M}_2$ denote the domain of prover-messages in round 1 and round 2 of the base IOP, respectively, and notice that in this case the domain of round-1 and round-2 messages of $\mathcal{P}$ are $\mathcal{M}_1^{t_\lambda}$ and $\mathcal{M}_2^{t_\lambda}$, respectively. Similarly, let $\mathcal{C}_1, \mathcal{C}_2$ be the domain of round-1 and round-2 random verifier challenges in the base IOP, and notice that $\mathcal{C}_1^{t_\lambda}, \mathcal{C}_2^{t_\lambda}$ are the domain of round-1 and round-2 random verifier challenges in the parallel-repetition IOP. Then:

- If $x \notin \mathcal{L}$ then $(x, \emptyset) \in \mathcal{D}$ by RBR soundness of the base IOP (and the 1st property of Definition 4.16), and so $(x, \mathcal{T}) \in \mathcal{D}^{\otimes 2}$ where $\mathcal{T}$ consists of t_λ empty transcripts, by the definition of $\mathcal{D}^{\otimes 2}$.
- If $(x, \mathcal{T} = (\mathcal{T}_1, \ldots, \mathcal{T}_{t_\lambda})) \in \mathcal{D}^{\otimes 2}$ and $\mathcal{T} \in \mathsf{PartTr}(2)$ then by definition of $\mathcal{D}^{\otimes 2}$, $\mathcal{T}_j \in \mathsf{PartTr}_{\mathsf{IOP}}(2)$ for every $j \in [t_\lambda]$, and $(x, \mathcal{T}_{j^*}) \in \mathcal{D}$ for some $j^* \in [t_\lambda]$. Therefore, by the RBR soundness of the base IOP (and the 2nd property of Definition 4.16), $\mathcal{V}_{\mathsf{IOP}}(x, \mathcal{T}_{j^*}) = 0$. In this case, $\mathcal{V}(x, \mathcal{T}) = 0$ since $\mathcal{V}$ accepts if and only if all executions of $\mathcal{V}_{\mathsf{IOP}}$ accept.

- If $(x, \mathcal{T} = (\mathcal{T}_1, \ldots, \mathcal{T}_{t_\lambda})) \in \mathsf{PartTr}(0)$ and $(x, \mathcal{T}) \in \mathcal{D}^{\otimes 2}$, we need to bound the probability that there exists a $\Pi = (\Pi_1, \ldots, \Pi_{t_\lambda}) \in \mathcal{M}_1^{t_\lambda}$ such that

$$\Pr_{r=(r_1,\ldots,r_{t_\lambda}) \leftarrow \mathcal{C}_1^{t_\lambda}} \left[(x, (\mathcal{T}_j, \Pi_j, r_j)_{j \in [t_\lambda]}) \notin \mathcal{D}^{\otimes 2} \right].$$

  The analysis follows from Hoeffding's bound, using a similar argument to the last bullet in the RBR proof of Theorem 5.4. Specifically, fix some round-1 prover message $(\Pi_1, \ldots, \Pi_{t_\lambda}) \in \mathcal{M}_1^{t_\lambda}$. Let X_j be the indicator of the event that the jth execution moved from doomed to not doomed in round 1 with prover message Π_j, i.e., the event (over r_j) that $(x, (\mathcal{T}_j, \Pi_j, r_j)) \notin \mathcal{D}$. From the RBR soundness of the base IOP, we know that $\mathbb{E}[X_j] \leq \epsilon_{\mathsf{RBR}}$ for all $j \in [t_\lambda]$, and for $Z = \frac{1}{t_\lambda} \sum_{j=1}^{t_\lambda} X_j$, we have $\mathbb{E}[Z] \leq \epsilon_{\mathsf{RBR}}$. Then:

$$\begin{aligned} &\Pr\left[(x, \mathcal{T}, \Pi, r) \notin \mathcal{D}^{\otimes 2}\right] \\ &\qquad = \Pr\left[\sum_{j=1}^{t_\lambda} X_j > (\epsilon_{\mathsf{RBR}} + \delta) t_\lambda\right] = \Pr\left[Z > (\epsilon_{\mathsf{RBR}} + \delta)\right] \\ &\qquad \geq \Pr\left[Z \geq \epsilon_{\mathsf{RBR}} + \delta\right] \leq \exp\left(-2\delta^2 t_\lambda\right) \end{aligned}$$

  where the rightmost inequality follows from Hoeffding's bound (Theorem 4.2).

- If $(x, \mathcal{T} = (\mathcal{T}_1, \ldots, \mathcal{T}_{t_\lambda})) \in \mathsf{PartTr}(1)$ and $(x, \mathcal{T}) \in \mathcal{D}^{\otimes 2}$, then by the definition of $\mathcal{D}^{\otimes 2}$, $|\{j \in [t_\lambda] : (x, \mathcal{T}_j) \in \mathcal{D}\}| \geq (1 - (\epsilon_{\mathsf{RBR}} + \delta)) t_\lambda$, and let $S := \{j \in [t_\lambda] : (x, \mathcal{T}_j) \in \mathcal{D}\}$ denote the set of doomed base executions. For every $j \in S$ and $\Pi_j \in \mathcal{M}_2$, the RBR soundness of the base IOP (and the 3rd bullet of Definition 4.16) guarantees that

$$\Pr_{r_j \leftarrow \mathcal{C}_2} \left[(x, \mathcal{T}_j, \Pi_j, r_j) \notin \mathcal{D}\right] \leq \epsilon_{\mathsf{RBR}}$$

  Since the random base-IOP verifier challenges r_j are sampled independently from $\mathcal{C}_2$, we have that

$$\Pr_{(r_1,\ldots,r_{t_\lambda}) \leftarrow \mathcal{C}_2} \left[(x, (\mathcal{T}_j, \Pi_j, r_j)_{j \in [t_\lambda]}) \notin \mathcal{D}^{\otimes 2} \right] \leq \Pi_{j \in S} \Pr_{r_j \leftarrow \mathcal{C}_2} \left[(x, \mathcal{T}_j, \Pi_j, r_j) \notin \mathcal{D} \right] \leq \epsilon_{\mathsf{RBR}}^{|S|} \leq \epsilon_{\mathsf{RBR}}^{(1-(\epsilon_{\mathsf{RBR}}+\delta))t_\lambda}$$

Knowledge Extraction. The proof is similar to the proof of knowledge extraction in Theorem 5.4. Assume the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has knowledge extraction with error ϵ_{E}, and let E denote the extractor whose existence is guaranteed by Definition 4.17. The extractor $\mathsf{E}^{\otimes}$ for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ operates identically to the extractor for the general case in Theorem 5.4, and it therefore holds that if $(x, \mathsf{E}^{\otimes}(x, \Pi_1)) \notin \mathcal{R}$ then $(x, \mathsf{E}(x, \Pi^i)) \notin \mathcal{R}$ for all $k \in [t_\lambda]$, which, for every k, implies that

$$\Pr_{r_k \leftarrow \mathcal{C}_1} \left[(x, \Pi^k, r_k) \notin \mathcal{D}\right] \leq \epsilon_{\mathsf{E}}. \tag{3}$$

We need to show that

$$\Pr_{r \leftarrow \mathcal{C}_1} \left[(x, \Pi_1, r) \notin \mathcal{D}^{2\otimes}\right] \leq \exp\left(-2\left(\epsilon_{\mathsf{RBR}} + \delta - \epsilon_{\mathsf{E}}\right)^2 t_\lambda\right)$$

By the definition of the threshold doomed set fro 2-round IOPs (Definition 5.6), $(x, \Pi_1, r) \notin \mathcal{D}^{2\otimes}$ iff the number of non-doomed underling base-IOP round-1 transcripts is more than

$$t_\lambda - ((1 - (\epsilon_{\mathsf{RBR}} + \delta)) t_\lambda) = (\epsilon_{\mathsf{RBR}} + \delta) t_\lambda.$$

The analysis now follows a similar Hoeffding-based analysis as in the extraction proof of Theorem 5.4. Specifically, let X_k be the indicator of the event that $(x, \Pi^k, r_k) \notin \mathcal{D}$, then by Eq. (3), $\mathbb{E}[X_k] \leq \epsilon_{\mathsf{E}}$ for all $k \in [t_\lambda]$, and for $Z = \frac{1}{t_\lambda}\sum_{k=1}^{t_\lambda} X_k$, we have $\mathbb{E}[Z] \leq \epsilon_{\mathsf{E}}$. Then:

$$\begin{aligned}\Pr\left[(x,\Pi_1,r) \notin \mathcal{D}^{2\otimes}\right] &= \Pr\left[\sum_{k=1}^{t_\lambda} X_k > (\epsilon_{\mathsf{RBR}}+\delta)t_\lambda\right] \leq \Pr\left[\sum_{k=1}^{t_\lambda} X_k \geq (\epsilon_{\mathsf{RBR}}+\delta)t_\lambda\right] \\ &= \Pr\left[Z \geq \epsilon_{\mathsf{RBR}}+\delta\right] = \Pr\left[Z \geq \epsilon_{\mathsf{E}} + (\epsilon_{\mathsf{RBR}}+\delta-\epsilon_{\mathsf{E}})\right] \leq \exp\left(-2\left(\epsilon_{\mathsf{RBR}}+\delta-\epsilon_{\mathsf{E}}\right)^2 t_\lambda\right)\end{aligned}$$

where the rightmost inequality follows from Hoeffding's bound (Theorem 4.2).

■

5.1 Applications: Sounder Linear-Time ZK IOPs and Arguments

In this Section, we show applications of our composition theorem (Theorem 5.4) to improving the soundness error of existing semi-HVZK-IOPs and CZK arguments.

Sounder Linear-Time semi-HVZK-IOPs. IOPs with (HV)ZK guarantees often achieve only a constant soundness error (e.g., [BCGV16, BCL22, RW24]). While the soundness error in standard IOPs can be easily amplified by repetition, for ZK-IOPs one must be careful, since ZK might not generally be preserved under repetition. It is instructive to note that while for the special case of *PCPs* with full-fledged ZK, one can easily reduce the soundness error by repeating the verification procedure (so long as the increased query complexity of the honest verifier does not exceed the ZK parameter of the PCP), this cannot generally be done for ZK-IOPs, since the verifier cannot generally perform *independent* executions of the verification procedure without communicating with the prover.

Theorem 5.4 implies that the soundness and RBR soundness errors in any semi-HVZK-IOP can be reduced, while still guaranteeing semi-HVZK. In particular, applying Theorem 5.4 to the semi-HVZK-IOPs of [BCL22, Thm. 9.1] yields semi-HVZK-IOPs for R1CS with a *linear-time prover* and an *arbitrarily small* constant soundness error. In more detail, Bootle, Chiesa and Liu [BCL22, Thm. 9.1] construct semi-HVZK-IOPs the NP-complete language R1CS[17] with a linear time prover, in which the soundness error is a fixed constant (that depends on the sparsity of the input, the field size, and the soundness errors of the building blocks). By repeating the protocol $t_\kappa := O(1)$ times in parallel we can reduce the soundness error to any desired constant, while preserving linear prover time. We note that the IOPs of [BCL22] are over large fields and in the holographic setting (in which an instance-witness pair is associated with a public "index", and the verifier is given oracle access to an encoding of this index; this is useful to obtain sublinear verification time). However, our theorems are still applicable, as we now explain. First, semi-HVZK over bit strings implies semi-HVZK over larger alphabets (in the ZK analysis, one can interpret every field element as a bit string; this does not affect the efficiency of the system since this is done only in the ZK analysis). Second, the index is a function of the public parameters of the problem, which the ZK guerantee does not try to hide (the simulator receives the index as input). Therefore, ZK can be analyzed when the verifier has full access to the index as part of her explicit input (again, this is done only in the ZK analysis and therefore doesn't affect the efficiency of the scheme. More specifically, in the IOPs of [BCL22, Thm 9.1], the index is obtained by applying a ZK code *with encoding randomness fixed to the all-0 string* to an efficiently-computable function of the public parameters of the problem. Therefore, this can be locally computed efficiently by both the verifier and the simulator.

[17] In the R1CS problem, the input consists of matrices A, B and C and a vector v, and the question is whether there exists a vector z such that $(A(u,z)) \star (B(u,z)) = C(u,z)$, where $\star$ represents point-wise multiplication.

Sounder Linear-Time CZK Arguments. Bootle et al. [BCL22, Thm. 2] construct semi-HVZK-IOPs for R1CS with a *linear-time prover*, and combine these with Kilian's transformation and linear-time computable hash functions to construct succinct CZK arguments for R1CS with a linear-time prover. The soundness error of their construction is some unspecified constant, and is inherited from the soundness error of their semi-HVZK-IOPs [BCL22, Thm. 9.1]. The soundness analysis for succinct arguments constructed from IOPs (without ZK) from Kilian's construction was given only recently in [BWW24]. Their analysis shows that the soundness error of the resultant argument is $\epsilon + \epsilon_h + \epsilon_{\mathsf{IOP}}$, where ϵ is any positive constant, ϵ_{IOP} is the soundness error of the underlying IOP, and ϵ_h is (roughly) the collision resistance error of the underlying hash against adversaries running in time $\mathsf{poly}(n, \lambda, 1/\epsilon)$ (more accurately, the position binding of the vector commitment). The soundness error of [BCL22]'s linear-time CZK arguments can be reduced to any constant $\epsilon' \in (0, 1)$ as follows. First, we reduce the soundness error in their semi-HVZK-IOPs (without violating semi-HVZK) to $\epsilon'/3$ by applying Theorem 5.4 with a constant repetition parameter as discussed above (this does not affect the asymptotic parameters, and in particular, preserves linear prover time). Then, we use this amplified semi-HVZK-IOPs in the construction of the CZK args of [BCL22, Thm. 2] together with the linear-time hash functions of [AHI+17, Cor. 4.11], which have $\mathsf{negl}(\lambda) < \epsilon'/3$ soundness error (and in particular against adversaries running in time $\mathsf{poly}(n, \lambda, 1/(\epsilon'/3)) = \mathsf{poly}(n, \lambda)$). Finally, we set $\epsilon = \epsilon'/3$ in the soundness analysis of [BWW24].

6 CZK Arguments from HVZK-IOPs and EEs

In this section, we describe a compiler from semi-HVZK-IOPs and EE commitments to CZK arguments with setup. Our compiler follows the general blueprint used in prior works [Kil92, BCS16, CDGS22], but employs an EE scheme instead of a vector commitment (cf. Section 3).

Theorem 6.1 (CZK Arguments from semi-HVZK-IOPs and EEs)**.** *Assume that:*

- $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ *is a (statistical) semi-HVZK-IOP with soundness error* ϵ_{S} *and completeness error* ϵ_{C}*, and*
- EE *is an EE commitment scheme with extraction error* ϵ_{EE}*,*

then the construction given in Figure 1 is a CZK argument system, with completeness error ϵ_{C}*, and soundness error* $\epsilon_{\mathsf{S}} + \epsilon_{\mathsf{EE}}$*.*

Moreover, the ZK simulation is black-box and straight-line.[18] *Furthermore, if the IOP has* l *rounds and the per-round proof length is* ℓ*, then the argument system has* $l + 1$ *rounds, and a CRS of size* $l \cdot \ell \cdot \mathsf{poly}(\kappa)$*, where* κ *is a security parameter. Furthermore, in each of the first* l *rounds, the prover message has length* $\ell \cdot \mathsf{poly}(\kappa)$*, and in the last round the prover message has length* $q \cdot \mathsf{poly}(\kappa)$*, where* q *denotes the query complexity of* $\mathcal{V}_{\mathsf{IOP}}$*.*

Proof: The completeness property is proved in Lemma 6.5, soundness is proved in Lemma 6.7 (with an extension to the *adaptive* setting in Lemma 6.14), and zero-knowledge is analyzed in Lemma 6.11. The analysis of the parameter is discussed in Lemma 6.6. ■

We use Theorem 6.1 to obtain CZK arguments, assuming PKE (to instantiate the EE scheme).

Corollary 6.2 (CZK Arguments from PKE)**.** *Assume that public-key encryption exists. Then any NP language has a 3-round CZK argument with setup with* $\mathsf{negl}(\kappa)$ *soundness error, where* κ *is a security parameter. Moreover, the CRS and prover messages have size* $\mathsf{poly}(\kappa, n)$*, and the prover and verifier run in* $\mathsf{poly}(\kappa, n)$ *time, where* n *is the input length.*

[18] A Simulator is straight-line if it doesn't rewind the verifier.

CZK Arguments with Setup from semi-HVZK-IOPs and EE Commitments

Let $\mathcal{L}$ be an NP language with a corresponding NP relation $\mathcal{R} = \mathcal{R}(x, w)$. The argument system for $\mathcal{L}$ uses an EE commitment scheme $\mathsf{EE} = (\mathsf{EE.setup}, \mathsf{EE.TDsetup}, \mathsf{EE.Com}, \mathsf{EE.Decom}, \mathsf{EE.Ver}, \mathsf{EE.Ext})$, and an l-round semi-HVZK-IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ for $\mathcal{L}$, as building blocks. We use ℓ to denote the length of oracle messages sent by $\mathcal{P}_{\mathsf{IOP}}$.[a]

The argument system is executed between a prover $\mathcal{P}$ that has input x with corresponding witness w, and a verifier $\mathcal{V}$ that has input x. Both parties also have as input a CRS crs (which consists of $l \cdot \ell$ CRSs for the EE commitment scheme, generated for security parameter κ).

The argument system operates as follows:

1. **Setup:** on input a security parameter 1^κ, the setup algorithm generates $l \cdot \ell$ CRSs crs^i_j for the EE commitment scheme by executing, for every $1 \leq i \leq l$ and every $1 \leq j \leq \ell$, $(\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa)$. The setup algorithm outputs $\mathsf{crs} := (\mathsf{crs}^i_j)_{i \in [l], j \in [\ell]}$.
2. **Emulating IOP's communication phase:** For every $1 \leq i \leq l$:
   (a) **Emulating round-i prover message:**[b] $\mathcal{P}$ emulates $\mathcal{P}_{\mathsf{IOP}}$ to generate the message $\Pi^i = (\Pi^i(1), \ldots, \Pi^i(\ell))$ which $\mathcal{P}_{\mathsf{IOP}}$ sends to $\mathcal{V}_{\mathsf{IOP}}$ in the ith IOP round.
   (b) **Committing to $\mathcal{P}_{\mathsf{IOP}}$'s message:** for every $1 \leq j \leq \ell$, $\mathcal{P}$ generates a commitment $(\mathsf{c}^i_j, \sigma^i_j) \leftarrow \mathsf{EE.Com}(\mathsf{crs}^i_j, \Pi^i(j))$ to $\Pi^i(j)$, and let $\mathsf{c}^i := (\mathsf{c}^i_1, \ldots, \mathsf{c}^i_\ell)$. Then, $\mathcal{P}$ sends c^i to $\mathcal{V}$.
   (c) **Emulating round-i verifier message:** $\mathcal{V}$ emulates $\mathcal{V}_{\mathsf{IOP}}$ to generate $\mathcal{V}_{\mathsf{IOP}}$'s round-$i$ random challenge r_i, and sends r_i to $\mathcal{P}$.
3. **Emulating IOP's query phase:**
   (a) **Emulating oracle answers;**
      i. **Determining $\mathcal{V}_{\mathsf{IOP}}$'s oracle queries (prover side):** $\mathcal{P}$ determines the queries $\mathcal{Q}$ which $\mathcal{V}_{\mathsf{IOP}}$, with randomness $r := (r_1, \ldots, r_l)$, makes to $\Pi^1, \ldots, \Pi^l$.
      ii. **Decommitting to the oracle answers:** $\mathcal{P}$ sends $\{(\sigma^i_j, \Pi^i(j)) : q := (i, j) \in \mathcal{Q}\}$ to $\mathcal{V}$, where $q = (i, j)$ denotes that $\mathcal{V}$ queries the jth bit of Π^i.
   (b) **Output:**
      i. **Determining $\mathcal{V}_{\mathsf{IOP}}$'s oracle queries (verifier side):** $\mathcal{V}$ determines the queries $\mathcal{Q}'$ which $\mathcal{V}_{\mathsf{IOP}}$, with randomness $r := (r_1, \ldots, r_l)$, makes to her oracles $\Pi^1, \ldots, \Pi^l$.
      ii. **Checking Commitments:** Let $\left\{(\tilde{\sigma}^i_j, \tilde{\Pi}^i(j)) : (i, j) \in \mathcal{Q}'\right\}$ denote the values obtained from $\mathcal{P}$ in Step 3(a)ii. Then for every $(i, j) \in \mathcal{Q}'$, $\mathcal{V}$ checks that $\mathsf{EE.Ver}(\mathsf{crs}^i_j, \mathsf{c}^i_j, \tilde{\sigma}^i_j, \tilde{\Pi}^i(j))$ outputs 1, and otherwise rejects.
      iii. **Computing IOP output:** $\mathcal{V}$ emulates $\mathcal{V}_{\mathsf{IOP}}$ with $\left\{\tilde{\Pi}^i(j) : (i, j) \in \mathcal{Q}'\right\}$ as the answers to $\mathcal{V}_{\mathsf{IOP}}$'s oracle queries, and outputs whatever $\mathcal{V}_{\mathsf{IOP}}$ outputs.

[a] We assume without loss of generality that all prover messages have the same length.
[b] In case the IOP takes as input a statistical security parameter λ, we set $\lambda := \kappa$.

Figure 1: A Compiler from semi-HVZK-IOPs and EE Commitments to CZK Arguments with Setup

Proof: For the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$, we apply Theorem 5.4 to the IOP of [BCS16, Thm. 4.1]. Specifically, Ben-Sasson et al. [BCGV16, Thm. 4.1] design a $\mathsf{poly}(n)$-length 2-round IOP for NP with soundness error $1/2$ with an honest verifier that makes $\mathsf{poly}(\log n)$ queries, and the prover and verifier run in time $\mathsf{poly}(n)$. Moreover, their IOP has perfect n-ZK, i.e., perfect ZK against *any*

verifier making at most n queries.[19] In particular, their IOP has semi-HVZK, and so by Theorem 5.4, the parallel-repetition $t_\lambda = \log^2 \kappa$ times of this IOP gives a semi-HVZK-IOP with $\mathsf{negl}(\kappa)$ soundness error.[20]

We then apply Theorem 6.1 to the semi-HVZK-IOPs described above, and the EE of Theorem 4.6. The parameters then follow directly from Thoerem 6.1. ■

6.1 Analysis

We first set some notation.

Notation 6.3 ([CDGS22, page 12]). *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be an IOP system. For $q \in \mathbb{N}$, $v \in \{0,1\}^q$, input $x \in \{0,1\}^*$, and randomness $r \in \{0,1\}^*$, we use $\mathcal{V}_{\mathsf{IOP}}^{[\mathcal{Q},v]}(x;r)$ to denote $\mathcal{V}_{\mathsf{IOP}}$'s decision bit (i.e., output) in an execution on x, in which she uses randomness r, makes queries $\mathcal{Q}$, and the ith query in $\mathcal{Q}$ is answered with v_i. (If $\mathcal{V}_{\mathsf{IOP}}$ on input x with randomness r makes a query which is not in $\mathcal{Q}$, or $q < |\mathcal{Q}|$, her output is fixed to 0.)*

Notation 6.4 (Query Sets and Stateful Provers). *In the following, it will be useful to partition $\mathcal{V}_{\mathsf{IOP}}$'s query set $\mathcal{Q}$ as $\mathcal{Q} := \bigcup_{i\in[l]} \mathcal{Q}_i$, where $\mathcal{Q}_i$ is the set of queries which $\mathcal{V}_{\mathsf{IOP}}$ makes to the ith proof oracle (i.e., the restriction of $\mathcal{Q}$ to queries to Π^i).*

It will sometimes be useful to explicitly describe a malicious prover $\mathcal{P}^*$ as a stateful *algorithm which, in each round, on input its current state and the next verifier challenge, outputs a round-i commitment and an updated state, formally: $(\mathsf{c}^i, \mathsf{st}_i^*) \leftarrow \mathcal{P}^*(\mathsf{st}_{i-1}^*, r_{i-1})$. We set $\mathsf{st}_0^* := \mathsf{crs}$. In the final round, the prover outputs the purported verifier query sets $(\mathcal{Q}_i)_{i\in[l]}$, the purported IOP answers $(\Pi^i(j))_{i\in[l],j\in\mathcal{Q}_i}$ and the openings $(\sigma_j^i)_{i\in[l],j\in\mathcal{Q}_i}$, formally: $(\mathcal{Q}_i, (\Pi^i(j), \sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]} \leftarrow \mathcal{P}^*(\mathsf{st}_l^*, r_l)$. To abbreviate notations, we somewhat abuse notation and write, for some $i \in [l]$,*

$$(\mathcal{Q}_i, (\Pi^i(j), \sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]} \leftarrow \mathcal{P}^*(\mathsf{st}_i^*, r_1, \ldots, r_l).$$

This is possible because given his ith-round state st_i^*, and all the random verifier challenges, the prover can locally generate his states in the following rounds (and therefore also his commitments, and — following the final round — the query sets and commitment openings).

6.1.1 Completeness and Parameter Analysis

Lemma 6.5 (Completeness). *If $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ is complete with error ϵ_{C} (Definition 4.11), and EE has correctness (Definition 4.4), then the construction given in Figure 1 is complete with error ϵ_{C}.*

Proof: Let $\Pi_1, \ldots, \Pi_l$ denote the proof oracles generated by $\mathcal{P}_{\mathsf{IOP}}$ during the executionon $(x,w) \in \mathcal{R}$, let $r_1, \ldots, r_l$ denote the random challenges generated for $\mathcal{V}_{\mathsf{IOP}}$, and for every $i \in [l]$, $j \in [\ell]$, let $(\mathsf{c}_j^i, \sigma_j^i)$ denote the commitment and opening generated for $\Pi^i(j)$. Furthermore, let $\mathcal{Q}$ denote the queries which $\mathcal{V}_{\mathsf{IOP}}$ makes to her oracles $\Pi^i, \ldots, \Pi^l$. The completeness of the IOP system guarantees that $\Pr\left[\mathcal{V}_{\mathsf{IOP}}^{[\mathcal{Q},(\Pi^i(j))_{(i,j)\in\mathcal{Q}}]}(x; r_1, \ldots, r_l) = 1\right] \geq 1 - \epsilon_{\mathsf{C}}$. Moreover, the correctness of the EE guarantees that for every $(i,j) \in \mathcal{Q}$, $\mathsf{EE.Ver}(\mathsf{crs}_j^i, \mathsf{c}_j^i, \sigma_j^i, \Pi^i(j)) = 1$. Therefore, it follows directly from the definition of $\mathcal{V}$ that she accepts with probability at least $1 - \epsilon_{\mathsf{C}}$. ■

[19] This is obtained by instantiating [BCGV16, Thm. 4.1] with knowledge bound parameter n.

[20] We note that an alternative route towards obtaining semi-HVZK-IOPs with negligible soundness error would be to apply [VW26, Thm. 9.1] to the parallel-repetition of [BCGV16, Thm. 4.1]. We chose to use Theorem 5.4 since [VW26] do not explicitly analyze the soundness error of the parallel repetition.

Lemma 6.6 (Parameter Analysis). *If $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ is an l-round IOP with proof length ℓ, and* EE *is an EE scheme, then the construction given in Figure 1 has $l+1$ rounds, and CRS of size $l \cdot \ell \cdot \mathsf{poly}(\kappa)$. Furthermore, in each of the first l rounds, the prover message has length $\ell \cdot \mathsf{poly}(\kappa)$, and in the last round the prover message has length $q \cdot \mathsf{poly}(\kappa)$, where q denotes the query complexity of $\mathcal{V}_{\mathsf{IOP}}$.*

Proof: The round complexity is $l+1$ because $\mathcal{P}, \mathcal{V}$ emulate the l rounds of the underlying IOP, followed by a single additional message from $\mathcal{P}$ to $\mathcal{V}$ (the decommitments to the queries of $\mathcal{V}_{\mathsf{IOP}}$). In the first l rounds, each prover message consists of ℓ commitments, each of size $\mathsf{poly}(\kappa)$, so the total length of the prover message in each of these rounds is $\ell \cdot \mathsf{poly}(\kappa)$. In the last round, the prover sends q decommitments, each of size $\mathsf{poly}(\kappa)$, so overall the last message has size $q \cdot \mathsf{poly}(\kappa)$. As for the CRS, it consists of $l \cdot \ell$ CRSs for the underlying EE scheme, each of which has size $\mathsf{poly}(\kappa)$, so overall the CRS has length $l \cdot \ell \cdot \mathsf{poly}(\kappa)$. ■

6.1.2 Soundness

Lemma 6.7 (Soundness). *If $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ is sound with error ϵ_{S} (Definition 4.11), and* EE *has extractability (Definition 4.4) with error ϵ_{EE}, then the construction given in Figure 1 has soundness error $\epsilon_{\mathsf{S}} + \epsilon_{\mathsf{EE}}$.*

Remark 6.8 (Comparison with [CDGS22]). *Recently, Chiesa et al. [CDGS22] gave a soundness analysis for non-ZK interactive arguments obtained by combining IOPs with vector commitments. Their interactive arguments are* succinct*, and consequently cannot be fully extractable. A majority of their analysis therefore focuses on extracting (partial) IOP prover messages from the succinct commitments. Our focus is on* CZK *arguments, and we require* full extraction *to enable us to apply (in Section 7) the CIH-based approach to construct NIZKs. The full extraction guarantee enables us to reduce the soundness of the CZK arguments directly to the soundness of the IOP system and the extractability of the EE scheme, allowing us to aviod the more complicated analysis needed in [CDGS22].*

Proof of Lemma 6.7: Fix $x \notin \mathcal{L}$, and let $\mathcal{P}^*$ be a (possibly malicious) PPT prover. We claim that $\mathcal{V}$ accepts x in the execution of the CZK argument of Figure 1 with $\mathcal{P}^*$ only with probability $\epsilon_{\mathsf{S}} + \epsilon_{\mathsf{EE}}$, where the probability is over the coin tosses used by the setup algorithm, $\mathcal{P}^*$ and $\mathcal{V}$. That is, by the definition of the argument system, we need to bound the following probability (using Notations 6.3 and 6.4):

$$\Pr\left[\begin{array}{l} \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \forall i\in[l]\,\forall j\in\mathcal{Q}_i,\ \mathsf{EE.Ver}(\mathsf{crs}_j^i,\mathsf{c}_j^i,\sigma_j^i,\Pi^i(j))=1 \end{array} : \begin{array}{l} \forall i\in[l]\,,j\in[\ell]\,,\ (\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa) \\ \mathsf{st}_0^*:=(\mathsf{crs}_j^i)_{i,j},\ r_0:=\perp \\ \text{For } i\in[l]: \\ \quad (\mathsf{c}^i,\mathsf{st}_i^*)\leftarrow\mathcal{P}^*(x,\mathsf{st}_{i-1}^*,r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \\ (\mathcal{Q}_i,(\Pi^i(j),\sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]}\leftarrow\mathcal{P}^*(x,\mathsf{st}_l,r_l) \end{array}\right] \tag{4}$$

For every $i \in [l]\,, j \in [\ell]$, set $\tilde{\Pi}^i(j) := \mathsf{EE.Ext}(\tau_j^i, \mathsf{c}_j^i)$, where c_j^i is the j'th commitment in c^i. By the law of total probability, Eq. (4) can be upper bounded by the sum of the probabilities of the following two events:

Event (1): IOP soundness is violated:

$$\Pr\left[\begin{array}{l} \\ \mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \forall i\in[l]\,\forall j\in\mathcal{Q}_i,\ \mathsf{EE.Ver}(\mathsf{crs}_j^i,\mathsf{c}_j^i,\sigma_j^i,\Pi^i(j))=1 \\ \\ \end{array} : \begin{array}{l} \forall i\in[l]\,,j\in[\ell]\,,\ (\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa) \\ \mathsf{st}_0^*:=(\mathsf{crs}_j^i)_{i,j},\ r_0:=\perp \\ \text{For } i\in[l]: \\ \quad (\mathsf{c}^i,\mathsf{st}_i^*)\leftarrow\mathcal{P}^*(x,\mathsf{st}_{i-1}^*,r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \\ (\mathcal{Q}_i,(\Pi^i(j),\sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]}\leftarrow\mathcal{P}^*(x,\mathsf{st}_l,r_l) \end{array}\right] \tag{5}$$

where $\mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x;r_1,\ldots,r_l)$ is the output of $\mathcal{V}_{\mathsf{IOP}}$ on input x when she uses randomness $r_1,\ldots,r_l$, and has oracle access to the oracle proofs $(\tilde{\Pi}^i)_{i\in[l]}$.

Event (2): EE extraction is violated:

$$\Pr\left[\begin{array}{l} \\ \mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x;r_1,\ldots,r_l)\neq 1 \\ \wedge\, \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \forall i\in[l]\,\forall j\in\mathcal{Q}_i,\ \mathsf{EE.Ver}(\mathsf{crs}_j^i,\mathsf{c}_j^i,\sigma_j^i,\Pi^i(j))=1 \\ \\ \end{array} : \begin{array}{l} \forall i\in[l]\,,j\in[\ell]\,,\ (\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa) \\ \mathsf{st}_0^*:=(\mathsf{crs}_j^i)_{i,j},\ r_0:=\perp \\ \text{For } i\in[l]: \\ \quad (\mathsf{c}^i,\mathsf{st}_i^*)\leftarrow\mathcal{P}^*(x,\mathsf{st}_{i-1}^*,r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \\ (\mathcal{Q}_i,(\Pi^i(j),\sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]}\leftarrow\mathcal{P}^*(x,\mathsf{st}_l,r_l) \end{array}\right] \tag{6}$$

We show in Lemma 6.9 below that the probability in Equation (5) is at most ϵ_{S} by the soundness of the IOP system. We show in Lemma 6.10 below that the probability in Equation (6) is upper bounded by ϵ_{EE}, by the extractability of EE. Thus, we conclude that overall $\mathcal{P}^*$ succeeds with probability $\epsilon_{\mathsf{S}}+\epsilon_{\mathsf{EE}}$. ■

Lemma 6.9 (Bounding Probability of Event (1))**.** *Assume that $(\mathcal{P}_{\mathsf{IOP}},\mathcal{V}_{\mathsf{IOP}})$ is sound with error ϵ_{S} (Definition 4.11). Then for every $x\notin\mathcal{L}$ and any PPT prover $\mathcal{P}^*$, Eq. (5) is upper bounded by ϵ_{S}, namely:*

$$\Pr\left[\begin{array}{l} \\ \mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \forall i\in[l]\,\forall j\in\mathcal{Q}_i,\ \mathsf{EE.Ver}(\mathsf{crs}_j^i,\mathsf{c}_j^i,\sigma_j^i,\Pi^i(j))=1 \\ \\ \end{array} : \begin{array}{l} \forall i\in[l]\,,j\in[\ell]\,,\ (\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa) \\ \mathsf{st}_0^*:=(\mathsf{crs}_j^i)_{i,j},\ r_0:=\perp \\ \textit{For } i\in[l]: \\ \quad (\mathsf{c}^i,\mathsf{st}_i^*)\leftarrow\mathcal{P}^*(x,\mathsf{st}_{i-1}^*,r_{i-1}) \\ \quad \textit{Sample uniformly random } r_i \\ (\mathcal{Q}_i,(\Pi^i(j),\sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]}\leftarrow\mathcal{P}^*(x,\mathsf{st}_l,r_l) \\ \forall i\in[l]\,,j\in[\ell]\,,\ \tilde{\Pi}^i(j):=\mathsf{EE.Ext}(\tau_j^i,\mathsf{c}_j^i) \end{array}\right]\leq\epsilon_{\mathsf{S}}$$

Proof: We prove the lemma by a reduction to the soundness of the underlying IOP system. Towards that end, we describe a malicious IOP prover $\mathcal{P}^*_{\mathsf{IOP}}$.

$\mathcal{P}^*_{\mathsf{IOP}}$ on input $1^\kappa, x$ operates as follows:

1. **Initialization:**
   (a) For every $i\in[l]\,,j\in[\ell]$, sample $(\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa)$, and let $\mathsf{crs}:=(\mathsf{crs}_j^i)_{i\in[l],j\in[\ell]}$.
   (b) Set $\mathsf{st}'_0:=(\mathsf{crs},(\tau_j^i)_{i\in[l],j\in[\ell]})$, $r_0:=\perp$, and $\mathsf{st}_0:=\mathsf{crs}$.
2. **Emulating $\mathcal{P}^*$:** For every $i\in[l]$, given x, the state st'_{i-1} and a verifier challenge r_{i-1}:

(a) Emulate $\mathcal{P}^*$ with input $\mathsf{st}_{i-1}, x, r_{i-1}$ to obtain a commitment $\mathsf{c}^i = (\mathsf{c}^i_1, \ldots, \mathsf{c}^i_\ell)$ and updated state st_i.

(b) For every $1 \le j \le \ell$, compute $\tilde{\Pi}^i(j) := \mathsf{EE.Ext}(\tau^i_j, \mathsf{c}^i_j)$.

(c) Set $\mathsf{st}'_i := (\mathsf{st}'_{i-1}, \mathsf{st}_i, \mathsf{c}^i, r_{i-1}, \tilde{\Pi}^i)$. Let $\tilde{\Pi}^i = (\tilde{\Pi}^i(j))_{j\in[\ell]}$.

(d) output $\tilde{\Pi}^i$ as the i'th IOP prover message, and st'_i as the updated state.

The probability in Equation (5) is upper bounded by

$$\Pr\left[\mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x; r_1, \ldots, r_l) = 1 \ : \begin{array}{l} \forall i \in [l], j \in [\ell], \ (\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa) \\ \mathsf{st}^*_0 := (\mathsf{crs}^i_j)_{i,j}, \ r_0 := \perp \\ \text{For } i \in [l]: \\ \quad (\mathsf{c}^i, \mathsf{st}^*_i) \leftarrow \mathcal{P}^*(x, \mathsf{st}^*_{i-1}, r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \\ (\mathcal{Q}_i, (\Pi^i(j), \sigma^i_j)_{j\in\mathcal{Q}_i})_{i\in[l]} \leftarrow \mathcal{P}^*(x, \mathsf{st}_l, r_l) \\ \forall i \in [l], j \in [\ell], \ \tilde{\Pi}^i(j) := \mathsf{EE.Ext}(\tau^i_j, \mathsf{c}^i_j) \end{array}\right]$$

which, by the definition of $\mathcal{P}^*_{\mathsf{IOP}}$, is exactly

$$\Pr\left[\mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x; r_1, \ldots, r_l) = 1 \ : \begin{array}{l} \mathsf{st}'_0 := (\mathsf{crs}^i_j, \tau^i_j)_{i,j}, \ r_0 := \perp \\ \text{For } i \in [l]: \\ \quad (\tilde{\Pi}^i, \mathsf{st}'_i) \leftarrow \mathcal{P}^*_{\mathsf{IOP}}(x, \mathsf{st}'_{i-1}, r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \end{array}\right]$$

The latter is at most ϵ_{S} due to the IOP soundness guarantee. ■

Lemma 6.10 (Bounding Probability of Event (2))**.** *Assume that* EE *has extractability error* ϵ_{EE}*. Then Eq. (6) is upper bounded by* ϵ_{EE}*, namely:*

$$\Pr\left[\begin{array}{l} \mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x; r_1, \ldots, r_l) \neq 1 \\ \wedge \ \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i, (\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x; r_1, \ldots, r_l) = 1 \\ \wedge \ \forall i \in [l] \, \forall j \in \mathcal{Q}_i, \ \mathsf{EE.Ver}(\mathsf{crs}^i_j, \mathsf{c}^i_j, \sigma^i_j, \Pi^i(j)) = 1 \end{array} : \begin{array}{l} \forall i \in [l], j \in [\ell], \ (\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa) \\ \mathsf{st}^*_0 := (\mathsf{crs}^i_j)_{i,j}, \ r_0 := \perp \\ \textit{For } i \in [l]: \\ \quad (\mathsf{c}^i, \mathsf{st}^*_i) \leftarrow \mathcal{P}^*(x, \mathsf{st}^*_{i-1}, r_{i-1}) \\ \quad \textit{Sample uniformly random } r_i \\ (\mathcal{Q}_i, (\Pi^i(j), \sigma^i_j)_{j\in\mathcal{Q}_i})_{i\in[l]} \leftarrow \mathcal{P}^*(x, \mathsf{st}_l, r_l) \\ \forall i \in [l], j \in [\ell], \ \tilde{\Pi}^i(j) := \mathsf{EE.Ext}(\tau^i_j, \mathsf{c}^i_j) \end{array}\right] \le \epsilon_{\mathsf{EE}}$$

Proof: We show that if the event in Eq. (6) happens, then the extractability of EE is violated, namely there exist $\mathsf{crs}, \tau, \mathsf{c}, \sigma, b$ (as in Definition 4.4) such that $\mathsf{EE.Ver}(\mathsf{crs}, \mathsf{c}, \sigma, b) = 1 \wedge \mathsf{EE.Ext}(\tau, \mathsf{c}) \neq b$. Since the latter can only happen with probability ϵ_{EE} (by the extractability of EE), this bounds Eq. (6).

Assume the event in Eq. (6) happens. Notice that in both of the executions $\mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x; r_1, \ldots, r_l)$ and $\mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i, (\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x; r_1, \ldots, r_l)$, $\mathcal{V}_{\mathsf{IOP}}$ makes the same oracle queries. Indeed, by definition, $\mathcal{V}_{\mathsf{IOP}}$'s queries are determines solely by x and her randomness, which is the same in both executions. Therefore, since $\mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x; r_1, \ldots, r_l)$ rejects, one of her queries must have received a different response than in the execution of $\mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i, (\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x; r_1, \ldots, r_l)$. That is, there must be

$i^* \in [l], j^* \in [\ell]$ such that $\tilde{\Pi}^{i^*}(j^*) \neq \Pi^{i^*}(j^*)$ and $\mathcal{V}_{\mathsf{IOP}}(x; r_1, \ldots, r_l)$ makes the query $q = (i^*, j^*)$. We show that extractability is violated for $\mathsf{crs}_{j^*}^{i^*}, \tau_{j^*}^{i^*}, \mathsf{c}_{j^*}^{i^*}, \sigma_{j^*}^{i^*}, \Pi^{i^*}(j^*)$. Indeed, by the definition of $\tilde{\Pi}^{i^*}$, we have $\tilde{\Pi}^{i^*}(j^*) = \mathsf{EE.Ext}(\tau_{j^*}^{i^*}, \mathsf{c}_{j^*}^{i^*})$. However, since the event in Eq. (6) happens, then $\mathsf{EE.Ver}(\mathsf{crs}, \mathsf{c}_{j^*}^{i^*}, \sigma_{j^*}^{i^*}, \Pi^{i^*}(j^*)) = 1$ because $j^* \in \mathcal{Q}_{i^*}$ (since $\mathcal{V}_{\mathsf{IOP}}$ made the query (i^*, j^*)). This violates extractability, because $\Pi^{i^*}(j^*) \neq \tilde{\Pi}^{i^*}(j^*)$. ■

6.1.3 Zero Knowledge

Lemma 6.11 (Zero Knowledge). *If $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has (statistical) semi-HVZK (Definition 4.14), and EE has mode indistinguishability (Definition 4.4), then the construction given in Figure 1 has (black-box, straight-line) CZK.*

Proof: We prove the lemma for the case that the IOP has statistical semi-HVZK, the proof for perfect semi-HVZK follows similarly. We define a black-box, straight-line, PPT simulator Sim that simulates the view of any (possibly malicious) PPT verifier $\mathcal{V}^*$. Sim has black-box access to $\mathcal{V}^*$, and uses a simulator $\mathsf{Sim}_{\mathsf{in}}$ for the semi-HVZK property of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$.

On input $1^\kappa, x$, and given black-box, straight-line, access to $\mathcal{V}^*$, Sim operates as follows:

1. **Generating the CRS.** Sim Generates $l \cdot \ell$ trapdoor-mode trapdoor CRSs for the EE, together with equivocal commitments, by executing for every $1 \leq i \leq l$ and every $1 \leq j \leq \ell$: $(\widetilde{\mathsf{tdcrs}}_j^i, \tilde{\mathsf{c}}_j^i, \tilde{\sigma}_j^{i,0}, \tilde{\sigma}_j^{i,1}) \leftarrow \mathsf{EE.TDsetup}(1^\kappa)$.[21] Then, Sim sets $\widetilde{\mathsf{tdcrs}} = (\widetilde{\mathsf{tdcrs}}_j^i)_{i \in [l], j \in [\ell]}$, and sends $\widetilde{\mathsf{tdcrs}}$ to $\mathcal{V}^*$.

2. **Emulating IOP's communication phase:** For every $1 \leq i \leq \ell$, Sim sends $\tilde{\mathsf{c}}^i := (\tilde{\mathsf{c}}_1^i, \ldots, \tilde{\mathsf{c}}_\ell^i)$ to $\mathcal{V}^*$ as the round-i prover message, and obtains from her a challenge $\tilde{r}_i$.

3. **Emulating IOP's query phase:**

   (a) **Simulating IOP answers.** Let $\tilde{r} := (\tilde{r}_1, \ldots, \tilde{r}_\ell)$. Sim emulates $\mathsf{Sim}_{\mathsf{in}}(\tilde{r}, x)$ to simulate $\mathcal{V}_{\mathsf{IOP}}$'s view when using randomness $\tilde{r}$. Let $(a_q)_{q \in \tilde{\mathcal{Q}}}$ denote the simulated answers to $\mathcal{V}_{\mathsf{IOP}}$'s queries to the proof oracles, where $\tilde{\mathcal{Q}}$ denotes the oracle queries which $\mathcal{V}_{\mathsf{IOP}}$ makes when using randomness $\tilde{r}$.

   (b) **Equivocating commitment openings.** Sim sets $\mathsf{op} := \left\{(\tilde{\sigma}_j^{i,a_q}, a_q) \ : \ q = (i,j) \in \tilde{\mathcal{Q}}\right\}$ as the commitment openings.

4. **Output.** Sim outputs $(\widetilde{\mathsf{crs}}, x, \tilde{r}, (\tilde{\mathsf{c}}^i)_{i \in [l]}, \mathsf{op})$.

We now prove that for every PPT verifier $\mathcal{V}^*$, and every $(x, w) \in \mathcal{R}$, the real and simulated views are computationally close. We prove this through a sequence of hybrids.

$\mathcal{H}_0$**:** this is the simulated view which Sim outputs in Step 4.

$\mathcal{H}_1$**:** In $\mathcal{H}_1$, we replace the simulated IOP bits (generated in Step 3a of the simulation) with the actual IOP bits generated by the IOP prover $\mathcal{P}_{\mathsf{IOP}}$ for input $(1^\kappa, x, w)$, when the verifier challenges are $\tilde{r}$.

Claim: $\mathcal{H}_0 \approx \mathcal{H}_1$.

Proof: *This follows from the statistical semi-HVZK property of the IOP system $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$.* Indeed, let Let $\mathcal{D}$ be a PPT distinguisher between $\mathcal{H}_0, \mathcal{H}_1$. We use $\mathcal{D}$ to define a distinguisher $\mathcal{D}_{\mathsf{IOP}}$

[21] Recall that the mode-indistinguishability property of EE guarantees that $\tilde{\sigma}_j^{i,b}$ can be used to open $\tilde{\mathsf{c}}_j^i$ to bit b.

obtaining the same distinguishing advantage between the real and simulated view of $\mathcal{V}_{\mathsf{IOP}}$, namely advantage $\epsilon_{\mathsf{ZK}}(\kappa)$ for some negligible function ϵ_{ZK}. $\mathcal{D}_{\mathsf{IOP}}$ is defined as follows.

The input of $\mathcal{D}_{\mathsf{IOP}}$ is the (real or simulated) view of $\mathcal{V}_{\mathsf{IOP}}$, which consists of $1^\kappa, x$, her randomness r (determining her oracle queries $\mathcal{Q}$), and the answers $(a_q)_{q\in\mathcal{Q}}$ to her oracle queries. $\mathcal{D}_{\mathsf{IOP}}$ execute Step 1 of the simulation (with security parameter κ) to obtain trapdoor CRSs $(\widetilde{\mathsf{tdcrs}}^i_j)_{i\in[l],j\in[\ell]}$, together with equivocal commitments $(\tilde{\mathsf{c}}^i_j)_{i\in[l],j\in[\ell]}$ and openings $(\tilde{\sigma}^{i,0}_j, \tilde{\sigma}^{i,1}_j)_{i\in[l],j\in[\ell]}$, where $\tilde{\sigma}^{i,b}_j$ can be used to open $\tilde{\mathsf{c}}^i_j$ to bit b. Let $\widetilde{\mathsf{tdcrs}} := (\widetilde{\mathsf{tdcrs}}^i_j)_{i\in[l],j\in[\ell]}$, $\tilde{\mathsf{c}} := (\tilde{\mathsf{c}}^i_j)_{i\in[l],j\in[\ell]}$, and $\mathsf{op} := (a_q, \tilde{\sigma}^{i,a_q}_j)_{q=(i,j)\in\mathcal{Q}}$. $\mathcal{D}_{\mathsf{IOP}}$ then emulates $\mathcal{D}$ on input $(\widetilde{\mathsf{tdcrs}}, x, r, \tilde{\mathsf{c}}, \mathsf{op})$ and outputs whatever $\mathcal{D}$ outputs. Notice that if $\mathcal{D}_{\mathsf{IOP}}$ is executed with the simulated view then it emulates $\mathcal{D}$ on $\mathcal{H}_0$, otherwise it emulates $\mathcal{D}$ on $\mathcal{H}_1$. Thus, $\mathcal{D}_{\mathsf{IOP}}$'s distinguishing advantage is identical to that of $\mathcal{D}$, and so by the statistical semi-HVZK property of the IOP, this advantage is negligible.

$\mathcal{H}_2$: In $\mathcal{H}_2$, we modify the setup and the opening of the commitments, as follows:

- We replace the trapdoor-mode CRS with a standard-mode CRS for the EE scheme in Step 1 of the simulation. That is, for every $1 \le i \le l$ and every $1 \le j \le \ell$ we generate $(\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa)$ and set $\mathsf{crs} := (\mathsf{crs}^i_j)_{i\in[l],j\in[\ell]}$.
- We replace the equivocal commitments $\tilde{\mathsf{c}}^i_j$ with real commitments c^i_j to the real IOP bits. This is done by a sequential process, where (unlike in the simulation) for every round i we emulate the real-world prover (with the real witness w) to obtain the round-i proof oracle Π^i, and only then generate the round-i commitments $\mathsf{c}^i_1, \ldots, \mathsf{c}^i_\ell$ by computing, for every $1 \le j \le \ell$, $(\mathsf{c}^i_j, \sigma^i_j) \leftarrow \mathsf{EE.Com}(\mathsf{crs}^i_j, \Pi^i(j))$. We then set $\mathsf{c}^i := (\mathsf{c}^i_1, \ldots, \mathsf{c}^i_\ell)$, send c^i to $\mathcal{V}^*$, and obtain from her the round-i challenge r_i (which is used to generate the round-$(i+1)$ proof oracle). Let $r := (r_1, \ldots, r_l)$ denote the randomness obtained from $\mathcal{V}^*$ throughout this process.
- We replace the equivocated opening of the equivocal commitments $\tilde{c}^i_j$ with honestly-generated openings for the real commitments c^i_j, i.e., for every $q = (i,j) \in \mathcal{Q}$ (where $\mathcal{Q}$ denotes the queries which $\mathcal{V}_{\mathsf{IOP}}$ makes to her oracles when using randomness r), we provide σ^i_j as the opening of c^i_j to Π^i_j.

Claim: $\mathcal{H}_1 \approx \mathcal{H}_2$.

Proof: *This follows from the mode indistinguishability of* EE *(Definition 4.4).*

Assume towards contradiction that there exists a (possibly non-uniform) PPT distinguisher $\mathcal{D}$ obtaining a non-negligible distinguishing advantage $\epsilon = \epsilon(\kappa)$ between $\mathcal{H}_1, \mathcal{H}_2$. We defined a sequence of hybrids $\mathcal{H}^0, \ldots, \mathcal{H}^{l\cdot\ell}$ where $\mathcal{H}^0 = \mathcal{H}_1$, and for every $1 \le s \le l \cdot \ell, s = (i,j)$, $\mathcal{H}^s$ is obtained from $\mathcal{H}^{s-1}$ by replacing the (i,j)'th EE-CRS from a trapdoor-mode CRS $\widetilde{\mathsf{tdcrs}}^i_j$ to a real-mode CRS crs^i_j, replacing the (i,j)'th commitment from an equivocal commitment $\tilde{\mathsf{c}}^i_j$ to a real commitment c^i_j (to $\Pi^i(j)$), and if $(i,j) \in \mathcal{Q}$, also replacing the opening of c^i_j from $\tilde{\sigma}^{i,\Pi^i(j)}_j$ to σ^i_j. Then $\mathcal{H}^{l\cdot\ell} = \mathcal{H}_2$, and by the negation assumption there exists an $s^* = (i^*, j^*) \in [l \cdot \ell]$ such that $\mathcal{D}$ obtains non-negligible advantage $\epsilon' := \epsilon/(l \cdot \ell)$ in distinguishing between $\mathcal{H}^{s^*-1}$ and $\mathcal{H}^{s^*}$. (We note that ϵ' is non-negligible, since both l and ℓ are $\mathsf{poly}(\kappa)$.) We use this to contradict mode indistinguishability of EE.

Let $\mathcal{Q}_{s^*-1}, \mathcal{Q}_{s^*}$ denote $\mathcal{V}_{\mathsf{IOP}}$'s queries in $\mathcal{H}^{s^*-1}, \mathcal{H}^{s^*}$, respectively. (These are random variables whose distribution is induced by the distribution over real and trapdoor-mode CRSs and commitments, respectively.) We use $\mathcal{D}$ to contradict the mode indistinguishability of EE.

Towards that end, consider the following non-uniform PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that has $(i^*, j^*), x, w$ hard wired into it, has black-box access to $\mathcal{V}^*$ (alternatively, $\mathcal{V}^*$'s code can be hard-wired into $\mathcal{A}$) and operates as follows.

1. $\mathcal{A}_1$ is given $1^\kappa, \mathsf{crs}_{j^*}^{i^*}$ as input, where $\mathsf{crs}_{j^*}^{i^*}$ was either honestly generated using $\mathsf{EE.setup}$, or is a trapdoor-mode CRS generated using $\mathsf{EE.TDsetup}$. It operates as follows.
   (a) For every $s := (i,j)$ such that $s < s^*$, $\mathcal{A}_1$ honestly generates the (i,j)'th CRS crs_j^i together with a trapdoor σ_j^i by executing $(\mathsf{crs}_j^i, \sigma_j^i) \leftarrow \mathsf{EE.setup}(1^\kappa)$.
   For every $s := (i,j)$ such that $s > s^*$, $\mathcal{A}_1$ samples the (i,j)'th CRS as a trapdoor-mode CRS tdcrs_j^i together with equivocal commitment $\tilde{\mathsf{c}}_j^i$ and openings $\tilde{\sigma}_j^{i,0}, \tilde{\sigma}_j^{i,1}$ by executing $(\mathsf{tdcrs}_j^i, \tilde{\mathsf{c}}_j^i, \tilde{\sigma}_j^{i,0}, \tilde{\sigma}_j^{i,1}) \leftarrow \mathsf{EE.TDsetup}(1^\kappa)$.
   Let $\mathsf{crs} := ((\mathsf{crs}_j^i)_{(i,j)=s<s^*}, \mathsf{crs}_{j^*}^{i^*}, (\mathsf{tdcrs}_j^i)_{(i,j)=s>s^*})$.
   (b) $\mathcal{A}_1$ then emulates an execution between the honest $\mathcal{P}$ with input crs, x, w and $\mathcal{V}^*$ with input crs, x. More specifically, in this execution $\mathcal{A}_1$ chooses the randomness of both parties. $\mathcal{A}_1$ emulates the honest $\mathcal{P}_{\mathsf{IOP}}$, and commits to its messages, where in round $i, i < i^*$, $\mathcal{A}_1$ honestly commits to the honestly-generated IOP bits. It then feeds this to $\mathcal{V}^*$ as the round-i messages from $\mathcal{P}$, and obtains a challenge r_i from $\mathcal{V}^*$. At the onset of the i^*'th round, $\mathcal{A}_1$ computes $\Pi^{i^*}(j^*)$ (this is possible because $\mathcal{A}_1$ knows w, and $r_1, \ldots, r_{i^*-1}$ have already been determined). It then outputs $\Pi_{j^*}^{i^*}$ as the bit b, and the state st includes $1^\kappa, \mathsf{crs}$, the randomness used to generate the real/trapdoor CRSs for all $s \neq (i^*, j^*)$, the randomness used for $\mathcal{P}_{\mathsf{IOP}}$'s and $\mathcal{V}^*$'s emulation, and the randomness used to generate the commitments up to round i^*.
2. $\mathcal{A}_2$ is given as input the state st, a commitment $\mathsf{c}_{j^*}^{i^*}$ — which was either honestly generated as a commitment to $\Pi^{i^*}(j^*)$, or is an equivocal commitment — and an opening $\sigma_{j^*}^{i^*}$, which was either honestly generated or is an equivocated openning for the bit $b = \Pi^{i^*}(j^*)$. $\mathcal{A}_2$ operates as follows.
   (a) $\mathcal{A}_2$ emulates an execution between the honest $\mathcal{P}$ and $\mathcal{V}^*$. The parties have the same CRS as in $\mathcal{A}_1$'s emulation, and use the same random strings used in $\mathcal{A}_1$'s emulation (as reported in st). Moreover, $\mathcal{A}_2$ uses the same randomness as $\mathcal{A}_1$ to generate the commitments in rounds $1, \ldots, i^* - 1$. At round i^*, for every $j < j^*$, $\mathcal{A}_2$ honestly commits to $\Pi^{i^*}(j)$. Next, it uses $\mathsf{c}_{j^*}^{i^*}$ as the commitment to the j^*'th bit of the i^*'th oracle. Then, for every $j > j^*$ it uses the equivocated commitment $\tilde{\mathsf{c}}_j^{i^*}$ as the commitment to the j'th bit of the i^*'th oracle. In round $i > i^*$, it uses the equivocated commitments generated by $\mathcal{A}_1$. At the end of the emulation, $\mathcal{A}_2$ holds the randomness $r := (r_1, \ldots, r_l)$ which $\mathcal{V}^*$ provided throughout the simulation.
   (b) $\mathcal{A}_2$ uses r to determine the set $\mathcal{Q}$ of $\mathcal{V}_{\mathsf{IOP}}$'s oracle queries. It then sets $\mathcal{V}^*$'s view V to include r, the honestly-generated commitments $(c_j^i)_{s:=(i,j),s<s^*}$, the challenge commitment $c_{j^*}^{i^*}$, the equivocated commitments $(\tilde{\mathsf{c}}_j^i)_{s:=(i,j),s>s^*}$. The view V also includes, for every $q = (i,j), q < s^*$ the honestly-generated opening σ_j^i to $\Pi^i(j)$, and for every $q = (i,j), q > s^*$, the appropriate equivocated opening $\sigma_j^{i,\Pi^i(j)}$ and the bit $\Pi^i(j)$, where $\Pi^i(j)$ is the j'th bit of the i'th proof oracle generated by $\mathcal{P}_{\mathsf{IOP}}$ in his emulation. Finally, if $(i^*, j^*) \in \mathcal{Q}$ then V also includes $\sigma_{j^*}^{i^*}$ and $\Pi^{i^*}(j^*)$.
   (c) $\mathcal{A}_2$ emulates $\mathcal{D}$ on V and outputs whatever $\mathcal{D}$ outputs.

We now show that when $\mathcal{A}$ is given honestly-generated CRS, commitment and opening, then it emulates $\mathcal{D}$ on $\mathcal{H}^{s^*}$, otherwise it emulates $\mathcal{D}$ on $\mathcal{H}^{s^*-1}$. Thus, $\mathcal{A}$'s distinguishing advantage

is ϵ', which is non-negligible. This contradicts the mode indistinguishability of EE. To show this, we consider 3 possible cases.

CASE (1): $q^* := (i^*, j^*) \in \mathcal{Q}_{s^*-1}$ BUT $q^* \notin \mathcal{Q}_{s^*}$, OR VICE VERSA. We claim that this happens only with negligible probability. Indeed, otherwise we can design an adversary $\mathcal{A}' = (\mathcal{A}_1, \mathcal{A}'_2)$ where $\mathcal{A}'_2$ operates similarly to $\mathcal{A}_2$, except that once it obtains r in Step 2a, it halts and outputs 1 if $(i^*, j^*) \in \mathcal{Q}$, otherwise it outputs 0. If this event happens with non-negligible probability, then $\mathcal{A}'$'s advantage in the mode indistinguishability game is non negligible, contradicting mode indistinguishability. Therefore, except with negligible probability $(j^*, i^*) \in \mathcal{Q}_{s^*} \Leftrightarrow (j^*, i^*) \in \mathcal{Q}_{s^*-1}$.

In the following, we condition on the event that $(j^*, i^*) \in \mathcal{Q}_{s^*} \Leftrightarrow (j^*, i^*) \in \mathcal{Q}_{s^*-1}$.

CASE (2): $(i^*, j^*) \in \mathcal{Q}_{s^*-1} \cap \mathcal{Q}_{s^*}$. In this case, $(i^*, j^*) \in \mathcal{Q}$, and indistinguishability again follows directly from the mode indistinguishability property, because if $\mathcal{A}$ is given honestly-generated CRS, commitment and opening, then it emulates $\mathcal{D}$ on $\mathcal{H}^{s^*}$, otherwise it emulates $\mathcal{D}$ on $\mathcal{H}^{s^*-1}$.

CASE (3): $(i^*, j^*) \notin \mathcal{Q}_{s^*-1}$. In this case, $(i^*, j^*) \notin \mathcal{Q}$ and also $(i^*, j^*) \notin \mathcal{Q}_{s^*}$. The analysis here is similar to case (2) above, except that we modify Step 2b such that V does not include the opening $\sigma^{i^*}_{j^*}$ and $\Pi^{i^*}(j^*)$. We again conclude that indistinguishability follows from the mode indistinguishability property.

We conclude the proof by noting that $\mathcal{H}_2$ is the real-world view of $\mathcal{V}^*$. ■

6.2 Achieving Adaptive Soundness

In this section we show that our CZK arguments of Figure 1 achieve *adaptive* soundness, if the underlying IOP also has adaptive soundness. The latter soundness notion is meaningful when the malicious IOP prover is given an auxiliary input, which is sampled according to some distribution $\mathcal{D}$. In such cases, adaptivity means the prover should not be able to cheat, even if he chooses the input statement *after* seeing the auxiliary input. We formalize this notion following [CDGS22, Def. 3.14]:

Definition 6.12 (Adaptive soundness — IOPs)**.** *Let $\epsilon \in (0, 1)$. We say that an IOP system $(\mathcal{P}, \mathcal{V})$ for language $\mathcal{L}$ has* adaptive soundness with error ϵ for distribution $\mathcal{D}$ *if for every (possibly malicious and unbounded) prover $\mathcal{P}^*$, the following holds:*

$$\Pr\left[\begin{matrix}(\mathcal{P}^*(\mathsf{st}, x), \mathcal{V}(x)) = 1 \\ \wedge\ x \notin L\end{matrix} : \begin{matrix}\mathsf{aux} \leftarrow \mathcal{D} \\ (\mathsf{st}, x) \leftarrow \mathcal{P}^*(\mathsf{aux})\end{matrix}\right] \leq \epsilon$$

where $(\mathcal{P}^*(\mathsf{st}, x), \mathcal{V}(x))$ denotes $\mathcal{V}$'s output in the interaction with $\mathcal{P}^*$ when she has input x.

The following notation will be useful:

Notation 6.13. *For $l, \ell \in \mathbb{N}$, let $\mathcal{D}_{l,\ell}$ denote the distribution obtained by sampling $(\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa)$ for every $i \in [l]\,, j \in [\ell]$, and outputting $(\mathsf{crs}^i_j)_{i\in[l], j\in[\ell]}$.*

We now prove the following:

Lemma 6.14 (Adaptive Soundness — CZK of Figure 1)**.** *Let κ be a security parameter, and $\epsilon_{\mathsf{S}} > 0$. If $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has adaptive soundness with error ϵ_{S} (Definition 6.12) for the distribution $\mathcal{D}_{l,\ell}$ of Notation 6.13, and EE has extractability (Definition 4.4) with error ϵ_{EE}, then the construction given in Figure 1 has adaptive soundness with error $\epsilon_{\mathsf{S}} + \epsilon_{\mathsf{EE}}$.*

The proof of Lemma 6.14 follows a similar route to the proof of Lemma 6.7.

Proof of Lemma 6.14 (sketch): Let $\mathcal{P}^*$ be a (possibly malicious) PPT prover in the CZK argument system of Figure 1. We need to upper bound the following distribution by $\epsilon_{\mathsf{S}} + \epsilon_{\mathsf{EE}}$:

$$\Pr\left[\begin{array}{l} \\ \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \forall i\in[l]\,\forall j\in\mathcal{Q}_i,\ \mathsf{EE.Ver}(\mathsf{crs}_j^i,\mathsf{c}_j^i,\sigma_j^i,\Pi^i(j))=1 \\ \wedge\, x\notin\mathcal{L} \\ \\ \end{array} : \begin{array}{l} \forall i\in[l]\,,j\in[\ell]\,,\ (\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa) \\ r_0:=\perp \\ (\mathsf{st}_0^*,x)\leftarrow\mathcal{P}^*((\mathsf{crs}_j^i)_{i\in[l],j\in[\ell]}) \\ \text{For } i\in[l]: \\ \quad (\mathsf{c}^i,\mathsf{st}_i^*)\leftarrow\mathcal{P}^*(x,\mathsf{st}_{i-1}^*,r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \\ (\mathcal{Q}_i,(\Pi^i(j),\sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]}\leftarrow\mathcal{P}^*(x,\mathsf{st}_l,r_l) \end{array}\right] \tag{7}$$

Similar to the proof of Lemma 6.7, the probability in Eq. (7) can be upper bounded by the sum of the probabilities of the follow disjoint events:

Event (1): IOP soundness is violated:

$$\Pr\left[\begin{array}{l} \\ \mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \forall i\in[l]\,\forall j\in\mathcal{Q}_i,\ \mathsf{EE.Ver}(\mathsf{crs}_j^i,\mathsf{c}_j^i,\sigma_j^i,\Pi^i(j))=1 \\ \wedge\, x\notin\mathcal{L} \\ \\ \end{array} : \begin{array}{l} \forall i\in[l]\,,j\in[\ell]\,,\ (\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa) \\ r_0:=\perp \\ (\mathsf{st}_0^*,x)\leftarrow\mathcal{P}^*((\mathsf{crs}_j^i)_{i\in[l],j\in[\ell]}) \\ \text{For } i\in[l]: \\ \quad (\mathsf{c}^i,\mathsf{st}_i^*)\leftarrow\mathcal{P}^*(x,\mathsf{st}_{i-1}^*,r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \\ (\mathcal{Q}_i,(\Pi^i(j),\sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]}\leftarrow\mathcal{P}^*(x,\mathsf{st}_l,r_l) \\ \forall i\in[l]\,,j\in[\ell]\,,\ \tilde{\Pi}^i(j):=\mathsf{EE.Ext}(\tau_j^i,\mathsf{c}_j^i) \end{array}\right] \tag{8}$$

Event (2): EE extraction is violated:

$$\Pr\left[\begin{array}{l} \\ \mathcal{V}_{\mathsf{IOP}}^{(\tilde{\Pi}^i)_{i\in[l]}}(x;r_1,\ldots,r_l)\neq 1 \\ \wedge\, \mathcal{V}_{\mathsf{IOP}}^{[(\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i}))_{i\in[l]}]}(x;r_1,\ldots,r_l)=1 \\ \wedge\, \forall i\in[l]\,\forall j\in\mathcal{Q}_i,\ \mathsf{EE.Ver}(\mathsf{crs}_j^i,\mathsf{c}_j^i,\sigma_j^i,\Pi^i(j))=1 \\ \wedge\, x\notin\mathcal{L} \\ \\ \end{array} : \begin{array}{l} \forall i\in[l]\,,j\in[\ell]\,,\ (\mathsf{crs}_j^i,\tau_j^i)\leftarrow\mathsf{EE.setup}(1^\kappa) \\ r_0:=\perp \\ (\mathsf{st}_0^*,x)\leftarrow\mathcal{P}^*((\mathsf{crs}_j^i)_{i\in[l],j\in[\ell]}) \\ \text{For } i\in[l]: \\ \quad (\mathsf{c}^i,\mathsf{st}_i^*)\leftarrow\mathcal{P}^*(x,\mathsf{st}_{i-1}^*,r_{i-1}) \\ \quad \text{Sample uniformly random } r_i \\ (\mathcal{Q}_i,(\Pi^i(j),\sigma_j^i)_{j\in\mathcal{Q}_i})_{i\in[l]}\leftarrow\mathcal{P}^*(x,\mathsf{st}_l,r_l) \\ \forall i\in[l]\,,j\in[\ell]\,,\ \tilde{\Pi}^i(j):=\mathsf{EE.Ext}(\tau_j^i,\mathsf{c}_j^i) \end{array}\right] \tag{9}$$

The probability in Eq. (8) is bounded by ϵ_{S} similarly to the proof of Lemma 6.9, by a reduction to the *adaptive* IOP soundness. Specifically, we modify Step 1 of the malicious IOP prover $\mathcal{P}^*_{\mathsf{IOP}}$ such that it receives the CRS crs sampled according to $\mathcal{D}_{l,\ell}$ (instead of sampling it himself), and then emulates $\mathcal{P}^*$ with input crs to obtain an input x. The remainder of the proof remains unchanged. We then bound the probability in Eq. (9) by ϵ_{EE} using the exact same proof as in Lemma 6.10. ∎

7 NIZK from CZK Arguments and CI Hash

In this section, we combine CZK arguments with CIH to obtain a *provably secure* construction of NIZK arguments from standard assumptions.

7.1 A BAD Relation and a CIH for BAD

In this section we describe a bad relation for CZK arguments constructed from IOPs and EE commitments. Jumping ahead, our NIZK construction will use a CIH for this bad relation.

A Bad-Challenge Relation for Arguments Obtained from Parallel-Repetition-Based semi-HVZK-IOPs and EEs. We will define a "bad challenge" relation BAD which is associated with an IOP system $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ obtained via parallel-repetition (as in Section 5), and EE commitments. Intuitively, this relation captures bad challenges that allow the adversary to violate soundness. In a multi-round IOP, a bad challenge is one that lets the prover take a partial transcript in a doomed state (as in Definition 4.16) into an extended transcript (i.e., a partial transcript together with a response to the bad challenge) that is not doomed. Since we would like to apply Fiat-Shamir on an IOP-based CZK argument, we define a bad relation with respect to the *underlying IOP* (and EE commitments). The *extractability* of the commitment scheme used in the CZK argument allows us to do this in a meaningful way. Then, using a CIH for this bad relation, and relying on the round-by-round soundness of the IOP, we will be able to prove soundness of the NIZK.

In more detail, for Fiat-Shamir to be sound, we need a CIH that is correlation-intractable with respect to the bad relation. It is therefore natural to use a parallel-repetition version of the IOP, since the corresponding bad relation will then be an (approximate) product relation (this is needed for a CIH to exist via the results of [HLR21]). A similar approach was used for non-interactive arguments constructed *from PCPs* in, e.g., [CJJ21, GLS22]. However, when considering an *interactive* oracle proof as the underlying proof system, one must also account for partial transcripts (as in [CCH+19]). Thus, a bad relation for an IOP should combine aspects of the bad relations from both the PCP and the interactive proofs settings. We use $\mathsf{PartTr}_{\mathsf{CZK}}(i)$ to denote the set of round-i partial transcripts of the CZK argument system; and $\mathsf{PartTr}_{\mathsf{IOP}}(i)$, and $\mathsf{PartTr}^{\otimes}(i)$ to denote the set of round-i partial transcripts of the underlying base IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$, and the parallel repetition IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ respectively. Consider the following relation:

$$\mathsf{BAD}_{\kappa,\tau} := \left\{ ((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}) \ : \ \begin{array}{c} i \in [l] \cup \{0\}, \mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i) : \mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}, \\ \mathsf{c}^j = (\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\ell), \{\Pi^j(s) = \mathsf{EE.Ext}(\tau^j_s, \mathsf{c}^j_s)\}_{j\in[i+1], s\in[\ell]}, \\ \Pi^j = \Pi^j(1) \mid \ldots \mid \Pi^j(\ell), \mathcal{T} = (\Pi^j, r^j)_{j\in[i]} \in \mathsf{PartTr}^{\otimes}(i), \\ (x, \mathcal{T}) \in \mathcal{D}^{\otimes} \wedge (x, \mathcal{T} \mid \Pi^{i+1} \mid r^{i+1}) \notin \mathcal{D}^{\otimes} \end{array} \right\},$$

where l denotes the round complexity of the CZK argument, ℓ is the length of each prover message, $\tau = (\tau^1_1, \ldots, \tau^l_\ell)$ is a list of trapdoors (generated as part of $\mathsf{EE.setup}$), and $\mathcal{D}^{\otimes}$ is the threshold doomed set of the parallel-repetition IOP (see Definition 5.2).

To see why defining the bad relation in this way makes sense, recall that for a CIH to exist via the results of [HLR21], the relation should be: (i) an (approximate) product relation (Definition 4.21) (ii) sparse (Definition 4.22), and (iii) efficient product-verifiable (Definition 4.23). When $\mathcal{D}^{\otimes}$ is the *threshold* doomed set, the resultant relation is both an approximate product relation, and sparse. Indeed, the threshold used to define $\mathcal{D}^{\otimes}$ guarantees that for sufficiently many $j \in [t_\kappa]$, we have $(x, \mathcal{T}_j) \in \mathcal{D}$ (where $\mathcal{T}_j$ is the partial transcript in the j'th repetition). Moreover, for every such j, the set of "witnesses" r^{i+1}_j (for membership in the bad relation) corresponds exactly to challenges

leading the j'th partial transcript from doomed to not doomed. The number of such witnesses is small due to the RBR soundness of the underlying base-IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$.

Unfortunately, this bad relation does not quite work, because it does not have efficient product-verifiability. Intuitively, this is because given $\left((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}\right)$, one should be able to *efficiently* determine whether or not the induced transcripts in the underlying base-IOP are doomed or not (which will then determine whether the transcript in the parallel-repetition is doomed). We therefore require the following additional property from the base-IOP: one can efficiently check whether a given *non-empty* transcript is doomed or not. We call this property "doom checkability", and formally define it in Definition 7.2 below. Notice that if the base-IOP has doom checkability, then the bad relation will have efficient product verifiability for every $\left((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}\right)$ *such that* $\mathcal{T}_{\mathsf{CZK}}$ *is not empty*, since doom checkability guarantees that we can efficiently check whether a non-empty given transcript of the base-IOP is doomed or not. However, doom checkability does not handle *empty transcripts*, and this is *inherent* to any NP-complete language (unless P=NP). Indeed, since for $x \notin \mathcal{L}$ the empty transcript is doomed (from RBR soundness), but for $x \in \mathcal{L}$ it is not (from completeness), an efficient algorithm checking whether $(x, \mathcal{T})$ is doomed for an *empty* $\mathcal{T}$ can be used to decide the language! Luckily, if the base-IOP has extraction, then we can efficiently check whether the empty transcript is doomed or not, by running the extractor. This discussion is summarized and formalized in the following definition of the bad relation.

Definition 7.1 ("Bad" Relation For semi-HVZK-IOP)**.** *Let $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ be an IOP system for a relation $\mathcal{R}$ resulting from parallel repetition of a base-IOP that is RBR sound (as in Construction 5.1). Let ℓ be the length of prover messages,*[22] *and let $\mathcal{D}^{\otimes}$ be the threshold doomed set for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ (Definition 5.2). Assume that the base-IOP has knowledge extraction with extractor* E*. Let* $\mathsf{EE} = (\mathsf{EE.setup}, \mathsf{EE.TDsetup}, \mathsf{EE.Com}, \mathsf{EE.Decom}, \mathsf{EE.Ver}, \mathsf{EE.Ext})$ *be an EE scheme. The* bad-challenge relation BAD *for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ and* EE *is defined as follows. For any security parameter κ, and any* $\mathsf{crs} := (\mathsf{crs}^i_j)_{i\in[l],j\in[\ell]}, \tau = (\tau^i_j)_{i\in[l],j\in[\ell]}$ *s.t. for every* $1 \le j \le \ell$ *and* $1 \le i \le l$, $(\mathsf{crs}^i_j, \tau^i_j) \in \mathsf{Supp}(\mathsf{EE.setup}(1^\kappa))$,

$$\mathsf{BAD}^{\ge 1}_{\kappa,\tau} := \left\{ \left((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}\right) \ : \ \begin{array}{c} i \in [l], \mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i) : \mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}, \\ \mathsf{c}^j = (\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\ell), \{\Pi^j(s) = \mathsf{EE.Ext}(\tau^j_s, \mathsf{c}^j_s)\}_{j\in[i+1],s\in[\ell]}, \\ \Pi^j = \Pi^j(1) \mid \ldots \mid \Pi^j(\ell), \mathcal{T} = \{\Pi^j, r^j\}_{j\in[i]}, \\ (x, \mathcal{T}) \in \mathcal{D}^{\otimes} \wedge (x, \mathcal{T} \mid \Pi^{i+1} \mid r^{i+1}) \notin \mathcal{D}^{\otimes} \end{array} \right\},$$

and

$$\mathsf{BAD}^{=0}_{\kappa,\tau} := \left\{ \left((x, 0, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^1); r^1\right) \ : \ \begin{array}{c} \mathcal{T}_{\mathsf{CZK}} = \emptyset, \mathsf{c}^1 = (\mathsf{c}^1_1, \ldots, \mathsf{c}^1_\ell), \{\Pi^1(s) = \mathsf{EE.Ext}(\tau^1_s, \mathsf{c}^1_s)\}_{s\in[\ell]}, \\ \Pi^1 = \Pi^1(1) \mid \ldots \mid \Pi^1(\ell), r^1 = (r^1_1, \ldots, r^1_{t_\kappa}), \\ \textit{Parse } \Pi^1 \textit{ as } \Pi^1 = \{\Pi^1_1, \ldots, \Pi^1_{t_\kappa}\}, |\Pi^1_k| = u, k \in [t_\kappa] \\ \forall k \in [t_\kappa], (x, \mathsf{E}(x, \Pi^1_k)) \notin \mathcal{R} \wedge (x, \Pi^1 \mid r^1) \notin \mathcal{D}^{\otimes} \end{array} \right\},$$

Let $\mathsf{BAD}_{\kappa,\tau} := \mathsf{BAD}^{\ge 1}_{\kappa,\tau} \cup \mathsf{BAD}^{=0}_{\kappa,\tau}$. *Let* $\mathsf{BAD}_\kappa = \{\mathsf{BAD}_\tau\}_\kappa$ *and* $\mathsf{BAD} = \{\mathsf{BAD}_\kappa\}_\kappa$.

Discussion. The bad relation for an IOP-based CZK argument combines aspects of the bad relations for both PCP-based arguments and interactive arguments. In particular, we require extraction of IOP messages from the commitments in the transcript, so the underlying proof system should have knowledge extraction (as in [CJJ21]).[23] We additionally rely on the existence of a "doomed" set (as in [CCH+19]), capturing the interactive nature of the CZK argument. We note that *all* these

[22] Note that $\ell = t_\kappa \cdot u$ where u is the message length of the underlying base-IOP, and t_κ is the number of repetitions.

[23] As discussed in Section 4.3, our knowledge extraction notion for IOPs generalizes the knowledge extraction notion for the special case of PCPs, considered in [CJJ21].

properties (EE extraction; and extraction and doom-checkability of the IOP) seem to be required to define a bad relation for which a CIH exist (through Theorem 4.28). Indeed, EE extraction is needed for the IOP transcript to be well-defined from the commitments, IOP extraction is needed to determine whether the verifier randomness in round $i = 1$ is "bad", and doom-checkability is needed to enable us to check whether at a given round $i > 1$, the partial transcript moved from doomed to not doomed (which is needed to obtain efficient product verifiability).

We now formalize the notion of doom-checkability discussed above which, roughly, requires that one can efficiently determine whether a given non-empty partial transcript $\mathcal{T}$ for an input x is in the doomed set of the IOP. A similar property was defined for Sigma protocols with unique bad challenges in [CLW18, Def. 3.6] (under the name "efficiently-searchable relation"). We call this property *doom-checkability*. Formally:

Definition 7.2 (Doom-Checkability of an IOP)**.** *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be an IOP for a relation $\mathcal{R}$ satisfying round-by-round soundness with error ϵ_{RBR}, and let $\mathcal{D}$ be the doomed set for Π (Definition 4.16). We say that Π satisfies* doom-checkability *if there exists a polynomial-sized circuit C such that for any x, and any non-empty (partial) transcript $\mathcal{T}$ for x (Definition 4.15), $(x, \mathcal{T}) \in \mathcal{D}$ if and only if $C(x, \mathcal{T}) = 1$.*

Remark 7.3 (Doom-Checkability Implies Doom-Checkability of Parallel Repetition)**.** *We note that the parallel-repetition transformation of Construction 5.1 preserves doom-checkability with respect to the threshold doomed sets $\mathcal{D}^{\otimes}, \mathcal{D}^{\otimes 2}$ of Definitions 5.2 and 5.6, respectively. Indeed, by definition, determining whether or not $(x, \mathcal{T}) \in \mathcal{D}^{\otimes}$ only requires computing the number of partial transcripts of the underlying base-IOP that are doomed. For each of the base-IOP transcripts, this can be done efficiently due to the doom-checkability of the base IOP. It therefore remains to explain how to efficiently compute the required threshold. For the doomed set $\mathcal{D}^{\otimes 2}$ of Definition 5.6, the threshold depends only on $t_\kappa, \epsilon_{\mathsf{RBR}}, \delta$, which are parameters of the system, and can therefore be hard-wired into the doomed-checkability circuit $C^{\otimes}$ of the parallel-repetition IOP. For the doomed set $\mathcal{D}^{\otimes}$ of Definition 5.2, the threshold τ depends only on the number of rounds l, the number of repetitions t_κ, and i where $\mathcal{T} \in \mathsf{PartTr}^{\otimes}(i)$. (More specifically, the threshold for a round-i partial transcript is $1 + \frac{l-i}{l}(t_\kappa - 1)$.) l and t_κ are fully defined by the description of the IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$, and i can be efficiently computed from $\mathcal{T}$. Therefore, the circuit $C^{\otimes}$ for doom-checkability of the parallel repetition computes the threshold τ, interprets the given transcript as the concatenation of t_κ individual transcripts, runs the doom-checkable circuit C for $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ on each of these t_κ transcripts, and outputs 1 if and only if C outputs 1 on τ of the t_κ transcripts.*

Next, we show that assuming LWE, there exist parallel-repetition-based (semi-HVZK) IOPs whose corresponding bad relation has a CIH.

Theorem 7.4 (CIH for IOP)**.** *Let $\kappa \in \mathbb{N}$ be a security parameter. Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be an l-round IOP for $\mathcal{L}$ (Definition 4.11) that satisfies doom-checkability, with RBR soundness error ϵ_{RBR}, and knowledge extraction with error ϵ_{E}, such that $\frac{1}{l} - \max\{\epsilon_{\mathsf{RBR}}, \epsilon_{\mathsf{E}}\} \geq \kappa^{-O(1)}$. Let $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ be the t_κ-fold parallel repetition of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ (Construction 5.1) for $t_\kappa \geq \max\{\frac{64 \cdot \kappa}{\Delta^3}, l + 1\}$ s.t. $t_\kappa - 1$ is not divisible by l, where $\Delta = \frac{1}{l} - \max\{\epsilon_{\mathsf{RBR}}, \epsilon_{\mathsf{E}}\}$,*

Then, under the LWE assumption, there exists a hash family h which is correlation intractable for the bad relation BAD for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ of Definition 7.1, and h can be evaluated in time $\mathsf{poly}(\kappa, |\mathcal{R}|)$, where $|\mathcal{R}|$ is the size of the verification circuit of $\mathcal{R}$.

Proof: Let $\mathsf{BAD} = \{\mathsf{BAD}_\kappa \subseteq X_\kappa \times Y_\kappa^{t_\kappa}\}_\kappa$. We first define a new relation $\widehat{\mathsf{BAD}} = \{\widehat{\mathsf{BAD}}_\kappa \subseteq X_\kappa \times Y_\kappa^{t_\kappa}\}_\kappa$ over the same domain and range as BAD. $\widehat{\mathsf{BAD}}$ is defined as follows, where $\theta := (t_\kappa - 1)/l$:

$$\widehat{\mathsf{BAD}}^{\geq 1}_{\kappa,\tau} := \left\{ \left((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1});r^{i+1}\right) \ : \ \begin{array}{c} i\in[l],\mathcal{T}_{\mathsf{CZK}}\in\mathsf{PartTr}_{\mathsf{CZK}}(i):\mathcal{T}_{\mathsf{CZK}}=\{\mathsf{c}^j,r^j\}_{j\in[i]},\\ \mathsf{c}^j=(\mathsf{c}^j_1,\ldots,\mathsf{c}^j_\ell),\{\Pi^j(s)=\mathsf{EE.Ext}(\tau^j_s,\mathsf{c}^j_s)\}_{j\in[i+1],s\in[\ell]},\\ \Pi^j=\Pi^j(1)\mid\ldots\mid\Pi^j(\ell),\mathcal{T}=\{\Pi^j,r^j\}_{j\in[i]},\\ (x,\mathcal{T})\in\mathcal{D}^{\otimes},\text{ and }\exists\,\Gamma\subseteq[t_\kappa]\text{ s.t. }|\Gamma|\geq\theta,\\ \forall k\in\Gamma,\ (x,\mathcal{T}_k)\in\mathcal{D}_{\mathsf{IOP}}\wedge(x,\mathcal{T}_k\mid\Pi^{i+1}_k\ |r^{i+1}_k)\notin\mathcal{D}_{\mathsf{IOP}}\end{array}\right\}$$

Let $\widehat{\mathsf{BAD}}_{\kappa,\tau} := \widehat{\mathsf{BAD}}^{\geq 1}_{\kappa,\tau}\cup\mathsf{BAD}^{=0}_{\kappa,\tau}$, let $\widehat{\mathsf{BAD}}_{\kappa}=\{\widehat{\mathsf{BAD}}_{\kappa,\tau}\}_\tau$, and $\widehat{\mathsf{BAD}}=\{\widehat{\mathsf{BAD}}_{\kappa}\}_\kappa$.

We will now prove that $\widehat{\mathsf{BAD}}\supseteq\mathsf{BAD}$ by showing that for every κ,τ, $\widehat{\mathsf{BAD}}^{\geq 1}_{\kappa,\tau}\supseteq\mathsf{BAD}^{\geq 1}_{\kappa,\tau}$. Then, we will show that there exists a CIH family h for $\widehat{\mathsf{BAD}}$, and since $\mathsf{BAD}\subseteq\widehat{\mathsf{BAD}}$, we will conclude that h is CI for BAD.

Step(1): $\widehat{\mathsf{BAD}}\supseteq\mathsf{BAD}$. Fix a security parameter κ, $\mathsf{crs}:=(\mathsf{crs}^i_j)_{i\in[l],j\in[\ell]}$, and $\tau=(\tau^i_j)_{i\in[l],j\in[\ell]}$ s.t. for every $1\leq j\leq\ell$ and $1\leq i\leq l$, $(\mathsf{crs}^i_j,\tau^i_j)\in\mathsf{Supp}(\mathsf{EE.setup}(1^\kappa))$. This fixes relations $\widehat{\mathsf{BAD}}_{\kappa,\tau}\in\widehat{\mathsf{BAD}}_\kappa$, and $\mathsf{BAD}_{\kappa,\tau}\in\mathsf{BAD}_\kappa$. For every κ, we have that the relations $\mathsf{BAD}_{\kappa,\tau}$ and $\widehat{\mathsf{BAD}}_{\kappa,\tau}$ are defined over the same domain and range.

We now show that $((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1});r^{i+1})\in\mathsf{BAD}^{\geq 1}_{\kappa,\tau}\implies((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1});r^{i+1})\in\widehat{\mathsf{BAD}}^{\geq 1}_{\kappa,\tau}$. Given $((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1});r^{i+1})\in\mathsf{BAD}^{\geq 1}_{\kappa,\tau}$, let $\mathcal{T}_{\mathsf{CZK}}=\{\mathsf{c}^j,r^j\}_{j\in[i]}$, and $\mathsf{c}^j=(\mathsf{c}^j_1,\ldots,\mathsf{c}^j_\ell)$. Let $\{\Pi^j(s)=\mathsf{EE.Ext}(\tau^j_s,\mathsf{c}^j_s)\}_{j\in[i+1],s\in[\ell]}$, and $\Pi^j=\Pi^j(1)\mid\ldots\mid\Pi^j(\ell)$, for $j\in[i+1]$. Parse Π^j as $\Pi^j=(\Pi^j_1,\ldots,\Pi^j_{t_\kappa})$ where $|\Pi^j_k|=u$ for all $k\in[t_\kappa]$. That is, Π^j consists of the jth round oracle message, which is composed of the jth round oracle messages of t_κ repetitions of the underlying base-IOP. Similarly, parse r^j as $\{r^j=(r^j_1,\ldots,r^j_{t_\kappa})\}_{j\in[i+1]}$, and let $\mathcal{T}_k=(\Pi^j_k,r^j_k)_{k\in[i]}\in\mathsf{PartTr}_{\mathsf{IOP}}(i)$ for all $k\in[t_\kappa]$. (Recall that $\mathsf{PartTr}_{\mathsf{IOP}}(i)$ denotes the set of round-i partial transcripts of $(\mathcal{P}_{\mathsf{IOP}},\mathcal{V}_{\mathsf{IOP}})$.) Let $\mathcal{T}=\{\Pi^j,r^j\}_{j\in[i]}$. Since $((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1});r^{i+1})\in\mathsf{BAD}^{\geq 1}_{\kappa,\tau}$, we have, (1) $(x,\mathcal{T})\in\mathcal{D}^{\otimes}$ and (2) $(x,\mathcal{T}\mid\Pi^{i+1}\ |r^{i+1})\notin\mathcal{D}^{\otimes}$. $(x,\mathcal{T})\in\mathcal{D}^{\otimes}$ implies that $|\{k\in[t_\kappa]\ :\ (x,\mathcal{T}_k)\in\mathcal{D}_{\mathsf{IOP}}\}|\geq 1+\frac{l-i}{l}(t_\kappa-1)$. $(x,\mathcal{T}|\Pi^{i+1}|r^{i+1})\notin\mathcal{D}^{\otimes}$ implies $|\{k\in[t_\kappa]\ :\ (x,\mathcal{T}_k,\Pi_k,r_k)\notin\mathcal{D}_{\mathsf{IOP}}\}|\geq\frac{(i+1)(t_\kappa-1)}{l}$ (see Remark 5.3).

Therefore, it must be the case that at least $\frac{t_\kappa-1}{l}$ independent executions out of d executions switched from doomed to not doomed, where d is number of doomed repetitions in the ith round, $\mathsf{d}:=|\{k\in[t_\kappa]\ :\ (x,\mathcal{T}_k)\in\mathcal{D}_{\mathsf{IOP}}\}|>1+\frac{l-i}{l}(t_\kappa-1)$ (again, see remark 5.3). That is, $\left|k\in[t_\kappa]:(x,\mathcal{T}_k)\in\mathcal{D}_{\mathsf{IOP}}\wedge(x,\mathcal{T}_k\mid\Pi^{i+1}_k\ |r^{i+1}_k)\notin\mathcal{D}_{\mathsf{IOP}}\right|\geq(t_\kappa-1)/l$. Therefore, $((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1});r^{i+1})\in\widehat{\mathsf{BAD}}^{\geq 1}_{\kappa,\tau}$. This shows that every instance in $\mathsf{BAD}^{\geq 1}_{\kappa,\tau}$ for some κ,τ is in $\widehat{\mathsf{BAD}}^{\geq 1}_{\kappa,\tau}$ and therefore in $\widehat{\mathsf{BAD}}_{\kappa,\tau}$. Every instance in $\mathsf{BAD}^{=0}_{\kappa,\tau}$ is in $\widehat{\mathsf{BAD}}_{\kappa,\tau}$ by definition. Therefore, $\mathsf{BAD}_{\kappa,\tau}\subseteq\widehat{\mathsf{BAD}}_{\kappa,\tau}$ for every κ,τ.

Step(2): A CIH for $\widehat{\mathsf{BAD}}$. To show that there exists a CIH for the relation $\widehat{\mathsf{BAD}}$, we show that $\widehat{\mathsf{BAD}}$ is an ensemble of α-approximate product relations that are product-verifiable with product sparsity at most $\rho<\alpha$, and then invoke Theorem 4.28. In the following, fix some security parameter κ, $\mathsf{crs}:=(\mathsf{crs}^i_j)_{i\in[l],j\in[\ell]}$, and $\tau=(\tau^i_j)_{i\in[l],j\in[\ell]}$ s.t. for every $1\leq j\leq\ell$ and $1\leq i\leq l$, $(\mathsf{crs}^i_j,\tau^i_j)\in\mathsf{Supp}(\mathsf{EE.setup}(1^\kappa))$. This fixes a relation $\widehat{\mathsf{BAD}}_{\kappa,\tau}\in\widehat{\mathsf{BAD}}$. Let E denote the knowledge extractor for $(\mathcal{P}_{\mathsf{IOP}},\mathcal{V}_{\mathsf{IOP}})$ with respect to which doom-checkability is defined.

$\widehat{\mathsf{BAD}}_{\kappa,\tau}$ is an approximate product relation for $\alpha=\frac{t_\kappa-1}{lt_\kappa}$. We will show that for any $i\in\{0,1,\ldots,l\}$, x, $\mathcal{T}_{\mathsf{CZK}}\in\mathsf{PartTr}_{\mathsf{CZK}}(i)$, and any c^{i+1}, the set $R_{i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}:=\{r^{i+1}:((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}),r^{i+1})\in\widehat{\mathsf{BAD}}_{\kappa,\tau}\}$ has a decomposition $\{R_{1,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}},\ldots,R_{t_\kappa,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\}$, for some

$R_{1,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}},\ldots,R_{t_\kappa,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$, that satisfies Definition 4.21. If $i \geq 1$, we use the notation from the definition of the bad relation. Namely, let $\mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}$, and $\mathsf{c}^j = (\mathsf{c}_1^j,\ldots,\mathsf{c}_\ell^j)$. Let $\{\Pi^j(s) = \mathsf{EE.Ext}(\tau_s^j,\mathsf{c}_s^j)\}_{j\in[i+1],s\in[\ell]}$, and $\Pi^j = \Pi^j(1) \mid\ldots\mid \Pi^j(\ell)$, for $j \in [i+1]$. Parse Π^j as $\Pi^j = (\Pi_1^j,\ldots,\Pi_{t_\kappa}^j)$ where $|\Pi_k^j| = u$ for all $k \in [t_\kappa]$. That is, Π^j consists of the jth round oracle message, which is composed of the jth round oracle messages of t_κ repetitions of the underlying base-IOP. Similarly, parse r^j as $\{r^j = (r_1^j,\ldots,r_{t_\kappa}^j)\}_{j\in[i+1]}$, and let $\mathcal{T}_k = (\Pi_k^j, r_k^j)_{j\in[i]} \in \mathsf{PartTr}_{\mathsf{IOP}}(i)$ for all $k \in [t_\kappa]$. (Recall that $\mathsf{PartTr}_{\mathsf{IOP}}(i)$ denotes the set of round-i partial transcripts of $(\mathcal{P}_{\mathsf{IOP}},\mathcal{V}_{\mathsf{IOP}})$.) With this notation in place, we proceed to prove that $\widehat{\mathsf{BAD}}_{\kappa,\tau}$ is an α-approximate product relation.

We now consider two possible cases, based on the value of i. Assume first that $i \geq 1$. For every $k \in [t_\kappa]$, we define the set $R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$ to be the set of random strings r_k^{i+1} for which $\big((x,\mathcal{T}) \in \mathcal{D}^{\otimes} \wedge (x,\mathcal{T}_k) \in \mathcal{D}_{\mathsf{IOP}} \wedge (x,\mathcal{T}_k|\Pi_k^{i+1}|r_k^{i+1}) \notin \mathcal{D}_{\mathsf{IOP}}\big)$.

By definition of $\widehat{\mathsf{BAD}}_{\kappa,\tau}$, if $r^{i+1} \in R_{i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$, and $i \geq 1$ then $((x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1});r^{i+1}) \in \widehat{\mathsf{BAD}}^{\geq 1}_{\kappa,\tau}$, and therefore (1) $(x,\mathcal{T}) \in \mathcal{D}^{\otimes}$ and (2) $\big|k \in [t_\kappa] : (x,\mathcal{T}_k) \in \mathcal{D}_{\mathsf{IOP}} \wedge (x,\mathcal{T}_k \mid \Pi_k^{i+1}\ |r_k^{i+1}) \notin \mathcal{D}_{\mathsf{IOP}}\big| \geq (t_\kappa - 1)/l$. Therefore, any $r^{i+1} \in R_{i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$ is such that $\big|k \in [t_\kappa] : r_k^{i+1} \in R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\big| \geq (t_\kappa - 1)/l = \alpha t_\kappa$.

In the other direction, if r^{i+1} is such that $\big|k \in [t_\kappa] : r_k^{i+1} \in R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\big| \geq \alpha \cdot t_\kappa = (t_\kappa - 1)/l \geq 1$, then $\big|k \in [t_\kappa] : (x,\mathcal{T}) \in \mathcal{D}^{\otimes} \wedge (x,\mathcal{T}_k) \in \mathcal{D}_{\mathsf{IOP}}) \wedge (x,\mathcal{T}_k \mid \Pi_k^{i+1}\ |r_k^{i+1}) \notin \mathcal{D}_{\mathsf{IOP}}\big| \geq (t_\kappa - 1)/l$. Hence $r^{i+1} \in R_{i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$.

Next, assume that $i = 0$. Then for every $1 \leq k \leq t_\kappa$, we define the set $R_{k,0,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^1}$ to be the set of random strings r_k^1 for which (1) $(x,\mathsf{E}(x,\Pi_{k'}^1)) \notin \mathcal{R}$ for all $k' \in [t_\kappa]$, and (2) $(x,\Pi_k^1|r_k^1) \notin \mathcal{D}_{\mathsf{IOP}}$. If $r^1 \in R_{0,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^1}$ then by definition of $\mathsf{BAD}^{=0}_{\kappa,\tau}$, $(x,\mathsf{E}(x,\Pi_{k'}^1)) \notin \mathcal{R}$ for all $k' \in [t_\kappa]$, and $(x,\Pi^1\ |r^1) \notin \mathcal{D}^{\otimes}$. Moreover, $(x,\Pi^1\ |r^1) \notin \mathcal{D}^{\otimes}$ implies that $\big|k \in [t_\kappa] : (x,\Pi_k^1\ |r_k^1) \notin \mathcal{D}_{\mathsf{IOP}}\big| > (i+1)(t_\kappa-1)/l = (t_\kappa-1)/l = \alpha \cdot t_\kappa$ (see Remark 5.3, setting $i+1=1$ as per the case assumption). In the other direction, if r^1 is such that $\big|k \in [t_\kappa] : r_k^1 \in R_{k,0,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^1}\big| \geq (t_\kappa-1)/l \geq 1$, then, $(x,\mathsf{E}(x,\Pi_k^1)) \notin \mathcal{R}$ for every $k' \in [t_\kappa]$, and $\big|k \in [t_\kappa] : (x,\Pi_k^1\ |r_k^1) \notin \mathcal{D}_{\mathsf{IOP}}\big| \geq \lceil(t_\kappa-1)/l\rceil > (t_\kappa-1)/l$[24]. Then, by definition of $\mathcal{D}^{\otimes}$, $(x,\Pi^1\ |r^1) \notin \mathcal{D}^{\otimes}$, and hence $r^1 \in R_{0,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^1}$.

$\widehat{\mathsf{BAD}}_{\kappa,\tau}$ is sparse with sparsity $\rho = \max\{\epsilon_{\mathsf{RBR}},\epsilon_{\mathsf{E}}\}$. We show that for any round $i \in \{0,1,\ldots,l\}$, x, any partial transcript $\mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i)$, and any commitment c^{i+1}, the corresponding sets $\mathcal{R}_{1,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}},\ldots,\mathcal{R}_{t_\kappa,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}} \subseteq Y$ have size $\leq \rho|Y|$, where $\rho = \max\{\epsilon_{\mathsf{RBR}},\epsilon_{\mathsf{E}}\}$, the maximum between round-by-round soundness error and the knowledge extraction error of the underlying base-IOP $(\mathcal{P}_{\mathsf{IOP}},\mathcal{V}_{\mathsf{IOP}})$, and Y is the randomness space for each round of $\mathcal{V}_{\mathsf{IOP}}$'s messages. Let $\mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}$, and $\mathsf{c}^{i+1} = (\mathsf{c}_1^{i+1},\ldots,\mathsf{c}_\ell^{i+1})$. Let $\{\Pi^j(s) = \mathsf{EE.Ext}(\tau_s^j,\mathsf{c}_s^j)\}_{j\in[i+1],s\in[\ell]}$, and $\Pi^j = \Pi^j(1) \mid\ldots\mid \Pi^j(\ell)$, for $j \in [i+1]$. Parse Π^j as $\Pi^j = \{\Pi_1^j,\ldots,\Pi_{t_\kappa}^j\}, |\Pi_k^j| = u, k \in [t_\kappa]$. That is, Π^j consists of the $j$the round oracle message, which is composed of the jth round oracle messages of t_κ repetitions of the underlying IOP. Parse r^j as $\{r^j = (r_1^j,\ldots,r_{t_\kappa}^j)\}_{j\in[i+1]}$, and let $\{\mathcal{T}_k = (\Pi_k^j, r_k^j)_{j\in[i+1]}\}_{k\in[t_\kappa]}$.

Towards contradiction, assume that there exists a $1 \leq k \leq t_\kappa$ such that $\big|\mathcal{R}_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\big| > \rho|Y|$. We consider two possible cases, based on the value of i. Assume first that $i \geq 1$. By definition, $\mathcal{R}_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$ consists exactly of the subset of random strings $\tilde{r}_k^{i+1} \in Y$ such that $(x,\mathcal{T}) \in \mathcal{D}^{\otimes} \wedge (x,\mathcal{T}_k) \in \mathcal{D}_{\mathsf{IOP}} \wedge (x,\mathcal{T}_k|\Pi_k^{i+1}|\tilde{r}_k^{i+1}) \notin \mathcal{D}_{\mathsf{IOP}}$. Since $\big|\mathcal{R}_{k,i,x,\mathcal{T},\mathsf{c}^{i+1}}\big| > \rho|Y|$ then for Π_k^{i+1} we have:

$$\Pr_{r\leftarrow Y}\left[(x,\mathcal{T}_k|\Pi_k^{i+1}|r) \notin \mathcal{D}_{\mathsf{IOP}}\right] > \rho = \max\{\epsilon_{\mathsf{RBR}},\epsilon_{\mathsf{E}}\} \geq \epsilon_{\mathsf{RBR}}$$

This contradicts the round-by-round-soundness of the underlying IOP.

[24] We set t_κ such that l does not divide $t_\kappa - 1$, and hence $\big|k \in [t_\kappa] : (x,\Pi_k^1\ |r_k^1) \notin \mathcal{D}_{\mathsf{IOP}}\big| \geq \lceil(t_\kappa-1)/l\rceil > (t_\kappa-1)/l$.

Next, assume that $i = 0$. By definition, $\mathcal{R}_{k,0,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^0}$ consists exactly of the subset of random strings $\tilde{r}^1_k$ such that $(x, \mathsf{E}(x, \Pi^1_{k'})) \notin \mathcal{R}$ for all $k' \in [t_\kappa]$ and $(x, \Pi^1_k | \tilde{r}^1_k) \notin \mathcal{D}_{\mathsf{IOP}}$. We consider three possible cases. First, if $x \notin \mathcal{L}$, then the negation assumption implies that

$$\Pr_{r \leftarrow Y}\left[(x, \Pi^1_k | r) \notin \mathcal{D}_{\mathsf{IOP}}\right] > \rho = \max\{\epsilon_{\mathsf{RBR}}, \epsilon_{\mathsf{E}}\} \tag{10}$$

which violates the RBR soundness of the base IOP, since $x \notin \mathcal{L}$ implies that the empty transcript is doomed. Second, if $x \in \mathcal{L}$ and $(x, \mathsf{E}(x, \Pi^1_{k'})) \in \mathcal{R}$ for some $k' \in [t_\kappa]$, then $\mathcal{R}_{k,0,x,\mathcal{T},\mathsf{c}^1} = \emptyset$ (and so satisfies sparsity). Finally, if $x \in \mathcal{L}$ and $(x, \mathsf{E}(x, \Pi^1_{k'})) \notin \mathcal{R}$ for all $k' \in [t_\kappa]$, then Eq. (10) contradicts the ϵ_{E}-extractability of the base IOP for $k' = k$. Therefore, $\left|\mathcal{R}_{j,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\right| \leq \rho\,|Y|$.

$\widehat{\mathsf{BAD}}_{\kappa,\tau}$ is efficient product verifiable. $\widehat{\mathsf{BAD}}_{\kappa,\tau}$ **is efficient product verifiable.** Consider a circuit $C_{\kappa,\tau}$ that has κ, τ hard-coded into it, and takes as input $(x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1})$, r^{i+1}_k, and $k \in [t_\kappa]$. Since $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ is doom-checkable, there exists a poly-size circuit C_{IOP} such that for every non-empty $\mathcal{T}_k \in \mathsf{PartTr}_{\mathsf{IOP}}(i)$, $C_{\mathsf{IOP}}(x, \mathcal{T}_k) = 1$ if and only if $(x, \mathcal{T}_k) \in \mathcal{D}_{\mathsf{IOP}}$. Doom-checkability of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ implies that $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$ is doom-checkable (Remark 7.3), and hence there exists a poly-size circuit $C^\otimes$ such that for every non-empty $\mathcal{T} \in \mathsf{PartTr}^\otimes(i)$, $C^\otimes(x, \mathcal{T}) = 1$ if and only if $(x, \mathcal{T}) \in \mathcal{D}^\otimes$. $C_{\kappa,\tau}$ works as follows: It parses $\mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}$, where each $\mathsf{c}^j = (\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\ell)$. It then runs the EE extractor $\{\Pi^j(s) = \mathsf{EE.Ext}(\tau^j_s, \mathsf{c}^j_s)\}_{j\in[i+1],s\in[\ell]}$, and sets $\Pi^j := \Pi^j(1) \,|\ldots|\, \Pi^j(\ell)$, for every $j \in [i+1]$. It then parses Π^j as $\Pi^j = \{\Pi^j_1, \ldots, \Pi^j_{t_\kappa}\}, |\Pi^j_k| = u, k \in [t_\kappa]$. Next, it parses r^j as $\{r^j = (r^j_1, \ldots, r^j_{t_\kappa})\}_{j\in[i+1]}$. It let $\mathcal{T} = \{\Pi^j, r^j\}_{j\in[i]}$, and $\mathcal{T}_k = (\Pi^j_k, r^j_k)_{j\in[i]}$. Then, it computes its output in one of two ways, depending on the value of i.
Case (1): $i \geq 1$: $C_{\kappa,\tau}$ computes $b_1 = C_{\mathsf{IOP}}(x, \mathcal{T}_k)$, $b_2 = C_{\mathsf{IOP}}(x, \mathcal{T}_k | \Pi^{i+1}_k | r^{i+1}_k)$, and sets $b' = b_1 \wedge \bar{b}_2$. $C_{\kappa,\tau}$ then computes $b = C^\otimes(x, \mathcal{T})$. $C_{\kappa,\tau}$ outputs $b \wedge b'$.
Case (2): $i = 0$: let C denote the verification circuit of $\mathcal{R}$. Then $C_{\kappa,\tau}$ computes $b_{1,k'} = C(x, \mathsf{E}(x, \Pi^1_{k'}))$ for all $k' \in [t_\kappa]$, $b_1 = \vee_{k'} b_{1,k'}$, and $b_2 = C_{\mathsf{IOP}}(x, \Pi^1_k | r^1_k)$, and outputs $\bar{b}_1 \wedge \bar{b}_2$.
Correctness follows from doom-checkability and extractability of the base-IOP.

We now analyze the size of $C_{\kappa,\tau}$. Let $T_1(\kappa)$ be the size of the EE extractor, $T_2(\kappa), T^\otimes_2(\kappa)$ be the sizes of $C_{\mathsf{IOP}}, C^\otimes$ (respectively), T_3 denote the size of the verification circuit C, and T_4 denote the size of E. Then $C_{\kappa,\tau}$ has size $T = O(l \cdot \ell \cdot T_1 + 3T_2 + T^\otimes_2 + t_\kappa(T_3 + T_4)) = \mathsf{poly}(\kappa, l, \ell)$.

Setting the Parameters. Set $t_\kappa := \max\{\lceil \frac{64\cdot\kappa}{\Delta^3} \rceil, l + 1\}$. We have shown that for every $\kappa, \mathsf{crs} := (\mathsf{crs}^i_j)_{i\in[l],j\in[\ell]}$, and $\tau = (\tau^i_j)_{i\in[l],j\in[\ell]}$ s.t. $(\mathsf{crs}^i_j, \tau^i_j) \in \mathsf{Supp}(\mathsf{EE.setup}(1^\kappa))$ for every $1 \leq j \leq \ell$ and $1 \leq i \leq l$, $\widehat{\mathsf{BAD}}_{\kappa,\tau} \subseteq X_\kappa \times Y^{t_\kappa}_\kappa$ satisfies the following properties: (1) it is T-time product verifiable with sparsity $\rho = \max\{\epsilon_{\mathsf{RBR}}, \epsilon_{\mathsf{E}}\}$; (2) an approximate product relation with approximation parameter $\alpha = \frac{t_\kappa - 1}{lt_\kappa}$ (3) $T = \mathsf{poly}(\kappa, l, \ell)$, and $\log|X_\kappa|, T(\kappa), t_\kappa$ are all $\mathsf{poly}(\kappa)$. To see why (3) holds, recall that $|X_\kappa| = 2^{\mathsf{poly}(\kappa,n)}$ because it consists of an instance, a round number, partial transcript, and $\mathsf{poly}(n, \kappa)$ commitments, each of size $\mathsf{poly}(n, \kappa)$, so overall all these sets have size $\mathsf{poly}(\kappa)$ (here, we use the fact that $l, \ell = \mathsf{poly}(n, \kappa)$ and n, κ are polynomially related).

We verify that the hypotheses for the large alphabet approximate CI theorem (Theorem 4.28) hold. We have $\alpha - \rho = \frac{t_\kappa - 1}{lt_\kappa} - \max\{\epsilon_{\mathsf{RBR}}, \epsilon_{\mathsf{E}}\} = \frac{1}{l} - \frac{1}{lt_\kappa} - \max\{\epsilon_{\mathsf{RBR}}, \epsilon_{\mathsf{E}}\} = \Delta - \frac{1}{lt_\kappa}$. We claim that $\frac{1}{lt_\kappa} \leq \frac{\Delta}{2}$, which yields $\alpha - \rho \geq \frac{\Delta}{2}$. To see this, set $C := 64$, so we have $t_\kappa \geq \frac{C\cdot\kappa}{\Delta^3}$, and hence $\frac{1}{lt_\kappa} \leq \frac{\Delta^3}{C\cdot\kappa\cdot l}$. Since $\Delta \leq \frac{1}{l}$, $\Delta^2 \leq \frac{1}{l^2}$. Therefore, $\frac{\Delta^3}{C\cdot\kappa\cdot l} = \Delta\frac{\Delta^2}{C\cdot\kappa\cdot l} \leq \frac{\Delta}{C\cdot\kappa\cdot l^3}$. Hence the right hand side is at most $\Delta/2$ as claimed (since $64 \cdot \kappa \cdot l > 2$). Therefore, $\alpha - \rho \geq \frac{\Delta}{2} \geq \kappa^{-O(1)}$. We need $t_\kappa \geq \frac{8\kappa}{(\alpha-\rho)^3}$. Since $\alpha - \rho \geq \frac{\Delta}{2}$, $\frac{8\kappa}{(\alpha-\rho)^3} \leq \frac{8\kappa}{(\Delta/2)^3} = \frac{64\kappa}{\Delta^3}$.

Now, by Theorem 4.28 there exists a hash family h that is correlation-intractable for $\widehat{\mathsf{BAD}}$ under the LWE assumption, and h can be evaluated in time $\mathsf{poly}(\kappa, \log|X_\kappa|, t_\kappa, T) = \mathsf{poly}(\kappa, |\mathcal{R}|)$. Since $\widehat{\mathsf{BAD}} \supseteq \mathsf{BAD}$, h is correlation-intractable for BAD.

■

We show an example of a semi-HVZK-IOP for NP that has the properties specified in Theorem 7.4 (i.e., satisfies doom-checkability, RBR soundness and knowledge extraction), in Appendix B.

Handling IOPs with Imperfect Completeness. When using the CIH of Theorem 7.4 to construct NIZKs in Section 7.2, we will apply the CIH to an underlying IOP with *imperfect completeness*. To argue completeness, we would then need the CIH to have an additional property of k-wise independence of outputs (Definition 4.3), which will be used to claim that when the CIH inputs are sampled honestly (i.e., randomly and independently for each IOP round) then the CIH output does not skew the distribution "too much". The existence of such a CIH follows from Theorem 7.4 using [CCH+19, Remark 5.2]. For self containment, we provide the details below.

In the following, let $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ be an l-round IOP with the properties stated in Theorem 7.4, let BAD be its bad relation (of Definition 7.1), and $\widehat{\mathsf{BAD}}$ the relation defined in the proof of Theorem 7.4. Let h' be an l-wise independent function as in Definition 4.3 (such functions exist unconditionally). For every security parameter κ and every $K' \in \mathsf{Supp}(\mathsf{h}')$, define $\mathsf{BAD}^{\star}_{\kappa,K'} := \left\{(x,r) \ : \ (x, r \oplus K'(x)) \in \widehat{\mathsf{BAD}}\right\}$.

Lemma 7.5. *Let $l \in \mathbb{N}$. Then under the conditions of Theorem 7.4, there exists a hash family h that is both l-wise independent, and correlation intractable for the bad relation BAD of Definition 7.1.*

Proof (sketch): Let h' be a family of l-wise independent hash functions. We define a new bad relation $\mathsf{BAD}^{\star}$ as follows: let κ, τ be as in Definition 7.1. For every $K' \in \mathsf{Supp}(\mathsf{h}')$, define:

$$\mathsf{BAD}^{\star}_{\kappa,\tau,K'} := \left\{(x,r) \ : \ (x, r \oplus K'(x)) \in \widehat{\mathsf{BAD}}_{\kappa,\tau}\right\}$$

and let $\mathsf{BAD}^{\star}_{\kappa} = \left\{\mathsf{BAD}^{\star}_{\kappa,\tau,K'}\right\}_{\tau,K'}$ and $\mathsf{BAD}^{\star} = \{\mathsf{BAD}^{\star}\}_{\kappa}$.

We prove the lemma in two steps. First, we show that if there exists a CIH h for $\mathsf{BAD}^{\star}$ then there exists an l-wise independent CIH h'' for BAD. Then, we show that the existence of h follows from Theorem 7.4.

Step (1): $\exists$ CIH h for $\mathsf{BAD}^{\star}$ $\Rightarrow$ $\exists$ l-wise independent CIH h'' for BAD. Assume that h is a CIH for $\mathsf{BAD}^{\star}$. The CIH h'' is defined as follows:

- $\mathsf{h}''.\mathsf{Gen}(1^{\kappa})$: samples $K \leftarrow \mathsf{h}.\mathsf{Gen}(1^{\kappa})$ and $K' \leftarrow \mathsf{h}'$ and outputs $K'' := (K, K')$.
- $\mathsf{h}''.\mathsf{Eval}((K,K'),x)$: outputs $\mathsf{h}.\mathsf{Eval}(K,x) \oplus K'(x)$.

Notice first that since $K' \leftarrow \mathsf{h}'.\mathsf{Gen}(1^{\kappa})$ and h' is l-wise independent, so is h''.

We now show that h'' is CI for $\widehat{\mathsf{BAD}}$. As noted in the proof of Theorem 7.4, this implies that h'' is also CIH for BAD. Let $\mathcal{A}'' = \{\mathcal{A}_{\kappa}\}_{\kappa}$ be a (non uniform) polynomial time adversary that breaks CI of h'' for $\widehat{\mathsf{BAD}}$. That is, there exist a polynomial $p(\kappa)$, and infinitely many κ, τ, such that $\mathcal{A}''_{\kappa}$ breaks CI of h'' for $\widehat{\mathsf{BAD}}_{\kappa,\tau}$ with probability $\geq 1/p(\kappa)$. We derive a contradiction to the CI of h for $\mathsf{BAD}^{\star}$, by using $\mathcal{A}''$ to construct a (non uniform) polynomial-time adversary $\mathcal{A} = \{\mathcal{A}_{\kappa}\}_{\kappa}$ that, for every such κ, τ, breaks the CI of h for $\mathsf{BAD}^{\star}_{\kappa,\tau,K'_{\kappa}}$ with the same probability $1/p(\kappa)$, where K'_{κ} to be the value that maximizes

$$\Pr[(x, \mathsf{h}.\mathsf{Eval}(K,x)) \in \mathsf{BAD}^{\star}_{\kappa,\tau,K'} \ : \ K \leftarrow \mathsf{h}.\mathsf{Gen}(1^{\kappa}),\ K' \leftarrow \mathsf{h}',\ x \leftarrow \mathcal{A}''_{\kappa}(K,K')].$$

$\mathcal{A}_{\kappa}$ has K'_{κ} hard-wired. On input K it runs $\mathcal{A}''_{\kappa}$ on input (K, K'_{κ}) and outputs whatever $\mathcal{A}''_{\kappa}$ outputs. The fact that $\mathcal{A}$ is polynomial time follows directly from its description. We now analyze its success

probability:

$$\begin{aligned}
&\Pr[(x,\mathsf{h.Eval}(K,x)) \in \mathsf{BAD}^{\star}_{\kappa,\tau,K'_\kappa} \ : \ K \leftarrow \mathsf{h.Gen}(1^\kappa),\ x \leftarrow \mathcal{A}_\kappa(K)] \\
&\quad =^{(1)} \Pr[(x,\mathsf{h.Eval}(K,x)) \in \mathsf{BAD}^{\star}_{\kappa,\tau,K'_\kappa} \ : \ K \leftarrow \mathsf{h.Gen}(1^\kappa), x \leftarrow \mathcal{A}''_\kappa(K,K'_\kappa)] \\
&\quad \geq^{(2)} \Pr[(x,\mathsf{h.Eval}(K,x)) \in \mathsf{BAD}^{\star}_{\kappa,\tau,K'} \ : \ K \leftarrow \mathsf{h.Gen}(1^\kappa),\ K' \leftarrow \mathsf{h}',\ x \leftarrow \mathcal{A}''_\kappa(K,K')] \\
&\quad =^{(3)} \Pr[(x,\mathsf{h.Eval}(K,x) \oplus K'(x)) \in \widehat{\mathsf{BAD}}_{\kappa,\tau} \ : \ K \leftarrow \mathsf{h.Gen}(1^\kappa),\ K' \leftarrow \mathsf{h}',\ x \leftarrow \mathcal{A}''_\kappa(K,K')] \\
&=^{(4)} \Pr[(x,\mathsf{h}''.\mathsf{Eval}(K'',x)) \in \widehat{\mathsf{BAD}}_{\kappa,\tau} \ : \ K'' := (K,K') \leftarrow \mathsf{h}''.\mathsf{Gen}(1^\kappa),\ x \leftarrow \mathcal{A}''_\kappa(K,K')] \geq 1/p(\kappa)
\end{aligned}$$

where the first equality follows from the definition of $\mathcal{A}$, the second inequality follows from the choice of K'_κ that maximizes the probability on the RHS, the third equality follows from the definition of $\mathsf{BAD}^\star$, the forth equality follows from the definition of h'', and the last inequality follows from the assumption on the success probability of $\mathcal{A}''$.

Step (2): $\exists$ CIH h for $\mathsf{BAD}^\star$. The proof follows similarly to the proof of Theorem 7.4; we only sketch the differences.

As in the proof of Theorem 7.4, fix some $\kappa, \mathsf{crs}, \tau$, but now we also fix some $K' \in \mathsf{Supp}(\mathsf{h}')$. This fixes a relation $\mathsf{BAD}^\star_{\kappa,\tau,K'}$.

To see that $\mathsf{BAD}^\star_{\kappa,\tau,K'}$ is an α-approximate product relation for $\alpha = (t_\kappa - 1)/lt_\kappa$ (the same approximation parameter as in the proof of Theorem 7.4), fix some $i \in \{0,1,\ldots,l\}$, x, $\mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i)$, and c^{i+1}. For every $k \in [t_\kappa]$, we define the set $R^\star_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$ as a shifted version of the set $R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$ from the proof of Theorem 7.4, as follows: we interpret $K'(x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1})$ as the concatenation of t_κ strings, and let $K'(x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1})|_k$ denote the k'th string in this concatenation. Then:

$$R^\star_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}} := \{r_k \ : \ r_k \oplus K'(x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1})|_k \in R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\}.$$

We then have:

$$\begin{aligned}
r \in R^\star_{i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}} &\Leftrightarrow^{(1)} r \oplus K'(x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}) \in R_{i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}} \\
&\Leftrightarrow^{(2)} \left|\{k \in [t_\kappa] \ : \ r_k \oplus K'(x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1})|_k \in R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\right| \geq \alpha t_\kappa \\
&\Leftrightarrow^{(3)} \left|\{k \in [t_\kappa] \ : \ r_k \in R^\star_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\}\right| \geq \alpha t_\kappa
\end{aligned}$$

where (1) follows from the definition of $\mathsf{BAD}^\star$ and the definition of $R_{i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$ in the proof of Theorem 7.4, (2) follows from the proof of Theorem 7.4, and (3) follows from the definition of the sets $R^\star_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$.

To see why $\mathsf{BAD}^\star_{\kappa,\tau,K'}$ satisfies sparsity, notice that since $i, x, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}$ and K' are fixed, so is $K'(i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1})$. Therefore, for every $k \in [t_\kappa]$, $R^\star_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$ is obtained by applying a shift on $R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}$. Consequently, $\left|R^\star_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\right| = \left|R_{k,i,x,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1}}\right|$, so sparsity follows directly from the proof of Theorem 7.4.

To see why $\mathsf{BAD}^\star_{\kappa,\tau,K'}$ is efficient product verifiable, we define a circuit $C^\star_{\kappa,\tau,K'}$ that has κ, τ, K' hard-wired into it. On input $i, x, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}, r_k^{i+1}, k$ it first computes $r_k^{\star,i+1} := r_k^{i+1} \oplus K'(x,i,\mathcal{T}_{\mathsf{CZK}},\mathsf{c}^{i+1})|_k$, then runs the circuit $C_{\kappa,\tau}$ from the proof of Theorem 7.4 on $i, x, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}, r_k^{\star,i+1}, k$. This circuit still has polynomial size because evaluating K' can be done efficiently.

Finally, we set the parameters exactly as in the proof of Theorem 7.4.

7.2 NIZK Construction

We now compile the CZK arguments from Figure 1 into a NIZK argument using a CI hash function for the bad relation of Definition 7.1. The NIZK is given in Figure 2. We note that the compilation to NIZK is specific to the CZK arguments of Figure 1, and the proofs of soundness and ZK are not black-box in the structure of the CZK argument. This is inherent to our construction, since the CIH depends (through the bad relation) on properties of the IOP underlying the CZK argument. For the proofs to be self-contained, we show the NIZK with the rolled-out CZK argument in Figure 3, with the steps of the CZK argument unrolled in blue: here, a semi-HVZK-IOP is compiled into a NIZK using an EE commitment and a CI hash function.

NIZK from CZK Arguments and CI Hash Functions

Let $\mathcal{L}$ be an NP language with a corresponding NP relation $\mathcal{R} = \mathcal{R}(x, w)$. The NIZK for $\mathcal{L}$ uses the l-round CZK arguments with setup $\mathsf{CZK} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{CZK}}, \mathcal{V}_{\mathsf{CZK}})$ from Figure 1, and a CI Hash function $\mathsf{Hash} = (\mathsf{Gen}, \mathsf{Eval})$ for the relation BAD of Definition 7.1 that is also l-wise independent. The prover $\mathcal{P}$ has input x with corresponding witness w, and the verifier $\mathcal{V}$ that has input x. Both parties also have as input a CRS crs generated by the NIZK Setup algorithm. The Setup, prover and verifier algorithms of the NIZK argument system are as follows.

1. **Setup:** On input 1^κ, the setup algorithm executes the CIH generation algorithm $\mathsf{Hash.Gen}$ to generate a hash key K, $K \leftarrow \mathsf{Hash.Gen}(1^\kappa)$, and runs the CZK setup $\mathsf{crs}' \leftarrow \mathsf{CZK.Setup}(1^\kappa)$. It outputs $\mathsf{crs} := (K, \mathsf{crs}')$.
2. **Prover Algorithm:** The prover $\mathcal{P}$ on input (crs, x, w) does the following:
   - For every $1 \le i \le l$:
     (a) Emulates $\mathcal{P}_{\mathsf{CZK}}$ on (crs', x, w) to generate the commitment c^i to the IOP message Π^i.[a]
     (b) Computes $r^i := \mathsf{Hash.Eval}(K, (x, i-1, \mathsf{c}^1, r^1, \ldots \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i))$.
   - $\mathcal{P}$ emulates $\mathcal{P}_{\mathsf{CZK}}$ to determine a set of queries $\mathcal{Q}$, determined by $r := (r^1, \ldots, r^l)$, and responses $\{(\sigma^i_j, \Pi^i(j)) \ : \ q := (i,j) \in \mathcal{Q}\}$.
     $\mathcal{P}$ sends the proof $\pi := ((\mathsf{c}^i)_{i\in[l]}, \{(\sigma^i_j, \Pi^i(j)) \ : \ q := (i,j) \in \mathcal{Q}\})$ to $\mathcal{V}$.
3. **Verifier Algorithm:** On input $\mathsf{crs} = (K, \mathsf{crs}'), x$ and proof π, $\mathcal{V}$ computes $r = (r^1, \ldots, r^l)$ where $r^i := \mathsf{Hash.Eval}(K, (x, i-1, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i)))$ for every $i \in [l]$, and determines the query set $\mathcal{Q}$. She then parses π as $\pi := ((\mathsf{c}^i)_{i\in[l]}, \{(\sigma^i_j, \Pi^i(j)) \ : \ q := (i,j) \in \mathcal{Q}\})$. $\mathcal{V}$ invokes $\mathcal{V}_{\mathsf{CZK}}$ on $(\mathsf{crs}', x, r, (\mathsf{c}^i)_{i\in[l]}, \{(\sigma^i_j, \Pi^i(j)) \ : \ q := (i,j) \in \mathcal{Q}\})$ with randomness $r = (r^1, \ldots, r^l)$, and outputs whatever $\mathcal{V}_{\mathsf{CZK}}$ outputs.

[a] In case the IOP takes as input a statistical security parameter λ, we set $\lambda := \kappa$.

Figure 2: Compiler from the CZK Argument of Figure 1 to NIZK

The NIZKs of Figures 2 and 3 satisfy the following properties (we state the theorem for the rolled-out NIZK description in Figure 3):

Theorem 7.6. *Let $(\mathcal{P}_{\mathsf{CZK}}, \mathcal{V}_{\mathsf{CZK}})$ be the l-round CZK of Figure 1, where $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$ is the underlying IOP obtained as a t_κ-fold parallel-repetition of an IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ with the following properties:*

- *has (statistical) semi-HVZK,*
- *satisfies doom-checkability,*[25]

[25] Notice that in this case $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$ also has doom checkability with respect to the threshold doomed set, see Remark 7.3.

NIZK from Semi-HVZK-IOPs, EEs, and CI Hash Functions

Let $\mathcal{L}$ be an NP language with a corresponding NP relation $\mathcal{R} = \mathcal{R}(x, w)$. The NIZK for $\mathcal{L}$ uses the l-round CZK argument with setup $\mathsf{CZK} = (\mathsf{Setup}, \mathcal{P}_{\mathsf{CZK}}, \mathcal{V}_{\mathsf{CZK}})$ from Figure 1 — which uses an EE commitment scheme $\mathsf{EE} = (\mathsf{EE.setup}, \mathsf{EE.TDsetup}, \mathsf{EE.Com}, \mathsf{EE.Decom}, \mathsf{EE.Ver}, \mathsf{EE.Ext})$, an l-round (statistical) semi-HVZK-IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ for $\mathcal{L}$ with proof length ℓ, and a CI Hash function $\mathsf{Hash} = (\mathsf{Gen}, \mathsf{Eval})$ for the relation BAD of Definition 7.1 that is also l-wise independent. The prover $\mathcal{P}$ has input x with corresponding witness w, and the verifier $\mathcal{V}$ that has input x. Both parties also have as input a CRS crs generated by the NIZK Setup algorithm. The Setup, prover and verifier algorithms of the NIZK argument system are as follows.

1. **Setup:** On input 1^{κ}, the setup algorithm executes the CIH generation algorithm $\mathsf{Hash.Gen}$ to generate a hash key K, $K \leftarrow \mathsf{Hash.Gen}(1^{\kappa})$. Then, it generates $l \cdot \ell$ CRSs crs_j^i for the EE commitment scheme by executing, for every $1 \le i \le l$ and every $1 \le j \le \ell$, $(\mathsf{crs}_j^i, \tau_j^i) \leftarrow \mathsf{EE.setup}(1^{\kappa})$. Output crs where $\mathsf{crs} := (K, \mathsf{crs}' := (\mathsf{crs}_j^i)_{i \in [l], j \in [\ell]})$.
2. **Prover Algorithm:** The prover $\mathcal{P}$ on input (crs, x, w) does the following:
   - For every $1 \le i \le l$:
     (a) Emulate $\mathcal{P}^{\otimes}$ to generate the message $\Pi^i = (\Pi^i(1), \ldots, \Pi^i(\ell))$.[a] For every $1 \le j \le \ell$, generate a commitment $(\mathsf{c}_j^i, \sigma_j^i) \leftarrow \mathsf{EE.Com}(\mathsf{crs}_j^i, \Pi^i(j))$ to $\Pi^i(j)$, and let $\mathsf{c}^i := (\mathsf{c}_1^i, \ldots, \mathsf{c}_\ell^i)$.
     (b) Computes $r^i := \mathsf{Hash.Eval}(K, (x, i-1, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i))$.
   - Determine the queries $\mathcal{Q}$ which $\mathcal{V}^{\otimes}$, with randomness $r := (r^1, \ldots, r^l)$, makes to $\Pi^1, \ldots, \Pi^l$. Compute $\{(\sigma_j^i, \Pi^i(j)) : q := (i,j) \in \mathcal{Q}\}$, where $q = (i,j)$ denotes a query to the jth bit of Π^i.
     $\mathcal{P}$ sends the proof $\pi := ((\mathsf{c}^i)_{i \in [l]}, \{(\sigma_j^i, \Pi_j^i) : q := (i,j) \in \mathcal{Q}\})$ to $\mathcal{V}$.
3. **Verifier Algorithm:** On input $\mathsf{crs} = (K, \mathsf{crs}'), x$ and proof π, $\mathcal{V}$ parses π as $\pi := ((\mathsf{c}^i)_{i \in [l]}, \{(\sigma_j^i, \Pi^i(j)) : q := (i,j) \in \mathcal{Q}\})$. Then, she computes $r = (r^1, \ldots, r^l)$ where $r^i := \mathsf{Hash.Eval}(K, (x, i-1, , \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i)))$ and determines the query set $\mathcal{Q}$. For every $(i,j) \in \mathcal{Q}$, $\mathcal{V}$ checks that $\mathsf{EE.Ver}(\mathsf{crs}_j^i, \mathsf{c}_j^i, \sigma_j^i, \Pi^i(j))$ outputs 1, and otherwise rejects. Then, she emulates $\mathcal{V}^{\otimes}$ with randomness $r = (r^1, \ldots, r^l)$, and $\{\Pi^i(j) : (i,j) \in \mathcal{Q}\}$ as the answers to $\mathcal{V}^{\otimes}$'s oracle queries, and outputs whatever $\mathcal{V}^{\otimes}$ outputs.

[a] In case the IOP takes as input a statistical security parameter λ, we set $\lambda := \kappa$.

Figure 3: A Rolled-Out NIZK Compiler from Figure 2

- *has completeness error ϵ_{C}, RBR soundness error ϵ_{RBR}, and knowledge extraction with error ϵ_{E}.*

Let EE *denote the underlying EE commitment scheme with extraction error ϵ_{EE}. Let* Hash *be an l-wise independent CIH for the* BAD *relation of the IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ (cf. Definition 7.1) such that any adversary in the CI game has advantage ϵ_{H}*

Then, the construction given in Figure 3 is a NIZK argument, with completeness error $t_{\kappa} \cdot \epsilon_{\mathsf{C}}$, soundness error $l \cdot \ell \cdot \epsilon_{\mathsf{EE}} + \epsilon_{\mathsf{H}}$, where ℓ is the length of messages sent by $\mathcal{P}^{\otimes}$, and $\epsilon_{\mathsf{E}}^{\otimes}$ is the knowledge extraction error of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$. Furthermore, the proof size of the NIZK is $|\pi| = l \cdot \ell \cdot \mathsf{poly}(\kappa)$.

Proof: **Completeness.** The only difference between the execution of the CZK argument and the NIZK is in the choice of randomness $r = (r^1, \ldots, r^l)$, which is no longer sampled randomly, but rather is sampled using the CIH. However, since the CIH is l-wise independent, correct distribution of verifier messages is guaranteed and hence r is still uniformly random in the NIZK execution.

Therefore, the completeness error of the NIZK is identical to the completeness error of the CZK argument. By Theorem 6.1, the latter is exactly the completeness error of the parallel repetition IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$, which by Theorem 5.4 is $t_\kappa \cdot \epsilon_{\mathsf{C}}$.

Soundness. Let $\mathcal{A}$ be a non-uniform PPT adversary that achieves advantage $\epsilon(\kappa)$ in the soundness game for the NIZK of Figure 3. We use $\mathcal{A}$ to construct a non-uniform PPT adversary $\mathcal{A}'$ that violates correlation-intractability of Hash for the relation BAD of Definition 7.1.
$\mathcal{A}'$, on input a CIH key K, operates as follow:

1. **Generate the CRS for $\mathcal{A}$:** For every $1 \leq i \leq l$ and every $1 \leq j \leq \ell$, run $(\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa)$. Set $\tau = (\tau^i_j)_{i\in[l],j\in[\ell]}$ and $\mathsf{crs} := (\mathsf{crs}^i_j)_{i\in[l],j\in[\ell]}$.
2. **Emulate $\mathcal{A}$:** Invoke $\mathcal{A}$ on input (K, crs) to receive (x, π).
3. **Output:**
   - **Determining NIZK transcript.** Parse π as $\pi := \left(\{\mathsf{c}^i\}_{i\in[l]}, \left\{(\sigma^i_j, \Pi^i_j) \ : \ q := (i,j) \in \mathcal{Q}\right\}\right)$ for some set $\mathcal{Q}$, and compute $r = (r^1, \ldots, r^l)$ where $r^i := \mathsf{Hash.Eval}\left(K, (x, i-1, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i)\right)$ for every $1 \leq i \leq l$.
   We can assume that π has this structure, and that $\mathcal{Q}$ is the query set for $\mathcal{V}^{\otimes}$ determined by r, otherwise $\mathcal{V}$ would reject.
   Let $\mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^i, r^i\}_{i\in[l]}$.
   - **Extracting IOP transcript.** $\mathcal{A}'$ invokes the EE extractor to obtain $\{\tilde{\Pi}^i(s) = \mathsf{EE.Ext}(\tau^i_s, \mathsf{c}^i_s)\}_{i\in[l],s\in[\ell]}$, where $\mathsf{c}^i = (\mathsf{c}^i_1, \ldots, \mathsf{c}^i_\ell)$ for every $i \in [l]$, and let $\tilde{\Pi}^i = \tilde{\Pi}^i(1)|\ldots|\tilde{\Pi}^i(\ell)$. Let $\tilde{\mathcal{T}} = \{(\tilde{\Pi}^i, r^i)\}_{i\in[l]}$. *Intuitively, $\tilde{\mathcal{T}}$ is a purported transcript of the parallel-repetition IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$.*
   - **Finding the first non-doomed transcript.** Let $C^{\otimes}$ denote the doom-checkable circuit for the threshold doomed set $\mathcal{D}^{\otimes}$ of the parallel-repetition IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$.[26] For every $1 \leq i \leq l$, let $\tilde{\mathcal{T}} \restriction_i$ denote the round-i partial transcript of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ induced by $\tilde{\mathcal{T}}$. Similarly, let $\mathcal{T}_{\mathsf{CZK}} \restriction_i$ denote the round-i partial transcript of the NIZK $(\mathcal{P}, \mathcal{V})$ induced by $\mathcal{T}_{\mathsf{CZK}}$. If there exists an $0 \leq i \leq l-1$, such that $C^{\otimes}(x, \tilde{\mathcal{T}} \restriction_{i+1}) = 0$, then output $(x, i, \mathcal{T}_{\mathsf{CZK}} \restriction_i, \mathsf{c}^{i+1})$ for the minimal such i. If no such i exists, output $(x, 0, \emptyset, \mathsf{c}^1)$.

We now argue that $\mathcal{A}'$ succeeds in breaking CI with probability $\epsilon - l\cdot\ell\cdot\epsilon_{\mathsf{EE}}$. Therefore, $\epsilon \leq l\cdot\ell\cdot\epsilon_{\mathsf{EE}} + \epsilon_{\mathsf{H}}$ (otherwise $\mathcal{A}'$ violates the ϵ_{H}-CI of Hash.) Assume that $\mathcal{A}$ succeeded in breaking NIZK-soundness. The proof proceeds in 3 steps. First, we show that if $\mathcal{A}$ breaks NIZK soundness then $\Pi^i(j) = \tilde{\Pi}^i(j)$ for all $(i,j) \in \mathcal{Q}$ except with probability $l \cdot \ell \cdot \epsilon_{\mathsf{EE}}$, an event which we denote by G. Second, we show that conditioned on G, the full transcript $\tilde{\mathcal{T}}$ is not doomed. Finally, we show by a case analysis that $\mathcal{A}'$ breaks CI with the claimed probability.

Step (1): Bounding the probability of breaking EE extraction. We claim that except with probability $l \cdot \ell \cdot \epsilon_{\mathsf{EE}}$,$\Pi^i(j) = \tilde{\Pi}^i(j)$ for all $i \in [l]$ and all $j \in \mathcal{Q}_i$, where $\mathcal{Q}_i := \{j : (i,j) \in \mathcal{Q}\}$. If not, then there exist $i^* \in [l]$ and $j^* \in \mathcal{Q}_{i^*}$ such that $\Pi^{i^*}(j^*) \neq \tilde{\Pi}^{i^*}(j^*)$. However, in this case we have $\mathsf{EE.Ver}(\mathsf{crs}^i_j, \mathsf{c}^i_j, \sigma^i_j, \Pi^i(j)) = 1$ (since the NIZK verifier accepts, and she checks all the commitment openings), but $\mathsf{EE.Ext}(\tau^i_j, \mathsf{c}^i_j) = \tilde{\Pi}^i(j) \neq \Pi^i(j)$. By extractability of EE, $\Pr\left[\mathsf{EE.Ver}(\mathsf{crs}, \mathsf{c}, \sigma, b) = 1 \wedge \mathsf{EE.Ext}(\tau_1, \mathsf{c}) \neq b\right] \leq \epsilon_{\mathsf{EE}}$ for all $(\mathsf{crs}, \tau) \in \mathsf{Supp}(\mathsf{EE.setup}(1^\kappa))$ c, σ, b. Therefore, using the union bound, this event happens with probability at most $l \cdot \ell \cdot \epsilon_{\mathsf{EE}}$. (Here, we use

[26] Doom-checkability of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ with respect to $\mathcal{D}^{\otimes}$ follows from the doom-checkability of the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$; see Remark 7.3.

the fact that $|Q|$ is bounded by the total length of prover messages, namely $l \cdot \ell$.) In the following, we condition on the event that $\Pi^i(j) = \tilde{\Pi}^i(j)$ for all $i \in [l]$ and all $j \in \mathcal{Q}_i$. We call this event G (for "good").

Step (2): Full transcript is not doomed. We show that conditioned on G, the full transcript $\tilde{\mathcal{T}}$ is not doomed. Otherwise, by the definition of a doomed state, we have (using Notation 6.3)

$$0 = (\mathcal{V}^{\otimes})^{([\mathcal{Q}_i,\tilde{\Pi}^i(j)])_{j\in\mathcal{Q}_i})]})_{i\in[l]}}(x; r_1, \ldots, r_l) = (\mathcal{V}^{\otimes})^{([\mathcal{Q}_i,(\Pi^i(j))_{j\in\mathcal{Q}_i})])_{i\in[l]}}(x; r_1, \ldots, r_l),$$

where the $\Pi^i(j)$'s are the commitment openings which $\mathcal{A}$ provides to the NIZK verifier in the NIZK execution. Indeed, the right equality holds because the two executions (with Π^i and $\tilde{\Pi}^i$) are with *the same* verifier randomness $r_1, \ldots, r_l$, so the verifier $\mathcal{V}^{\otimes}$ makes the same queries in both executions, and the *oracle answers* are also the same (since we've conditioned on G), so the verifier has the same output. The left equality holds by the definition of RBR soundness (Definition 4.16, bullet 2), because the full transcript is doomed. This contradicts the assumption that $\mathcal{A}$ succeeded in breaking soundness, because the NIZK verifier $\mathcal{V}$ accepts only if the IOP verifier $\mathcal{V}^{\otimes}$ accepts. We summarize this step of the proof by noting that since the full transcript is not doomed, $\mathcal{A}'$'s output is *never* computed by the fall-back case "if no such i exists...".

Step (3): $\mathcal{A}'$ breaks CI (via case analysis). We now consider two possible cases. Assume first that the round-1 partial transcript $\tilde{\mathcal{T}} \restriction_1 = (\tilde{\Pi}^1, r^1)$ is not doomed. By the doom-checkability guarantee of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$, and $\mathcal{A}'$'s construction, in this case $\mathcal{A}'$ outputs $(x, 0, \emptyset, \mathsf{c}^1)$. (We stress that this output results from the fact that $C^{\otimes}(x, \tilde{\mathcal{T}} \restriction_1) = 0$, and not because of the fall-back case.) We show that in this case, $\mathcal{A}'$ succeeded in breaking CI. Parse $\tilde{\Pi}^1$ as $\tilde{\Pi}^1 = \{\tilde{\Pi}^1_1, \ldots, \tilde{\Pi}^1_{t_\kappa}\}$, where the message lengths of the IOPs $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ and $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ are u and $\ell = u \cdot t_\kappa$, respectively, and $\left|\tilde{\Pi}^1_k\right| = u$ for every $k \in [t_\kappa]$. (Intuitively, $\tilde{\Pi}^1_k$ is the round-1 prover message sent in the k'th repetition of the base-IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$.) Since $x \notin \mathcal{L}$ (because $\mathcal{A}$ broke NIZK soundness) then for any purported witness w, we have $(x, w) \notin \mathcal{R}$. In particular, $(x, \mathsf{E}(x, \tilde{\Pi}^1_k)) \notin \mathcal{R}$ for any $k \in [t_\kappa]$, where E is the knowledge extractor of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. Moreover, since $\tilde{\mathcal{T}} \restriction_1$ is not doomed then $(x, \Pi_1 | r_1) \notin \mathcal{D}^{\otimes}$. Therefore, $\mathcal{A}'$'s output is in $\mathsf{BAD}^{=0}_{\kappa,\tau}$ (by definition) and therefore in BAD, and $\mathcal{A}'$ violated CI.

Next, assume the round-1 partial transcript *is* doomed, and recall that we have shown in step (2) above that the full transcript $\tilde{\mathcal{T}}$ (i.e., the round-l transcript) is not doomed. Therefore, by the doom-checkability guarantee of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ and $\mathcal{A}'$'s construction, $\mathcal{A}'$ will output $(x, i, \mathcal{T}_{\mathsf{CZK}} \restriction_i, \mathsf{c}^{i+1})$ for some $i > 1$. Indeed, $\mathcal{A}'$ only outputs the round-1 transcript if it is not doomed (which the case assumption precludes) or if no round-i partial transcript was not doomed (which is not possible since the round-l transcript is not doomed). We show that in this case, $\left((x, i, \mathcal{T}_{\mathsf{CZK}} \restriction_i, \mathsf{c}^{i+1}), r^{i+1}\right) \in \mathsf{BAD}$, so $\mathcal{A}'$ broke CI. For this, we need to prove that: (1) $(x, \tilde{\mathcal{T}} \restriction_i | \tilde{\Pi}^{i+1} | r^{i+1}) \notin \mathcal{D}^{\otimes}$, and (2) $(x, \tilde{\mathcal{T}} \restriction_i) \in \mathcal{D}^{\otimes}$. (1) follows directly from the description of $\mathcal{A}'$ and the doom checkability guarantee, because $i > 1$. (2) follows from the minimality of i, and the fact that any transcript is either doomed or not.

Zero-knowledge. We define a PPT simulator Sim that simulates the joint distribution of the CRS and NIZK proof generated by the honest prover. Sim uses a simulator $\mathsf{Sim}_{\mathsf{in}}$ for the (statistical) semi-HVZK property of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$.

Sim, on input $(1^\kappa, x)$, operates as follows:

1. **Generating the CRS.** Sim Generates $l \cdot \ell$ trapdoor-mode trapdoor CRSs for the EE, together with equivocal commitments, by running for every $1 \le i \le l$ and every $1 \le j \le \ell$: $(\widetilde{\mathsf{tdcrs}}^i_j, \tilde{\mathsf{c}}^i_j, \tilde{\sigma}^{i,0}_j, \tilde{\sigma}^{i,1}_j) \leftarrow \mathsf{EE.TDsetup}(1^\kappa)$. It honestly generates a CIH key by running $K \leftarrow \mathsf{Hash.Gen}(1^\kappa)$. Then, Sim sets $\widetilde{\mathsf{tdcrs}} = (K, (\widetilde{\mathsf{tdcrs}}^i_j)_{i\in[l],j\in[\ell]})$.

2. **Emulating Prover messages and Challenges:** For every $1 \leq i \leq \ell$, Sim sets $\tilde{\mathsf{c}}^i := (\tilde{\mathsf{c}}^i_1, \ldots, \tilde{\mathsf{c}}^i_\ell)$ as the round-i CZK argument prover message. Then, Sim honestly generates queries as follows. For every $1 \leq i \leq l$, $\tilde{r}_i = \mathsf{Hash.Eval}(K, (x, i-1, \tilde{\mathsf{c}}^1, \tilde{r}^1, \ldots, \tilde{\mathsf{c}}^{i-1}, \tilde{r}^{i-1}, \tilde{\mathsf{c}}^i))$.

3. **Emulating responses:**

   (a) **Simulating IOP answers.** Let $\tilde{r} := (\tilde{r}_1, \ldots, \tilde{r}_\ell)$. Sim emulates $\mathsf{Sim}_{\mathsf{in}}(1^\kappa, \tilde{r}, x)$ to simulate $\mathcal{V}^\otimes$'s view when using randomness $\tilde{r}$. Let $(a_q)_{q \in \tilde{\mathcal{Q}}}$ denote the simulated answers to $\mathcal{V}^\otimes$'s queries to the proof oracles, where $\tilde{\mathcal{Q}}$ denotes the oracle queries which $\mathcal{V}^\otimes$ makes when using randomness $\tilde{r}$.

   (b) **Equivocating commitment openings.** Sim sets $\mathsf{op} := \left\{(\tilde{\sigma}^{i,a_q}_j, a_q) \ : \ q = (i,j) \in \tilde{\mathcal{Q}}\right\}$ as the commitment openings.

4. **Output.** Sim outputs $(\widetilde{\mathsf{tdcrs}}, (\tilde{\mathsf{c}}^i)_{i \in [l]}, \mathsf{op})$.

We now show that for every PPT distinguisher, and every $(x, w) \in \mathcal{R}$, the real and simulated views are computationally close. We prove this through a sequence of hybrids.

$\mathcal{H}_0$**:** This is the simulated view which Sim outputs in Step 4.

$\mathcal{H}_1$**:** In $\mathcal{H}_1$, we replace the simulated IOP bits generated in Step 3a with the actual IOP bits generated by the IOP prover $\mathcal{P}^\otimes$ for input (x, w), when the verifier challenges are $\tilde{r}$.

Claim: $\mathcal{H}_0 \approx \mathcal{H}_1$.

This follows from the (statistical) semi-HVZK property of the IOP system $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$. Let $\mathcal{D}$ be a PPT distinguisher between $\mathcal{H}_0$ and $\mathcal{H}_1$. We use $\mathcal{D}$ to define a distinguisher $\mathcal{D}^\otimes$ obtaining the same distinguishing advantage between the real and simulated view of $\mathcal{V}^\otimes$. $\mathcal{D}^\otimes$ works as follows. The input of $\mathcal{D}^\otimes$ is the real or simulated view of $\mathcal{V}^\otimes$ with randomness $\tilde{r}$ for input x and security parameter κ. $\mathcal{D}^\otimes$ executes Step 1 of the simulation (with security parameter κ) to obtain trapdoor CRSs $(\widetilde{\mathsf{tdcrs}}^i_j)_{i \in [l], j \in [\ell]}$, together with equivocal commitments $(\tilde{\mathsf{c}}^i_j)_{i \in [l], j \in [\ell]}$ and openings $(\tilde{\sigma}^{i,0}_j, \tilde{\sigma}^{i,1}_j)_{i \in [l], j \in [\ell]}$, where $\tilde{\sigma}^{i,b}_j$ can be used to open $\tilde{\mathsf{c}}^i_j$ to bit b. Let $\widetilde{\mathsf{tdcrs}} := (\widetilde{\mathsf{tdcrs}}^i_j)_{i \in [l], j \in [\ell]}$ and $\tilde{\mathsf{c}} := (\tilde{\mathsf{c}}^i_j)_{i \in [l], j \in [\ell]}$. Let $\mathsf{op} := (a_q, \tilde{\sigma}^{i,a_q}_j)_{q=(i,j) \in \tilde{\mathcal{Q}}}$, where $\tilde{\mathcal{Q}}$ are the queries that $\mathcal{V}^\otimes$ makes on input $1^\kappa, x$ when using randomness $\tilde{r}$, and $\{a_q \ : \ q \in \tilde{\mathcal{Q}}\}$ are the oracle answers reported in the (real or simulated) view of $\mathcal{V}^\otimes$. $\mathcal{D}^\otimes$ then emulates $\mathcal{D}$ on input $(\widetilde{\mathsf{tdcrs}}, \tilde{\mathsf{c}}, \mathsf{op})$ and outputs whatever $\mathcal{D}$ outputs. Notice that if $\mathcal{D}^\otimes$ is executed with the simulated view then it emulates $\mathcal{D}$ on $\mathcal{H}_0$, otherwise it emulates $\mathcal{D}$ on $\mathcal{H}_1$. Thus, $\mathcal{D}^\otimes$'s distinguishing advantage is identical to that of $\mathcal{D}$, and so by the (statistical) semi-HVZK property of the IOP, this advantage is $\mathsf{negl}(\kappa)$ (if the IOP has *perfect* semi-HVZK, then the analysis shows the distinguishing advantage here is 0).

$\mathcal{H}_2$**:** In $\mathcal{H}_2$, we modify the setup and the opening of the commitments, as follows:

- We replace the trapdoor-mode CRS with a standard-mode CRS for the EE scheme in Step 1 of the simulation: for every $1 \leq i \leq l$ and every $1 \leq j \leq \ell$ generate $(\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa)$ and set $\mathsf{crs} := (\mathsf{crs}^i_j)_{i \in [l], j \in [\ell]}$.
- We replace the equivocal commitments $\tilde{\mathsf{c}}^i_j$ with real commitments c^i_j to the real IOP bits. For every round i we emulate the real-world prover (with the witness w) to obtain the round-i proof oracle Π^i, and then generate the round-i commitments $\mathsf{c}^i_1, \ldots, \mathsf{c}^i_\ell$ by computing, for every $1 \leq j \leq \ell$, $(\mathsf{c}^i_j, \sigma^i_j) \leftarrow \mathsf{EE.Com}(\mathsf{crs}^i_j, \Pi^i(j))$. We then set $\mathsf{c}^i := (\mathsf{c}^i_1, \ldots, \mathsf{c}^i_\ell)$,

and compute the round-i challenge $r_i = \mathsf{Hash.Eval}(K, (x, i-1, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i))$. Let $r := (r^1, \ldots, r^l)$ denote the randomness obtained throughout this process.

- We replace the equivocated opening of the equivocal commitments $\tilde{c}^i_j$ with honestly-generated openings for the real commitments c^i_j, i.e., for every $q = (i,j) \in \mathcal{Q}$ (where $\mathcal{Q}$ denotes the queries which $\mathcal{V}^{\otimes}$ makes to her oracles when using randomness r), we provide σ^i_j as the opening of c^i_j to Π^i_j.

Claim: $\mathcal{H}_1 \approx \mathcal{H}_2$.

Assume towards contradiction that there exists a (possibly non-uniform) PPT distinguisher $\mathcal{D}$ obtaining a non-negligible distinguishing advantage $\epsilon = \epsilon(\kappa)$ between $\mathcal{H}_1$ and $\mathcal{H}_2$. We define a sequence of hybrids $\mathcal{H}^0, \ldots, \mathcal{H}^{l \cdot \ell}$ where $\mathcal{H}^0 = \mathcal{H}_1$, and for every $1 \leq s \leq l \cdot \ell, s = (i,j)$, $\mathcal{H}^s$ is obtained from $\mathcal{H}^{s-1}$ by replacing the (i,j)'th EE-CRS from a trapdoor-mode CRS $\widetilde{\mathsf{tdcrs}}^i_j$ to a real-mode CRS crs^i_j, replacing the (i,j)'th commitment from an equivocal commitment $\tilde{\mathsf{c}}^i_j$ to a real commitment c^i_j (to $\Pi^i(j)$), and if $(i,j) \in \mathcal{Q}$, also replacing the opening of c^i_j from $\tilde{\sigma}^{i,\Pi^i(j)}_j$ to σ^i_j. Then $\mathcal{H}^{l \cdot \ell} = \mathcal{H}_2$, and by the negation assumption there exists an $s^* = (i^*, j^*) \in [l \cdot \ell]$ such that $\mathcal{D}$ obtains non-negligible advantage $\epsilon' := \epsilon/(l \cdot \ell)$ in distinguishing between $\mathcal{H}^{s^*-1}$ and $\mathcal{H}^{s^*}$. (We note that ϵ' is non-negligible, since both l and ℓ are $\mathsf{poly}(\kappa)$.) We use this to contradict mode indistinguishability of EE.

Let $\mathcal{Q}_{s^*-1}, \mathcal{Q}_{s^*}$ denote $\mathcal{V}^{\otimes}$'s queries in $\mathcal{H}^{s^*-1}, \mathcal{H}^{s^*}$, respectively. (These are random variables whose distribution is induced by the distribution over real and trapdoor-mode CRSs and commitments, respectively, as well as verifier randomness.) We use $\mathcal{D}$ to contradict the mode indistinguishability of EE. Towards that end, consider the following non-uniform PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that has $(i^*, j^*), x, w$ hard wired into it, and operates as follows.

1. $\mathcal{A}_1$ is given $1^\kappa, \mathsf{crs}^{i^*}_{j^*}$ as input, where $\mathsf{crs}^{i^*}_{j^*}$ was either honestly generated using $\mathsf{EE.setup}$, or is a trapdoor-mode CRS generated using $\mathsf{EE.TDsetup}$. It operates as follows.
   (a) For every $s := (i,j)$ such that $s < s^*$, $\mathcal{A}_1$ honestly generates the (i,j)'th CRS crs^i_j together with an extraction trapdoor τ^i_j by executing $(\mathsf{crs}^i_j, \tau^i_j) \leftarrow \mathsf{EE.setup}(1^\kappa)$.
   For every $s := (i,j)$ such that $s > s^*$, $\mathcal{A}_1$ samples the (i,j)'th CRS as a trapdoor-mode CRS tdcrs^i_j together with equivocal commitment $\tilde{\mathsf{c}}^i_j$ and openings $\tilde{\sigma}^{i,0}_j, \tilde{\sigma}^{i,1}_j$ by executing $(\mathsf{tdcrs}^i_j, \tilde{\mathsf{c}}^i_j, \tilde{\sigma}^{i,0}_j, \tilde{\sigma}^{i,1}_j) \leftarrow \mathsf{EE.TDsetup}(1^\kappa)$.
   Let $\mathsf{crs} := ((\mathsf{crs}^i_j)_{(i,j)=s<s^*}, \mathsf{crs}^{i^*}_{j^*}, (\mathsf{tdcrs}^i_j)_{(i,j)=s>s^*})$.
   (b) $\mathcal{A}_1$ then emulates the execution of an honest $\mathcal{P}_{\mathsf{NIZK}}$ with input crs, x, w. More specifically, $\mathcal{A}_1$ emulates the honest $\mathcal{P}^{\otimes}$, and commits to its messages, where in round $i, i < i^*$, $\mathcal{A}_1$ honestly commits to the honestly-generated IOP bits. It then computes challenge r_i as $r_i = \mathsf{Hash.Eval}(K, (x, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i))$. At the onset of the i^*'th round, $\mathcal{A}_1$ computes $\Pi^{i^*}(j^*)$ (this is possible because $\mathcal{A}_1$ knows w, and $r_1, \ldots, r_{i^*-1}$ have already been determined). It then outputs $\Pi^{i^*}_{j^*}$ as the bit b, and the state st includes $1^\kappa, \mathsf{crs}$, the randomness used to generate the real/trapdoor CRSs for all $s \neq (i^*, j^*)$, the randomness used for $\mathcal{P}^{\otimes}$'s emulation, and the randomness used to generate the commitments up to round i^*.
2. $\mathcal{A}_2$ is given as input the state st, a commitment $\mathsf{c}^{i^*}_{j^*}$ — which was either honestly generated as a commitment to $\Pi^{i^*}(j^*)$, or is an equivocal commitment — and an opening $\sigma^{i^*}_{j^*}$, which was either honestly generated or is an equivocated commitment for the bit $b = \Pi^{i^*}(j^*)$. $\mathcal{A}_2$ operates as follows.

(a) $\mathcal{A}_2$ emulates an execution of the honest $\mathcal{P}^{\otimes}$, with the same random string used in $\mathcal{A}_1$'s emulation (as reported in st). Moreover, $\mathcal{A}_2$ uses the same CRS as in $\mathcal{A}_1$'s emulation, and the same randomness as $\mathcal{A}_1$, to generate the commitments in rounds $1, \ldots, i^* - 1$. At round i^*, for every $j < j^*$, $\mathcal{A}_2$ honestly commits to $\Pi^{i^*}(j)$. Next, it uses $\mathsf{c}^{i^*}_{j^*}$ as the commitment to the j^*'th bit of the i^*'th oracle. Then, for every $j > j^*$ it uses the equivocated commitment $\tilde{\mathsf{c}}^{i^*}_j$ as the commitment to the j'th bit of the i^*'th oracle. In round $i > i^*$, it uses the equivocated commitments generated by $\mathcal{A}_1$. In each round $i \in [l]$, $\mathcal{A}_2$ uses the CIH to derive $\mathcal{V}^{\otimes}$'s randomness in round i, as in the NIZK execution. In particular, at the end of the emulation, $\mathcal{A}_2$ holds the randomness $r := (r^1, \ldots, r^l)$ computed as $r^i = \mathsf{Hash.Eval}(K, (x, i-1, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i))$.

(b) $\mathcal{A}_2$ uses r to determine the set $\mathcal{Q}$ of $\mathcal{V}^{\otimes}$'s oracle queries. It then sets the proof π to include the honestly-generated commitments $(c^i_j)_{s:=(i,j),s<s^*}$, the challenge commitment $c^{i^*}_{j^*}$, and the equivocated commitments $(\tilde{\mathsf{c}}^i_j)_{s:=(i,j),s>s^*}$. The proof π also includes, for every $q = (i,j), q < s^*$ the honestly-generated opening σ^i_j to $\Pi^i(j)$, and for every $q = (i,j), q > s^*$, the appropriate equivocated opening $\tilde{\sigma}^{i,\Pi^i(j)}_j$ and the bit $\Pi^i(j)$, where $\Pi^i(j)$ is the j'th bit of the i'th proof oracle generated by $\mathcal{P}^{\otimes}$ in his emulation. Finally, if $(i^*, j^*) \in \mathcal{Q}$ then π also includes $\sigma^{i^*}_{j^*}$ and $\Pi^{i^*}(j^*)$.

(c) $\mathcal{A}_2$ emulates $\mathcal{D}$ on crs, π and outputs whatever $\mathcal{D}$ outputs.

$\mathcal{A}$ is efficient because the EE and Hash algorithms are efficient, and so is $\mathcal{D}$. We now show analyze the distinguishing advantage of $\mathcal{A}$ by considering 3 possible cases.

CASE (1): $q^* := (i^*, j^*) \in \mathcal{Q}_{s^*-1}$ BUT $q^* \notin \mathcal{Q}_{s^*}$, OR VISE VERSA. We claim that this happens only with negligible probability. Indeed, otherwise we can design an adversary $\mathcal{A}' = (\mathcal{A}_1, \mathcal{A}'_2)$ where $\mathcal{A}'_2$ operates similarly to $\mathcal{A}_2$, except that once it obtains r in Step 2a, it halts and outputs 1 if $(i^*, j^*) \in \mathcal{Q}$, otherwise it outputs 0. If this event happens with non-negligible probability, then $\mathcal{A}'$'s advantage in the mode indistinguishability game is non negligible, contradicting mode indistinguishability. Therefore, except with negligible probability $(j^*, i^*) \in \mathcal{Q}_{s^*} \Leftrightarrow (j^*, i^*) \in \mathcal{Q}_{s^*-1}$.

In the following, we condition on the event that $(j^*, i^*) \in \mathcal{Q}_{s^*} \Leftrightarrow (j^*, i^*) \in \mathcal{Q}_{s^*-1}$. We will show that conditioned on this event, $\mathcal{A}'$ obtains the same distinguishing advantage as $\mathcal{D}$, so overall $\mathcal{A}'$'s distinguishing advantage is $\epsilon' - \mathsf{negl}(\kappa)$ which is non negligible by the negation assumption.

CASE (2): $(i^*, j^*) \in \mathcal{Q}_{s^*-1} \cap \mathcal{Q}_{s^*}$. In this case, $(i^*, j^*) \in \mathcal{Q}$, and indistinguishability again follows directly from the mode indistinguishability property, because if $\mathcal{A}$ is given honestly-generated CRS, commitment and opening, then it emulates $\mathcal{D}$ on $\mathcal{H}^{s^*}$, otherwise it emulates $\mathcal{D}$ on $\mathcal{H}^{s^*-1}$.

CASE (3): $(i^*, j^*) \notin \mathcal{Q}_{s^*-1}$. In this case, $(i^*, j^*) \notin \mathcal{Q}$ and also $(i^*, j^*) \notin \mathcal{Q}_{s^*}$. The analysis here is similar to case (2) above, except that we modify Step 2b such that π does not include the opening $\sigma^{i^*}_{j^*}$ and $\Pi^{i^*}(j^*)$. We again conclude that indistinguishability follows from the mode indistinguishability property.

We conclude the CZK proof by noting that $\mathcal{H}_2$ is the real-world view of $\mathcal{V}_{\mathsf{NIZK}}$.

Proof size. By construction, the NIZK proof consists of $\pi := ((\mathsf{c}^i)_{i\in[l]}, \{(\sigma^i_j, \Pi^i(j)) : q := (i,j) \in \mathcal{Q}\})$. Each commitment c^i is to the ith round IOP message of $\mathcal{P}^{\otimes}$ which is of length ℓ. The length of the commitment per bit is $\mathsf{poly}(\kappa)$, and therefore $|\mathsf{c}^i| = \ell \cdot \mathsf{poly}(\kappa)$. The commitment openings σ per query are $\mathsf{poly}(\kappa)$ long, and there are $|\mathcal{Q}|$ many openings (where $|\mathcal{Q}| = t_\kappa \cdot |\mathcal{Q}|_{\mathsf{IOP}}$). Thus, the proof

size is $|\pi| = l \cdot \ell \cdot \mathsf{poly}(\kappa) + |\mathcal{Q}| \cdot \mathsf{poly}(\kappa)$. The size of the query set is at most the total size of prover messages. Therefore, $|\mathcal{Q}| \leq l \cdot \ell$, and we have that $|\pi| = O(l \cdot \ell) \cdot \mathsf{poly}(\kappa) = l \cdot \ell \cdot \mathsf{poly}(\kappa)$. ■

NIZK AoK. Under the same conditions as in Theorem 7.6, we show that the construction given in Figure 3 is in fact a NIZKAoK:

Theorem 7.7. *Assume the conditions stated in Theorem 7.6 hold. Then the proof system given in Figure 3 is a NIZK AoK with knowledge extraction error* $l \cdot \ell \cdot \epsilon_{\mathsf{EE}} + \epsilon_{\mathsf{H}} + \epsilon_{\mathsf{E}}^{\otimes}$.

Proof (sketch): We describe the trapdoor setup algorithm and prove CRS indistinguishability and extraction.

TdSetup: On input 1^κ, the trapdoor-setup algorithm executes the CIH generation algorithm $\mathsf{Hash.Gen}$ to generate a hash key K, $K \leftarrow \mathsf{Hash.Gen}(1^\kappa)$. Then, it generates $l \cdot \ell$ CRSs crs_j^i for the EE commitment scheme by executing, for every $1 \leq i \leq l$ and every $1 \leq j \leq \ell$, $(\mathsf{crs}_j^i, \tau_j^i) \leftarrow \mathsf{EE.setup}(1^\kappa)$. It outputs $(\mathsf{crs}, \mathsf{td})$, where $\mathsf{crs} := (K, \mathsf{crs}' := (\mathsf{crs}_j^i)_{i\in[l],j\in[\ell]})$, $\mathsf{td} := (\tau_j^i)_{i\in[l],j\in[\ell]}$.

CRS Indistinguishability holds since crs is generated identically in Setup and TdSetup.

Extraction. We now construct an extractor E and then argue its success probability. E on input (td, x, π) works as follows:

- Parse $\mathsf{td} = (\tau_j^i)_{i\in[l],j\in[\ell]}$. Parse π as $\pi = \left(\{\mathsf{c}^i\}_{i\in[l]}, \left\{(\sigma_j^i, \Pi_j^i) \ : \ q := (i,j) \in \mathcal{Q}\right\}\right)$ for some set $\mathcal{Q}$, and compute $r = (r^1, \ldots, r^l)$ where $r^i := \mathsf{Hash.Eval}\left(K, (x, i-1, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i)\right)$ for every $1 \leq i \leq l$. If π does not have this structure, E aborts. (E works only when the verifier accepts, and we can assume the verifier rejects if π does not have this structure).
- E then invokes the EE extractor to obtain $\{\tilde{\Pi}^i(s) = \mathsf{EE.Ext}(\tau_s^i, \mathsf{c}_s^i)\}_{i\in[l],s\in[\ell]}$, where $\mathsf{c}^i = (\mathsf{c}_1^i, \ldots, \mathsf{c}_\ell^i)$ for every $i \in [l]$, and let $\tilde{\Pi}^i = \tilde{\Pi}^i(1)|\ldots|\tilde{\Pi}^i(\ell)$. Let $\tilde{\mathcal{T}} = \{(\tilde{\Pi}^i, r^i)\}_{i\in[l]}$. *Intuitively,* $\tilde{\mathcal{T}}$ *is a purported transcript of the parallel-repetition IOP* $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$.
- E invokes the IOP extractor $\mathsf{E}^\otimes$ to obtain $w^* \leftarrow \mathsf{E}^\otimes(x, \tilde{\Pi}^1)$. If $(x, w^*) \in \mathcal{R}$, E outputs w^*. If not, it outputs $\perp$.

We now show that E succeeds except with probability at most $l \cdot \ell \cdot \epsilon_{\mathsf{EE}} + \epsilon_{\mathsf{H}} + \epsilon_{\mathsf{E}}^{\otimes}$. The analysis is very similar to the soundness analysis in the proof of Theorem 7.6, we write it here in full for better readability. Let $\mathcal{A}$ be a non-uniform PPT adversary that achieves advantage $\epsilon(\kappa)$ in the extraction game for the NIZK of Figure 3.

We use $\mathcal{A}$ to construct a non-uniform PPT adversary $\mathcal{A}'$ that violates correlation-intractability of Hash for the relation BAD of Definition 7.1. ($\mathcal{A}'$ is defined identically to the soundness proof.)
$\mathcal{A}'$, on input a CIH key K, operates as follow:

1. **Generate the CRS for $\mathcal{A}$:** For every $1 \leq i \leq l$ and every $1 \leq j \leq \ell$, run $(\mathsf{crs}_j^i, \tau_j^i) \leftarrow \mathsf{EE.setup}(1^\kappa)$. Set $\mathsf{td} = (\tau_j^i)_{i\in[l],j\in[\ell]}$ and $\mathsf{crs} := (\mathsf{crs}_j^i)_{i\in[l],j\in[\ell]}$.
2. **Emulate $\mathcal{A}$:** Invoke $\mathcal{A}$ on input (K, crs) to receive (x, π).
3. **Output:**
   - **Determining NIZK transcript.** Parse π as $\pi := \left(\{\mathsf{c}^i\}_{i\in[l]}, \left\{(\sigma_j^i, \Pi_j^i) \ : \ q := (i,j) \in \mathcal{Q}\right\}\right)$ for some set $\mathcal{Q}$, and compute $r = (r^1, \ldots, r^l)$ where $r^i := \mathsf{Hash.Eval}\left(K, (x, i-1, \mathsf{c}^1, r^1, \ldots, \mathsf{c}^{i-1}, r^{i-1}, \mathsf{c}^i)\right)$ for every $1 \leq i \leq l$.
   We can assume that π has this structure, and that $\mathcal{Q}$ is the query set for $\mathcal{V}^\otimes$ determined by r, otherwise $\mathcal{V}$ would reject.
   Let $\mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^i, r^i\}_{i\in[l]}$.

- **Extracting IOP transcript.** $\mathcal{A}'$ invokes the EE extractor to obtain $\{\tilde{\Pi}^i(s) = \mathsf{EE.Ext}(\tau_s^i, \mathsf{c}_s^i)\}_{i\in[l],s\in[\ell]}$, where $\mathsf{c}^i = (\mathsf{c}_1^i, \ldots, \mathsf{c}_\ell^i)$ for every $i \in [l]$, and let $\tilde{\Pi}^i = \tilde{\Pi}^i(1)| \ldots |\tilde{\Pi}^i(\ell)$. Let $\tilde{\mathcal{T}} = \{(\tilde{\Pi}^i, r^i)\}_{i\in[l]}$. *Intuitively, $\tilde{\mathcal{T}}$ is a purported transcript of the parallel-repetition IOP $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$.*
- **Finding the first non-doomed transcript.** Let $C^\otimes$ denote the doom-checkable circuit for the threshold doomed set $\mathcal{D}^\otimes$ of the parallel-repetition IOP $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$.[27] For every $1 \le i \le l$, let $\tilde{\mathcal{T}} \restriction_i$ denote the round-i partial transcript of $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$ induced by $\tilde{\mathcal{T}}$. Similarly, let $\mathcal{T}_{\mathsf{CZK}} \restriction_i$ denote the round-i partial transcript of the NIZK $(\mathcal{P}, \mathcal{V})$ induced by $\mathcal{T}_{\mathsf{CZK}}$. If there exists an $0 \le i \le l-1$, such that $C^\otimes(x, \tilde{\mathcal{T}} \restriction_{i+1}) = 0$, then output $(x, i, \mathcal{T}_{\mathsf{CZK}} \restriction_i, \mathsf{c}^{i+1})$ for the minimal such i. If no such i exists, output $(x, 0, \emptyset, \mathsf{c}^1)$.

We now argue that $\mathcal{A}'$ succeeds in breaking CI with probability $\epsilon - l \cdot \ell \cdot \epsilon_{\mathsf{EE}} - \epsilon_{\mathsf{E}}^\otimes$. Therefore, $\epsilon \le l \cdot \ell \cdot \epsilon_{\mathsf{EE}} + \epsilon_{\mathsf{E}}^\otimes + \epsilon_{\mathsf{H}}$ (otherwise $\mathcal{A}'$ violates the ϵ_{H}-CI of Hash.) Assume that $\mathcal{A}$ succeeded in breaking NIZK-extraction. The proof proceeds in 3 steps. First, we show that if $\mathcal{A}$ breaks NIZK extraction then $\Pi^i(j) = \tilde{\Pi}^i(j)$ for all $(i,j) \in \mathcal{Q}$ except with probability $l \cdot \ell \cdot \epsilon_{\mathsf{EE}}$, an event which we denote by G. Second, we show that conditioned on G, the full transcript $\tilde{\mathcal{T}}$ is doomed. Finally, we show by a case analysis that $\mathcal{A}'$ breaks CI with the claimed probability. The first two steps follow identically to the corresponding steps in the soundness analysis, the difference is only in the third step.

Step (1): Bounding the probability of breaking EE extraction. We claim that except with probability $l \cdot \ell \cdot \epsilon_{\mathsf{EE}}$, $\Pi^i(j) = \tilde{\Pi}^i(j)$ for all $i \in [l]$ and all $j \in \mathcal{Q}_i$, where $\mathcal{Q}_i := \{j : (i,j) \in \mathcal{Q}\}$. The argument here is exactly as in the proof of this step in the soundness analysis in the proof of Theorem 7.6. In the following, we condition on the event that $\Pi^i(j) = \tilde{\Pi}^i(j)$ for all $i \in [l]$ and all $j \in \mathcal{Q}_i$. We call this event G (for "good").

Step (2): Full transcript is not doomed. We show that conditioned on G, the full transcript $\tilde{\mathcal{T}}$ is not doomed. Otherwise, by the definition of a doomed state, we have (using Notation 6.3)

$$0 = (\mathcal{V}^\otimes)^{([\mathcal{Q}_i, \tilde{\Pi}^i(j)])_{j\in\mathcal{Q}_i})]_{i\in[l]}}(x; r_1, \ldots, r_l) = (\mathcal{V}^\otimes)^{([\mathcal{Q}_i, (\Pi^i(j))_{j\in\mathcal{Q}_i})]_{i\in[l]}}(x; r_1, \ldots, r_l),$$

where the $\Pi^i(j)$'s are the commitment openings which $\mathcal{A}$ provides to the NIZK verifier in the NIZK execution. The proof of the this equation is identical to the proof of step (2) in the soundness analysis. This equation then contradicts the assumption that $\mathcal{A}$ succeeded in breaking extraction, because the NIZK verifier $\mathcal{V}$ accepts only if the IOP verifier $\mathcal{V}^\otimes$ accepts.

We summarize this step of the proof by noting that since the full transcript is not doomed, $\mathcal{A}'$'s output is *never* computed by the fall-back case "if no such i exists...".

Step (3): $\mathcal{A}'$ breaks CI (via case analysis). We now consider two possible cases. Assume first that the round-1 partial transcript $\tilde{\mathcal{T}} \restriction_1 = (\tilde{\Pi}^1, r^1)$ is not doomed. By the doom-checkability guarantee of $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$, and $\mathcal{A}'$'s construction, in this case $\mathcal{A}'$ outputs $(x, 0, \emptyset, \mathsf{c}^1)$. (We stress that this output results from the fact that $C^\otimes(x, \tilde{\mathcal{T}} \restriction_1) = 0$, and not because of the fall-back case.)

Parse $\tilde{\Pi}^1$ as $\tilde{\Pi}^1 = \{\tilde{\Pi}_1^1, \ldots, \tilde{\Pi}_{t_\kappa}^1\}$, where the message lengths of the IOPs $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ and $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$ are u and $\ell = u \cdot t_\kappa$, respectively, and $\left|\tilde{\Pi}_k^1\right| = u$ for every $k \in [t_\kappa]$. (Intuitively, $\tilde{\Pi}_k^1$ is the round-1 prover message sent in the k'th repetition of the base-IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$.)

Since $\mathcal{A}$ broke NIZK extraction, $(x, \mathsf{E}(x, \pi)) \notin \mathcal{R}$, and therefore, by construction of E, $(x, \mathsf{E}^\otimes(x, \tilde{\Pi}^1)) \notin \mathcal{R}$ (note that we have conditioned on the EE extractor succeeding). From the knowledge extraction property of $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$, we have that for every $x, \Pi^1 \in \{0,1\}^*$, if $(x, \mathsf{E}^\otimes(x, \Pi^1)) \notin \mathcal{R}$

[27] Similar to the soundness proof, doom-checkability of $(\mathcal{P}^\otimes, \mathcal{V}^\otimes)$ with respect to $\mathcal{D}^\otimes$ follows from the doom-checkability of the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$; see Remark 7.3.

then

$$\Pr_{r \leftarrow \mathcal{C}_1}\left[(x, \Pi^1, r) \notin \mathcal{D}^{\otimes}\right] \leq \epsilon_{\mathsf{E}}^{\otimes}.$$

Therefore, we are in the case that the round-1 partial transcript is not doomed only with probability $\epsilon_{\mathsf{E}}^{\otimes}$.[28] Next, assume the round-1 partial transcript *is* doomed, and recall that we have shown in step (2) above that the full transcript $\tilde{\mathcal{T}}$ (i.e., the round-l transcript) is not doomed. Therefore, by the doom-checkability guarantee of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ and $\mathcal{A}''$s construction, $\mathcal{A}'$ will output $(x, i, \mathcal{T}_{\mathsf{CZK}} \restriction_i, \mathsf{c}^{i+1})$ for some $i > 1$. Indeed, $\mathcal{A}'$ only outputs the round-1 transcript if it is not doomed (which occurs with probability $\leq \epsilon_{\mathsf{E}}^{\otimes}$) or if no round-$i$ partial transcript was not doomed (which is not possible since the round-l transcript is not doomed). To complete the proof, we show that if $\mathcal{A}'$ outputs $(x, i, \mathcal{T}_{\mathsf{CZK}} \restriction_i, \mathsf{c}^{i+1})$ for some $i > 1$, then $\left((x, i, \mathcal{T}_{\mathsf{CZK}} \restriction_i, \mathsf{c}^{i+1}), r^{i+1}\right) \in \mathsf{BAD}$, so $\mathcal{A}'$ broke CI. For this, we need to prove that: (1) $(x, \tilde{\mathcal{T}} \restriction_i |\tilde{\Pi}^{i+1}| r^{i+1}) \notin \mathcal{D}^{\otimes}$, and (2) $(x, \tilde{\mathcal{T}} \restriction_i) \in \mathcal{D}^{\otimes}$. (1) follows directly from the description of $\mathcal{A}'$ and the doom checkability guarantee, because $i > 1$. (2) follows from the minimality of i, and the fact that any transcript is either doomed or not. ■

Instantiation Example: NIZKs Approaching Witness Length. To demonstrate how to use our framework, we instantiate Theorem 7.6 with the base IOP of Lemma B.17 and the EE commitments of Theorem 4.6. This yields NIZK arguments whose length approaches the witness length for relations verifiable in $\mathsf{AC}^0[\oplus]$, namely by constant-depth and polynomial size boolean circuits over $\wedge, \vee, \neg, \oplus$ gates of unbounded fan in. Formally:

Corollary 7.8. *Assuming (leveled)* LWE*, there exist NIZK arguments for all languages in* $\mathsf{AC}^0[\oplus]$*, with proof size* $m \cdot \mathsf{poly}\log(|C|, \kappa)$ *where* C *is an arithmetic circuit for the* NP *verification function,* m *is the witness length, and* κ *is a computational security parameter.*

We note that $\mathsf{AC}^0[\oplus]$ contain several interesting languages, e.g., 3SAT. Thus, Corollary 7.8 gives a NIZK for 3SAT whose length approaches the length of the satisfying assignment, which could be significantly shorter than the length of the 3CNF (whose size can be quadratic in the size of the assignment).

Proof of Corollary 7.8 (Sketch). The corollary follows from Theorem 7.6 by instantiating the NIZK of Figure 3 with an appropriate CZK argument. Specifically, the CZK is instantiated with the EE commitments of Theorem 4.6, and the IOP $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ is instantiated as the t_κ-fold parallel-repetition of the base IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ given in Lemma B.17, for $t_\kappa = \lceil \frac{2^{15} \cdot \kappa}{27} \rceil$.

Lemma B.17 guarantees that $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ has 2 rounds, doom checkability, statistical semi-HVZK, completeness and knowledge extraction with negligible error, and soundness and RBR soundness with error $1/8$. In particular, the requirement of Theorem 7.4 hold for our choice of t_κ, since $\Delta := \frac{1}{2} - \frac{1}{8} = \frac{3}{8} \geq \kappa^{-O(1)}$, $\Delta^3 = \frac{27}{8^3} = \frac{27}{2^9}$ and so $t_\kappa = \lceil \frac{2^{15} \cdot \kappa}{27} \rceil = \frac{2^6 \cdot \kappa}{\Delta^3} = \max\{\frac{64 \cdot \kappa}{\Delta^3}, 3\}$. Therefore, Lemma 7.5 guarantees that there exists an l-wise independent CIH for the bad relation BAD of Definition 7.1. We can therefore apply Theorem 7.6 to this instantiation of the NIZK of Figure 3.

By Theorem 7.6, the proof size of the NIZK is $l \cdot \ell \cdot \mathsf{poly}(\kappa)$, where $\ell = u \cdot t_\kappa$ and u is the length of messages generated by $\mathcal{P}_{\mathsf{IOP}}$. The IOP of Lemma B.17 has $l = 2$ rounds and proof size $u = m \cdot \kappa \cdot \mathsf{poly}\log(|C|)$ (because in Figure 3, we set the statistical security parameter of the IOP to be κ). The resulting NIZK proof size is therefore $|\pi| = l \cdot t_\kappa \cdot u \cdot \mathsf{poly}(\kappa) = l \cdot t_\kappa \cdot m \cdot \kappa \cdot \mathsf{poly}\log(|C|) \cdot \mathsf{poly}(\kappa)$. For $t_\kappa = \mathsf{poly}(\kappa)$, and $l = 2$, we get $|\pi| = m \cdot \mathsf{poly}(\log(|C|), \kappa)$. ■

[28] This is the main difference from the soundness analysis: in the soundness analysis, if the round-1 transcript was not doomed then $\mathcal{A}'$ broke CI since it found an instance of $\mathsf{BAD}^{=0}$. In the extraction analysis, we can instead argue that if NIZK-extraction fails, then IOP-extraction also fails, in which case the round-1 transcript is not doomed only with small probability.

Remark 7.9 (Prior Works on Rate-1 NIZK Arguments and NIZK Arguments Approaching Witness Length)**.** *Afshar et al. [ACG+24] and Branco et al. [BDS25] recently constructed rate-1 NIZKs for NP from LWE, namely NIZKs whose length is $|w| + o(|w|) \cdot \mathsf{poly}(\kappa)$. These NIZKs subsume Corollary 7.8 as they construct shorter NIZKs for a larger class of relations from the same assumptions. Prior to these works, Ghosal, Lou and Sahai [GLS22, Thm. 1.3] employed the MPCitH paradigm to construct NIZKs for NP from LWE, whose length is roughly $O(|C|)$, where C is the verification circuit of the relation. They also obtain NIZKs for NP with length $|w| + \mathsf{poly}(\kappa)$ (w is the NP witness), assuming FHE. Corollary 7.8 can be interpreted as a scaling-down of this result: it shows that by restricting the relation class from NP to $\mathsf{AC}^0[\oplus]$, we can obtain NIZKs whose proof approaches the witness length assuming only (leveled) LWE. This also demonstrates the potential advantage of replacing PCPs with IOPs in the FS+CIH paradigm. Ishai, Kushilevitz, Ostrovsky and Sahai [IKOS09, Appendix A] construct unconditional NIZK arguments whose proof approaches the witness length, in the* correlated randomness *model, namely where the prover and verifier obtain private correlated (but crucially, different) randomness sampled honestly in a preprocessing phase. Thus, their construction is not directly comparable to ours', due to the different preprocessing model.*

References

[ACG+24] Abtin Afshar, Jiaqi Cheng, Rishab Goyal, Aayush Yadav, and Saikumar Yadugiri. Encrypted RAM delegation: Applications to rate-1 extractable arguments, homomorphic NIZKs, MPC, and more. Cryptology ePrint Archive, Report 2024/1806, 2024. https://eprint.iacr.org/2024/1806.

[AH18] Kurt M. Alonso and Jordi Herrera-Joancomartí. Monero - privacy in the blockchain. Cryptology ePrint Archive, Paper 2023/1256, 2018. https://eprint.iacr.org/2018/535.

[AHI+17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. Low-complexity cryptographic hash functions. In *ITCS*, pages 7:1–7:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. https://eprint.iacr.org/2017/036.

[ALM+92] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. In *FOCS*, pages 14–23, 1992.

[AS92] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; A new characterization of NP. In *FOCS*, pages 2–13, 1992.

[BCG+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from Bitcoin. In *S&P*, pages 459–474. IEEE Computer Society, 2014.

[BCG+17] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. Interactive oracle proofs with constant rate and query complexity. In *ICALP*, pages 40:1–40:15, 2017.

[BCGV16] Eli Ben-Sasson, Alessandro Chiesa, Ariel Gabizon, and Madars Virza. Quasi-linear size zero knowledge from linear-algebraic PCPs. In *TCC 2016-A, Proceedings, Part II*, pages 33–64. Springer, 2016. https://eprint.iacr.org/2016/021.

[BCL22] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge IOPs with linear-time prover and polylogarithmic-time verifier. In *EUROCRYPT, Proceedings, Part II*, pages 275–304. Springer, 2022. https://eprint.iacr.org/2020/1527.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *TCC 2016-B, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 31–60, 2016.

[BDG+13] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. Why "Fiat-Shamir for proofs" lacks a proof. In *TCC*, pages 182–201. Springer, 2013.

[BDS25] Pedro Branco, Nico Döttling, and Akshayaram Srinivasan. Rate-1 statistical non-interactive zero-knowledge. In *CRYPTO, Proceedings, Part VII*, pages 514–547. Springer, 2025.

[BG08] Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM J. Comput.*, 38(5):1661–1694, 2008.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO, Proceedings*, Lecture Notes in Computer Science, pages 1–18. Springer, 2001.

[BGTZ23] Alexander R. Block, Albert Garreta, Pratyush Ranjan Tiwari, and Michał Zając. On soundness notions for interactive oracle proofs. Cryptology ePrint Archive, Paper 2023/1256, 2023. https://eprint.iacr.org/2023/1256.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS*, pages 309–325. ACM, 2012.

[BI05] Omer Barkol and Yuval Ishai. Secure computation of constant-depth circuits with applications to database search problems. In *CRYPTO*, Lecture Notes in Computer Science, pages 395–411. Springer, 2005.

[BISW18] Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Quasi-optimal SNARGs via linear multi-prover interactive proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT, Proceedings, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 222–255. Springer, 2018.

[BKK+16] Eli Ben-Sasson, Yohay Kaplan, Swastik Kopparty, Or Meir, and Henning Stichtenoth. Constant rate PCPs for circuit-SAT with sublinear query complexity. *Journal of the ACM*, 63(4):32:1–32:57, 2016.

[BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In *CRYPTO, Proceedings, Part III*, pages 738–767. Springer, 2020.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *CRYPTO*, pages 868–886. Springer, 2012.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106. IEEE Computer Society, 2011.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, pages 1–12. ACM, 2014.

[BWW24] Eli Bradley, Brent Waters, and David J. Wu. Batch arguments to NIZKs from one-way functions. In *TCC, Proceedings, Part II*, pages 431–463. Springer, 2024. https://eprint.iacr.org/2023/1938.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In *STOC*, pages 1082–1090. ACM, 2019. https://eprint.iacr.org/2018/1004.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In *TCC 2016-A, Proceedings, Part I*, pages 389–415. Springer, 2016.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In *EUROCRYPT, Proceedings, Part I*, pages 91–122. Springer, 2018. https://eprint.iacr.org/2018/131.

[CDGS22] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, and Nicholas Spooner. Concrete security for succinct arguments from vector commitments. Cryptology ePrint Archive, Report 2023/1737, 2022. https://eprint.iacr.org/2023/1737.

[CFW26] Alessandro Chiesa, Giacomo Fenzi, and Guy Weissenberg. Zero-knowledge IOPPs for constrained interleaved codes. Cryptology ePrint Archive, Report 2026/391, 2026. https://eprint.iacr.org/2026/391.

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for P from LWE. In *FOCS*, pages 68–79. IEEE, 2021. https://eprint.iacr.org/2021/808.

[CKU20] Geoffroy Couteau, Shuichi Katsumata, and Bogdan Ursu. Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. In *EUROCRYPT, Proceedings, Part III*, pages 442–471. Springer, 2020.

[CLW18] Ran Canetti, Alex Lombardi, and Daniel Wichs. Non-interactive zero knowledge and correlation intractability from circular-secure FHE. Cryptology ePrint Archive, Report 2018/1248, 2018. https://eprint.iacr.org/2018/1248.

[CPV20] Michele Ciampi, Roberto Parisella, and Daniele Venturi. On adaptive security of delayed-input sigma protocols and Fiat-Shamir NIZKs. In *SCN*, pages 670–690. Springer, 2020.

[CRT24] Lijie Chen, Ron D. Rothblum, and Roei Tell. Fiat-Shamir in the plain model from derandomization (or: Do efficient algorithms believe that NP = PSPACE?). *Electron. Colloquium Comput. Complex.*, TR24-116, 2024.

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

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194. Springer, 1986.

[FS11] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *Journal of Computer and System Sciences*, 77(1):91–106, 2011.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory*, 31(4):469–472, 1985.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *FOCS*, pages 102–113. IEEE Computer Society, 2003.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In *STOC*, pages 113–122. ACM, 2008.

[GLS22] Riddhi Ghosal, Paul Lou, and Amit Sahai. Efficient NIZKs from LWE via polynomial reconstruction and "MPC in the Head"'. In *ASIACRYPT, Proceedings, Part II*, volume 13792 of *Lecture Notes in Computer Science*, pages 496–521. Springer, 2022.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO. Proceedings, Part I*, pages 75–92. Springer, 2013.

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In *FOCS*, pages 850–858. IEEE Computer Society, 2018. https://eprint.iacr.org/2018/385.

[HLR21] Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. Fiat-Shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge). In *STOC*, pages 750–760. ACM, 2021. https://eprint.iacr.org/2021/286.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *STOC*, pages 21–30. ACM, 2007.

[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.

[IMS12] Yuval Ishai, Mohammad Mahmoody, and Amit Sahai. On efficient zero-knowledge PCPs. In *TCC*, pages 151–168, 2012.

[IMSX15] Yuval Ishai, Mohammad Mahmoody, Amit Sahai, and David Xiao. On zero-knowledge PCPs : Limitations , simplifications , and applications, 2015. https://api.semanticscholar.org/CorpusID:52059436.

[JJ21] Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In *EUROCRYPT, Proceedings, Part I*, pages 3–32. Springer, 2021.

[JKKZ21] Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Yun Zhang. SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE. In *STOC*, pages 708–721. ACM, 2021.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, pages 723–732. ACM, 1992.

[KLR06] Eyal Kushilevitz, Yehuda Lindell, and Tal Rabin. Information-theoretically secure protocols and security under composition. In *STOC*, pages 109–118. ACM, 2006.

[KR08] Yael Tauman Kalai and Ran Raz. Interactive PCP. In *ICALP*, pages 536–547, 2008.

[KRR17a] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In *CRYPTO, Proceedings, Part II*, pages 224–251. Springer, 2017. https://eprint.iacr.org/2016/303.

[KRR17b] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In *CRYPTO, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 224–251. Springer, 2017.

[KRS25] Dmitry Khovratovich, Ron D. Rothblum, and Lev Soukhanov. How to prove false statements: Practical attacks on Fiat-Shamir. In *CRYPTO, Proceedings, Part VI*, Lecture Notes in Computer Science, pages 3–26. Springer, 2025.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.

[Nao89] Moni Naor. Bit commitment using pseudo-randomness. In *CRYPTO*, pages 128–136. Springer, 1989.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In *CRYPTO, Proceedings, Part I*, pages 89–114. Springer, 2019.

[Reg03] Oded Regev. New lattice based cryptographic constructions. In *STOC*, pages 407–416. ACM, 2003.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 84–93. ACM, 2005.

[RR20] Noga Ron-Zewi and Ron D. Rothblum. Local proofs approaching the witness length [extended abstract]. In *FOCS*, pages 846–857, 2020.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *STOC*, pages 49–62. ACM, 2016.

[RW24] Noga Ron-Zewi and Mor Weiss. Zero-knowledge IOPs approaching witness length. In *CRYPTO, Proceedings, Part X*, pages 105–137. Springer, 2024. https://eprint.iacr.org/2024/816.

[VW26] Himanshu Vashishth and Mor Weiss. Composition theorems for zero-knowledge IOPs. Cryptology ePrint Archive, Report 2026/307, 2026. https://eprint.iacr.org/2026/307.

[WW25] Brent Waters and David J. Wu. A pure indistinguishability obfuscation approach to adaptively-sound SNARGs for NP. In *CRYPTO, Proceedings, Part VII*, volume 16006 of *Lecture Notes in Computer Science*, pages 292–326. Springer, 2025.

A Parallel Repetition of semi-HVZK IOP with Implicit Inputs

In this section we extend our results from Section 5 on parallel-repetition of semi-HVZK-IOPs to capture also settings in which the IOP has *implicit* inputs. In this setting (defined in Section A.1 below), part of the input is given to the verifier *implicitly*. That is, the input is divided into two parts

$x = (x_{\text{exp}}, x_{\text{imp}})$, where the verifier explicitly reads the *explicit input* x_{exp}, but only has *oracle access* to the *implicit input* x_{imp}. This captures, as a special case, the setting considered thorughout the paper, in which the verifier receives the entire input. Then, we show in Section A.2 that parallel repetition of semi-HVZK-IOPs *with implicit* inputs preserves the semi-HVZK property.

A.1 (Semi-HVZK) IOPs with Implicit Inputs

In this section, we generalize the definition of semi-HVZK-IOPs (Section 4.3) to the case in which part of the input is given explicitly.

Definition A.1 (semi-HVZK-IOPs (Implicit Input Setting))**.** *A semi-Honest-Verifier Zero-Knowledge l-round (public coin) Interactive Oracle Proof (semi-HVZK-IOP) with soundness error ϵ_{S} and completeness error ϵ_{C} in the implicit input setting for an NP language $\mathcal{L}$ (with corresponding NP relation $\mathcal{R}$) is a pair $(\mathcal{P}, \mathcal{V})$ of PPT algorithms satisfying the following properties.*

- **Syntax.**
  - ***Inputs:*** *Prover $\mathcal{P}$ receives as input $x = (x_{\text{exp}}, x_{\text{imp}})$ with a corresponding witness w. Verifier $\mathcal{V}$ receives as input x_{exp} and $1^{|x_{\text{imp}}|}$, and is also given oracle access to x_{imp}.*
  - **Protocol Structure.** *The protocol consists of a* Communication Phase *followed by a* Query Phase*. In the* Communication phase*, $\mathcal{P}$ and $\mathcal{V}$ interact for l rounds, where $\mathcal{V}$'s messages are uniformly random strings $r_1, \ldots, r_l$, and $\mathcal{P}$'s messages are* oracle *messages. In the* Query phase*, $\mathcal{V}$ makes (non-adaptive) queries to the implicit input x_{imp} and to $\mathcal{P}$'s oracle messages. These queries are determined solely by the explicit input x_{exp}, $1^{|x_{\text{imp}}|}$, and $\mathcal{V}$'s randomness string $r := (r_1, \ldots, r_l)$. $\mathcal{V}$ then deterministically decides whether to accept or reject based on the answers to these queries, the explicit input x_{exp}, and the randomness strings $r_1, \ldots, r_l$.*
- **Semantics.** *The protocol satisfies the following semantic properties:*
  - **Completeness:** *If $(x, w) \in \mathcal{R}$, then when $\mathcal{V}$ interacts with $\mathcal{P}$, it accepts with probability at least $1 - \epsilon_{\mathsf{C}}$.*
  - **Soundness:** *If $x \notin \mathcal{L}$, then for any (possibly unbounded) prover strategy $\mathcal{P}^*$, when $\mathcal{V}$ interacts with $\mathcal{P}^*$, it accepts with probability at most ϵ_{S}.*
  - **Semi Honest-Verifier Zero-Knowledge (semi-HVZK):** *there exists a PPT simulator* Sim *such that for every $((x_{\text{exp}}, x_{\text{imp}}), w) \in \mathcal{R}$, and every possible randomness string r for $\mathcal{V}$, the following are identically distributed.*
    - *The view $\mathsf{View}_{\mathcal{V}}(x_{\text{exp}}, x_{\text{imp}}, w; r)$, which is $\mathcal{V}$'s view conditioned on $\mathcal{V}$ using randomness r. (Notice that the view now consists of $x_{\text{exp}}, 1^{|x_{\text{imp}}|}$, $\mathcal{V}^*$'s random coins, and the answers to $\mathcal{V}^*$'s queries to x_{imp} and $\mathcal{P}$'s oracle messages.)*
    - *The output $\mathsf{Sim}(r, x_{\text{exp}}, 1^{|x_{\text{imp}}|}, x_{\text{imp}}(\mathcal{Q}_r))$ of the simulator, when given the verifier's randomness r, the explicit input x_{exp}, and the answers $x_{\text{imp}}(\mathcal{Q}_r)$ to the queries $\mathcal{Q}_r$ which $\mathcal{V}$, when using randomness r, makes to the implicit input x_{imp}.*

Similar to Definition 4.11, we extend the definition to allow the parties to take as input a statistical security parameter λ (in unary), where $\epsilon_{\mathsf{C}}, \epsilon_{\mathsf{S}}$ might depend on λ. Similarly, we consider also statistical *semi-HVZK, where the simulator is also given 1^{λ} as input, and the simulated view is $\mathsf{negl}(\lambda)$-statistically close to the real world view (conditioned on r).*

Notation A.2. *Let $(\mathcal{P}, \mathcal{V})$ be an IOP for an NP language $\mathcal{L}$. For $x = (x_{\text{exp}}, x_{\text{imp}}) \in \mathcal{L}$, and a choice r of randomness of $\mathcal{V}$, we use $\mathcal{Q}_r$ to denote the queries which $\mathcal{V}$, when using randomness r, makes to x_{imp}.*[29]

[29] We note that $\mathcal{Q}_r$ is well defined, regardless of $\mathcal{P}$'s randomness, since $\mathcal{V}$ is non adaptive (by definition).

Remark A.3. *We note that the semi-HVZK property differs between the settings with and without implicit inputs. Indeed, in the latter, semi-HVZK guarantees that the interaction reveals to a (semi) honest verifier nothing about the witness (the view can be simulated given x alone). On the other hand, when part of the input is given implicitly, semi-HVZK guarantees that the verifier learns nothing about w, and* only *the* physical bits *it queried from x_{imp} (the view can be simulated given these bits, with no additional access to x_{imp} and with no access to w).*

A.2 The Composition Theorem

In this section, we prove that parallel-composition of IOPs preserves semi-HVZK, also with implicit inputs.

Construction A.4 (Parallel Repetition IOP $(\mathcal{P}, \mathcal{V})$)**.** *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be a semi-HVZK-IOP for a language $\mathcal{L}$, and let $t_\lambda \in \mathbb{N}$. We construct the t_λ-fold parallel repetition $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ of $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$, which is an IOP for $\mathcal{L}$, as follows. The prover $\mathcal{P}$ receives as input $x = (x_{\text{exp}}, x_{\text{imp}})$ with a corresponding witness w. the verifier $\mathcal{V}$ receives as input $x_{\text{exp}}, 1^{|x_{\text{imp}}|}$ and has oracle access to x_{imp}. $\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes}$ emulate t_λ random and independent copies of $\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}}$ (respectively), with inputs (x, w) and $x_{\text{exp}}, 1^{|x_{\text{imp}}|}$, respectively (as in Construction 5.1).*

Theorem A.5 (IOP Composition theorem)**.** *Let $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ be an l-round IOP for $\mathcal{L}$ in the implicit input setting with soundness error ϵ_{S} and completeness error ϵ_{C}, that has (statistical) semi-HVZK with implicit inputs (Definition A.1). Then the t_λ-fold parallel repetition $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ (Construction A.4) is a (statistical) semi-HVZK-IOP for $\mathcal{L}$ with soundness error $\epsilon_{\mathsf{S}}^{t_\lambda}$ and completeness error $t_\lambda \cdot \epsilon_{\mathsf{C}}$.*

Proof: The completeness and soundness properties are proven exactly as in the proof of Theorem 5.4. We proceed to prove (statistical) semi-HVZK.

(Statistical) Semi-HVZK. We show that $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ is statistical semi-HVZK whenever the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$ is statistical semi-HVZK. The proof for perfect semi-HVZK follows similarly. Let $\mathsf{Sim}_{\mathsf{IOP}}$ be the simulator whose existence is guaranteed by the semi-HVZK property $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. That is, there exists a negligible function $\epsilon_{\mathsf{ZK}}(\lambda)$ such that for every possible randomness r_{IOP} of the honest verifier $\mathcal{V}_{\mathsf{IOP}}$, and any $x = (x_{\text{exp}}, x_{\text{imp}}) \in \mathcal{L}$, Sim on input $1^\lambda, r_{\mathsf{IOP}}, x_{\text{exp}}, 1^{|x_{\text{imp}}|}$, and $x_{\text{imp}}(\mathcal{Q})$ (i.e., the restriction of x_{imp} to the queries $\mathcal{Q}$ which $\mathcal{V}_{\mathsf{IOP}}$ makes when using randomness r_{IOP}), can simulate the view of the honest verifier $\mathcal{V}_{\mathsf{IOP}}$ when using randomness r_{IOP}, and the real and simulated views are $\epsilon_{\mathsf{ZK}}(\lambda)$-statistically close.

We now describe a simulator Sim for the parallel repetition. Sim is given as input $1^\lambda, r, x_{\text{exp}}, 1^{|x_{\text{imp}}|}$, and $x_{\text{imp}}(\mathcal{Q})$. Sim parses r as $r = (r^1, \ldots, r^{t_\lambda})$ where $r^j := (r_1^j, \ldots, r_l^j)$ for each $j \in [t_\lambda]$. Sim invokes $\mathsf{Sim}_{\mathsf{IOP}}$ on input $(1^\lambda, r^j, x_{\text{exp}}, 1^{|x_{\text{imp}}|}, x_{\text{imp}}(\mathcal{Q}_{r^j}))$ for each $j \in [t_\lambda]$ where $x_{\text{imp}}(\mathcal{Q}_{r^j})$ is the restriction of x_{imp} to the queries $\mathcal{Q}_{r^j}$ which $\mathcal{V}_{\mathsf{IOP}}$ makes when using randomness r^j. Each of these invocations is executed independently with uniform and independent randomness for $\mathsf{Sim}_{\mathsf{IOP}}$. Sim outputs $\{\mathsf{Sim}_{\mathsf{IOP}}(1^\lambda, r^j, x_{\text{exp}}, 1^{|x_{\text{imp}}|}, x_{\text{imp}}(\mathcal{Q}_{r^j}))\}_{j \in [t_\lambda]}$.

Let $\mathcal{R} = \mathcal{R}(x, w)$ denote the relation corresponding to $\mathcal{L}$. We show that for any $(x = (x_{\text{exp}}, x_{\text{imp}}), w) \in \mathcal{R}$, and any randomness r for the honest $\mathcal{V}^{\otimes}$, the output of Sim is $t_\lambda \cdot \epsilon_{\mathsf{ZK}}(\lambda)$-statistically close to the real-world view of $\mathcal{V}^{\otimes}$ when using randomness r. That is, $\mathsf{View}_{\mathcal{V}^{\otimes}}(\lambda, x_{\text{exp}}, x_{\text{imp}}, w; r)$ and $\mathsf{Sim}(1^\lambda, r, x_{\text{exp}}, 1^{|x_{\text{imp}}|}, x_{\text{imp}}(\mathcal{Q}))$ are $t_\lambda \cdot \epsilon_{\mathsf{ZK}}(\lambda)$-statistically close.

We show this via a hybrid argument where we replace the views of the underlying IOP from real to simulated one at a time. Specifically, we define hybrids $\mathcal{H}_0, \ldots, \mathcal{H}_{t_\lambda}$ where

$$\mathcal{H}_i = \Big(\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}\left(\lambda, x_{\text{exp}}, x_{\text{imp}}, w; r^1\right), \ldots, \mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}\left(\lambda, x_{\text{exp}}, x_{\text{imp}}, w; r^i\right), \\ \mathsf{Sim}_{\mathsf{IOP}}(1^\lambda, r^{i+1}, x_{\text{exp}}, 1^{|x_{\text{imp}}|}, x_{\text{imp}}(\mathcal{Q}_{r^{i+1}})), \ldots, \mathsf{Sim}_{\mathsf{IOP}}(1^\lambda, r^{t_\lambda}, x_{\text{exp}}, 1^{|x_{\text{imp}}|}, x_{\text{imp}})\mathcal{Q}_{r^{t_\lambda}}))\Big)$$

Then $\mathcal{H}_{t_\lambda}$ is the view of $\mathcal{V}^{\otimes}$ in the real execution, and $\mathcal{H}_0$ is the output of Sim. For the sake of contradiction, assume that $\mathsf{SD}(\mathcal{H}_0, \mathcal{H}_{t_\lambda}) > t_\lambda \cdot \epsilon_{\mathsf{ZK}}(\lambda)$. Then, there exists a distinguisher $\mathcal{D}$ that obtained advantage $> t_\lambda \cdot \epsilon_{\mathsf{ZK}}(\lambda)$ in distinguishing between $\mathcal{H}_0$ and $\mathcal{H}_{t_\lambda}$. Therefore, there exists an $1 \leq i^* \leq t_\lambda$ such that $\mathcal{D}$ obtains advantage $> \epsilon_{\mathsf{ZK}}(\lambda)$ in distinguishing between $\mathcal{H}_{i^*-1}, \mathcal{H}_{i^*}$. We show that this contradicts the semi-HVZK property of the underlying IOP $(\mathcal{P}_{\mathsf{IOP}}, \mathcal{V}_{\mathsf{IOP}})$. We construct a non-uniform distinguisher that has $\lambda, x = (x_{\mathrm{exp}}, x_{\mathrm{imp}}), w, i^*, r = (r^1, \ldots, r^{t_\lambda})$ hard-coded. It takes as input a random variable V that is either $\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}\left(\lambda, x_{\mathrm{exp}}, x_{\mathrm{imp}}, w; r^{i^*}\right)$ or $\mathsf{Sim}_{\mathsf{IOP}}(1^\lambda, r^{i^*}, x_{\mathrm{exp}}, 1^{|x_{\mathrm{imp}}|}, x_{\mathrm{imp}}(\mathcal{Q}_{r^{i^*}}))$. The distinguisher honestly generates $i^* - 1$ IOP views $\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}\left(\lambda, x_{\mathrm{exp}}, x_{\mathrm{imp}}, w; r^i\right)$, for $i \in [i^* - 1]$ using random strings r^i, and (x, w) (by emulating an honest execution of $\mathcal{P}_{\mathsf{IOP}}$ with $\mathcal{V}_{\mathsf{IOP}}$). Then, it invokes $t_\lambda - i^*$ random and independent copies of $\mathsf{Sim}_{\mathsf{IOP}}$ to generate simulated views $\mathsf{Sim}_{\mathsf{IOP}}(1^\lambda, r^i, x_{\mathrm{exp}}, 1^{|x_{\mathrm{imp}}|}, x_{\mathrm{imp}}(\mathcal{Q}_{r^i}))$ for $i \in \{i^*, \ldots, t_\lambda\}$ using random strings r^i. Now, the distinguisher invokes $\mathcal{D}$ on $\left\{\{\mathsf{View}_{\mathcal{V}_{\mathsf{IOP}}}\left(\lambda, x_{\mathrm{exp}}, x_{\mathrm{imp}}, w; r^i\right)\}_{i\in[i^*-1]}, V, \{\mathsf{Sim}_{\mathsf{IOP}}(1^\lambda, r^i, x_{\mathrm{exp}}, 1^{|x_{\mathrm{imp}}|}, x_{\mathrm{imp}}(\mathcal{Q}_{r^i}))\}_{i\in\{i^*,\ldots,t_\lambda\}}\right\}$, and outputs whatever $\mathcal{D}$ outputs. ■

B A Base IOP for Our Constructions

In this section, we construct a "base IOP" whose parallel-repetition will be used to instantiate our CZK arguments in Section 6, which in turn are used in our NIZK construction in Section 7. The base IOP construction uses the "MPC-in-the-Head" (MPCitH) paradigm of [IKOS07]. We first give some background on MPC protocols in Appendix B.1, then describe a generic IOP construction from MPCitHead (Appendix B.2). Appendix B.2 also contains a discussion of different approaches towards constructing NIZK via MPCitH, and a comparison with a recent construction of [GLS22]. The base IOP used in Sections 6 and 7 is obtained in Appendix B.3 by instantiating the generic construction (of Appendix B.2) with an MPC protocol of [BI05].

B.1 MPC Preliminaries

Let $k \in \mathbb{N}$. We consider a k-party MPC protocol Π_{MPC} for a function $f(x_1, \ldots, x_k)$, which enables k mutually-distrusting parties $\mathsf{P}_1, \ldots, \mathsf{P}_k$ to jointly evaluate f on their inputs, while preserving the secrecy of the inputs, and the correctness of the outputs, even in the presence of (few) corrupted parties. This property can be formalized in different ways, depending on the type and number of corrupted parties. For our purposes, it suffices for Π_{MPC} to have rather mild guarantees, specifically, to be correct when all parties are honest, and hide the honest parties' inputs in the presence of few passive corruptions, as defined below.

Definition B.1 (Correctness). *A k-party protocol Π_{MPC} for a function $f(x_1, \ldots, x_k)$ is* correct with error ϵ_{C} *if for all $x_1, \ldots, x_k$:*

$$\Pr[\exists i \in [k] \text{ s.t. } \mathsf{P}_i \text{ outputs } y \neq f(x_1, \ldots, x_k) \text{ in } \Pi_{\mathsf{MPC}}] \leq \epsilon_{\mathsf{C}}$$

where the probability is over the random inputs of all parties.

The *view* View_i of party P_i in Π_{MPC} consists of P_i's inputs, random inputs, and the messages he received from the other parties in Π_{MPC}'s execution. (Messages that P_i sent, and his output, can be computed from his view.)

Definition B.2 (Privacy). *Let $1 \leq t < k$. A k-party protocol Π_{MPC} for $f(x_1, \ldots, x_k)$ is* statistically t-private with error ϵ_{ZK} *if there exists a PPT simulator* Sim *such that for every $x_1, \ldots, x_k$ and every subset*

$T \subset [k]$ *of size* $|T| \leq t$*, the joint view* $(\mathsf{View}_i)_{i\in T}$ *of the parties in* T *is* ϵ_{ZK}*-statistically close to the output* $\mathsf{Sim}(T, (x_i)_{i\in T}, f(x_1, \ldots, x_k))$ *of* Sim*.*

Definition B.3 (Consistent views)**.** *A pair* $\mathsf{View}_i, \mathsf{View}_j$ *of views of parties* $\mathsf{P}_i, \mathsf{P}_j$ *in* Π_{MPC} *are* consistent *(with respect to* Π_{MPC}*) if the outgoing messages from* P_i *to* P_j *which are implicit in* View_i *are identical to the incoming messages from* P_i *to* P_j *reported in* View_j*, and vice versa.*

Definition B.4 (Efficiency Measures)**.** *A* k*-party protocol* Π_{MPC} *has communication complexity* c *is the total number of bits exchanged in the protocol is at most* c*. Similarly,* Π_{MPC} *has randomness complexity* r *if the total number of coins tossed throughout the protocol (over all parties) is at most* r*.*

Following [IKOS07], we use MPC protocols for a special type of "verification" functions of the following form.

Definition B.5 (Verification function for relation $\mathcal{R}$)**.** *For an NP relation* $\mathcal{R} = \mathcal{R}(x, w)$*, the corresponding* k*-party verification function is defined as follows:* $f_x(w_1, \ldots, w_k) := \mathcal{R}(x, w_1 \oplus \ldots \oplus w_k)$*. (This is actually a family of functions, parameterized by an instance* x*. We can also consider a single function that takes* x *as input from all parties, but this description will be more convenient for us.)*

B.2 An IOP From MPC in the Head

Our IOP is described in Figure 4. It is inspired by the ZK proofs of [IKOS07] based on statistically-correct MPC protocols [IKOS09, Fig. 2]. Their construction (which uses commitments to obtain ZK proofs in the plain model) naturally yields an IOP when the commitments are replaced with the query restriction on the verifier. This natural interpretation of their construction is presented in Figure 4, and completeness and semi-HVZK are analyzed similarly to [IKOS09]. We then show that this construction also satisfies the other properties needed for the "IOP+CIH => NIZK" paradigm, namely RBR soundness, doom checkability and knowledge extraction (these properties were not considered in [IKOS09]); see the digest at the end of the section for further details and comparison with [IKOS09].
We now analyze the properties of the protocol.

Lemma B.6 (Completeness)**.** *If* Π_{MPC} *is an* ϵ_{C}*-correct protocol for* $\mathcal{R}$ *then the IOP of Figure 4 has completeness error* ϵ_{C}*.*

Proof: Let $(x, w) \in \mathcal{R}$. From the ϵ_{C}-correctness of Π_{MPC}, when it is executed with hard-wired x and any $w_1, \ldots, w_k$ such that $w_1 \oplus \ldots \oplus w_k = w$, then except with probability ϵ_{C}, all parties in Π_{MPC} output 1. In an honest execution of the IOP, all the views are consistent with each other and with $x, r_1, \ldots, r_k$. Moreover, $r_1, \ldots, r_k$ are distributed uniformly at random, so except with probability ϵ_{C}, all the views lead to output 1. Therefore, the verifier will never reject due to an inconsistency, and might only reject if there is a view that leads to output 0. As discussed above, this happens with probability at most ϵ_{C}. ■

Lemma B.7 (Semi-HVZK)**.** *Let* $\lambda \in \mathbb{N}$ *be a statistical security parameter, and* $\epsilon_{\mathsf{ZK}}(\lambda) = \mathsf{negl}(\lambda)$*. If* Π_{MPC} *is statistically 2-private with error* $\epsilon_{\mathsf{ZK}}(\lambda)$*, then the IOP of Figure 4 has statistical semi-HVZK.*

Proof: Let $\mathsf{Sim}_{\mathsf{MPC}}$ be the simulator whose existence is guaranteed by the statistical 2-privacy of Π_{MPC}. We describe a simulator Sim for $(\mathcal{P}_{\mathsf{B}}, \mathcal{V}_{\mathsf{B}})$ which on input $(1^\lambda, x, (r^V = (r_1^V, \ldots, r_k^V), i, j))$ operates as follows:

1. Samples w_i, w_j uniformly at random. Then, Sim emulates $\mathsf{Sim}_{\mathsf{MPC}}$ with the set $T := \{i, j\}$ of corrupted parties, the inputs w_i, w_j, and the output 1, to obtain simulated views $\mathsf{View}_i, \mathsf{View}_j$.

Semi-HVZK IOPs from MPC in the Head

Let $k \in \mathbb{N}$,[a] let λ be a statistical security parameter, let $\mathcal{R} = \mathcal{R}(x, w)$ be an NP relation, and let Π_{MPC} be a k-party protocol for the verification function f of $\mathcal{R}$ (cf. Definition B.5).[b]

The IOP is executed between a prover $\mathcal{P}_{\mathsf{B}}$ that has input x with corresponding witness w, and a verifier $\mathcal{V}_{\mathsf{B}}$ that has input x (both parties also take 1^{λ} as input), and operates as follows:

1. $\mathcal{P}_{\mathsf{B}}$ samples $w_1, \ldots, w_k$ uniformly at random such that $w_1 \oplus \ldots \oplus w_k = w$. For every $i \in [k]$, he also samples randomness r_i^P for the emulation of P_i in Π_{MPC}. The prover then sends $\pi_1 := \left(w_i, r_i^P\right)_{i \in [k]}$ to the verifier.[c]
2. For every $i \in [k]$, the verifier $\mathcal{V}_{\mathsf{B}}$ samples randomness r_i^V for the emulation of P_i in Π_{MPC}, and sends $\left(r_i^V\right)_{i \in [k]}$ to the prover.
3. $\mathcal{P}_{\mathsf{B}}$ emulates Π_{MPC} on inputs $w_1, \ldots, w_k$ with randomness $r_1 := r_1^P \oplus r_1^V, \ldots, r_k := r_k^P \oplus r_k^V$, to generate the views $\mathsf{View}_1, \ldots, \mathsf{View}_k$ of the parties $\mathsf{P}_1, \ldots, \mathsf{P}_k$. He then sends $\pi_2 := (\mathsf{View}_1, \ldots, \mathsf{View}_k)$ to the verifier.
4. The verifier picks random $i \neq j \in [k]$ and sends them to the prover.
5. **Output.** The verifier queries $(w_i, r_i^P), (w_j, r_j^P), \mathsf{View}_i, \mathsf{View}_j$, and accepts if and only if $\mathsf{View}_i, \mathsf{View}_j$ are consistent (as in Definition B.3) with each other, the inputs (randomness, respectively) reported in $\mathsf{View}_i, \mathsf{View}_j$ are w_i, w_j ($r_i^P \oplus r_i^V, r_j^P \oplus r_j^V$, respectively), and the outputs (which can be computed from the views) are 1.

[a] The exact value of k is set in the proof of Lemma B.17 below.

[b] As described, the IOP is *non uniform* in the sense that for every input x the verification function, and consequently also the MPC protocol, is different. However, the construction is actually uniform when the underlying Π_{MPC} takes as input also the instance x and then evaluates $\mathcal{R}$'s verification circuit with x hard-wired. The MPC protocol we use to instantiate our IOP will have this property.

[c] For simplicity, the IOP is described over a large alphabet (where every $\left(w_i, r_i^P\right)$ is a symbol). This naturally yields an IOP over bits where each symbol is replaced with a corresponding bitstring. Since this transformation preserves semi-HVZK, we chose to present the messages over the natural (large) alphabet.

Figure 4: A Base IOP with semi-HVZK Based on the MPC in the Head Paradigm

2. Let r_i, r_j be the random inputs reported in $\mathsf{View}_i, \mathsf{View}_j$, respectively. Then Sim sets $r_i^P := r_i \oplus r_i^V$ and $r_j^P := r_j \oplus r_j^V$. It then outputs $(x, (r^V = (r_1^V, \ldots, r_k^V), i, j), (w_i, w_j, r_i^P, r_j^P, \mathsf{View}_i, \mathsf{View}_j))$.

Fix some $x \in \mathcal{L}_{\mathcal{R}}$, and a random string $(r^V = (r_1^V, \ldots, r_k^V), i, j)$ for $\mathcal{V}_{\mathsf{B}}$. We now prove that Sim's output given $(x, (r^V, i, j))$ is $\epsilon_{\mathsf{ZK}}(\lambda)$-statistically close to the real-world view $\mathsf{View}_{\mathcal{V}\mathsf{B}}(x, w; (r^V, i, j))$ of $\mathcal{V}_{\mathsf{B}}$, conditioned on her using randomness (r^V, i, j).

We do so by comparing both distributions to the output of the following hybrid simulator $\mathsf{Sim}_{\mathcal{H}}$. $\mathsf{Sim}_{\mathcal{H}}$ is also given the witness w as input, and operates identically to Sim, except that in Step 1 it samples $w_1, \ldots, w_k$ uniformly at random subject to the constraint that $\oplus_{i \in [k]} w_i = w$.

Notice first that the outputs of $\mathsf{Sim}, \mathsf{Sim}_{\mathcal{H}}$ are identically distributed. Indeed, w_i, w_j are distributed identically in the executions of $\mathsf{Sim}, \mathsf{Sim}_{\mathcal{H}}$, and that is the only difference between the two executions.

Next, we show that $\mathsf{Sim}_{\mathcal{H}}$'s output is ϵ_{ZK}-statistically close to the real-world view of $\mathcal{V}_{\mathsf{B}}$, conditioned on her using the randomness r^V, i, j. Notice that w_i, w_j are distributed identically in the real world view and in $\mathsf{Sim}_{\mathcal{H}}$'s simulation, so we can condition on them. Under this conditioning, the statistical 2-privacy of Π_{MPC} guarantees that $\mathsf{View}_i, \mathsf{View}_j$ are ϵ_{ZK}-statistically close in the real and simulated views, namely $(w_i, w_j, r_i, r_j, \mathsf{View}_i, \mathsf{View}_j)$ are ϵ_{ZK}-statistically close in a real-world

execution of Π_{MPC} and an execution of $\mathsf{Sim}_{\mathsf{MPC}}$. The only difference between these and the real and simulated (by $\mathsf{Sim}_{\mathcal{H}}$) views of $\mathcal{V}_{\mathsf{B}}$ is that r_i, r_j are replaced with r_i^P, r_j^P. However, the latter are fully determined by the former (and the fixed r_i^V, r_j^V). Since applying a function cannot increase the statistical distance, we conclude that $\mathsf{Sim}_{\mathcal{H}}$'s output is ϵ_{ZK}-statistically close the real-world view of $\mathcal{V}_{\mathsf{B}}$. ■

Remark B.8. *Let $\varepsilon_{\mathsf{IKOS}}$ denote the soundness error obtained in [IKOS09] when using a* perfectly *correct protocol Π_{MPC} (i.e., when the correctness error is 0). Ishai et al. show that $\varepsilon_{\mathsf{IKOS}} = 1 - 1/\binom{k}{2}$ [IKOS09, Thm. 3.1]. They further show that if Π_{MPC} is ϵ_{C}-correct then the soundness error of the IOP of Figure 4 is $\varepsilon_{\mathsf{IKOS}} + \epsilon_{\mathsf{C}}$ [IKOS09, Thm. 3.3].*[30]

Next, we analyze the RBR soundness guarantee of the IOP. Recall that the RBR soundness is with respect to a doomed set of (partial) transcripts. The following definition of "bad" randomness for the MPC protocol will be useful.

Definition B.9 (Bad Randomness for MPC Protocols)**.** *Let Π_{MPC} be a k-party protocol for a k-party function f. We say that $r = (r_1, \ldots, r_k)$ is* bad for inputs $y_1, \ldots, y_k$ with respect to f *if, in an honest execution of Π_{MPC} on inputs $y_1, \ldots, y_k$ in which the parties use the randomness $r_1, \ldots, r_l$, there exists a party P_i whose output is not $f(y_1, \ldots, y_k)$.*

Using the notion of bad randomness from Definition B.9, we define the doomed set $\mathcal{D}_{\mathsf{B}}$ as follows:

Definition B.10 (Doomed Set for Base IOP)**.** *The Doomed set $\mathcal{D}_{\mathsf{B}}$ for the base IOP of Figure 4 is defined as follows.*

- *If $x \notin \mathcal{L}_{\mathcal{R}}$ then $(x, \emptyset) \in \mathcal{D}_{\mathsf{B}}$, where $\emptyset$ is the empty transcript.*
- *If $\mathcal{T} \in \mathsf{PartTr}(1)$, i.e., $\mathcal{T} = \left(\left(w_i, r_i^P\right)_{i\in[k]}, \left(r_i^V\right)_{i\in[k]}\right)$, then $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$ if and only if $(x, w_1 \oplus \ldots \oplus w_k) \notin \mathcal{R}$, and $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is not bad for $(w_1, \ldots, w_k)$ with respect to f_x (as in Definition B.9).*
- *If $\mathcal{T} \in \mathsf{PartTr}(2)$ then $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$ if and only if $\mathcal{V}_{\mathsf{B}}(x, \mathcal{T}) = 0$.*

We now prove that the base IOP of Figure 4 has RBR soundness with respect to the doomed set $\mathcal{D}_{\mathsf{B}}$ of Definition B.10.

Lemma B.11 (RBR Soundness)**.** *If Π_{MPC} is an ϵ_{C}-correct protocol for $\mathcal{R}$ then the IOP of Figure 4 has RBR soundness with error $\varepsilon_{\mathsf{IKOS}} + \epsilon_{\mathsf{C}}$ (cf. Remark B.8).*

Proof: We prove that the IOP satisfies the properties of Definition 4.16 with respect to the doomed set $\mathcal{D}_{\mathsf{B}}$ of Definition B.10. Let $\mathcal{T}$ be a (partial) transcript in an execution of the IOP when the verifier has input x.

1. If $x \notin \mathcal{L}_{\mathcal{R}}$ then $(x, \emptyset) \in \mathcal{D}_{\mathsf{B}}$ by definition.
2. If $\mathcal{T} \in \mathsf{PartTr}(2)$ and $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$ then $\mathcal{V}_{\mathsf{B}}(x, \mathcal{T}) = 0$ by definition.

[30] We note that [IKOS09, Thm. 3.3] is stated for protocols with a negligible correctness error, but the proof extends also to the case of a non-negligible ϵ_{C}.

3. If $\mathcal{T} \in \mathsf{PartTr}(0)$, i.e., $\mathcal{T} = \emptyset$, and $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$, then $x \notin \mathcal{L}_{\mathcal{R}}$ by definition of $\mathcal{D}_{\mathsf{B}}$. In this case, for every first prover message $\pi_1 = \left(w_i, r_i^P\right)_{i\in[k]}$, we have $(x, w_1 \oplus \ldots \oplus w_k) \notin \mathcal{R}$. Therefore, $(x, \mathcal{T}, \pi_1, (r_1^V, \ldots, r_k^V))$ will *not* be doomed, only if $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ *is* bad for $w_1, \ldots, w_k$ with respect to f_x. By the definition of bad randomness (Definition B.9), in this case when Π_{MPC} is executed with inputs $w_1, \ldots, w_k$ and randomness $r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V$, there exists a party whose output is incorrect. The ϵ_{C}-correctness of Π_{MPC} guarantees that this happens only with probability ϵ_{C} over the choice of $r_1^V, \ldots, r_k^V$, since $r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V$ are uniformly random when $r_1^V, \ldots, r_k^V$ are. By the definition of $\mathcal{D}_{\mathsf{B}}$, we can conclude that:

$$\Pr_{(r_1^V,\ldots,r_k^V)}\left[(x, \mathcal{T}, \pi_1, (r_1^V, \ldots, r_k^V)) \notin \mathcal{D}_{\mathsf{B}}\right] \leq \epsilon_{\mathsf{C}} \leq \varepsilon_{\mathsf{IKOS}} + \epsilon_{\mathsf{C}}$$

4. If $\mathcal{T} \in \mathsf{PartTr}(1)$, i.e., $\mathcal{T} = \left(\left((w_i, r_i^P)\right)_{i\in[k]}, (r_i^V)_{i\in[k]}\right)$, and $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$, then by the definition of $\mathcal{D}_{\mathsf{B}}$ we have: (1) $(x, w_1 \oplus \ldots \oplus w_k) \notin \mathcal{R}$, and (2) $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is not bad for $w_1, \ldots, w_k$ with respect to f_x. Let $\pi_2 = (\mathsf{View}_1, \ldots, \mathsf{View}_k)$ be a round-2 prover message. We need to show that

$$\Pr_{i,j}\left[\mathcal{V}_{\mathsf{B}}(x, \mathcal{T}, \pi_2, (i, j)) = 1\right] \leq \varepsilon_{\mathsf{IKOS}} + \epsilon_{\mathsf{C}}$$

since the full transcript is doomed if and only if it causes $\mathcal{V}_{\mathsf{B}}$ to reject. Since $(x, w_1 \oplus \ldots \oplus w_k) \notin \mathcal{R}$, then $f_x(w_1, \ldots, w_k) = 0$. Since $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is not bad for $w_1, \ldots, w_k$ with respect to f_x, then in an honest execution of Π_{MPC} on $w_1, \ldots, w_k$ with randomness $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$, all parties output 0. We consider two possible cases, depending on the structure of π_2. First, if all views in π_2 are pairwise consistent, and consistent with $w_1, \ldots, w_k$ and $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$, then all parties output 0 in Π_{MPC} (because $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is not bad) and so $\mathcal{V}_{\mathsf{B}}$ rejects in this case with probability 1. Otherwise, there exists a pair of views $\mathsf{View}_i, \mathsf{View}_j$ that are either inconsistent with each other, or WLOG View_i is inconsistent with w_i or $r_i^P \oplus r_i^V$. In this case, [IKOS09, Thm. 3.3] shows that the verifier rejects except with probability $\varepsilon_{\mathsf{IKOS}} + \epsilon_{\mathsf{C}}$. Overall, in this case the verifier accepts with probability at most $\varepsilon_{\mathsf{IKOS}} + \epsilon_{\mathsf{C}}$.

■

Lemma B.12 (Knowledge Extraction). *If Π_{MPC} is an ϵ_{C}-correct protocol for $\mathcal{R}$ then the IOP of Figure 4 has knowledge extraction with error ϵ_{C}.*

Proof: The extractor E on input $x, \pi_1 = \left(w_i, r_i^P\right)_{i\in[k]}$ outputs $w := w_1 \oplus \ldots \oplus w_k$. (We can assume π_1 has this structure, otherwise $\mathcal{V}_{\mathsf{B}}$ will reject.) Then E is deterministic and efficient. It remains to show that if $(x, w) \notin \mathcal{R}$ then

$$\Pr_{r^V=(r_1^V,\ldots,r_k^V)}\left[(x, \pi_1, (r_1^V, \ldots, r_k^V)) \notin \mathcal{D}_{\mathsf{B}}\right] \leq \varepsilon.$$

Recall from Definition B.10 that $(x, \pi_1, r_V) \in \mathcal{D}_{\mathsf{B}}$ if and only if $(x, w_1 \oplus \ldots \oplus w_k) \notin \mathcal{R}$, and $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is not bad for $(w_1, \ldots, w_k)$ with respect to f_x (as in Definition B.9). Since E outputs $w_1 \oplus \ldots \oplus w_k$, and we have assumed that it fails to extract a valid witness, then $(x, \pi_1, r_V) \notin \mathcal{D}_{\mathsf{B}}$ implies that $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is bad for $(w_1, \ldots, w_k)$ with respect to f_x. By the definition of bad randomness, this means that in an execution of Π_{MPC} with inputs $w_1, \ldots, w_k$ and random inputs $r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V$, there exists a party P_i whose output is not $f_x(w_1, \ldots, w_k)$. By the ϵ_{C}-correctness of Π_{MPC}, the probability over random $r_1^V, \ldots, r_k^V$ that $r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V$ is bad, is upper bounded by ϵ_{C} (because in this case $r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V$ are uniformly random). ■

Lemma B.13 (Doom Checkability). *The IOP of Figure 4 has doom checkability with respect to the doomed set $\mathcal{D}_{\mathsf{B}}$ of Definition B.10.*

Proof: We describe a polynomial-sized circuit C that given x and a non-empty transcript $\mathcal{T}$, outputs 1 if and only if $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$. C on input $(x, \mathcal{T})$ operates as follows:

- If $\mathcal{T} \in \mathsf{PartTr}(1)$ then $\mathcal{T} = \left(\left(w_i, r_i^P\right)_{i\in[k]}, \left(r_i^V\right)_{i\in[k]} \right)$. C emulates Π_{MPC}, using w_i and $r_i^P \oplus r_i^V$ as the input and randomness of P_i, respectively. It outputs 1 if and only if $(x, w_1 \oplus \ldots \oplus w_k) \notin \mathcal{R}$ and all parties P_i output 0 in the emulation.
- If $\mathcal{T} \in \mathsf{PartTr}(2)$, then C outputs $1 - \mathcal{V}_{\mathsf{B}}(x, \mathcal{T})$.

Clearly, C has polynomial size. We now argue that its output is correct, by considering two possible cases. First, if $\mathcal{T} \in \mathsf{PartTr}(2)$ then by the definition of $\mathcal{D}_{\mathsf{B}}$, $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$ if and only if $\mathcal{V}_{\mathsf{B}}(x, \mathcal{T}) = 0$ which happens if and only if C outputs 1. Second, if $\mathcal{T} \in \mathsf{PartTr}(1)$ then by the definition of $\mathcal{D}_{\mathsf{B}}$, $(x, \mathcal{T}) \in \mathcal{D}_{\mathsf{B}}$ if and only if (i) $(x, w_1 \oplus \ldots \oplus w_k) \notin \mathcal{R}$ (which C checks directly), and (ii) $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is not bad for $(w_1, \ldots, w_k)$ with respect to f_x. The latter implies that all parties output $f_x(w_1, \ldots, w_k) = \mathcal{R}(x, w_1 \oplus \ldots \oplus w_k) = 0$ in an execution of Π_{MPC} in which P_i has input w_i and uses randomness $r_i^P \oplus r_i^V$. Since C outputs 1 only if all parties P_i outputs $0 = f_x(w_1, \ldots, w_k)$ in the emulation, this implies that C outputs 1 only when $(r_1^P \oplus r_1^V, \ldots, r_k^P \oplus r_k^V)$ is not bad. ■

Digest: Using MPCitH-Based IOPs to Construct NIZKs. The MPCitH paradigm of [IKOS07] was first introduced in the context of ZK proofs. However, when relying on *perfectly correct* MPC protocols, their results can be interpreted as giving a (semi-HV) ZK PCP (see [IKOS09, Section 5]). Their ZK proofs based on *statistically-correct* protocols ([IKOS09, Section 3.1]) can be interpreted as yielding a (semi-HV) ZK *IOP*; Figure 4 formally presents this interpretation. Our analysis of Figure 4 shows that the construction in fact possesses additional useful properties that were not considered by [IKOS09], specifically, RBR soundness, knowledge extraction, and doom checkability. These properties are instrumental to using the resultant semiHVZK-IOP to construct NIZKs. Ishai et al. [IKOS07] also construct ZK proofs based on *robust* MPC protocols. Roughly, these are protocols in which correctness holds even if a small subset of parties is *actively* trying to change the computation outcome by deviating from the protocol. (More accurately, the guarantee is that if there is no input that is (1) consistent with the honest parties' inputs, and (2) causes the function to output 1, then all honest parties output 0 even if the malicious parties deviate from the protocol specification.) Such *perfectly* robust protocols were recently used by [GLS22] to design (semi-HV) ZK PCPs which they then use to construct NIZKs. Relying on perfect correctness/robustness significantly simplifies the analysis of knowledge extraction, RBR soundness, and doom checkability. Intuitively, this is because the first prover message fully determines the entire protocol execution, including the (possibly malicious) behavior of the corrupted parties. Combining the construction of Figure 4 with the results of [GLS22] shows that applying the MPCitH paradigm to either *perfectly robust* or *statistically correct* MPC protocols yields PCPs/IOPs that can be used to construct NIZKs by combining Fiat-Shamir with CIHs. We show in Appendix B.3 below that relying on statistically-correct protocols can have efficiency advantages compared to using perfectly-robust protocols. (See Remark B.14 for further discussion and comparison with [GLS22].)

It is natural to ask whether one can use *statistically robust* protocols to construct NIZK in this way, similar to how they were used to construct ZK proofs in [IKOS07]. The obstacle to using such protocols in the NIZK context lies in proving *doom-checkability* (or, more generally, efficient product verifiability). In more detail, The MPCitH paradigm is applicable to statistically robust MPC protocols in which the execution is divided into two phases: Phase 1 and Phase 2, as follows. Following Phase 1 the parties obtain a public uniformly random string r which is used in Phase 2.

Roughly, robustness of the protocol is guaranteed so long as r is unpredictable in Phase 1.[31] The IOP induced by such 2-phase protocols differs slightly from the one described in Figure 4. Specifically, the prover chooses the entire random strings $r_1, \ldots, r_k$ for all parties in Step 1, and the verifier chooses r in Step 2. The prover then generates the views using the inputs $w_1, \ldots, w_k$ and random strings $r_1, \ldots, r_k$ he sent in the first round, and the verifier-chosen r. The doomed set can then be defined similarly to Definition B.10, with respect to a modified notion of "bad randomness" where $(r_1, \ldots, r_k, \mathsf{r})$ is considered bad for $y_1, \ldots, y_k$ if there exists an execution with $\leq t$ corrupted parties where the honest parties use $r_1, \ldots, r_k, \mathsf{r}$, in which some honest party's output is not $f(y_1, \ldots, y_k)$. The issue is that doom-checkability does not seem to hold with respect to this doomed set. Specifically, to check whether $(r_1, \ldots, r_k, \mathsf{r})$ is bad, we need to consider all executions, with all (possibly malicious and unbounded) adversaries $\mathcal{A}$. It is not clear how one can *efficiently* check whether there exists such an execution (that causes an honest party to have incorrect output), since we cannot even efficiently iterate all possible malicious strategies (the adversary may be unbounded). This issue does not arise when using statistically *correct* protocols, since there the inputs and randomness fully determine the execution, and this suffices for doom-checkability, as we show in the proof of Lemma B.13.

Remark B.14 (Comparison with [GLS22])**.** *As discussed above, Ghosal, Lou, and Sahai [GLS22] recently used MPCitH to design semi-HVZK PCPs which they then use to construct NIZKs via the "Fiat-Shamir + CIH" paradigm. Although both works use MPCitH, our construction differs significantly from theirs:*

- *[GLS22] rely on perfectly-robust protocols, whereas we use statistically-correct protocols. Thus, they obtain a PCP, whereas we obtain an IOP. We note that proving doom checkability (more generally, efficient product verifiability) for a PCP is significantly easier than for IOPs. Indeed, in a PCP the first (and only) prover message fully determines the entire execution, and so one can use the next message function of the MPC protocol to easily check membership in the doomed set/bad relation.*

- *The IOP used in our NIZK construction is a parallel-repetition version of a base-IOP constructed from an MPC protocol with a small (poly-logarithmic) number of parties (see Lemma B.17 below), whereas [GLS22] do not employ parallel repetition, instead using a protocol with $\tilde{O}(\kappa)$ parties. Parallel repetition is used in our construction to get a bad relation that is a product relation, and to reduce the soundness error (which is quite large when using correct-only protocols). In contrast, the PCP of [GLS22] has a negligible soundness error with no amplification (using robust protocols in MPCitH yields a negligible soundness error). Moreover, they obtain a product relation without repetition using a variant of MPC where the verifier compares a view with part of the communication transcript (i.e., all messages exchanged in the protocol), instead of comparing pairs of views. This yields a bad relation that is naturally a product relation of sufficient arity (the arity in [GLS22] corresponds to the number of MPC parties). Such a strategy would not work for us, since we need to keep the number of parties small to obtain the desired efficiency advantages in our NIZKs of Corollary 7.8. We note that the product relation of [GLS22] has better arity compared to ours', specifically, in [GLS22] the product is over $\tilde{O}(\kappa)$ sets, whereas in our construction it is over $O(\kappa^4)$ sets.*

- *Using their PCPs, [GLS22, Thm. 1.3] construct NIZKs for NP from LWE, whose length is roughly $O(|C|)$, where C is the verification circuit of the relation. Using FHE, they also obtain NIZKs for NP*

[31] The reason 2-phase protocols are needed is because soundness of the IOP does not follow immediately from standard robustness. Indeed, in the MPC setting, the adversary sees only the random strings of corrupted parties, whereas in the context of MPC in the Head, a malicious prover knows the random strings used by all parties (even if only some of them behave maliciously in the underlying MPC). This is not a problem when the protocol is perfectly robust, since even though the prover knows all random strings, there does not exist a "bad" choice of randomness which he can exploit to cheat.

with length $|w| + \mathsf{poly}(\kappa)$, where w is the NP witness. Corollary 7.8 can be interpreted as a scaling-down of this result: it shows that by scaling down the class of relations (from NP to $\mathsf{AC}^0[\oplus]$), we can obtain NIZKs whose proof approaches the witness length assuming only (leveled) LWE.

We further note that our goals and results differ quite significantly from [GLS22]. Specifically, our goal is to show when the FS+CIH paradigm can be applied to IOPs instead of PCPs. *We show this is possible provided the IOP satisfies some mild additional properties (midly-small RBR soundness error, knowledge extraction, doom checkability). We then use MPCitH to obtain a specific instantiation of an IOP satisfying these properties. In contrast, [GLS22] focuses on making the standard construction,* based on PCPs, *more efficient by giving an MPCitH-based PCP and designing a more efficient CIH for its bad relation. While both works use MPCitH as a tool, the main goal and technical contributions are focused on obtaining more efficient/general building blocks by relying on either* recurrent *list-recoverable codes (this enables constructing CIHs more efficiently than in [HLR21], which rely on standard list-recoverable codes) in [GLS22], or IOPs in the current work.*

B.3 Instantiating the semi-HVZK-IOP

We now instantiate the IOP construction of Figure 4 with a specific MPC protocol. The IOP obtained in this manner will serve as the base-IOP in our interactive (and non-interactive) ZK arguments.

We use an MPC protocol for $\mathsf{AC}^0[\oplus]$ circuits due to Barkol and Ishai [BI05], in which the total communication and randomness complexity is quasi-linear in the input length. We first formally defined the complexity class $\mathsf{AC}^0[\oplus]$.

Definition B.15. *$\mathsf{AC}^0[\oplus]$ is the complexity class of all circuits of polynomial size and constant depth over $\wedge, \vee, \neg, \oplus$ gates of unbounded fan-in.*

Theorem B.16 (MPC protocols for $\mathsf{AC}^0[\oplus]$ [BI05])**.** *Let λ be a statistical security parameter, and let $C \in \mathsf{AC}^0[\oplus]$ be a circuit of size s with input length m. Then there exists a $k = \mathsf{poly}\log(s)$ such that for any partition of C's inputs between k parties, there exists a negligible function $\epsilon(\lambda)$, and a k-party MPC protocol Π_{MPC} for $C(x_1, \ldots, x_m)$ such that Π_{MPC} is correct with error $\epsilon(\lambda)$ and statistically 2-private with error $\epsilon(\lambda)$. Moreover, the communication complexity and randomness complexity of Π_{MPC} are at most $m \cdot \lambda \cdot \mathsf{poly}\log(s)$.*

Lemma B.17. *Let $\mathcal{R}$ be an NP-relation verifiable in $\mathsf{AC}^0[\oplus]$. Then there exists a 2-round public-coin IOP system $(\mathcal{P}_{\mathsf{B}}, \mathcal{V}_{\mathsf{B}})$ with statistical semi-HVZK, completeness and knowledge extraction with $\mathsf{negl}(\lambda)$ errors, soundness error 1/8, RBR soundness error 1/8 with respect to the doomed set of Definition B.10, and doom checkability. Moreover, the proof length is $m \cdot \lambda \cdot \mathsf{poly}\log(s)$, where m is the witness length, λ is a statistical security parameter, and $s = \mathsf{poly}(\lambda)$ is the size of $\mathcal{R}$'s verification circuit.*

Proof: Let C be the verification circuit of $\mathcal{R}$, and let $s := |C|$. We instantiate the IOP system $(\mathcal{P}_B, \mathcal{V}_B)$ of Figure 4 with the MPC protocol Π_{MPC} of Theorem B.16. Then the IOP has statistical semi-HVZK by Lemma B.7, completeness with negligible $\epsilon(\lambda)$ error by Lemma B.6, doom checkability with respect to the doomed set of Definition B.10 by Lemma B.13, knowledge extraction with negligible error $\epsilon(\lambda)$ by Lemma B.12, soundness error $\epsilon(\lambda) + \varepsilon_{\mathsf{IKOS}}$ by Remark B.8, and RBR soundness with error $\epsilon_B := 1 - 1/\binom{\mathsf{poly}\log s}{2} + \epsilon(\lambda) = 1 - 1/\binom{\mathsf{poly}\log s}{2}$ by Lemma B.11 (for a sufficiently large exponent in the $\mathsf{poly}\log(s)$ term on the RHS), i.e., the RBR soundness error ϵ_B satisfies $\epsilon_B = 1 - \frac{1}{\log^c s}$ for some constant $c \in \mathbb{N}$. Moreover, since the proof consists of the views of the MPC parties, which consist of their randomness and the exchanged messages, the total proof length is $m \cdot \lambda \cdot \mathsf{poly}\log(s)$, where m is the witness length.

Next, let $\alpha := 3$, and notice that $\exp(-\alpha) < 1/8$. Let $(\mathcal{P}, \mathcal{V})$ be the IOP obtained by applying Theorem 5.7 to $(\mathcal{P}_B, \mathcal{V}_B)$ with $\delta = \frac{1-\epsilon_B}{2} = \frac{1}{2\log^c s}$ and number of repetitions $t = \lceil 2\alpha \cdot \log^{2c} s \rceil = \lceil \frac{\alpha}{2\delta^2} \rceil = \mathsf{poly}\log(s)$. Then $(\mathcal{P}, \mathcal{V})$ has doom checkability (because $(\mathcal{P}_B, \mathcal{V}_B)$ has doom checkability,

see Remark 7.3), and Theorem 5.7 guarantees that $(\mathcal{P},\mathcal{V})$ has completeness error $t\cdot\epsilon(\lambda)=\mathsf{negl}(\lambda)$, statistical semi-HVZK, and soundness error

$$\epsilon_B^t=\left(1-\frac{1}{\log^c s}\right)^{\lceil 2\alpha\cdot\log^{2c}s\rceil}\leq\left(1-\frac{1}{\log^c s}\right)^{2\alpha\cdot\log^{2c}s}=\left(\left(1-\frac{1}{\log^c s}\right)^{\log^c s}\right)^{2\alpha\cdot\log^c s}$$
$$\leq\left(\exp\left(-1\right)\right)^{2\alpha\log^c s}=\left(\exp\left(-\alpha\right)\right)^{2\log^c s}\leq\exp\left(-\alpha\right)\leq\frac{1}{8}.$$

Moreover, the system has RBR soundness error $\varepsilon^{\otimes}:=\max\{\exp\left(-2\delta^2 t\right),\epsilon_B^{(1-(\epsilon_B+\delta))t}\}$ with respect to the doomed set of Definition 5.6, and $\epsilon^{\otimes}\leq 1/8$ because it is the maximum over

$$\exp\left(-2\delta^2 t\right)=\exp\left(-2\delta^2\cdot\lceil\frac{\alpha}{2\delta^2}\rceil\right)\leq\exp\left(-2\delta^2\cdot\frac{\alpha}{2\delta^2}\right)=\exp\left(-\alpha\right)\leq\frac{1}{8}$$

and

$$\epsilon_B^{(1-(\epsilon_B+\delta))t}=^{(1)}\epsilon_B^{(1-\epsilon_B)\cdot\frac{t}{2}}=^{(2)}\left(1-\frac{1}{\log^c s}\right)^{\frac{1}{\log^c s}\cdot\frac{t}{2}}=\left(\left(1-\frac{1}{\log^c s}\right)^{\log^c s}\right)^{\frac{t}{2\cdot\log^{2c}s}}$$
$$\leq\exp\left(-\frac{t}{2\cdot\log^{2c}s}\right)=^{(3)}\exp\left(-\frac{\lceil 2\alpha\cdot\log^{2c}s\rceil}{2\cdot\log^{2c}s}\right)\leq\exp\left(-\frac{2\alpha\cdot\log^{2c}s}{2\cdot\log^{2c}s}\right)=\exp\left(-\alpha\right)\leq\frac{1}{8}$$

where the equality denoted by (1) follows from our choice $\delta=\frac{1-\epsilon_B}{2}$, the equality denoted by (2) follows from the fact that $\epsilon_B=1-1/\log^c s$, and the equality denoted by (3) follows from our choice $t=\lceil 2\alpha\cdot\log^{2c}s\rceil$. Furthermore, Theorem 5.7 guarantees that $(\mathcal{P},\mathcal{V})$ has knowledge extraction with error

$$\exp\left(-2\left(\varepsilon_B+\delta-\varepsilon(\lambda)\right)^2 t_\kappa\right)=^{(1)}\exp\left(-2\left(\varepsilon_B+\frac{1-\varepsilon_B}{2}-\varepsilon(\lambda)\right)^2 t_\kappa\right)=\exp\left(-2\left(\frac{1}{2}+\frac{\varepsilon_B}{2}-\varepsilon(\lambda)\right)^2 t_\kappa\right)$$
$$\leq^{(2)}\exp\left(-2\left(\frac{1}{4}+\frac{\varepsilon_B}{2}\right)^2 t_\kappa\right)\leq^{(3)}\exp\left(-\frac{t_\kappa}{2}\right)=^{(4)}\exp\left(-\frac{\lceil 2\alpha\cdot\log^{2c}s\rceil}{2}\right)\leq\exp\left(-\frac{2\alpha\cdot\log^{2c}s}{2}\right)$$
$$\leq\exp\left(-\alpha\right)\leq\frac{1}{8}$$

where the equality denoted by (1) follows from our choice $\delta=\frac{1-\epsilon_B}{2}$, the inequality denoted by (2) follows from the fact that $\epsilon_B<1$ (so $\frac{1}{4}+\epsilon_B 2<\frac{1}{2}$), the inequality denoted by (3) holds for a sufficiently large λ, and the equality denoted by (4) follows from our choice $t=\lceil 2\alpha\cdot\log^{2c}s\rceil$. Finally, the execution consists of $t=\mathsf{poly}\log s$ executions of $(\mathcal{P}_B,\mathcal{V}_B)$, each with proof length $m\cdot\lambda\cdot\mathsf{poly}\log(s)$, and so the total proof length is $m\cdot\lambda\cdot\mathsf{poly}\log(s)$. ■

C Alternative Bad Relation Definitions

In this section, we discuss several alternative definitions of the $\mathcal{R}_{\mathsf{BAD}}^{\geq 1}$ component of the bad relation of Section 7.1. Recall that we prove the existence of a CIH for the bad relation of Section 7.1 by relying on the results of [HLR21]. For this, the bad relation should be: (i) an (approximate) product relation, (ii) sparse, and (iii) efficient product verifiable. Defining a bad relation which satisfies these three properties, and is also useful in the NIZK soundness reduction, is non trivial. We illustrate this by discussing several alternative definitions of the bad relation, each of which fails to achieve at least one of these properties, or is insufficient to claim NIZK soundness.

At a high level, a "witness" for an instance in the bad relation is a bad verifier challenge that takes a partial transcript $\mathcal{T}$ of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ in a doomed state into an extended transcript (i.e., the partial transcript $\mathcal{T}$ together with the bad challenge and the prover's response to it) that is not doomed. Formalizing this intuition necessitates in particular defining the doomed set $\mathcal{D}^{\otimes}$ for $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ which, as discussed in Section 2.3, can be defined in several was. We discuss several different "natural" bad relations that arise from different definitions of the doomed set $\mathcal{D}^{\otimes}$, and their limitations.

1. *OR doomed set.* The bad relation consists of round-$(i+1)$ partial transcripts (for some i) such that the round i transcript of at least one execution of the base IOP was doomed, and all the round-i partial transcripts of the base-IOP are not doomed. Formally:

$$\mathcal{R}_1 = \left\{ ((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}) \ : \ \begin{array}{c} i \in [l], \mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i) : \mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}, \\ \mathsf{c}^j = (\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\ell), \{\Pi^j(s) = \mathsf{EE.Ext}(\tau^j_s, \mathsf{c}^j_s)\}_{j\in[i+1], s\in[\ell]}, \\ \Pi^j = \Pi^j(1) \mid \ldots \mid \Pi^j(\ell), \mathcal{T} = (\Pi^j, r^j)_{j\in[i]} \in \mathsf{PartTr}^{\otimes}(i), \\ \forall j \in [i+1], \Pi^j = (\Pi^j_1, \ldots, \Pi^j_{t_\kappa}) \ \wedge \ r^j = (r^j_1, \ldots, r^j_{t_\kappa}) \\ \exists k \in [t_\kappa], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k) \in \mathcal{D}_{\mathsf{IOP}} \\ \forall k \in [t_\kappa], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k, \Pi^{i+1}_k, r^{i+1}_k) \notin \mathcal{D}_{\mathsf{IOP}} \end{array} \right\}.$$

   (where $\mathsf{PartTr}^{\otimes}(i), \mathsf{PartTr}(i)$ are the sets of round-i partial transcript of the parallel-repetition IOP and base IOP, respectively).

   The above bad relation is induced by the following definition of the doomed set:

$$\begin{aligned} (x, \mathcal{T}) \in \mathcal{D}^{\otimes} \text{ for } \mathcal{T} \in \mathsf{PartTr}^{\otimes}(i) \\ \Leftrightarrow \mathcal{T} = (\mathcal{T}_1 \ldots, \mathcal{T}_{t_\kappa}) \text{ and } \forall j \in [t_\kappa], \mathcal{T}_j \in \mathsf{PartTr}_{\mathsf{IOP}}(i) \\ \text{and } \exists k \in [t], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k) \in \mathcal{D}_{\mathsf{IOP}} \end{aligned}$$

   This bad relation $\mathcal{R}_1$ can be used in the NIZK soundness reduction. Indeed, for the prover to convince the verifier to accept an $x \notin \mathcal{L}$, the final transcript of the parallel-repetition must be non-doomed, which by the above definition of the doomed set means *all* the induced base-IOP transcripts are non-doomed. Since all these base-IOP transcripts start out doomed (because $x \in \mathcal{L}$) then there must be a round i in which the *last* base-IOP transcript moved from doomed to not doomed. The transcript of this round will be in $\mathcal{R}_1$ and can be used to reduce a NIZK prover that violates sound $\mathcal{P}^*$ to a CI-breaking adversary.

   Unfortunately, while this relation is a product relation, it is not sparse. This is because RBR soundness of the underlying base-IOP does not guarantee anything about a transcript staying non-doomed, namely, once a transcript becomes non-doomed, there is no upper bound on the number of "bad" verifier challenges. More specifically, the problem is with executions taking the partial transcripts of the base-IOP from doomed to not doomed *one at a time* (or, more generally, few in each round). For example, consider an execution in which *all but the first* base-IOP partial transcript move from doomed to not doomed in the second round. In this case, the round-2 partial transcript *of the parallel repetition* is still doomed. However, potentially *all* verifier challenges in repetitions $2, \ldots, t_\kappa$ of the base-IOP might be "bad" (this is indeed the case in, e.g., standard sumcheck IOPs). This discussion also suggest a cheating NIZK-prover strategy that violates soundness without giving rise to a CI-violating attack: simply move the underlying base-IOP transcripts from doomed to non-doomed one (or few) at a

time. Intuitively, our use of a dynamic threshold in the doomed set definition rules out such strategies. (Roughly, to move the parallel-repetition transcript from doomed to non-doomed there must be a round in which $\mathcal{P}^*$ moves "too many" base-IOP transcripts from doomed to non-doomed, where the definition of "too many" varies depending on the round number.)

2. *AND doomed set.* The bad relation consists of round-$(i+1)$ partial transcripts (for some i) such that the round i transcripts of *all* base-IOP repetitions were doomed, but at least one of the round-$(i+1)$ transcripts of the base-IOP is not doomed. Formally:

$$\mathcal{R}_2 = \left\{ ((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}) \ : \ \begin{array}{c} i \in [l], \mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i) : \mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}, \\ \mathsf{c}^j = (\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\ell), \{\Pi^j(s) = \mathsf{EE.Ext}(\tau^j_s, \mathsf{c}^j_s)\}_{j\in[i+1], s\in[\ell]}, \\ \Pi^j = \Pi^j(1) \mid \ldots \mid \Pi^j(\ell), \mathcal{T} = (\Pi^j, r^j)_{j\in[i]} \in \mathsf{PartTr}^{\otimes}(i), \\ \forall j \in [i+1], \Pi^j = (\Pi^j_1, \ldots, \Pi^j_{t_\kappa}) \ \wedge \ r^j = (r^j_1, \ldots, r^j_{t_\kappa}) \\ \forall k \in [t_\kappa], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k) \in \mathcal{D}_{\mathsf{IOP}} \\ \exists k \in [t_\kappa], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k, \Pi^{i+1}_k, r^{i+1}_k) \notin \mathcal{D}_{\mathsf{IOP}} \end{array} \right\}.$$

The above bad relation is induced by the following definition of doomed set.

$$\begin{aligned} (x, \mathcal{T}) \in \mathcal{D}^{\otimes} \text{ for } \mathcal{T} \in \mathsf{PartTr}^{\otimes}(i) \\ \Leftrightarrow \mathcal{T} = (\mathcal{T}_1 \ldots, \mathcal{T}_{t_\kappa}) \text{ and } \forall j \in [t_\kappa], \mathcal{T}_j \in \mathsf{PartTr}_{\mathsf{IOP}}(i) \\ \text{and } \forall k \in [t], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k) \in \mathcal{D}_{\mathsf{IOP}} \end{aligned}$$

This relation is not a product relation. Indeed, consider an execution in which the *first* base-IOP moves from doomed to not doomed in the second round with some randomness r^2_1. Then for every $(r^2_2, \ldots, r^2_{t_\kappa})$, we have that $(r^2_1, \ldots, r^2_{t_\kappa})$ should be a valid witness for $(x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1})$. Similarly, consider an execution in which the *second* base-IOP moves from doomed to not doomed in the second round with some randomness $\hat{r}^2_2$. Then for every $(\hat{r}^2_1, \hat{r}^2_3, \ldots, \hat{r}^2_{t_\kappa})$, we have that $(\hat{r}^2_1, \ldots, \hat{r}^2_{t_\kappa})$ should be a valid witness for $(x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1})$. But if $\mathcal{R}_2$ is a product relation, this implies that *any* $(\tilde{r}^2_1, \ldots, \tilde{r}^2_{t_\kappa})$ is a valid witness for $(x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1})$. The same argument can be applied to any of the t_κ base-IOP repetitions, so $\mathcal{R}_2$ is not even an approximate product relation. It is also not sparse, for similar reasons to the ones discussed for $\mathcal{R}_1$ above.

3. *Exact-count doomed set.* The bad relation consists of round-$(i+1)$ partial transcripts (for some i) such that at least m executions of the base IOP moved from doomed to not doomed in round-i. Formally:

$$\mathcal{R}_3 = \left\{ ((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}) \ : \ \begin{array}{c} i \in [l], \mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i) : \mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}, \\ \mathsf{c}^j = (\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\ell), \{\Pi^j(s) = \mathsf{EE.Ext}(\tau^j_s, \mathsf{c}^j_s)\}_{j\in[i+1], s\in[\ell]}, \\ \Pi^j = \Pi^j(1) \mid \ldots \mid \Pi^j(\ell), \mathcal{T} = \{\Pi^j, r^j\}_{j\in[i]}, \\ \exists \Gamma \subseteq [t_\kappa] \text{ s.t. } |\Gamma| \geq m, \\ \forall k \in \Gamma, (x, \mathcal{T}_k) \in \mathcal{D}_{\mathsf{IOP}} \wedge (x, \mathcal{T}_k \mid \Pi^{i+1}_k \mid r^{i+1}_k) \notin \mathcal{D}_{\mathsf{IOP}} \end{array} \right\}.$$

This relation is an approximate product relation, but it is not sparse, for reasons similar to $\mathcal{R}_2$.

4. The bad relation consists of transcripts in which *all* executions of the base-IOP move from

doomed to not-doomed *in the same round*. That is:

$$\mathcal{R}_4 = \left\{ ((x, i, \mathcal{T}_{\mathsf{CZK}}, \mathsf{c}^{i+1}); r^{i+1}) \ : \ \begin{array}{c} i \in [l], \mathcal{T}_{\mathsf{CZK}} \in \mathsf{PartTr}_{\mathsf{CZK}}(i) : \mathcal{T}_{\mathsf{CZK}} = \{\mathsf{c}^j, r^j\}_{j\in[i]}, \\ \mathsf{c}^j = (\mathsf{c}^j_1, \ldots, \mathsf{c}^j_\ell), \{\Pi^j(s) = \mathsf{EE.Ext}(\tau^j_s, \mathsf{c}^j_s)\}_{j\in[i+1], s\in[\ell]}, \\ \Pi^j = \Pi^j(1) \,|\, \ldots \,|\, \Pi^j(\ell), \mathcal{T} = (\Pi^j, r^j)_{j\in[i]} \in \mathsf{PartTr}^{\otimes}(i), \\ \forall j \in [i+1], \Pi^j = (\Pi^j_1, \ldots, \Pi^j_{t_\kappa}) \ \wedge \ r^j = (r^j_1, \ldots, r^j_{t_\kappa}) \\ \forall k \in [t_\kappa], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k) \in \mathcal{D}_{\mathsf{IOP}} \\ \forall k \in [t_\kappa], (x, \Pi^1_k, r^1_k, \ldots, \Pi^i_k, r^i_k, \Pi^{i+1}_k, r^{i+1}_k) \notin \mathcal{D}_{\mathsf{IOP}} \end{array} \right\}.$$

This relation is a product relation, and is sparse. However, it is not clear how to use it in the NIZK soundness reduction. Indeed, in the modified bad relation $\mathcal{R}_4$ described above, we ask that all underlying transcripts of the parallel repetition move from doomed to non-doomed at the same round. However, there could exist executions in which the verifier challenges take the underlying transcripts of the parallel-repetition IOP from doomed to non-doomed *one at a time in each round* (or, more generally, few at a time), resulting in an accepting transcript for a false statement. Such executions will violate NIZK-soundness without violating CI (since in no round will the transcript belong to the bad relation). That is, there might exist verifier challenges that violate soundness and can still be found by an efficient adversary, since they do not belong to the bad relation and so finding them efficiently does not violate CI. This could potentially be used to break NIZK soundness without violating CI of the CIH.

Our final bad relation is one that corresponds to a threshold doomed set as in [HLR21] where the number of base-IOPs that need to be doomed for a transcript of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ to be doomed depends on the round number. This gives sparsity by forcing many underlying base-IOP transcripts to flip from doomed to non-doomed in order for the transcript to become non-doomed. However, the "natural" bad relation – the entire transcript moving from doomed to non-doomed – with this threshold definition of a doomed set is still *not* an approximate product relation (since we cannot show the "only if" direction of Definition 4.21). Our final solution is to define a new bad relation ($\widehat{\mathsf{BAD}}$ on Page 48) that specifically asks for a threshold number of base-IOPs to flip from doomed to non-doomed. This will be an approximate product relation, and then we argue that the natural bad relation (that takes the transcript of $(\mathcal{P}^{\otimes}, \mathcal{V}^{\otimes})$ from doomed to non-doomed) is a subset of the new bad relation. Thus, the bad relation of interest *is contained* in a sparse approximate product relation for which there exists a CIH.