Round-optimal Black-box Commit-and-prove with Succinct Communication

Susumu Kiyoshima

NTT Research susumu.kiyoshima@ntt-research.com

July 6, 2020

Abstract

We give a four-round black-box construction of a commit-and-prove protocol with succinct communication. Our construction is WI and has constant soundness error, and it can be upgraded into a one that is ZK and has negligible soundness error by relying on a round-preserving transformation of Khurana et al. (TCC 2018). Our construction is obtained by combining the MPC-in-the-head technique of Ishai et al. (SICOMP 2009) with the two-round succinct argument of Kalai et al. (STOC 2014), and the main technical novelty lies in the analysis of the soundness—we show that, although the succinct argument of Kalai et al. does not necessarily provide soundness for \mathcal{NP} statements, it can be used in the MPC-in-the-head technique for proving the consistency of committed MPC views. Our construction is based on sub-exponentially hard collision-resistant hash functions, two-round PIRs, and two-round OTs.

This article is a full version of the following article: Round-optimal Black-box Commit-and-prove with Succinct Communication, in Proceedings of CRYPTO 2020, ©IACR 2020. Parts of this work were done while the author was a member of NTT Secure Platform Laboratories.

Contents

1	Introduction 1.1 Our Result	4 5 6	
2	Preliminaries 2.1 Notations and Conventions 2.2 Witness-indistinguishable Commit-and-prove Protocols 2.3 Secure Multi-party Computation 2.4 Probabilistically Checkable Proofs (PCPs) 2.5 Definitions from Kalai et al. [KRR14] and Subsequent Works	7 7 7 8 9 9	
3	Outline of Proof of Theorem 1 3.1 Building Block: Perfect 2-private MPC protocol Π		
4	Overview of Step 1 (Non-WI Scheme with Soundness against CNS Well-behaving Provers) 4.1 Preliminary: Overview of Analysis of KRR No-signaling PCP 4.2 Protocol Description	11 11 12 12	
5	Overview of Step 2 (Non-WI Scheme with Soundness against CNS Provers)	17	
6	Overview of Subsequent Steps of Proof of Theorem 1	18	
A	On Verification Time	20	
В	On Definition of Commit-and-prove Protocols B.1 Differences from Definition in Khurana et al. [KOS18]	20 20 21	
C	Additional Preliminaries C.1 Oblivious Transfer Protocols C.2 Private Information Retrieval C.3 Computational No-signaling (Parallel Version) C.4 Low-degree Extension (LDE) C.5 Threshold Verifiers	21 21 21 22 22 22	
D	Step 0: PCP for Checking View Consistency D.1 Preliminaries: Results from Kalai et al. [KRR14] and Subsequent Works	22 22 25	
E	Step 1: Non-WI Scheme with (1 – negl)-Soundness against Well-behaving CNS Provers E.1 Protocol Description E.2 Proof of Binding E.3 Proof of Soundness	28 28 28 33	
F	Step 2: Non-WI Scheme with (1 – negl)-Soundness against CNS Provers F.1 Protocol Description F.2 Proof of Binding F.3 Proof of Soundness	39 40 40 42	
G	Step 3: Non-WI Scheme with negl-Soundness against CNS ProversG.1 Protocol DescriptionG.2 Proof of BindingG.3 Proof of Soundness	43 43 43 45	
Н	Step 4: Non-WI Scheme with Standard negl-SoundnessH.1 PreliminariesH.2 Protocol DescriptionH.3 Proof of BindingH.4 Proof of 2-PrivacyH.5 Proof of Soundness	46 46 47 47 49 50	

Ι	Step	5: WI Scheme with Standard $O(1)$ -Soundness	51	
	I.1	Protocol Description	51	
	I.2	Proof of Binding	51	
		Proof of Witness Indistinguishability		
		Proof of Soundness		
J	ZK Scheme with Standard negl-Soundness			
	J.1	Protocol Description	57	
		Proof Sketch of Binding		
		Proof Sketch of Zero-knowledge		
		Proof Sketch of Soundness		
K	Lemmas from Kalai et al. [KRR14] and Subsequent Works			
		Lemmas on SelfCorr	59	
		Soundness Amplification Lemma		

1 Introduction

In this paper, we obtain a new *commit-and-prove protocol* by relying on techniques in the area of *succinct arguments*. We start by giving some backgrounds.

Succinct arguments. Informally speaking, a *succinct argument* is an argument system with small communication complexity and fast verification time—typically, when a statement about T-time deterministic or non-deterministic computation is proven, the communication complexity and the verification time are required to be polylogarithmic in T. (The security requirements are, as usual, completeness and computational soundness.) Succinct arguments are useful when resources for communication and verification are limited; for example, a direct application of succinct arguments is *delegating computation* [GKR15] (or *verifiable computation* [GGP10]), where a computationally weak client delegates heavy computations to a powerful server and the client uses succinct arguments to verify the correctness of the server's computation efficiently. It was shown that a four-round succinct argument for all statements in \mathcal{NP} can be obtained from collision-resistance hash functions [Kil92]. Since then, succinct arguments have been actively studied, and protocols with various properties have been proposed.

Among existing succinct arguments, the most relevant to this work is the one by Kalai et al. [KRR14] (KRR succinct argument in short), which has several desirable properties such as (1) being *doubly efficient* [GKR15] (i.e., not only the verifier but also the prover is efficient), (2) being a two-round protocol (i.e., the scheme consists of a single query message from the verifier and a single answer message from the prover), and (3) being proven secure under standard assumptions, especially without relying on unfalsifiable assumptions and random oracles. More concretely, when the statement is about the correctness of a *T*-time computation, the communication complexity and the verifier running time is polylogarithmic in *T* while the prover running time is polynomial in *T*, and the security is proven assuming the existence of private information retrieval (PIR) or fully homomorphic encryption (FHE).

Given the powerful properties of KRR succinct argument, it is natural to expect that it has many cryptographic applications. For example, since argument systems have been extensively used in the design of cryptographic protocols, one might expect that the efficiency of such cryptographic protocols can be improved by simply plugging in KRR succinct argument.

However, using KRR succinct argument in cryptographic applications is actually non-trivial. One difficulty is that the soundness of KRR succinct argument is currently proven only for some specific types of \mathcal{NP} statements [KP16, BHK17, BKK+18] (originally, its soundness was proven for statements in \mathcal{P} [KRR14]). Another difficulty is that it does not provide any privacy on witnesses when it is used for \mathcal{NP} statements.

Nonetheless, recent works showed that KRR succinct argument can be used in some cryptographic applications. For example, by cleverly combining KRR succinct argument with other cryptographic primitives, Bitansky et al. [BBK+16] obtained a three-round zero-knowledge argument against uniform cheating provers, Brakerski and Kalai [BK20] obtained a succinct private access control protocol for the access structures that can be expressed by monotone formulas, and Morgan et al. [MPP20] obtained a succinct non-interactive secure two-party computation protocol.

The number of applications is, however, still limited. A potential reason for this limitation is that the current techniques inherently use cryptographic primitives in non-black-box ways. Concretely, to hide the prover's witness, the current techniques use KRR succinct argument under other cryptographic protocols (such as garbling schemes) and thus require non-black-box accesses to the codes of the cryptographic primitives that underlies KRR succinct argument. Consequently, the current techniques cannot be used for applications where black-box uses of cryptographic protocols are desirable, such as the application to commit-and-prove protocols, which we discuss next.

Commit-and-prove protocols. Informally speaking, a commit-and-prove protocol is a commitment scheme in which the committer can prove a statement about the committed value without opening the commitment. Proofs by the committer are required to be *zero-knowledge* (ZK) or *witness-indistinguishable* (WI), where the former requires that the views of the receiver in the commit and prove phases can be simulated in polynomial time without knowing the committed value, and the latter requires that for any two messages and any statement such that both of the messages satisfy the statement, the receiver cannot tell which of the messages is committed even after receiving a proof on the statement. Commit-and-prove protocols were implicitly used by Goldreich et al. [GMW87] for obtaining a secure multi-party computation protocol with malicious security, and later formalized by Canetti et al. [CLOS02].

A desirable property of commit-and-prove protocols is that they are constructed in a black-box way, i.e., in a way that uses the underlying cryptographic primitives as black-box by accessing them only through their input/output interfaces. Indeed, this black-box construction property is essential when commit-and-prove protocols are used as a tool for enforcing honest behaviors on malicious parties without relying on non-black-box uses of the underlying cryptographic primitives (see, e.g., [GLOV12, LP12, GOSV14]).

Very recently, Hazay and Venkitasubramaniam [HV18] and Khurana et al. [KOS18] gave four-round black-box constructions of ZK commit-and-prove protocols, where the round complexity of a commit-and-prove protocol is defined as the sum of that of the commit phase and that of the prove phase. Their protocols are *round optimal* since the commit and prove phases of their commit-and-prove protocols can be thought of as black-box ZK arguments (where the prover first

commits to a witness and then proves the validity of the committed witness) and black-box ZK arguments are known to require at least four rounds [GK96]. Their protocols also have the *delayed-input property*, i.e., the property that statements to be proven on committed values can be chosen adaptively in the last round of the prove phase.

The commit-and-prove protocols by Hazay and Venkitasubramaniam [HV18] and Khurana et al. [KOS18] are not succinct in the sense that when the statement is expressed as a T-time predicate on the committed value, the communication complexity depends at least linearly on T. This is because both of their protocols were obtained via transformations from the three-round constant-sound commit-and-prove protocol of Hazay and Venkitasubramaniam [HV16], which is not succinct in the above sense.

1.1 Our Result

Our main result is a four-round black-box construction of a constant-sound WI commit-and-prove protocol with succinct communication complexity.

Theorem 1. Assume the existence of sub-exponentially hard versions of the following cryptographic primitives: a collision-resistant hash function family, a two-round oblivious transfer protocol, and a two-round private information retrieval protocol. Then, there exists a constant-sound WI commit-and-prove protocol with the following properties.

- 1. The round complexity is 4, and the protocol satisfies the delayed-input property and uses the above cryptographic primitives in a black-box way.
- 2. When the length of the committed value is n and the statement to be proven on the committed value is a T-time predicate, the communication complexity depends polynomially on $\log n$, $\log T$, and the security parameter.

Our commit-and-prove protocol uses a variant of KRR succinct argument (which is obtained from the private information retrieval protocol), and succinctness of our commit-and-prove protocol is inherited from that of KRR succinct argument. We assume sub-exponential hardness on the cryptographic primitives since we use complexity leveraging.

ZK and negligible soundness error. Given our constant-sound WI commit-and-prove protocol, we can use (a minor variant of) a transformation of Khurana et al. [KOS18] to transform it into a 4-round ZK commit-and-prove protocol with negligible soundness error. The resultant commit-and-prove protocol still satisfies the delayed-input property, the black-box uses of the underlying primitives, and the succinct communication complexity. (See Appendix J for details.)

Verification time. The verification of our commit-and-prove protocol is not succinct, i.e., the verifier running time depends polynomially on T. Although we might be able to make it succinct by appropriately modifying our protocol (see Appendix A for details), we do not explore this possibility in this work so that we can focus on our main purpose, i.e., on showing how to use KRR succinct argument in black-box constructions of commit-and-prove protocols.

Complexity leveraging. As mentioned above, we use complexity leveraging in the proof of Theorem 1. Although we might be able to avoid the use of complexity leveraging by using known techniques (e.g., by relying on extractable commitments [PW09]), we do not explore this possibility in this work for the same reason as above.

Comparison with existing schemes. As explained above, Hazay and Venkitasubramaniam [HV18] and Khurana et al. [KOS18] gave four-round black-box ZK commit-and-prove protocols with the delayed-input property. Their schemes rely on a weak primitive (injective one-way functions) but do not have succinct communication.

Goyal et al. [GOSV14] and Ishai and Weiss [IW14] studied black-box commit-and-prove protocols with succinct communication under slightly different definitions than ours. If their techniques are used to obtain schemes under our definitions, the resultant schemes will rely on a weak primitive (collision-resistant hash functions) but have round complexity larger than 4.2

Kalai and Paneth [KP16] observed that when messages are committed by using Merkle tree-hash, KRR succinct argument can be used for proving statements on the committed messages. The resultant scheme is succinct in terms of both communication complexity and verification time, but uses the underlying hash function in a non-black-box way and does not have privacy properties (which are not needed for the purpose of [KP16]).

¹For example, the definition in [IW14] considers non-deterministic statements on committed values but the statements are assumed to be fixed in the commit phase, whereas our definition considers deterministic statements but the statements are allowed to be chosen after the commit phase is completed.

²Roughly speaking, this is because in a setting where statements to be proven are chosen after the commit phase (e.g., the delayed-input setting), techniques in [GOSV14, IW14] require that (1) the commit phase has 2 rounds as it needs to be succinct and (2) the prove phase has 3 rounds as it has a commit-challenge-response structure.

1.2 Overview of Our Commit-and-prove Protocol

The overall approach is to combine KRR succinct argument with the MPC-in-the-head technique [IKOS09].

Let us first recall how we can obtain a non-succinct WI commit-and-prove protocol by using the MPC-in-the-head technique. Let $M \in \mathbb{N}$ be an arbitrary constant, Π be any 2-private semi-honest secure M-party computation protocol with perfect completeness,³ OT be any two-round 1-out-of- M^2 oblivious transfer (OT) protocol, SBCom be any statistically binding commitment scheme, and SHCom be any statistically hiding commitment scheme. We assume that the hiding property of SBCom can be broken in a quasi-polynomial time $T_{\rm SB}$, and the security of the other primitives holds against poly($T_{\rm SB}$)-time adversaries.

Commit phase. To commit to a message x_{COM} , the committer (1) chooses random $x_{\text{MPC}}^1, \dots, x_{\text{MPC}}^M$ such that $x_{\text{MPC}}^1 \oplus \dots \oplus x_{\text{MPC}}^M = x_{\text{COM}}$, (2) chooses randomness $r_{\text{MPC}}^1, \dots, r_{\text{MPC}}^M$ for the M parties of Π , and (3) commits to $\mathsf{St}_0^\mu \coloneqq (x_{\text{MPC}}^\mu, r_{\text{MPC}}^\mu)$ for each $\mu \in [M]$ by using SHCom. (Note that each St_0^μ can be thought of as an initial state of a party of Π .) For each $\mu \in [M]$, let $\mathsf{dec}_{\mathsf{SH}}^\mu$ denote the decommitment of SHCom for revealing St_0^μ .

Prove phase. In the first round, the receiver computes a receiver message of OT by using random $(\alpha, \beta) \in [M] \times [M]$ as the input, ⁴ and sends it to the committer.

In the second round, to prove $f(x_{\text{COM}}) = 1$ for a predicate f, the committer does the following. (1) Execute Π in the head by using $\mathsf{st}_0^1, \dots, \mathsf{st}_0^M$ as the initial states of the M parties and using $f': (y^1, \dots, y^M) \mapsto f(y^1 \oplus \dots \oplus y^M)$ as the functionality to be computed. Let $\mathsf{view}^1, \dots, \mathsf{view}^M$ be the views of the parties in this execution of Π . (2) For each $\mu \in [M]$, compute a commitment to $(\mathsf{dec}_{\mathtt{SH}}^\mu, \mathsf{view}^\mu)$ by using SBCom. Let $\mathsf{dec}_{\mathtt{SB}}^\mu$ be the decommitment of SBCom for revealing $(\mathsf{dec}_{\mathtt{SH}}^\mu, \mathsf{view}^\mu)$. (3) Compute a sender message of OT by using $\{(\mathsf{dec}_{\mathtt{SB}}^\mu, \mathsf{dec}_{\mathtt{SB}}^\nu)\}_{\mu,\nu \in [M]}$ as the input. (4) Send the commitments and the OT message to the receiver.

In the verification, the receiver (1) recovers dec_{sB}^{α} , dec_{sB}^{β} from the OT message, (2) checks that they are valid decommitments of SBCom for revealing dec_{sH}^{α} , $view^{\alpha}$, dec_{sH}^{β} , $view^{\beta}$ and that dec_{sH}^{α} , dec_{sH}^{β} are valid decommitments of SHCom for revealing st_{0}^{α} , st_{0}^{β} , and (3) checks the following two conditions on st_{0}^{α} , $view^{\alpha}$, st_{0}^{β} , $view^{\beta}$.

- 1. The views $view^{\alpha}$, $view^{\beta}$ are *consistent* in the sense that the messages that the party P^{α} receives from the party P^{β} in $view^{\alpha}$ is equal to the messages that P^{β} sends to P^{α} in $view^{\beta}$ and vice versa.
- 2. For each $\xi \in \{\alpha, \beta\}$, the view view ξ indicates that the initial state of P^{ξ} is S_0^{ξ} and the output is 1.

First, the constant soundness follows from the receiver security of OT and the perfect completeness of Π . Roughly speaking, this is because (1) the receiver security of OT guarantees that the committer can convince the verifier with high probability only when it commits to initial states and views that satisfy the above two conditions for every $\alpha, \beta \in [M]$, and (2) when the committed initial states and views satisfy the above two conditions for every $\alpha, \beta \in [M]$, the perfect completeness of Π guarantees $f(x_{\text{COM}}) = 1$, where x_{COM} is derived from the committed initial states. Next, the witness-indistinguishability follows from the receiver security of OT and the 2-privacy of Π . This is because the former guarantees that the receiver only learns the committed initial states and views of two parties and the latter guarantees the committed initial states and views of any two parties do not reveal any information about x_{COM} . Finally, this scheme is not succinct since the committer sends the initial states and views of Π (or more precisely the decommitments to them) via OT.

Now, to make the above scheme succinct, we combine it with KRR succinct argument. The idea is to let the committer send succinct arguments about the initial states and views (instead of the initial states and views themselves) via OT. That is, we let the committer prove that the above two conditions hold on the committed initial states and views of each pair of the parties, where a separate instance of KRR succinct argument is used for each pair of the parties, and let it send the resultant M^2 succinct arguments via OT. (Note that KRR succinct argument can naturally be combined with OT since it is a two-round protocol.) As a minor modification, we also let the committer use a succinct commitment scheme to commit to the initial states and the views.

Unfortunately, although the modifications are intuitive, proving the soundness of the resultant scheme is non-trivial. (In contrast, the WI property can be proven similarly to the WI property of the original scheme. The key point is that, although KRR succinct argument does not provide any witness privacy, we can still prove WI of the whole scheme since in each instance of KRR succinct argument, the witness—initial states and views of a pair of the parties—does not reveal any secret information anyway.)

A natural approach for proving the soundness would be to first prove the soundness of each instance of KRR succinct argument and then derive the soundness of the whole scheme from it. Indeed, if we can show that each of the M^2 instances of KRR succinct argument provides an argument-of-knowledge property (which allows us to extract the committed initial states and views from the cheating committer), we can easily prove the soundness of the whole scheme.

The problem of this approach is that KRR succinct argument is not known to provide soundness for all statements in \mathcal{NP} , and hence, does not necessarily provide soundness when it is used as above.

³Such an MPC protocol can be obtained unconditionally (e.g., the 2-private five-party protocol by Ben-Or et al. [BGW88, AL17]).

⁴We assume that the set $[M] \times [M]$ is identified with the set $[M^2]$ in a canonical way.

⁵Formally, complexity leveraging is required in this argument since the receiver security of OT needs to hold even against adversaries that extract the committed initial states and views by brute force.

Our actual approach is to show that, while each of the instances of KRR succinct argument does not necessarily provide soundness, they as a whole provide a meaningful notion of the soundness, which can be used to prove the soundness of the whole scheme. Specifically, by getting into the security proof of the soundness of KRR succinct argument, we show that when M^2 instances of KRR argument are used in parallel for proving the consistency of each pair of the committed views etc. as above, then they as a whole guarantee that the committed views are mutually consistent etc.

We give more detailed overviews of our approach from Section 3 to Section 6 after giving necessary definitions in Section 2.

2 Preliminaries

We assume familiarity with basic cryptographic primitives. Several additional definitions are given in Appendix C.

2.1 Notations and Conventions

We denote the security parameter by λ . We assume that every algorithm takes the security parameter as input, and often do not write it explicitly.

We identify a bit-string with a function in the following manner: a bit-string $x = (x_1, ..., x_n)$ is thought of as a function $x : [n] \to \{0, 1\}$ such that $x(i) = x_i$. More generally, for any finite field F, we identify a string over F with a function in the same manner. For a vector $\mathbf{v} = (v_1, ..., v_n)$ and a set $S \subseteq [n]$, we define $\mathbf{v}|_S$ by $\mathbf{v}|_S := \{v_i\}_{i \in S}$. Similarly, for a function $f : D \to R$ and a set $S \subseteq D$, we define $f|_S$ by $f|_S := \{f(i)\}_{i \in S}$.

For any two probabilistic interactive Turing machines A and B and any input x_A to A and x_B to B, we denote by $(out_A, out_B) \leftarrow \langle A(x_A), B(x_B) \rangle$ that the output of an interaction between $A(x_A)$ and $B(x_B)$ is (out_A, out_B) , where out_A is the output from A and out_B is the output from B.

2.2 Witness-indistinguishable Commit-and-prove Protocols

We give the definition of witness-indistinguishable commit-and-prove protocols. Our definition is based on the definition by Khurana et al. [KOS18] but is slightly different from it; see Appendix B for the differences.

A witness-indistinguishable (WI) commit-and-prove protocol $\langle C, R \rangle$ is a protocol between a committer C = (C.Com, C.Dec, C.Prv) and a receiver R = (R.Com, R.Dec, R.Prv), and it consists of three phases.

- 1. In the commit phase, C.Com takes a message $x \in \{0, 1\}^n$ as input and interacts with R.Com to commit to x.⁶ At the end of the interaction, C.Com outputs its internal state st_C and R.Com outputs the commitment com, which is the transcript of the commit phase.
- 2. In the prove phase, C.Prv takes a predicate f as input along with st_C , and interacts with R.Prv to prove that f(x) = 1 holds, where R.Prv takes (com, f) as input, At the end of the interaction, R.Prv outputs either 1 (accept) or 0 (reject).
- 3. In the open phase, C.Dec takes an index $i \in [n]$ as input along with st_C , and interacts with R.Dec to reveal the i-th bit of x, where R.Dec takes (com, i) as input. At the end of the interaction, R.Dec outputs either a bit x_i as the decommitted bit, or \bot (reject).

In this paper, we focus on a WI commit-and-prove protocol such that (1) both the prove phase and the open phase consist of two rounds, (2) the first round of the prove phase does not depend on the commitment com and the predicate f, and (3) the first round of the open phase does not depend on the commitment com. Because of (1) and (2), R.Prv can be split into two algorithms, R.Prv.Q and R.Prv.D, such that the prove phase proceeds as follows: $(Q, st_R) \leftarrow R.Prv.Q$; $\pi \leftarrow C.Prv(st_C, f, Q)$; $b \leftarrow R.Prv.D(st_R, com, f, \pi)$. Similarly, because of (1) and (3), R.Dec can be split into two algorithms, R.Dec.Q and R.Dec.D, such that the open phase proceeds as follows: $(Q, st_R) \leftarrow R.Dec.Q(i)$; dec $\leftarrow C.Dec(st_C, i, Q)$; $b \leftarrow R.Dec.D(st_R, com, dec)$.

WI commit-and-prove protocols need to satisfy the following security notions.

Definition 1 (Completeness). A commit-and-prove protocol $\langle C, R \rangle$ is complete if for any polynomial $n : \mathbb{N} \to \mathbb{N}$ and any $\lambda \in \mathbb{N}$, $x \in \{0, 1\}^{n(\lambda)}$, and $i \in [n(\lambda)]$,

$$\Pr \left[x_i = \tilde{x}_i \, \middle| \, \begin{array}{l} (\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}(x), \mathsf{R.Com} \rangle \\ (\bot, \tilde{x}_i) \leftarrow \langle \mathsf{C.Dec}(\mathsf{st}_C, i), \mathsf{R.Dec}(\mathsf{com}, i) \rangle \end{array} \right] = 1 \ .$$

Definition 2 (Binding). A commit-and-prove protocol $\langle C, R \rangle$ is (computationally) binding if for any polynomial $n : \mathbb{N} \to \mathbb{N}$, any PPT cheating committer $C^* = (C.Com^*, C.Dec^*)$, and any $\lambda \in \mathbb{N}$, the following binding condition holds with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com^*, R.Com \rangle$.

 $^{^6}$ We assume that the length of the message to be committed, n, is implicitly given to the receiver as input.

⁷We assume that Time(f) is known to the both parties in advance (where f is expressed as, e.g., a Turing machine).

- Binding Condition: For every $i \in [n(\lambda)]$, it holds $\Pr[b_{BAD} = 1] \leq \operatorname{negl}(\lambda)$ in the following probabilistic experiment $Exp^{\operatorname{bind}}(\mathsf{C.Dec}^*, \mathsf{st}_C, \mathsf{com}, i)$.
 - 1. For each $b \in \{0, 1\}$, sample Q_b by $(Q_b, \mathsf{st}_b) \leftarrow \mathsf{R.Dec.Q}(i)$.
 - 2. $Run \{ dec_b \}_{b \in \{0,1\}} \leftarrow C.Dec^*(st_C, i, \{Q_b\}_{b \in \{0,1\}}).$
 - 3. For each $b \in \{0, 1\}$, let $x_b^* \leftarrow \mathsf{R.Dec.D}(\mathsf{st}_b, \mathsf{com}, \mathsf{dec}_b)$.
 - 4. Output $b_{\scriptscriptstyle BAD} := 1$ if and only if $x_0^* \neq \bot \land x_1^* \neq \bot \land x_0^* \neq x_1^*$ holds.

Definition 3 (Soundness). Let $\epsilon : \mathbb{N} \to [0,1]$ be a function. A commit-and-prove protocol $\langle C,R \rangle$ is (computationally) ϵ -sound if for any constant $c \in \mathbb{N}$, there exists a PPT oracle Turing machine E (called an extractor) such that for any polynomial $n : \mathbb{N} \to \mathbb{N}$, any PPT cheating committer $C^* = (C.Com^*, C.Prv^*)$, and any sufficiently large $\lambda \in \mathbb{N}$, the following soundness condition holds with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com^*, R.Com \rangle$.

• Soundness Condition⁸: If it holds

$$\Pr \left[b = 1 \; \middle| \; \begin{array}{l} (Q, \mathsf{st}_R) \leftarrow \mathsf{R}.\mathsf{Prv.Q}; \; (f, \pi) \leftarrow \mathsf{C}.\mathsf{Prv}^*(\mathsf{st}_C, Q); \\ b \leftarrow \mathsf{R}.\mathsf{Prv.D}(\mathsf{st}_R, \mathsf{com}, f, \pi) \end{array} \right] \geq \epsilon(\lambda) + \frac{1}{\lambda^c} \;\; ,$$

then there exists $x^* = (x_1^*, \dots, x_n^*) \in \{0, 1\}^{n(\lambda)}$ such that

$$\forall i \in [n(\lambda)], \Pr\left[x_i = x_i^* \;\middle|\; (\bot, x_i) \leftarrow \langle E^{\text{C.Prv}^*(\text{st}_C, \cdot)}(\text{com}, i), \text{R.Dec}(\text{com}, i)\rangle\right] \geq 1 - \mathsf{negl}(\lambda)$$

and

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x^*)=0\end{array}\right| \begin{array}{c}(Q,\mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}; \ (f,\pi) \leftarrow \mathsf{C.Prv}^*(\mathsf{st}_C,Q);\\ b \leftarrow \mathsf{R.Prv.D}(\mathsf{st}_R,\mathsf{com},f,\pi)\end{array}\right] \leq \epsilon(\lambda) + \mathsf{negl}(\lambda) \ .$$

 $\langle C, R \rangle$ is said to be sound if it is ϵ -sound for a negligible function ϵ

Definition 4 (Witness Indistinguishability). $\langle C, R \rangle$ is witness-indistinguishable if for any polynomial $n : \mathbb{N} \to \mathbb{N}$, any two sequences $\{x_{\lambda}^0\}_{\lambda \in \mathbb{N}}$ and $\{x_{\lambda}^1\}_{\lambda \in \mathbb{N}}$ such that $x_{\lambda}^0, x_{\lambda}^1 \in \{0, 1\}^{n(\lambda)}$, any PPT cheating receiver $R^* = (\mathsf{R.Com}^*, \mathsf{R.Prv.Q}^*)$, the outputs of Experiment 0 and Experiment 1 are computationally indistinguishable.

- Experiment b ($b \in \{0, 1\}$).
 - 1. $(\operatorname{st}_C, \operatorname{st}_R) \leftarrow \langle \operatorname{C.Com}(x_1^b), \operatorname{R.Com}^*(x_1^0, x_1^1) \rangle$.
 - 2. $(f, Q, \mathsf{st}_R') \leftarrow \mathsf{R.Prv.Q}^*(\mathsf{st}_R)$. If $f(x_3^0) \neq 1$ or $f(x_3^1) \neq 1$, abort.
 - 3. $\pi \leftarrow \text{C.Prv}(\text{st}_C, f, Q)$.
 - 4. Output (st'_p, π) .

2.3 Secure Multi-party Computation

We recall the definition of secure multi-party computation (MPC) protocols based on the description by Ishai et al. [IKOS09]. (We assume that the readers are familiar with the concept of secure MPC protocols.)

The basic model that is used in this paper is the following. The number of parties is denoted by M. We focus on MPC protocols that realize any deterministic M-party functionality that outputs a single bit (which is obtained by all the parties), given the synchronous communication over secure point-to-point channels. We assume that every party implicitly takes as input the M-party functionality to be computed.

Recall that the *view* of a party in an execution of an MPC protocol consists of its input, its randomness, and all the incoming messages that it received from the other parties during the execution of the protocol. The consistency between a pair of views is defined as follows.

Definition 5 (Consistent Views). A pair of views $view^i$, $view^j$ is consistent (w.r.t. an MPC protocol Π for a functionality f) if the outgoing messages that are implicitly reported in $view^i$ are identical to the incoming messages that are reported in $view^j$ and vice versa.

We consider security against semi-honest adversaries. Concretely, we use the following two security notions.

Definition 6 (Perfect correctness). We say that an MPC protocol Π satisfies perfect correctness if for any deterministic M-party functionality f and for any private inputs to the parties, the probability that the output of some party in an honest execution of Π is different from the output of f is 0.

Definition 7 (2-privacy). We say that an MPC protocol Π satisfies perfect 2-privacy if for any deterministic M-party functionality f, there exists a PPT simulator S_{MPC} such that for any private inputs x_1, \ldots, x_M to the parties and every pair of corrupted parties, $T \subset [M]$ such that |T| = 2, the joint view $\mathsf{View}_T(x_1, \ldots, x_M)$ of the parties in T is identically distributed with $S_{MPC}(T, \{x_i\}_{i \in T}, f(x_1, \ldots, x_M))$.

⁸Roughly speaking, the soundness condition requires that if a cheating prover convinces the verifier with sufficiently high probability, then there exists a value x^* such that (1) the extractor can decommit com to x^* and (2) the cheating prover cannot prove false statements about x^* .

2.4 Probabilistically Checkable Proofs (PCPs)

We recall the definition of probabilistically checkable proofs (PCPs) based on the description by Brakerski et al. [BHK17]. Roughly speaking, PCPs are proof systems with which one can probabilistically verify the correctness of statements by reading only a few bits or symbols of the proof strings. A formal definition is given below.

Definition 8. A κ -query PCP system (P, V) for an NP language L, where V = (Q, D), satisfies the following.

• (Completeness) For all $\lambda \in \mathbb{N}$ and $x \in L$ (with witness w) such that $|x| \leq 2^{\lambda}$,

$$\Pr\left[\mathsf{D}(\mathsf{st},x,\pi|_{Q}) = 1 \, \middle| \, \begin{array}{c} (Q,\mathsf{st}) \leftarrow \mathsf{Q}(1^{\lambda}) \\ \pi \leftarrow \mathsf{P}(1^{\lambda},x,w) \end{array} \right] = 1 \ .$$

The PCP proof π is a string of characters over some alphabet Σ , and it can be thought that this string is indexed by a set Γ (by identifying Γ with [N] in a canonical way, where N is the length of the string) and $Q \subseteq \Gamma$. Alternatively, π can be thought of as a function from Γ to Σ .

• (Soundness) For all $\lambda \in \mathbb{N}$, all $x \notin L$ such that $|x| \leq 2^{\lambda}$, and all proof string π^* ,

$$\Pr\left[\mathsf{D}(\mathsf{st},x,\pi^*|_{Q}) \mid (Q,\mathsf{st}) \leftarrow \mathsf{Q}(1^{\lambda})\right] \leq \frac{1}{2}.$$

- (Query Efficiency) If $(Q, st) \leftarrow Q(1^{\lambda})$, then $|Q| \le \kappa(\lambda)$ and the combined run-time of Q and D is $poly(\lambda)$.
- (**Prover Efficiency**) The prover P runs in polynomial time, where its input is $(1^{\lambda}, x, w)$.

2.5 Definitions from Kalai et al. [KRR14] and Subsequent Works

2.5.1 Computational no-signaling (CNS).

We recall the definition of adaptive (computational) no-signaling [KRR14, BHK17].

Definition 9. Fix any alphabet $\{\Sigma_{\lambda}\}_{{\lambda}\in\mathbb{N}}$, any $\{N_{\lambda}\}_{{\lambda}\in\mathbb{N}}$ such that $N_{\lambda}\in\mathbb{N}$, any function $\kappa_{\max}:\mathbb{N}\to\mathbb{N}$ such that $\kappa_{\max}(\lambda)\leq N_{\lambda}$, and any algorithm Algo such that for any $\lambda\in\mathbb{N}$, on input a subset $Q\subset[N_{\lambda}]$ of size at most $\kappa_{\max}(\lambda)$, Algo outputs (the truth table of) a function $A:Q\to\Sigma\cup\{\bot\}$ with an auxiliary output out.

Then, the algorithm Algo is adaptive κ_{\max} -computational no-signaling (CNS) if for any PPT distinguisher \mathcal{D} , any sufficiently large $\lambda \in \mathbb{N}$, any $Q, S \subset [N_{\lambda}]$ such that $Q \subseteq S$ and $|S| \leq \kappa_{\max}(\lambda)$, and any $z \in \{0, 1\}^{\mathsf{poly}(\lambda)}$,

$$\left| \ \Pr\left[\mathcal{D}(\mathsf{out},A,z) = 1 \ | \ (\mathsf{out},A) \leftarrow \mathsf{Algo}(Q) \right] - \Pr\left[\mathcal{D}(\mathsf{out},A|_Q,z) = 1 \ | \ (\mathsf{out},A) \leftarrow \mathsf{Algo}(S) \right] \ \right| \leq \mathsf{negl}(\lambda) \ .$$

We remark that the above definition can be naturally extended for the case that Algo takes auxiliary inputs, as well as for the case that Algo takes multiple subsets as input and then outputs multiple functions (see Section C.3).

2.5.2 Adaptive local assignment generator.

We recall the definition of adaptive local assignment generators [PR14, BHK17].

Definition 10. For any function $\kappa_{\max} : \mathbb{N} \to \mathbb{N}$, an adaptive κ_{\max} -local assignment generator Assign on variables $\{V_{\lambda}\}_{{\lambda} \in \mathbb{N}}$ is an algorithm that takes as input a security parameter 1^{λ} and a set of at most $\kappa_{\max}(\lambda)$ queries $W \subseteq \{1, \ldots, |V_{\lambda}|\}$, and outputs a 3CNF formula φ on variables V_{λ} and assignments $A : W \to \{0, 1\}$ such that the following two properties hold.

- Everywhere Local Consistency. For every $\lambda \in \mathbb{N}$ and every set $W \subseteq \{1, \dots, |V_{\lambda}|\}$ such that $|W| \leq \kappa_{\max}(\lambda)$, with probability at least $1 \mathsf{negl}(\lambda)$ over sampling $(\varphi, A) \leftarrow \mathsf{Assign}(1^{\lambda}, W)$, the assignment A is "locally consistent" with the formula φ . That is, for any $i_1, i_2, i_3 \in W$, if φ has a clause whose variables are $v_{i_1}, v_{i_2}, v_{i_3}$, then this clause is satisfied with the assignment $A(i_1), A(i_2), A(i_3)$ with probability at least $1 \mathsf{negl}(\lambda)$.
- *Computational No-signaling*. Assign *is adaptive* κ_{max} -*CNS*.

2.5.3 No-signaling PCPs.

We recall the definition of (computational) no-signaling PCPs [KRR14, BHK17]. Essentially, no-signaling PCPs are PCP systems that are sound against no-signaling cheating provers. Specifically, for any function $\kappa_{max} : \mathbb{N} \to \mathbb{N}$, a PCP system (P, V) for a language L, where V = (Q, D), is adaptive κ_{max} -no-signaling sound with negligible soundness error if it satisfies the following.

• (No-signaling Soundness) For any adaptive κ_{max} -CNS cheating prover P^* and any $\lambda \in \mathbb{N}$,

$$\Pr\left[x^* \notin L \land \mathsf{D}(\mathsf{st}, x^*, \pi^*) = 1 \, \middle| \, \begin{array}{c} (Q, \mathsf{st}) \leftarrow \mathsf{Q}(1^\lambda) \\ (x^*, \pi^*) \leftarrow P^*(1^\lambda, Q) \end{array} \right] \leq \mathsf{negl}(\lambda) \enspace .$$

3 Outline of Proof of Theorem 1

As mentioned in Section 1.2, our commit-and-prove protocol uses the succinct argument of Kalai et al. [KRR14] (KRR succinct argument in short). Unfortunately, we do not use it modularly—we slightly modify a building block of KRR succinct argument (namely, their no-signaling PCP system) when constructing our protocol, and we see low-level parts of the analysis of KRR succinct argument when analyzing our protocol.

At a high level, KRR succinct argument is obtained in three steps, starting from a scheme with a weak soundness notion.

- 1. Obtain a PCP system such that no CNS adversary can break the soundness with overwhelming success probability.
- 2. Obtain a PCP system such that no CNS adversary can break the soundness with non-negligible success probability.
- 3. Obtain a succinct argument such that no adversary can break the soundness with non-negligible success probability.

Somewhat similarly, our commit-and-prove protocol is obtained in five steps, starting from a non-WI scheme with a weak soundness notion.

- 1. Obtain a non-WI scheme, $\langle C_1, R_1 \rangle$, such that no CNS "well-behaving" adversary can break the soundness with overwhelming success probability. (Well-behaving adversaries is the class of adversaries that we introduce later.)
- 2. Obtain a non-WI scheme, $\langle C_2, R_2 \rangle$, such that no CNS adversary can break the soundness with overwhelming success probability.
- 3. Obtain a non-WI scheme, $\langle C_3, R_3 \rangle$, such that no CNS adversary can break the soundness with <u>non-negligible</u> success probability.
- 4. Obtain a non-WI scheme, $\langle C_4, R_4 \rangle$, such that <u>no adversary</u> can break the soundness with non-negligible success probability.
- 5. Obtain a WI scheme, (C_5, R_5) , such that no adversary can break the soundness with constant success probability.

The most technically interesting step is the first step, and an extensive overview of this step is given in Section 4. Overviews of the other steps are given in Section 5 and Section 6. The formal proof is given in appendices (from Appendix D to Appendix I).

3.1 Building Block: Perfect 2-private MPC protocol Π

In addition to the cryptographic primitives that are listed in Theorem 1, we use a 2-private semi-honest secure M-party computation protocol Π with perfect completeness, where M is an arbitrary constant. (Note that such an MPC protocol can be obtained unconditionally; cf. Footnote 3.) We denote the parties of Π by $P^1, \ldots P^M$.

Simplifying assumptions on Π . For editorial simplicity, we make several simplifying assumptions on Π .

- The length of the initial state of each party is denoted by $n_{\text{st}} = n + n_{\text{MPC}}$, where n is the input length and n_{MPC} is the randomness length, and each party has n_{st} -bit internal state at the beginning of each round.
- Every party uses the same next-message function in every round.⁹
- Every party sends a 1-bit message to each party at the end of each round.
- Every party receives dummy incoming messages from all the parties at the beginning of the first round, and every party sends a dummy outgoing message to itself at the end of each round. (This assumption is made so that the next-message function always takes an $(n_{st} + M)$ -bit input, where the last M bits are the concatenation of the incoming messages.)
- The first bit of the final state of each party denotes the output of that party.

⁹The next-message function takes as input an internal state and incoming messages of a round, and it outputs the internal state and outgoing messages of the round. (We assume that the internal state implicitly includes all the incoming messages of the previous rounds.)

4 Overview of Step 1 (Non-WI Scheme with Soundness against CNS Wellbehaving Provers)

We give an extensive overview of our non-WI commit-and-prove protocol $\langle C_1, R_1 \rangle$, which is (1-negl)-sound against CNS "well-behaving" provers. At a high level, we follow the approach that we outline in Section 1.2. That is, we implement the MPC-in-the-head technique with the MPC protocol Π and a succinct argument. However, instead of using KRR succinct argument, we use a variant of the no-signaling PCP system (PCP.P_{KRR}, PCP.V_{KRR}) of Kalai et al. [KRR14] (which is the main building block of KRR succinct argument and is referred to as KRR no-signaling PCP in what follows), and we do not use any cryptographic primitives in this step so that we can focus on information theoretical arguments in the analysis. As a result, we can prove soundness only against very restricted provers, which we define as CNS well-behaving provers.

For simplicity, in this overview, we focus on static soundness, where the statement to be proven by the cheating prover is fixed at the beginning of the prove phase. We will also make several implicit oversimplifications in this overview.

4.1 Preliminary: Overview of Analysis of KRR No-signaling PCP

We start by briefly recalling the analysis of KRR no-signaling PCP (i.e., the analysis of its no-signaling soundness for statements in \mathcal{P}), focusing on the parts that are relevant to this work. ¹⁰

We first remark that KRR no-signaling PCP is a PCP system for 3SAT, so at the beginning the statement to be proven is converted into a 3SAT instance. Specifically, given any statement in \mathcal{P} of the form "(f, x) satisfies f(x) = 1" for some public function f and input x, first the function f is converted into a carefully designed Boolean circuit C that computes f, and next the statement is converted into a 3SAT instance φ that has the following properties.

- 1. φ has a variable for each of the wires in C, and the values that are assigned to these variables are interpreted as an assignment to the corresponding wires in C.
- 2. The clauses of φ checks that (1) for each gate in C, the assignment to its input and output wires is consistent with the computation of the gate, (2) the assignment to the input wires of C is equal to x, and (3) the assignment to the output wire of C is equal to 1.

Now, the analysis of KRR no-signaling PCP roughly consists of three parts.

The first part of the analysis shows that any successful CNS cheating prover for a statement (f, x) can be converted into a local assignment generator for the 3SAT instance φ that is obtained from (f, x) as above. That is, it shows that any successful CNS cheating prover can be converted into a probabilistic algorithm Assign such that (1) Assign takes as input a small-size subset of the variables of φ and it outputs an assignment to these variables, and (2) Assign is guaranteed to satisfy the following everywhere local consistency.

Everywhere local consistency. Assign does not make an assignment that violates any clause of φ . Specifically, when Assign is asked to make an assignment to the three variables that appear in a clause of φ , it makes an assignment that satisfies this clause.

(Actually, Assign is also guaranteed to be CNS, but we ignore it in this overview for simplicity.¹¹) We note that Assign does not necessarily comply with a single global assignment, that is, Assign can assign different values to the same variable depending on the randomness and the input. We also note that this part of the analysis holds even for statements in \mathcal{NP} . For simplicity, in this overview we assume that Assign does not err (i.e., the everywhere local consistency holds with probability 1).

The second part of the analysis shows that the local assignment generator Assign that is obtained in the first part is guaranteed to comply with a single global "correct" assignment. A bit more precisely, this part shows the following.

Let the correct assignment to a wire in C (or, equivalently, to a variable in φ) be defined as the assignment that is obtained by evaluating C on x, and let Assign be called *correct* on a wire in C (or variable in φ) if Assign makes the correct assignment to it whenever Assign is asked to make an assignment to it. Then, Assign is correct on any wire in C (or variable in φ), and in particular correct on the output wire of C.

Roughly speaking, the above is shown in two steps.

- 1. First, it is shown, by relying on a specific structure of C, that Assign is correct on any wire in C if Assign is correct on each input wire of C.
- 2. Next, it is observed that Assign is indeed correct on each input wire of C due to the everywhere local consistency and the definition of φ (which has clauses that check that the assignment to the input wires of C is equal to x).

¹⁰We follow the modularization by Paneth and Rothblum [PR14].

¹¹Concretely, in this overview we assume that Assign is perfect no-signaling, i.e., that the RHS of the equation in Definition 10 is 0 even against computationally unbounded distinguishers.

Finally, the last part of the analysis obtains the soundness by combining what are shown by the preceding two parts. In particular, it is observed that the existence of Assign as above implies f(x) = 1 since (1) on the one hand, Assign always assigns 1 to the output wire of C due to the everywhere local consistency and the definition of φ (which has clauses that check that the assignment to the output wire of C is 1), and (2) on the other hand, Assign always assigns f(x) to the output wire of C since what is shown by the second part implies that Assign is correct on the output wire of C.

Remark 1 (Difficulty in the case of NP statements). The above analysis does not work in general for statements in NP. A difficulty is that when the statement is in NP, it is unclear how we should define the correct assignment in the second part of the analysis. Indeed, on the one hand, the correct assignment can be naturally defined in the case of statements in P since there exists a unique assignment that any successful prover is supposed to use (namely the assignment that is derived from x); on the other hand, in the case of statements in NP, there does not exist a single such assignment. Jumping ahead, below we define well-behaving provers so that we can define the correct assignment naturally (while at the same time so that we can use cryptographic primitives later to force any prover to be well-behaving).

4.2 Protocol Description

In this overview, we consider the following protocol $\langle C_1, R_1 \rangle = (C.Com_1, C.Prv_1, R.Com_1, R.Prv.Q_1, R.Prv.D_1)$, which is slightly oversimplified from the actual protocol in Appendix E. (At this point, we temporarily ignore the open phase.) We warn that $\langle C_1, R_1 \rangle$ is not biding at all in the standard sense since the committer sends no message in the commit phase.

Commit Phase:

Round 1: Given x_{COM} as the value to be committed, C.Com₁ does the following.

- 1. Sample random $x_{\text{MPC}}^1, \dots, x_{\text{MPC}}^M$ such that $x_{\text{MPC}}^1 \oplus \dots \oplus x_{\text{MPC}}^M = x_{\text{COM}}$.
- 2. For each $\mu \in [M]$, define $x_{1,\text{in}}^{\mu}$ as follows: sample random $r_{\text{MPC}}^{\mu} \in \{0,1\}^{n_{\text{MPC}}}$ and let $\mathsf{st}_0^{\mu} \coloneqq x_{\text{MPC}}^{\mu} \| r_{\text{MPC}}^{\mu}$, i-msgs $_1^{\mu} \coloneqq \mathsf{st}_0^{\mu} \| \text{i-msgs}_1^{\mu}$.
- 3. Output an empty string as the commitment and store $\{x_{1 \text{ in}}^{\mu}\}_{\mu \in [M]}$ as the internal state.

Prove Phase:

Round 1 R.Prv. Q_1 does the following.

- 1. For each $\mu, \nu \in [M]$, obtain a set of queries $Q^{\mu,\nu}$ by running the verifier of KRR no-signaling PCP.
- 2. Output $\{Q^{\mu,\nu}\}_{\mu,\nu\in[M]}$ as the query.

Round 2: Given the statement f and the query $\{Q^{\mu,\nu}\}_{\mu,\nu\in[M]}$ as input, $C.Prv_1$ does the following.

- 1. Run the MPC protocol Π in the head for functionality f' and initial states $\{(\mathbf{st}_0^\mu, \mathbf{i-msgs}_1^\mu)\}_{\mu \in [M]}, \mathbf{i}^{-2}$ where f' is defined as $f': (y^1, \dots, y^M) \mapsto f(y^1 \oplus \dots \oplus y^M)$ and each $(\mathbf{st}_0^\mu, \mathbf{i-msgs}_1^\mu)$ is recovered from the internal state of the commit phase. Let $\{\mathbf{view}^\mu\}_{\mu \in [M]}$ be the view of the parties in this execution.
- 2. For each $\mu, \nu \in [M]$, obtain a PCP proof $\pi^{\mu:\nu}$ by running the prover of KRR no-signaling PCP on the 3SAT instance $\varphi^{\mu:\nu}$ that we will carefully design later—roughly speaking, $\varphi^{\mu:\nu}$ takes views of the parties P^{μ}, P^{ν} of Π as input, and checks that the views are consistent and that P^{μ} and P^{ν} output 1 in the views. (In an honest execution, C.Prv₁ uses (view^{μ}, view^{ν}) to obtain a satisfying assignment to $\varphi^{\mu:\nu}$ and then uses it to obtain $\pi^{\mu:\nu}$.)
- 3. Output $\{\pi^{\mu:\nu}|_{Q^{\mu:\nu}}\}_{\mu,\nu\in[M]}$ as the proof.

Verification: Given the statement f and the proof $\{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]}$ as input, R.Prv.D₁ does the following.

- 1. Verify each $\pi^{*\mu:\nu}$ by running the verifier of KRR no-signaling PCP, and let $b^{\mu:\nu}$ be the verification result.
- 2. Output 1 if and only if $b^{\mu:\nu} = 1$ for every $\mu, \nu \in [M]$.

4.3 Proof of Soundness

We give an overview of the proof of the soundness. To focus on the main technical idea, in this overview we consider a weak version of the soundness where the extractor is only required to extract a committed value (rather than decommit the commitment as required in Definition 3). Thus, for any successful cheating prover, the extractor is required to extract a value such that the cheating prover cannot prove false statements on it.

¹²Each i-msgs $_1^{\mu}$ is the dummy incoming messages of the first round (cf. Section 3.1).

4.3.1 Overall approach.

At a very high level, the proof consists of two parts.

The first part is to obtain an extractor. Toward this end, we first observe that, by borrowing analyses from Kalai et al. [KRR14], we can convert any successful CNS cheating prover against $\langle C_1, R_1 \rangle$ into a *parallel local assignment generator* p-Assign, which gives M^2 local assignments to the 3SAT instances $\{\varphi^{\mu\nu}\}_{\mu,\nu\in[M]}$ in parallel when it is given M^2 subsets of the variables as input. (To see this, observe that the prove phase of $\langle C_1, R_1 \rangle$ consists of M^2 parallel executions of KRR no-signaling PCP.) Then, we obtain an extractor by using p-Assign as follows.

- Note that since each $\varphi^{\mu:\nu}$ is a 3SAT instance that takes views of P^{μ} , P^{ν} as input, for any particular parts of P^{μ} and P^{ν} 's views, $\varphi^{\mu:\nu}$ has variables that are supposed to be assigned with these parts. In the following, when we say that p-Assign makes an assignment to particular parts of P^{μ} and P^{ν} 's views in $\varphi^{\mu:\nu}$, we mean that p-Assign makes an assignment to the variables that are supposed to be assigned with these parts in $\varphi^{\mu:\nu}$.
- Now, to extract the *i*-th bit of the committed value, the extractor obtains the *i*-th bit of each party's MPC input by asking p-Assign to make an assignment to the *i*-th bit of P^{μ} 's input in $\varphi^{\mu;\mu}$ for every $\mu \in [M]$, and then takes XOR of the obtained bits.

The second part is to show that any cheating prover cannot prove false statements on the extracted value. In this part, the analysis proceeds similarly to the analysis of KRR no-signaling PCP. That is, we first define the correct assignment for each of $\varphi^{\mu:\nu}$, and next show that p-Assign always makes the correct assignment to any variable in any of $\varphi^{\mu:\nu}$.

Unfortunately, we do not know how to prove the second part against CNS cheating provers in general, and thus, we further restrict the provers to be "well-behaving."

4.3.2 Well-behaving provers.

Roughly speaking, we define well-behaving provers as follows. Recall that the extractor is obtained by converting the cheating prover into a parallel local assignment generator. Now, we define well-behaving provers so that when we convert a successful CNS well-behaving prover into a parallel local assignment generator p-Assign, it satisfies the following two consistency properties.

Consistency on the initial states: Once the commit phase is completed, there exists a unique set of MPC initial states $\{(st_0^\mu, i\text{-msgs}_1^\mu)\}_{\mu\in[M]}$ such that p-Assign always makes assignments that are consistent with it (i.e., for any $\mu, \nu \in [M]$, when p-Assign is asked to make an assignment to any bit of the initial state of P^μ or P^ν in $\varphi^{\mu\nu}$, then p-Assign always assigns the corresponding bit of $(st_0^\mu, i\text{-msgs}_1^\mu)$ or $(st_0^\nu, i\text{-msgs}_1^\nu)$).

Consistency on the views: For every $\mu, \nu, \xi \in [M]$, when p-Assign is asked to make an assignment to any bit of P^{μ} 's view in both $\varphi^{\mu;\nu}$ and $\varphi^{\mu;\xi}$, then the value that p-Assign assigns to it in $\varphi^{\mu;\nu}$ is identical with the value that p-Assign assigns to it in $\varphi^{\mu;\xi}$. (The same holds for $\varphi^{\nu;\mu}$ and for $\varphi^{\mu;\nu}$ and $\varphi^{\xi;\mu}$ etc.)

Remark 2 (Intuition of the two consistency properties of p-Assign). Essentially, the above two consistency properties guarantee that p-Assign behaves as if it were obtained from an honest prover. This is because when p-Assign is indeed obtained from an honest prover, we can show that p-Assign always assigns the same MPC initial states once the commit phase is fixed, and assigns the same P^{μ} 's view in any $\varphi^{\mu;\nu}$ and $\varphi^{\mu;\xi}$. (Roughly speaking, this is because in an honest execution of $\langle C_1, R_1 \rangle$, a set of MPC initial states are fixed in the commit phase, and the same P^{μ} 's view is used for computing PCPs on any $\varphi^{\mu;\nu}$ and $\varphi^{\mu;\xi}$ in the prove phase.)

Before giving more details on the definition of well-behaving provers, we show that by restricting the provers to be well-behaving, we can complete the second part of the above overall approach, where our goal is to show that any cheating prover cannot prove false statements on the extracted value.

4.3.3 Showing that cheating prover cannot prove false statements.

As stated earlier, the analysis proceeds similarly to the analysis of KRR no-signaling PCP. That is, we first define the correct assignment for each of $\varphi^{\mu:\nu}$, and next show that p-Assign always makes the correct assignment to any variable in any of $\varphi^{\mu:\nu}$.

Step 1: Defining the correct assignments. We define the correct assignments for $\{\varphi^{\mu:\nu}\}_{\mu,\nu\in[M]}$ by relying on that p-Assign satisfies the consistency on the initial states. Recall that it guarantees that once the commit phase is completed, there exists a unique set of MPC initial states $\{(\mathsf{st}_0^\mu,\mathsf{i-msgs}_1^\mu)\}_{\mu\in[M]}$ such that p-Assign always makes local assignments that are consistent with it. Then, we first define *the correct views* $\{\mathsf{view}^\mu\}_{\mu\in[M]}$ as the views that are obtained by executing Π on these unique initial states $\{(\mathsf{st}_0^\mu,\mathsf{i-msgs}_1^\mu)\}_{\mu\in[M]}$, and then define *the correct assignment* for $\varphi^{\mu:\nu}$ ($\mu,\nu\in[M]$) as the assignment that is derived from the correct views ($\mathsf{view}^\mu,\mathsf{view}^\nu$) of P^μ,P^ν . (Recall that $\varphi^{\mu:\nu}$ is a 3SAT instance that takes views of P^μ,P^ν as input.)

From the definition, it is clear that p-Assign is correct on the initial states in every $\varphi^{\mu:\nu}$ (i.e., p-Assign always assigns the correct assignment to any bit of the initial states of P^{μ} , P^{ν} in $\varphi^{\mu:\nu}$ for every $\mu, \nu \in [M]$). Also, since the extractor extracts the committed value by taking XOR of the MPC inputs that are obtained from p-Assign, p-Assign's correctness on the initial states implies that the value that the extractor extracts is unique and is equal to the XOR of the MPC inputs that are used in the correct views.

Step 2: Showing that p-Assign is correct on every variable. At a high level, our approach is to apply the second part of the analysis of KRR no-signaling PCP (Section 4.1) on each party's next-message computation in a "round-by-round" manner. More concretely, our approach is to first show that p-Assign is correct on each of the variables that correspond to the internal states and incoming/outgoing messages of Round 1 of Π in every $\varphi^{\mu;\nu}$, next show it on each of the variables that correspond to those of Round 2 of Π in every $\varphi^{\mu;\nu}$, and so on.

Toward this end, we first remark that we design each 3SAT instance $\varphi^{\mu:\nu}$ carefully so that it has the following specific structure.

- 1. Let N_{round} be the round complexity of Π . Then, $\varphi^{\mu:\nu}$ has variables that can be partitioned into $4N_{\text{round}}$ sequences of variables, $\boldsymbol{w}_{1,\text{in}}^{\xi}, \boldsymbol{w}_{1,\text{out}}^{\xi}, \dots, \boldsymbol{w}_{N_{\text{round}},\text{in}}^{\xi}, \boldsymbol{w}_{N_{\text{round}},\text{out}}^{\xi}$ for $\xi \in \{\mu, \nu\}$, such that for each $\ell \in [N_{\text{round}}]$:
 - $\mathbf{w}_{\ell,\text{in}}^{\xi}$ is a sequence of variables such that the values that are assigned to them are interpreted as an internal state and incoming messages of \mathbf{P}^{ξ} at the beginning of Round ℓ .¹³
 - $\mathbf{w}_{\ell,\text{out}}^{\xi}$ is a sequence of variables such that the values that are assigned to them are interpreted as an internal state and outgoing messages of P^{ξ} at the end of Round ℓ .
- 2. $\varphi^{\mu:\nu}$ has clauses that check the following.
 - In each round, for each of P^{μ} and P^{ν} , its end state (i.e., its internal state at the end of the round) and outgoing messages are correctly derived from its start state (i.e., its internal state at the beginning of the round) and incoming messages.
 - In each round, for each of P^{μ} and P^{ν} , its start state is equal to its end state of the previous round.
 - In each round, P^{μ} 's incoming message from P^{ν} at the beginning of the round is equal to P^{ν} 's outgoing message to P^{μ} at the end of the previous round, and vise versa.
 - Both P^{μ} and P^{ν} output 1 in the last round.

We note that given consistent views of P^{μ} , P^{ν} in which they output 1, we can compute a satisfying assignment to the variables in $\varphi^{\mu;\nu}$ efficiently by obtaining each party's end state and outgoing messages of each round through the next-message function.

Now, we first show that if in every $\varphi^{\mu:\nu}$, p-Assign is correct on P^{μ} and P^{ν} 's start states and incoming messages in Round 1, then in every $\varphi^{\mu:\nu}$, p-Assign is also correct on P^{μ} and P^{ν} 's end states and outgoing messages in Round 1. A key observation on this step is that, essentially, what we need to show is that in every $\varphi^{\mu:\nu}$, for each $\xi \in \{\mu, \nu\}$, if p-Assign is correct on the input of P^{ξ} 's next-message computation of Round 1, then p-Assign is also correct on the output of it. Given this observation (and by designing the details of $\varphi^{\mu:\nu}$ appropriately), we can complete this step by just reusing the second part of the analysis of KRR no-signaling PCP, where it is shown that if Assign is correct on the input, then Assign is also correct on the output.

We next show that in every $\varphi^{\mu:\nu}$, if p-Assign is correct on P^{μ} and P^{ν} 's end states and outgoing messages in Round 1, then in every $\varphi^{\mu:\nu}$, p-Assign is also correct on P^{μ} and P^{ν} 's start states and incoming messages in Round 2. In this step, we consider three cases for each $\varphi^{\mu:\nu}$.

- Case 1. We first consider the correctness on P^{ξ} 's start state of Round 2 ($\xi \in \{\mu, \nu\}$). This case is easy and we just need to use the everywhere local consistency of p-Assign and the definition of $\varphi^{\mu;\nu}$. Specifically, since $\varphi^{\mu;\nu}$ has clauses that check that P^{ξ} 's start state of Round 2 is equal to its end state of Round 1, the everywhere local consistency of p-Assign guarantees that p-Assign assigns the same value on P^{ξ} 's start state of Round 2 and on P^{ξ} 's end state of Round 1, and thus, if p-Assign is correct on the latter, it is also correct on the former.
- Case 2. We next consider the correctness on P^{μ} 's incoming message from P^{ν} and P^{ν} 's incoming message from P^{μ} at the beginning of Round 2. Again, this case is easy and we just need to use the everywhere local consistency of p-Assign and the definition of $\varphi^{\mu,\nu}$ (which has clauses that check that the message that P^{μ} receives from P^{ν} at the beginning of Round 2 is equal to the one that P^{ν} sends to P^{μ} at the end of Round 1, and vise versa).

 $^{^{13}}$ We think that each round of Π starts when each party receives incoming messages from the other parties, and ends when each party sends outgoing messages to the other parties.

Case 3. We finally consider the correctness on P^{μ} and P^{ν} 's incoming messages from the parties other than P^{μ} and P^{ν} at the beginning of Round 2. This case is not straightforward, and we rely on that p-Assign satisfies the consistency on the views, which is guaranteed since p-Assign is obtained from a well-behaving prover. Let us consider, for example, P^{μ} 's incoming message from P^{ξ} ($\xi \notin \{\mu, \nu\}$). Then, since the consistency on the views guarantees that p-Assign assigns the same value in $\varphi^{\mu:\nu}$ and $\varphi^{\mu:\xi}$ as P^{μ} 's incoming message from P^{ξ} , if p-Assign is correct on it in $\varphi^{\mu:\xi}$, then p-Assign is also correct on it in $\varphi^{\mu:\nu}$. Then, since we showed in Case 2 that p-Assign is indeed correct on it in $\varphi^{\mu:\xi}$, we conclude that p-Assign is correct on it in $\varphi^{\mu:\nu}$. 14

By proceeding identically (and observing that, by definition, p-Assign is correct on P^{μ} and P^{ν} 's start states and incoming messages in Round 1 in every $\varphi^{\mu:\nu}$), we conclude that p-Assign is correct on any variable, and in particular correct on P^{μ} and P^{ν} 's final states in every $\varphi^{\mu:\nu}$.

Step 3: Obtaining soundness. On the one hand, the value that p-Assign assigns as the output of any party P^{μ} is always 1 due to the everywhere local consistency of p-Assign (recall that $\varphi^{\mu;\nu}$ has a clause that checks that P^{μ} 's output is 1). On the other hand, since p-Assign is correct on the output of P^{μ} , it is also equal to the value that P^{μ} outputs in the correct views. Thus, P^{μ} outputs 1 in the correct view, which means that the statement proven by the prover is true on the XOR of the MPC inputs of the correct views. From the definition of the extractor, it follows that the prover cannot prove false statements on the extracted value.

4.3.4 More details of well-behaving provers.

It remains to give an overview of the concrete definition of well-behaving provers. As we mentioned earlier, we define well-behaving provers so that when we convert a CNS well-behaving prover into a parallel local assignment generator p-Assign, then p-Assign has the aforementioned two consistency properties.

Before giving the definition of well-behaving provers, we give a few details about the construction of KRR no-signaling PCP.

- When a PCP proof π for a 3SAT instance φ is created by using a satisfying assignment x to φ , the PCP proof π contains an encoding of x, ¹⁵ i.e., there is a set of queries D(X) such that $\pi|_{D(X)}$ is an encoding of x.
- Furthermore, we can make sure that in our protocol, each PCP proof $\pi^{\mu:\nu}$ for $\varphi^{\mu:\nu}$ (where $\pi^{\mu:\nu}$ is created by using (view^{μ}, view^{ν})) contains encodings of $x^{\mu}_{1,\text{in}}$, $x^{\nu}_{1,\text{in}}$, view^{μ}, and view^{ν}, i.e., there are sets of queries $D(X^{\mu}_{1,\text{in}})$, $D(X^{\nu}_{1,\text{in}})$, $D(X^{\nu})$ such that:
 - $-\pi^{\mu:\nu}|_{D(X^{\mu}_{1:in})}$ and $\pi^{\mu:\nu}|_{D(X^{\nu}_{1:in})}$ are encodings of $x^{\mu}_{1:in}$ and $x^{\nu}_{1:in}$, respectively.
 - $-\pi^{\mu:\nu}|_{D(X^{\mu})}$ and $\pi^{\mu:\nu}|_{D(X^{\nu})}$ are encodings of view and view, respectively.

(Recall that $x_{1,\text{in}}^{\mu} \coloneqq \operatorname{st}_{0}^{\mu} \| \operatorname{i-msgs}_{1}^{\mu}$ and $x_{1,\text{in}}^{\nu} \coloneqq \operatorname{st}_{0}^{\nu} \| \operatorname{i-msgs}_{1}^{\nu}$ are the initial states and dummy incoming messages that are computed in the commit phase.)

Then, informally speaking, a CNS prover is said to be *well-behaving* if it satisfies the following two consistency properties.

Consistency on $D(X_{1,\text{in}}^{\mu})$. Once the commit phase is completed, the prover gives the same response to a query in $D(X_{1,\text{in}}^{\mu})$ ($\mu \in [M]$) in different invocations. More concretely, for any queries $\{Q_0^{\mu;\nu}\}_{\mu,\nu\in[M]}, \{Q_1^{\mu;\nu}\}_{\mu,\nu\in[M]}$, any $\alpha,\beta,\gamma,\delta\in[M]$ such that $\exists \xi \in \{\alpha,\beta\} \cap \{\gamma,\delta\}$, and any $q \in Q_0^{\alpha;\beta} \cap Q_1^{\gamma;\delta} \cap D(X_{1,\text{in}}^{\xi})$, we have $\pi_0^{*\alpha;\beta}(q) = \pi_1^{*\gamma;\delta}(q)$, where $\pi_0^{*\alpha;\beta}$ and $\pi_1^{*\gamma;\delta}$ are generated as follows.

- 1. $(st_C, com) \leftarrow \langle C.Com_1^*, R.Com_1 \rangle$
- 2. $(f_0, \{\pi_0^{*\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \text{C.Prv}_1^*(\text{st}_C, \{Q_0^{\mu:\nu}\}_{\mu,\nu\in[M]})$
- $3. \ (f_1, \{\pi_1^{*\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{C.Prv}_1^*(\mathsf{st}_C, \{Q_1^{\mu:\nu}\}_{\mu,\nu \in [M]})$

Consistency on $D(X^{\mu})$. The prover gives the same responses to a query in $D(X^{\mu})$ ($\mu \in [M]$) in a single invocation. More concretely, for any queries $\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}$, any $\alpha,\beta,\gamma,\delta\in[M]$ such that $\exists \xi\in\{\alpha,\beta\}\cap\{\gamma,\delta\}$, and any $q\in Q^{\alpha:\beta}\cap Q^{\gamma:\delta}\cap D(X^{\xi})$, we have $\pi^{*\alpha:\beta}(q)=\pi^{*\gamma:\delta}(q)$, where π^* is generated as follows.

- 1. $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C}.\mathsf{Com}_1^*, \mathsf{R}.\mathsf{Com}_1 \rangle$
- 2. $(f, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \text{C.Prv}_1^*(\text{st}_C, \{Q^{\mu:\nu}\}_{\mu,\nu\in[M]})$

¹⁴Note that we cannot use this argument if we try to reuse the analysis of Kalai et al. [KRR14] for each $\varphi^{\mu;\nu}$ individually (rather than in the round-by-round manner) since we show the correctness in $\varphi^{\mu;\nu}$ by using the correctness in $\varphi^{\mu;\varepsilon}$.

¹⁵Concretely, a low-degree extension of x (Section C.4).

To show that the above definition indeed implies the aforementioned two consistency properties of p-Assign, we need to see the details of p-Assign. Specifically, we rely on that p-Assign obtains local assignments by applying a procedure called *self-correction* on the cheating prover. In this overview, we do not give the details of self-correction, and we just note that p-Assign obtains local assignments in the following manner: p-Assign first creates some queries $Q^{\mu:\nu}$ for each $\mu, \nu \in [M]$ based on its input, next queries $\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}$ to the prover, and finally obtains the local assignments based on the prover's responses.

Now, at first sight, it seems trivial to show that the above definition of well-behaving provers implies the two consistency properties of p-Assign. Consider, for example, showing that the above definition implies that p-Assign has the consistency on the initial states. Then, since p-Assign obtains local assignments based on the prover's responses, and well-behaving provers are guaranteed to give unique responses to any queries on the initial states (i.e., any queries in $D(X_{1,\text{in}}^{\mu})$ ($\mu \in [M]$)), it seems trivial to show that p-Assign makes unique assignments on the initial states.

However, this intuition is wrong. For example, in the case of showing the consistency on the initial states, the problem is that even when making assignments on the initial states, p-Assign's queries to the prover includes those that are not in $D(X_{1 \text{ in}}^{\mu})$ ($\mu \in [M]$), and well-behaving provers' responses to such queries are not necessarily unique.

Fortunately, this problem can be solved relatively easily by using a technique in a previous work [HR18]. Specifically, by letting the verifier of KRR no-signaling PCP do several additional tests on the prover, we can show that it suffices to consider a modified version of p-Assign, which obtains local assignments on the initial states (resp., the views) based solely on the prover's responses to the queries in $D(X_{1,in}^{\mu})$ (resp., in $D(X^{\mu})$). On this modified version of p-Assign, it is indeed easy to show that the two consistency properties of well-behaving provers imply the two consistency properties of p-Assign by relying on analyses given in [KRR14].

4.3.5 Towards formal proof.

Finally, we discuss what modifications are needed to turn the above proof idea into a formal proof.

First, we need to modify the extractor so that it can open the commitment (instead of just extracting a committed value) as required in Definition 3; along the way, we also need to define the open phase of the protocol appropriately. Recall that in the above, the extractor uses the parallel local assignment generator p-Assign to extract a committed value. Motivated by this construction of the extractor, we follow the following overall approach: we define the open phase so that running p-Assign jointly with the receiver is sufficient for the committer to succeed in the open phase. To implement this approach, we rely on that, as mentioned above, p-Assign obtains local assignments in the following manner: p-Assign first creates some queries $Q^{\mu:\nu}$ for each $\mu, \nu \in [M]$ based on its input, next queries $\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}$ to the prover, and finally obtains the local assignments based on the prover's responses. Given this structure of p-Assign, we define the open phase as follows.

- 1. In the first round, the receiver computes queries as in p-Assign and sends them to the committer.
- 2. In the second round, the committer gives responses to the queries.
- 3. Finally, the receiver computes the local assignments from the responses as in p-Assign and then uses them to extract a committed value as in the extractor.

Then, we modify the extractor so that it simply forwards the queries from the receiver to the cheating prover and next forwards the responses from the cheating prover to the receiver. Since the extracted value is computed from the output of p-Assign just as before (the only difference is that now p-Assign is executed jointly between the extractor and the receiver), we can still prove that any CNS well-behaving cheating prover cannot prove false statements on the extracted value. Furthermore, we can show that the above open phase is strong enough to guarantee a meaningful binding property. Specifically, by letting the receiver make additional queries in the open phase,¹⁷ we can prove the binding property against CNS well-behaving decommitters, which are defined similarly to CNS well-behaving provers. (The proof of the binding property proceeds essentially in the same way as we show that p-Assign satisfies the consistency on the initial states in the proof of the soundness against well-behaving provers, where we show that once the commit phase is completed, the assignments by p-Assign on the MPC initial states—which define the committed value—are unique.)

Second, we need to consider the case that p-Assign can err (i.e., the everywhere local consistency does not necessarily hold with probability 1). Fortunately, this case is already handled in Kalai et al. [KRR14], and we can handle it identically. (Concretely, when showing that p-Assign is correct on every variable in the round-by-round way, we only show that p-Assign is correct on any variables that correspond to, say, the start state and incoming message of a round, we only show that p-Assign is correct on randomly chosen $\omega(\log \lambda)$ such variables. It is shown in [KRR14] that showing such average-case correctness is sufficient to prove the soundness.)

Third, we need to consider adaptive soundness, where the cheating prover chooses the statement to prove at the last round of the prove phase. Fortunately, adaptive soundness is already considered in previous works (e.g., [BHK17]), and we can handle it identically.

¹⁶Concretely, we use *layer-parallel low-degree tests* [HR18] to guarantee that the initial states (resp., the views) that are recovered through self-correction in p-Assign do not change when the queries are sampled from $D(X_{1,\text{in}}^{\mu})$ (resp., from $D(X^{\mu})$) rather than from D(X).

¹⁷Specifically, the receiver make queries for a low-degree test (just like the verifier of KRR succinct argument does) so that we can reuse analyses of Kalai et al. [KRR14] as in the proof of soundness.

5 Overview of Step 2 (Non-WI Scheme with Soundness against CNS Provers)

We give an overview of our non-WI commit-and-prove protocol $\langle C_2, R_2 \rangle$, which is (1 - negl)-sound against CNS provers. Our high-level approach is to upgrade the protocol $\langle C_1, R_1 \rangle$ that we give in Step 1 so that the soundness holds against any (not necessarily well-behaving) CNS provers. Recall that, roughly speaking, an adversary is well-behaving if for every $\mu \in [M]$,

- 1. it does not give different responses to a query in $D(X_{1,in}^{\mu})$ in different invocations, and
- 2. it does not give different responses to a query in $D(X^{\mu})$ in a single invocation,

where $D(X_{1,\text{in}}^{\mu})$ and $D(X^{\mu})$ are sets of queries such that in $\langle C_1, R_1 \rangle$, the prover is supposed to create PCPs $\{\pi^{\mu;\nu}\}_{\mu,\nu\in[M]}$ such that $\pi^{\mu;\nu}|_{D(X_{1,\text{in}}^{\mu})}$ is an encoding of $x_{1,\text{in}}^{\mu}$ and $\pi^{\mu;\nu}|_{D(X^{\mu})}$ is an encoding of view for every $\nu\in[M]$, where $x_{1,\text{in}}^{\mu}$ is the value that is fixed in the commit phase and view is the view that is fixed in the prove phase. Naturally, we enforce this behavior on the prover by relying on collision-resistant hash functions: we require the prover to publish the roots of the tree-hash of the encodings of $\{x_{1,\text{in}}^{\mu}\}_{\mu\in[M]}$ and $\{\text{view}^{\mu}\}_{\mu\in[M]}$, and also require it to give responses along with appropriate certificates when it is queried on these values.

More concretely, we consider the following protocol (which is slightly oversimplified from the actual protocol in Appendix F). In the following, for a hash function hf, we denote by TreeHash $_{hf}$ an algorithm that computes the Merkle tree-hash of the input.

Commit Phase

Round 1: R.Com₂ sends a hash function $hf \in \mathcal{H}$ to C.Com₂.

Round 2: Given $(x_{\text{COM}}, \text{hf})$ as input, C.Com_2 obtains $\{x_{1,\text{in}}^{\mu}\}_{\mu \in [M]}$ by running $\text{C.Com}_1(x_{\text{COM}})$, computes encodings $\{X_{1,\text{in}}^{\mu}\}_{\mu \in [M]}$ of them, and then outputs $\{\text{rt}_{1,\text{in}}^{\mu} := \text{TreeHash}_{\text{hf}}(X_{1,\text{in}}^{\mu})\}_{\mu \in [M]}$ as the commitment and store $(\text{hf}, \{X_{1,\text{in}}^{\mu}\}_{\mu \in [M]})$ as the internal state.

Prove Phase

Round 1: R.Prv.Q₂ works identically with R.Prv.Q₁. That is, R.Prv.Q₂ obtains $\{Q^{\mu,\nu}\}_{\mu,\nu\in[M]}$ just like R.Prv.Q₁ does, and outputs $\{Q^{\mu,\nu}\}_{\mu,\nu\in[M]}$ as the query.

Round 2: Given the statement f and the query $\{Q^{\mu,\nu}\}_{\mu,\nu\in[M]}$ as input, C.Prv₂ does the following.

- 1. Obtain $\{\mathsf{view}^{\mu}\}_{\mu\in[M]}$ and $\{\pi^{\mu:\nu}\}_{\mu,\nu\in[M]}$ just like C.Prv₁ does.
- 2. Compute encodings $\{X^{\mu}\}_{\mu\in[M]}$ of $\{\text{view}^{\mu}\}_{\mu\in[M]}$, and compute $\{\text{rt}^{\mu}:=\text{TreeHash}_{\text{hf}}(X^{\mu})\}_{\mu\in[M]}$.
- 3. Augment each $\pi^{\mu:\nu}$ as follows.
 - Augment each symbol in $\pi^{\mu:\nu}|_{D(X_{1,in}^{\xi})}$ $(\xi \in \{\mu, \nu\})$ with a certificate for opening $\mathsf{rt}_{1,in}^{\xi}$ to it.
 - Augment each symbol in $\pi^{\mu:\nu}|_{D(X^{\xi})\setminus D(X^{\xi}_{1:n})}$ ($\xi \in \{\mu, \nu\}$) with a certificate for opening rt^{ξ} to it.
- 4. Output $(\{\mathsf{rt}^{\mu}\}_{\mu \in [M]}, \{\pi^{\mu : \nu}|_{Q^{\mu : \nu}}\}_{\mu, \nu \in [M]})$ as the proof.

Verification: Given the commitment $\{\mathsf{rt}_{1,\mathsf{in}}^{\mu}\}_{\mu\in[M]}$, the statement f, and the proof $(\{\mathsf{rt}^{\mu}\}_{\mu\in[M]}, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]})$ as input, R.Prv.D₂ works identically with R.Prv.D₁ except that before the verification, each $\pi^{*\mu:\nu}$ is "filtered" as follows.

- Replace each symbol (x, cert) in $\pi^{*\mu:\nu}|_{D(X^{\xi}_{1,\text{in}})}$ $(\xi \in \{\mu, \nu\})$ with x if cert is a valid certificate for opening $\text{rt}_{1,\text{in}}^{\xi}$ to x, and replace it with \bot otherwise.
- Replace each symbol (x, cert) in $\pi^{*\mu,\nu}|_{D(X^{\xi})\setminus D(X^{\xi}_{1,\text{in}})}$ $(\xi \in \{\mu, \nu\})$ with x if cert is a valid certificate for opening rt^{ξ} to x, and replace it with \bot otherwise.

We prove the soundness of $\langle C_2, R_2 \rangle$ by relying on the soundness of $\langle C_1, R_1 \rangle$. Specifically, for any cheating committer-prover $C_2^* = (\mathsf{C.Com}_2^*, \mathsf{C.Prv}_2^*)$ against $\langle C_2, R_2 \rangle$, we consider the following cheating committer-prover $C_1^* = (\mathsf{C.Com}_1^*, \mathsf{C.Prv}_1^*)$ against $\langle C_1, R_1 \rangle$.

- Committer. C.Com₁* runs (st_C, com) $\leftarrow \langle \text{C.Com}_2^*, \text{R.Com}_2 \rangle$ internally, sends an empty string to R.Com₁ as the commitment, and stores (com, st_C) as the internal state.
- **Prover.** Given $(\text{com}, \text{st}_C)$ and $\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}$ as input, C.Prv_1^* first runs $(f, \{\text{rt}^{\mu}\}_{\mu\in[M]}, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \text{C.Prv}_2^*(\text{st}_C, \{Q^{\mu:\nu}\}_{\mu,\nu\in[M]})$. Then, C.Prv_1^* filters each $\pi^{*\mu:\nu}$ as in the verification of $\langle C_2, R_2 \rangle$, and sends $(f, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]})$ to R.Prv_1 as the proof.

It is straightforward to show that (1) C_1^* is successful if C_2^* is successful and (2) C_1^* is well-behaving CNS. (The latter follows from the binding property of TreeHash_{hf}.)

6 Overview of Subsequent Steps of Proof of Theorem 1

In Step 3, we upgrade the soundness to the one with negligible soundness error. Fortunately, this type of soundness amplification is already studied by Kalai et al. [KRR14] as mentioned in Section 3, and it suffices to apply their soundness amplification on the protocol $\langle C_2, R_2 \rangle$ that we obtained in Step 2. Concretely, in this step, we just borrow a soundness amplification technique from [KRR14, BHK17], which amplifies soundness by letting the verifier use a smaller threshold parameter for the PCP decision algorithm (i.e., letting the verifier tolerate a smaller number of failures on the tests that it applies on the prover).

In Step 4, we upgrade the soundness to the one against any (not necessarily CNS) adversaries. Again, this type of soundness amplification is already studied by Kalai et al. [KRR14] as mentioned in Section 3, and it suffices to apply their soundness amplification on the protocol $\langle C_3, R_3 \rangle$ that we obtained in Step 3. Concretely, in this step, we just borrow a transformation from [KRR14], which enforces CNS behavior on the committer by encrypting the verifier queries by PIR. (Intuitively, encrypting the verifier queries by PIR is helpful to enforce CNS behavior since it forces the prover to answer each query independently of the other queries.)

In Step 5, we add the WI property while tolerating that the soundness error increases to a constant. Toward this end, we augment the protocol $\langle C_4, R_4 \rangle$ that we obtained in Step 4 with commitment schemes and OT by using these two primitives as in the non-succinct protocol that we sketched in Section 1.2. The soundness and WI of the resultant protocol $\langle C_5, R_5 \rangle$ can be shown similarly to those of the non-succinct protocol in Section 1.2. That is, the soundness follows from the security of OT and the soundness of $\langle C_4, R_4 \rangle$, ¹⁸ and the WI property follows from the 2-privacy of $\langle C_4, R_4 \rangle$, which roughly guarantees that the verifier does not learn any secret information if it only obtains one of the M^2 KRR no-signaling PCP strings. (The 2-privacy of $\langle C_4, R_4 \rangle$, in turn, follows from the 2-privacy of the underlying MPC protocol Π .)

References

- [AL17] Gilad Asharov and Yehuda Lindell. A full proof of the BGW protocol for perfectly secure multiparty computation. *Journal of Cryptology*, 30(1):58–151, January 2017.
- [BBK⁺16] Nir Bitansky, Zvika Brakerski, Yael Tauman Kalai, Omer Paneth, and Vinod Vaikuntanathan. 3-message zero knowledge against human ignorance. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B*, *Part I*, volume 9985 of *LNCS*, pages 57–83. Springer, Heidelberg, October / November 2016.
- [BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988.
- [BHK16] Zvika Brakerski, Justin Holmgren, and Yael Kalai. Non-interactive RAM and batch NP delegation from any PIR. Cryptology ePrint Archive, Report 2016/459, 2016. http://eprint.iacr.org/2016/459.
- [BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, 49th ACM STOC, pages 474–482. ACM Press, June 2017.
- [BK18] Zvika Brakerski and Yael Tauman Kalai. Monotone batch NP-delegation with applications to access control. Cryptology ePrint Archive, Report 2018/375, 2018. https://eprint.iacr.org/2018/375.
- [BK20] Zvika Brakerski and Yael Kalai. Witness indistinguishability for any single-round argument with applications to access control. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 97–123. Springer, Heidelberg, May 2020.
- [BKK⁺18] Saikrishna Badrinarayanan, Yael Tauman Kalai, Dakshita Khurana, Amit Sahai, and Daniel Wichs. Succinct delegation for low-space non-deterministic computation. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 709–721. ACM Press, June 2018.
- [CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multiparty secure computation. In *34th ACM STOC*, pages 494–503. ACM Press, May 2002.
- [GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 465–482. Springer, Heidelberg, August 2010.
- [GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

¹⁸Formally, as in the case of the non-succinct protocol in Section 1.2, complexity leveraging is required.

- [GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *Journal of the ACM*, 62(4):27:1–27:64, 2015.
- [GLOV12] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *53rd FOCS*, pages 51–60. IEEE Computer Society Press, October 2012.
- [GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.
- [GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.
- [GOSV14] Vipul Goyal, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Black-box non-black-box zero knowledge. In David B. Shmoys, editor, *46th ACM STOC*, pages 515–524. ACM Press, May / June 2014.
- [HR18] Justin Holmgren and Ron Rothblum. Delegating computations with (almost) minimal time and space overhead. In Mikkel Thorup, editor, *59th FOCS*, pages 124–135. IEEE Computer Society Press, October 2018.
- [HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On the power of secure two-party computation. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016*, *Part II*, volume 9815 of *LNCS*, pages 397–429. Springer, Heidelberg, August 2016.
- [HV18] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. Round-optimal fully black-box zero-knowledge arguments from one-way permutations. In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 263–285. Springer, Heidelberg, November 2018.
- [IKOS09] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge proofs from secure multiparty computation. *SIAM Journal on Computing*, 39(3):1121–1152, 2009.
- [IW14] Yuval Ishai and Mor Weiss. Probabilistically checkable proofs of proximity with zero-knowledge. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 121–145. Springer, Heidelberg, February 2014.
- [Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In 24th ACM STOC, pages 723–732. ACM Press, May 1992.
- [Kiy18] Susumu Kiyoshima. No-signaling linear PCPs. Cryptology ePrint Archive, Report 2018/649, 2018. https://eprint.iacr.org/2018/649.
- [KOS18] Dakshita Khurana, Rafail Ostrovsky, and Akshayaram Srinivasan. Round optimal black-box "commit-and-prove". In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 286–313. Springer, Heidelberg, November 2018.
- [KP16] Yael Tauman Kalai and Omer Paneth. Delegating RAM computations. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 91–118. Springer, Heidelberg, October / November 2016.
- [KRR13] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: The power of no-signaling proofs. Cryptology ePrint Archive, Report 2013/862, 2013. http://eprint.iacr.org/2013/862.
- [KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In David B. Shmoys, editor, *46th ACM STOC*, pages 485–494. ACM Press, May / June 2014.
- [LP12] Huijia Lin and Rafael Pass. Black-box constructions of composable protocols without set-up. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 461–478. Springer, Heidelberg, August 2012.
- [MPP20] Andrew Morgan, Rafael Pass, and Antigoni Polychroniadou. Succinct non-interactive secure computation. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 216–245. Springer, Heidelberg, May 2020.
- [ORS15] Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015*, *Part II*, volume 9216 of *LNCS*, pages 339–358. Springer, Heidelberg, August 2015.
- [PR14] Omer Paneth and Guy N. Rothblum. Publicly verifiable non-interactive arguments for delegating computation. Cryptology ePrint Archive, Report 2014/981, 2014. http://eprint.iacr.org/2014/981.

- [PR17] Omer Paneth and Guy N. Rothblum. On zero-testable homomorphic encryption and publicly verifiable non-interactive arguments. Cryptology ePrint Archive, Report 2017/903, 2017. http://eprint.iacr.org/2017/903.
- [PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 403–418. Springer, Heidelberg, March 2009.
- [Sud00] Madhu Sudan. Probabilistically checkable proofs, 2000. http://madhu.seas.harvard.edu/MIT/pcp/pcp.ps.

A On Verification Time

The verification of our protocol is not succinct since we use a simpler version of KRR succinct argument where the verifier naively evaluates a low-degree extension (LDE) of the indicator function of a 3CNF formula whose size is polynomially related to the complexity of the statement. In Kalai et al. [KRR14] and subsequent works [BHK17, HR18], the verification is made succinct by observing that when the statement to be proven satisfies some conditions, the evaluation of the LDE can be either recursively delegated to the prover succinctly or locally performed by the verifier efficiently. In our protocol, KRR succinct argument is used for proving statements that are related to the next-message function of the underlying perfect 2-privacy MPC protocol (cf. Section 1.2, Section 4, and Appendix D). Thus, if we can show that the above-mentioned conditions are satisfied for a specific perfect 2-privacy MPC protocol, the verification of our protocol can be made succinct.

B On Definition of Commit-and-prove Protocols

B.1 Differences from Definition in Khurana et al. [KOS18]

Our definition of commit-and-prove protocols in Section 2.2 has several differences from the definition in Khurana et al. [KOS18]. First, our definition has several syntactical differences.

- Instead of thinking the prove phase as a part of the commit phase, we separate the prove phase from the commit phase.
- We focus on the case that each of the prove phase and the open phase consists of two rounds.

Next, our definition is stronger than the definition of Khurana et al. [KOS18] in the following points.

- We explicitly define the soundness and witness indistinguishability in the delayed-input setting, where the statement to be proven is chosen at the last round of the prove phase.
- In the definition of the soundness, we require the extractor to decommit the commitment to a value on which any committer cannot prove false statements. (In the definition in [KOS18], the extractor just outputs such a value without decommitment, with the guarantee that any committer cannot decommit the commitment to a value other than the extracted one.)

We think that requiring the extractor to decommit the commitment is important, as otherwise the definition would not prevent an attack where the committer gives an accepting proof on an invalid commitment (i.e., a commitment that cannot be opened to any value). (This is because even if such an attack is possible, we can still show that any committer cannot decommit the commitment to a value other than the extracted one, since an invalid commitment cannot be opened to any value.) We remark that such an attack is possible if a commit-and-prove protocol is naively executed in parallel multiple times.

Finally, our definition is weaker than the definition of Khurana et al. [KOS18] in the following points.

- In the definitions of the binding and the soundness, the extractor succeeds only on an overwhelming fraction of the executions of the commit phase, rather than on any execution of the commit phase.
- In the definition of the soundness, the extractor is allowed to depend on the success probability of the cheating committer.

¹⁹We remark that the schemes by Khurana et al. [KOS18] are designed to prevent such an attack. What we claim is that the definition by Khurana et al. [KOS18] does not prevent such an attack.

B.2 Rationale behind Our Definition

Since our definition of commit-and-prove protocols in Section 2.2 might look too cumbersome, we explain the rationale behind it.

First, binding and witness-indistinguishability are defined naturally, and the only complication is that we allow the open phase to be interactive in the definition of binding. To guarantee a stronger notion of binding, our definition considers an adversary that obtains two sets of receiver decommitment queries simultaneously (rather than obtain each of them separately).

Next, soundness is defined similarly to proof-of-knowledge of interactive proofs [GMR89]. A complication is that we define it so that it guarantees the adaptive delayed-input property, i.e., it holds against an adversary that chooses the statement to prove at the last round of the prove phase. To guarantee proof-of-knowledge with the adaptive delayed-input property, our definition requires that if a cheating prover convinces the verifier for a commitment com with sufficiently high probability, then there exists a value x^* such that (1) the extractor can decommit com to x^* and (2) the cheating prover cannot prove false statements about x^* . (We remark that the extractor is required to succeed in the decommitment of each bit of x^* with overwhelming probability so that we can obtain the whole x^* by repeatedly using the extractor.)

C Additional Preliminaries

C.1 Oblivious Transfer Protocols

We recall the definition of 1-out-of-*n* oblivious transfer (OT) protocols for a constant *n*, based on the description by Brakerski and Kalai [BK18] while straightforwardly generalizing the description from 1-out-of-2 OT to 1-out-of-*n* OT.

We focus on two-round OT protocols. They consist of three PPT algorithms, (OT_1, OT_2, OT_3) , such that OT_1 is used by the receiver to send the first-round message, OT_2 is used by the sender to send the second-round message, and OT_3 is used by the receiver to compute the output from the sender message.

Definition 11 (Correctness). We say that a 1-out-of-n OT protocol OT = (OT_1, OT_2, OT_3) satisfies correctness if for all $\lambda \in \mathbb{N}$, $i \in [n]$, and $(m_1, \ldots, m_n) \in (\{0, 1\}^{\lambda})^n$,

$$\Pr\left[\mathsf{OT}_{3}(\mathsf{st}_{o\tau},\mathsf{ot}_{3}) = m_{i} \middle| \begin{array}{c} (\mathsf{ot}_{1},\mathsf{st}_{o\tau}) \leftarrow \mathsf{OT}_{1}(1^{\lambda},i) \\ \mathsf{ot}_{2} \leftarrow \mathsf{OT}_{2}(1^{\lambda},\mathsf{ot}_{1},(m_{1},\ldots,m_{n})) \end{array} \right] = 1 .$$

Definition 12 (Receiver privacy²¹). We say that a 1-out-of-n OT protocol $OT = (OT_1, OT_2, OT_3)$ satisfies receiver privacy if for any PPT distinguisher \mathcal{D} and any $\lambda \in \mathbb{N}$,

$$\Pr\left[\mathcal{D}(\mathsf{ot}_1,i_b) = b \;\middle|\; \begin{array}{c} \textit{Sample random } b \in \{0,1\} \textit{ and } i_0,i_1 \in [n] \\ (\mathsf{ot}_1,\mathsf{st}_{o\tau}) \leftarrow \mathsf{OT}_1(1^\lambda,i_0) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \enspace .$$

Definition 13 (Sender privacy). We say that a 1-out-of-n OT protocol OT = (OT_1, OT_2, OT_3) satisfies sender privacy if for all $\lambda \in \mathbb{N}$ and for all $OT_1^* \in \{0, 1\}^{|OT_1|}$, there exists $i^* \in [n]$ such that for any $(m_1, \ldots, m_n) \in (\{0, 1\}^{\lambda})^n$ and $(m_1', \ldots, m_n') \in (\{0, 1\}^{\lambda})^n$ such that $m_{i^*} = m_{i^*}'$, the distributions $OT_2(1^{\lambda}, OT_1^*, (m_1, \ldots, m_n))$ and $OT_2(1^{\lambda}, OT_1^*, (m_1', \ldots, m_n'))$ are computationally indistinguishable. We denote by Ext_{OT} an arbitrary computationally unbounded procedure that produces i^* as above from OT_1^* , i.e., $i^* := Ext_{OT}(1^{\lambda}, OT_1^*)$

C.2 Private Information Retrieval

We recall the definition of private information retrieval (PIR) schemes, based on the description by Brakerski et al. [BHK17].

Definition 14. A 2-message private information retrieval (PIR) scheme over alphabet $\{\Sigma_{\lambda}\}_{{\lambda}\in\mathbb{N}}$, with $\log|\Sigma_{\lambda}| \leq \text{poly}(\lambda)$, is a tuple of PPT algorithms (PIR.Enc, PIR.Res, PIR.Dec) with the following syntax.

- $(q, \operatorname{st}_{PIR}) \leftarrow \operatorname{PIR.Enc}(1^{\lambda}, q, N)$: given 1^{λ} and some $q, N \in \mathbb{N}$ such that $q \leq N$, $\operatorname{PIR.Enc}$ outputs a query string q and a secret state st_{PIR} .
- $x \leftarrow \mathsf{PIR.Res}(1^\lambda, q, DB)$: given a query string q and a database $DB \in \Sigma_1^N$, $\mathsf{PIR.Res}$ outputs a response string x.
- $x \leftarrow \mathsf{PIR.Dec}(\mathsf{st}_{\mathsf{PIR}}, \mathbb{z})$: given a secret state $\mathsf{st}_{\mathsf{PIR}}$ and a response string \mathbb{z} , $\mathsf{PIR.Dec}$ outputs an element $x \in \Sigma_{\lambda}$.

We say that the scheme is a polylogarithmic PIR if $|\mathbf{x}| = \mathsf{poly}(\lambda, \log N)$. We say that it is (perfectly) correct if for all $q \leq N \leq 2^{\lambda}$ and $DB \in \Sigma_{\lambda}^{N}$ it holds that when setting $(\mathbb{Q}, \mathsf{st}_{PIR}) \leftarrow \mathsf{PIR}.\mathsf{Enc}(1^{\lambda}, q, N), \ \mathbf{x} \leftarrow \mathsf{PIR}.\mathsf{Res}(1^{\lambda}, \mathbb{Q}, DB), \ \mathbf{x} \leftarrow \mathsf{PIR}.\mathsf{Dec}(\mathsf{st}_{PIR}, \mathbf{x}),$ then $\mathbf{x} = DB(q)$ holds with probability 1. We say that the scheme is secure if for any sequence of $\{N_{\lambda}, q_{\lambda}, q_{\lambda}'\}_{\lambda \in \mathbb{N}}$ such that $q_{\lambda}, q_{\lambda}' \leq N_{\lambda}$, it holds $\mathbb{Q} \approx \mathbb{Q}'$, where $(\mathbb{Q}, \mathsf{st}_{PIR}) \leftarrow \mathsf{PIR}.\mathsf{Enc}(1^{\lambda}, q_{\lambda}, N_{\lambda})$ and $(\mathbb{Q}', \mathsf{st}'_{PIR}) \leftarrow \mathsf{PIR}.\mathsf{Enc}(1^{\lambda}, q_{\lambda}', N_{\lambda}).$

²⁰The adaptive delayed-input property is required to, e.g., upgrade our WI commit-and-prove protocol to a ZK one by using the transformation of Khurana et al. [KOS18].

²¹Our version of the receiver privacy might look unusual. Yet, it is easy to see that our version is implied by the standard one (e.g., the one in [BK18]).

C.3 Computational No-signaling (Parallel Version)

Definition 15. Fix any constant $M \in \mathbb{N}$, any alphabet $\{\Sigma_{\lambda}\}_{{\lambda} \in \mathbb{N}}$, any $\{N_{\lambda}\}_{{\lambda} \in \mathbb{N}}$ such that $N_{\lambda} \in \mathbb{N}$, any function $\kappa_{\max} : \mathbb{N} \to \mathbb{N}$ such that $\kappa_{\max}(\lambda) \leq N_{\lambda}$, and any algorithm Algo such that for any $\lambda \in \mathbb{N}$, on input subsets $Q^1, \ldots, Q^M \subset [N_{\lambda}]$ of size at most $\kappa_{\max}(\lambda)$, Algo outputs (the truth tables of) functions $A^1, \ldots, A^M : Q \to \Sigma \cup \{\bot\}$ with an auxiliary output out.

Then, the algorithm Algo is adaptive κ_{\max} -computational no-signaling (CNS) if for any PPT distinguisher \mathcal{D} , any sufficiently large $\lambda \in \mathbb{N}$, any $Q^i, S^i \subset [N_{\lambda}]$ such that $Q^i \subseteq S^i$ and $|S^i| \leq \kappa_{\max}(\lambda)$ for $\forall i \in [M]$, and any $z \in \{0, 1\}^{\mathsf{poly}(\lambda)}$,

$$\left| \begin{array}{l} \Pr\left[\mathcal{D}(\mathsf{out}, \{A^i\}_{i \in [M]}, z) = 1 \;\middle|\; (\mathsf{out}, \{A^i\}_{i \in [M]}) \leftarrow \mathsf{Algo}(\{Q^i\}_{i \in [M]}) \right] \\ -\Pr\left[\mathcal{D}(\mathsf{out}, \{A^i|_{Q^i}\}_{i \in [M]}, z) = 1 \;\middle|\; (\mathsf{out}, \{A^i\}_{i \in [M]}) \leftarrow \mathsf{Algo}(\{S^i\}_{i \in [M]}) \right] \end{array} \right| \leq \mathsf{negl}(\lambda) \enspace .$$

We remark that the above definition can be naturally extended for the case that Algo takes auxiliary inputs.

C.4 Low-degree Extension (LDE)

Let F be a finite field, and let H be a subset of F, and let m be an integer. Any function $f: H^m \to \{0, 1\}$ can be extended into a function $\hat{f}: F^m \to F$ such that \hat{f} satisfies $\hat{f}|_{H^m} \equiv f$ and is an m-variate polynomial of degree at most (|H| - 1) in each variable; the function \hat{f} is called a *low-degree extension* (LDE) of f.

Low-degree extensions of strings. An LDE of a binary string x of length N can be obtained by choosing H and m such that $N \le |H|^m$, identifying $\{1, \ldots, |H|^m\}$ with H^m in a canonical way, and viewing x as a function $x : H^m \to \{0, 1\}$ such that $x(i) = x_i$ for $\forall i \in [N]$ and x(i) = 0 for $\forall i \in \{N + 1, \ldots, |H|^m\}$

C.5 Threshold Verifiers

We recall the definition of threshold verifiers [BHK17], which are used in the description and analysis of the PCP system of Kalai et al. [KRR14].

Definition 16. Given a PCP verifier V = (Q, D), the t-of-n threshold verifier $(Q^{\otimes n}, D^{\geq t})$ (where both n and t may be functions of λ) is defined as follows. $Q^{\otimes n}$ takes a security parameter 1^{λ} as input and does the following:

- 1. Compute $(Q_i, st_i) \leftarrow Q(1^{\lambda})$ for each $i \in [n]$.
- 2. Output $\bigcup_{i=1}^{n} Q_i$ as the query and $((Q_1, \mathsf{st}_1), \ldots, (Q_n, \mathsf{st}_n))$ as the state.

 $\mathsf{D}^{\geq t}$ takes input $(((Q_1,\mathsf{st}_1),\ldots,(Q_n,\mathsf{st}_n)),x,\pi)$ and does the following:

- 1. Compute $y_i \leftarrow \mathsf{D}(\mathsf{st}_i, x, \pi|_{Q_i})$ for each $i \in [n]$.
- 2. Output 1 if at least t of the y_i 's are 1; otherwise outputs 0.

For the special case where t = n, we write $D^{\otimes n}$ instead of $D^{\geq n}$.

D Step 0: PCP for Checking View Consistency

Before starting to construct our commit-and-prove protocol, we first introduce the main building block, the PCP system (PCP.P, PCP.V) for checking the consistency of a pair of views of the MPC protocol Π. This PCP system is a variant of the PCP system of Kalai et al. [KRR14] (which is the main building block of their succinct argument), and the differences are that the statement to be proven is restricted to a specific form of 3SAT instances and that the verification includes several additional tests.

D.1 Preliminaries: Results from Kalai et al. [KRR14] and Subsequent Works

We recall some results from Kalai et al. [KRR14] and its subsequent works.

The first is a lemma that gives a PCP system (PCP.P_{KRR}, PCP.V_{KRR}) for 3SAT such that any successful adaptive CNS prover can be converted into an adaptive local assignment generator.

Lemma D.1. There exists a PCP system (PCP.P_{KRR}, PCP.V_{KRR}) for 3SAT, where PCP.V_{KRR} = (PCP.Q_{KRR}, PCP.D_{KRR}), that satisfies the following soundness property: There exists a PPT oracle machine Assign and a polynomial 22 κ_0 such that

 $^{^{22}\}kappa_0$ is a polynomial in λ and log N, where N is the number of the variables in the 3CNF formula given as the statement. Note that when N is expressed as a polynomial in λ , κ_0 is a polynomial in λ .

for every negligible function ϵ , every polynomial κ_{max} , and every adaptive $(\kappa_0 \cdot \kappa_{max})$ -CNS cheating prover PCP.P*, if it holds

$$\Pr \left[b = 1 \; \middle| \; \begin{array}{l} (\mathsf{st}, Q) \leftarrow \mathsf{PCP}.\mathsf{Q}_{\mathsf{KRR}}^{\otimes \lambda}(1^{\lambda}); \; (\varphi, \pi) \leftarrow \mathsf{PCP}.\mathsf{P}^*(Q) \\ b \coloneqq \mathsf{PCP}.\mathsf{D}_{\mathsf{KRR}}^{\geq \lambda - \zeta}(\mathsf{st}, \varphi, \pi) \end{array} \right] \geq 1 - \epsilon(\lambda),$$

for $\zeta = \omega(\log \lambda)$ for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of such λ), then $\mathsf{Assign}^{\mathsf{PCP},\mathsf{P}^*}$ is an adaptive κ_{max} -local assignment generator for every sufficiently large $\lambda \in \Lambda$. Moreover, the distribution of φ that is generated by $(\varphi, A) \leftarrow \mathsf{Assign}^{\mathsf{PCP},\mathsf{P}^*}(W)$ for any W is computationally indistinguishable from the distribution of φ that is generated by $\mathsf{PCP},\mathsf{P}^*$ as above.

П

The next is a lemma that describes properties of (PCP.P_{KRR}, PCP.V_{KRR}).

Lemma D.2. The PCP system (PCP.P_{KRR}, PCP.V_{KRR}) and the oracle machine Assign given in Lemma D.1 satisfy the following properties, where in the following, N is used to denote the number of the variables in the 3SAT instance that is given to (PCP.P_{KRR}, PCP.V_{KRR}) as the statement.

- (PCP.P_{KRR}, PCP.V_{KRR}) has \mathbf{F} , \mathbf{H} , and \mathbf{m} as parameters, where \mathbf{F} is a finite field of size $O(\log^2 N) \leq |\mathbf{F}| \leq \operatorname{polylog}(N)$, and \mathbf{H} and \mathbf{m} are defined as $\mathbf{H} := \{0, \dots, \Theta(\log N) 1\}$ and $\mathbf{m} := \lceil \log N / \log |\mathbf{H}| \rceil$ so that $|\mathbf{H}|^m \geq N$.
- When PCP.P_{KRR} generates a proof π for a 3SAT instance φ by using a satisfying assignment $\mathbf{x} \in \{0, 1\}^N$ as a witness, the proof π consists of several low-degree polynomials, and one of these polynomials, denoted as X, is an LDE (w.r.t. \mathbf{F} , \mathbf{H} , m) of \mathbf{x} .
- The query complexity of PCP.V_{KRR} is at most $O(m|F|^2) \le \text{polylog}(N)$, and the polynomial κ_0 given in Lemma D.1 satisfies $\kappa_0 = O(\lambda |F|^2)$.
- Assign is defined as Assign := SelfCorr_{m|H|,D(X)} for the oracle machine SelfCorr in Algorithm 1, where $D(X) := F^m$ is the domain of X.²³

Proof. See, e.g., the proof of [BHK16, Lemma 6].

The last is a technical lemma. Roughly speaking, it says that any function f can be converted into a 3CNF formula φ with the following property: φ is a 3CNF formula that checks whether the n-bit value x given as "the input value" and the n-bit value y given as "the output value" satisfy y = f(x); furthermore, it is carefully designed so that for any local assignment generator Assign for φ , if the assignment by Assign to "the input value" agrees with x, then the assignment by Assign to "the output value" must agree with f(x). The formal statement is given below.

Lemma D.3. Fix any polynomials N' and n. There exist PPT algorithms Aug, Aug⁻¹ and a polynomial²⁴ κ_{max} that satisfy the following properties.

Syntax. Fix the security parameter λ , and let $N' := N'(\lambda)$ and $n := n(\lambda)$. Aug takes as input a function $f : \{0, 1\}^n \to \{0, 1\}^n$ such that $\mathsf{Time}(f) = N'$, and outputs a 3CNF formula φ of $N = \mathsf{poly}(N')$ variables. Aug^{-1} takes as input a 3CNF formula φ , and outputs a function f such that $\mathsf{Aug}(f) = \varphi$ if such f exists, and outputs \perp otherwise.

Satisfiability. Given any f and $x = (x_1, ..., x_n) \in \{0, 1\}^n$, one can efficiently compute a satisfying assignment to the variables in $\varphi := \operatorname{Aug}(f)$. Furthermore, the resultant satisfying assignment satisfies the following.

- The assignment to the first n variables, $\mathbf{w}_{in} = (w_1, \dots, w_n)$, is $x = (x_1, \dots, x_n)$.
- There exists a sequence of variables $w_{in,LDE}$ such that the assignment to $w_{in,LDE}$ is the LDE of x w.r.t. some predetermined F', H', m'.
- There exists a sequence of variables $\mathbf{w}_{\text{out,LDE}}$ such that the assignment to $\mathbf{w}_{\text{out,LDE}}$ is the LDE of $f(x) = (y_1, \dots, y_n)$ w.r.t. $\mathbf{F}', \mathbf{H}', \mathbf{m}'$, and therefore there exists a sequence of n variables $\mathbf{w}_{\text{out}} \subset \mathbf{w}_{\text{out,LDE}}$ such that the assignment to \mathbf{w}_{out} is $f(x) = (y_1, \dots, y_n)$.

Let $I_{in} = \{1, ..., n\}$ denote the set of indices of the variables in \mathbf{w}_{in} , $I_{in,LDE}$ denote the set of indices of the variables in $\mathbf{w}_{in,LDE}$, $I_{out,LDE}$ denote the set of indices of the variables in $\mathbf{w}_{out,LDE}$, and I_{out} denote the set of indices of the variables in \mathbf{w}_{out} .

Security. If there exists a PPT machine Assign that is an adaptive κ_{max} -local assignment generator for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of such λ) and it always outputs a 3CNF formula φ such that $\operatorname{Aug}^{-1}(\varphi) \neq \bot$, then the following three claims hold.

²³Roughly speaking, SelfCorr $_{m|H|,D(X)}^{PCP,P^*}$ does "self-correction" (see, e.g., [Sud00, Lecture 2, Section 4]) on X by making queries to PCP.P*, where m|H| is an upper bound on the total degree of X.

 $^{^{24}\}kappa_{\rm max}$ is a polynomial in λ and log N just like κ_0 ; see Footnote 22.

 $^{^{25}}I_{\text{in}}$, $I_{\text{in,LDE}}$, $I_{\text{out,LDE}}$ and I_{out} only depend on N' and n, and thus are fixed before f, x are fixed.

Algorithm 1 Self-Correction Procedure SelfCorr $_{d,D}^{PCP,P}$

Parameter: $d \in \mathbb{N}$ is an integer, and $D \subseteq D(X)$ is a subset of the domain $D(X) = \mathbf{F}^m$ of X. (In this paper, D is always of the form $\{(c_1, \ldots, c_k, z_{k+1}, \ldots, z_m) \mid z_{k+1}, \ldots, z_m \in \mathbf{F}\} \subseteq \mathbf{F}^m$ for some constants $k \in \{0, \ldots, m\}$ and $c_1, \ldots, c_k \in \mathbf{F}$.) **Input:** $Q \subseteq [N]$, which is a subset of the indices of the variables in a 3SAT instance, and is thought of as a subset of $\mathbf{H}^m \subset \mathbf{F}^m = D(X)$ by a canonical mapping.

- 1. Run $(\tilde{Q}, \operatorname{st}_{Q}) \leftarrow \operatorname{SelfCorr.Q}_{D}(Q)$.
- 2. Run $(\varphi, \pi^*) \leftarrow \mathsf{PCP.P}^*(\tilde{Q})$.
- 3. Run $A := SelfCorr.Rec_d(st_O, \pi^*)$.
- 4. Output (φ, A) .

.....

Subroutine SelfCorr. $Q_D(Q)$:

- 1. For each $q \in Q$, choose λ random lines $L_{q,1}, \ldots, L_{q,\lambda} : F \to D$ such that each $L \in \{L_{q,1}, \ldots, L_{q,\lambda}\}$ satisfies L(0) = q. (Abort if $q \notin D$.)
- 2. Output $(\tilde{Q}, \operatorname{st}_Q)$, where $\tilde{Q} := \{L_{q,j}(t)\}_{q \in Q, j \in [\lambda], t \in F \setminus \{0\}}$ and $\operatorname{st}_Q := (Q, \{L_{q,j}\}_{q \in Q, j \in [\lambda]})$.

Subroutine SelfCorr.Rec_d(st_Q, X^*):

1. Define $A:Q\to F\cup\{\bot\}$ as follows. For each $q\in Q$, check that there exists $c_q\in F$ such that

$$\left|\left\{j \in [\lambda] \;\middle|\; \mathsf{Recon}_d\left(\left\{X^*(L_{q,j}(t))\right\}_{t \in F \setminus \{0\}}\right) = c_q\right\}\right| \geq 0.9\lambda \ ,$$

let $A(q) := c_q$ if such c_q exists, and let $A(q) := \bot$ otherwise.

2. Output A.

Subroutine Recon_d($\{z_t\}_{t \in F \setminus \{0\}}$):

- 1. Obtain a degree-d polynomial P that satisfies $P(t) = z_t$ for $\forall t \in F \setminus \{0\}$ through interpolation. (Output \bot if the interpolation fails or $\bot \in \{z_t\}_{t \in F \setminus \{0\}}$.)
- 2. Output P(0).

Claim D.1 (From average correctness of input to average correctness of output). There exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$ and every $x \in \{0, 1\}^n$, it holds

$$\Pr \left[\begin{array}{c} \mathsf{Correct}(S) \\ \land \neg \mathsf{Correct}(T) \end{array} \right| \left[\begin{array}{c} S \coloneqq \{s_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ \textit{where} \ s_i \leftarrow \mathcal{I}_{\mathsf{in},\mathsf{LDE}} \\ T \coloneqq \{t_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ \textit{where} \ t_i \leftarrow \mathcal{I}_{\mathsf{out},\mathsf{LDE}} \\ (\varphi, A) \leftarrow \mathsf{Assign}(S \cup T) \end{array} \right] \leq N \cdot \mathsf{negl}(\lambda) \ ,$$

where the events Correct(S) and Correct(T) are defined as follows. Given φ and $f := Aug^{-1}(\varphi)$, let the correct assignment A_{corr} (to the variables in φ) be the assignment that is obtained from x as above. Then, Correct(S) is the event that $A(s) = A_{corr}(s)$ holds for $\forall s \in S$ and Correct(T) is the event that $A(t) = A_{corr}(t)$ holds for $\forall t \in T$.

Claim D.2 (From average correctness of output to worst-case correctness of output). *There exists a negligible func*tion negl such that for every sufficiently large $\lambda \in \Lambda$, every $x \in \{0,1\}^n$, and every $i^* \in \mathcal{I}_{out}$, it holds

$$\Pr \left[\begin{array}{c|c} \mathsf{Correct}(T) & T \coloneqq \{t_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ \textit{where} \ t_i \leftarrow \mathcal{I}_{\mathsf{out},\mathsf{LDE}} \\ \land A(i^*) \neq A_{\mathsf{corr}}(i^*) & (\varphi, A) \leftarrow \mathsf{Assign}(T \cup \{i^*\}) \end{array} \right] \leq \mathsf{negl}(\lambda) \enspace ,$$

where Correct(T) and A_{corr} are defined as in Claim D.1.

Claim D.3 (From worst-case correctness of input to average correctness of input). There exist negligible functions negl, negl' such that for every sufficiently large $\lambda \in \Lambda$ and every $x \in \{0, 1\}^n$, if it holds

$$\Pr[A(i^*) \neq A_{corr}(i^*) \mid (\varphi, A) \leftarrow \mathsf{Assign}(\{i^*\})] \leq \mathsf{negl}(\lambda)$$

for every $i^* \in I_{in}$, it also holds

$$\Pr\left[\neg \mathsf{Correct}(S) \left| \begin{array}{c} S \coloneqq \{s_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ \textit{where} \ s_i \leftarrow \mathcal{I}_{\mathsf{in},\mathsf{LDE}} \\ (\varphi,A) \leftarrow \mathsf{Assign}(S) \end{array} \right] \leq \mathsf{negl'}(\lambda) \enspace ,$$

Proof. See, e.g., [PR14, Section 4.4] and [PR17, Section 4.4]. (In their terminologies, Aug first transforms f to an equivalent circuit C, then creates the "augmented version" of C, and then creates a 3CNF formula that checks the computation of the augmented C.)

D.2 Our PCP system (PCP.P, PCP.V)

Let f be an arbitrary functionality that is computable by Π .

D.2.1 3CNF formula $\varphi_f^{\mu:\nu}$.

For each $\mu, \nu \in [M]$, we introduce 3CNF formula $\varphi_f^{\mu;\nu}$, which takes views of the μ -th and ν -th parties P^{μ}, P^{ν} of Π as input and checks whether (1) these views are consistent and (2) both P^{μ} and P^{ν} output 1 in those views. Roughly speaking, we design $\varphi_f^{\mu;\nu}$ so that we can use Claim D.1 in a round-by-round way. The details are given below.

Let $\check{\mathsf{Next}}_{f,\Pi}$ be the next message function of Π and N_{round} be the round complexity of Π , where the functionality to be computed by Π is f. The input to $\mathsf{Next}_{f,\Pi}$ is of the form $(\mathsf{st}^i_{\ell-1},\mathsf{i-msgs}^i_\ell)$ for some $i \in [M]$ and $\ell \in [N_{\mathsf{round}}]$, where $\mathsf{st}^i_{\ell-1} \in \{0,1\}^{n_{\mathsf{st}}}$ is an internal state of party P^i at the beginning of the ℓ -th round and $\mathsf{i-msgs}^i_\ell = (\mathsf{msg}^{i-1}_\ell,\ldots,\mathsf{msg}^{i-M}_\ell) \in \{0,1\}^M$ is incoming messages that P^i receives at the beginning of the ℓ -th round. Given such input, $\mathsf{Next}_{f,\Pi}$ outputs $(\mathsf{st}^i_\ell,\mathsf{o-msgs}^i_\ell)$, where $\mathsf{st}^i_\ell \in \{0,1\}^{n_{\mathsf{st}}}$ is the internal state at the end of the ℓ -th round and $\mathsf{o-msgs}^i_\ell = (\mathsf{msg}^{i-1}_\ell,\ldots,\mathsf{msg}^{i-M}_\ell) \in \{0,1\}^M$ is the outgoing messages that P^i sends at the end of the ℓ -th round. Let $N_{\mathsf{TO}} := n_{\mathsf{st}} + M$ so that the input and the output of $\mathsf{Next}_{f,\Pi}$ are of length N_{TO} .

Let $\varphi_{f,\Pi}$ be the 3CNF Boolean formula that is obtained by $\varphi_{f,\Pi} := \operatorname{Aug}(\operatorname{Next}_{f,\Pi})$. Let N_{Aug} be the number of the variables in $\varphi_{f,\Pi}$.

Now, for each $\mu, \nu \in [M]$, we define 3CNF Boolean formula $\varphi_f^{\mu;\nu}$ as follows, where the number of the variables is $N := 2N_{\text{round}}N_{\text{Aug}}$ and they are denoted as $(\boldsymbol{w}_1^{\mu}, \cdots, \boldsymbol{w}_{N_{\text{round}}}^{\mu}, \boldsymbol{w}_1^{\nu}, \dots, \boldsymbol{w}_{N_{\text{round}}}^{\nu})$, where each $\boldsymbol{w}_{\ell}^{\xi}$ is a sequence of N_{Aug} variables and thus can be viewed as the variables in $\varphi_{f,\Pi}$. (As explained below in Remark 3, the assignment to each $\boldsymbol{w}_{\ell}^{\xi}$ is supposed to be derived from the start state and incoming messages of round ℓ of party P^{ξ} in the manner that is described in the "Satisfiability" paragraph in Lemma D.3.)

- 1. Let φ_1 be a 3CNF Boolean formula that checks $\varphi_{f,\Pi}(\mathbf{w}_{\ell}^{\xi}) = 1$ for $\forall \xi \in \{\mu, \nu\}, \ell \in [N_{\text{round}}]^{27}$
- 2. For each \mathbf{w}_{ℓ}^{ξ} ($\xi \in \{\mu, \nu\}$, $\ell \in [N_{\text{round}}]$), let $\mathbf{w}_{\ell, \text{in}}^{\xi} = (\mathbf{w}_{\ell, \text{in}}^{\xi}(1), \dots, \mathbf{w}_{\ell, \text{in}}^{\xi}(N_{\text{to}}))$ and $\mathbf{w}_{\ell, \text{out}}^{\xi} = (\mathbf{w}_{\ell, \text{out}}^{\xi}(1), \dots, \mathbf{w}_{\ell, \text{out}}^{\xi}(N_{\text{to}}))$ be the sequences of N_{to} variables that correspond to \mathbf{w}_{in} and \mathbf{w}_{out} in Lemma D.3 when \mathbf{w}_{ℓ}^{ξ} is viewed as the variables in $\varphi_{f,\Pi} = \text{Aug}(\text{Next}_{f,\Pi})$. Then, let φ_2 be a 3CNF Boolean formula that checks each of the following.²⁸
 - $\mathbf{w}_{\ell \text{ out}}^{\xi}(i) = \mathbf{w}_{\ell+1 \text{ in}}^{\xi}(i)$ holds for $\forall \xi \in \{\mu, \nu\}, \ell \in [N_{\text{round}} 1], i \in [n_{\text{st}}].$
 - $\mathbf{w}_{\ell \text{ out}}^{\mu}(n_{\text{st}} + \nu) = \mathbf{w}_{\ell+1 \text{ in}}^{\nu}(n_{\text{st}} + \mu)$ and $\mathbf{w}_{\ell \text{ out}}^{\nu}(n_{\text{st}} + \mu) = \mathbf{w}_{\ell+1 \text{ in}}^{\mu}(n_{\text{st}} + \nu)$ hold for $\forall \ell \in [N_{\text{round}} 1]$.
 - $\mathbf{w}_{N_{\text{round}},\text{Out}}^{\xi}(1) = 1 \text{ holds for } \forall \xi \in \{\mu, \nu\}.$
- 3. Finally, let $\varphi_f^{\mu;\nu}$ be the 3CNF Boolean formula that checks that φ_1 and φ_2 are satisfied by $(\mathbf{w}_1^{\mu},\cdots,\mathbf{w}_{N_{\text{round}}}^{\mu},\mathbf{w}_1^{\nu},\ldots,\mathbf{w}_{N_{\text{round}}}^{\nu})$.

We think that each variable in $\varphi_f^{\mu;\nu}$ is indexed by an element in $[M] \times [N_{\text{round}}] \times [N_{\text{Aug}}]$ in the natural way (i.e., in the way that the i-th variable in w_ℓ^ξ is indexed by (ξ,ℓ,i)). For each $\xi \in \{\mu,\nu\}$ and $\ell \in [N_{\text{round}}]$, let $I_{\ell,\text{in}}^\xi = \{(\xi,\ell,1),\ldots,(\xi,\ell,N_{\text{io}})\}$ be the set of the indices of the variables in $w_{\ell,\text{in},\text{LDE}}^\xi$, be the set of the indices of the variables in $w_{\ell,\text{in},\text{LDE}}^\xi$, and $I_{\ell,\text{out},\text{LDE}}^\xi$ be the set of the indices of the variables in $w_{\ell,\text{out},\text{LDE}}^\xi$, where $w_{\ell,\text{in},\text{LDE}}^\xi$ and $w_{\ell,\text{out},\text{LDE}}^\xi$ are the sequences of variables that correspond to $w_{\text{in},\text{LDE}}$ and $w_{\text{out},\text{LDE}}^\xi$ in Lemma D.3 when w_ℓ^ξ is viewed as the variables in $\varphi_{f,\Pi} = \text{Aug}(\text{Next}_{f,\Pi})$. For each $\xi \in \{\mu,\nu\}$, let $I^\xi := \{(\xi,1,1),\ldots,(\xi,N_{\text{round}},N_{\text{Aug}})\}$ be the set of the indices of the variables in $(w_1^\xi,\ldots,w_{N_{\text{cound}}}^\xi)$.

Remark 3. Given any views view^{μ} , view^{ν} of P^{μ} , P^{ν} , we can obtain an assignment to the variables $(\boldsymbol{w}_{1}^{\mu},\cdots,\boldsymbol{w}_{N_{\text{round}}}^{\mu},\boldsymbol{w}_{1}^{\nu},\ldots,\boldsymbol{w}_{N_{\text{round}}}^{\nu})$ in $\varphi_{f}^{\mu;\nu}$ as follows.

 $1. \ \text{Parse view}^{\xi} \ \text{as} \ (\mathsf{st}_0^{\xi}, \mathsf{i-msgs}_1^{\xi}, \dots, \mathsf{i-msgs}_{N_{\mathsf{round}}}^{\xi}) \ \text{for} \ \forall \xi \in \{\mu, \nu\}.$

 $^{^{26}}$ Recall that we assume for simplicity that each party uses the same next message function in every round in Π .

²⁷Intuitively, φ_1 checks whether each end state and outgoing messages are correctly computed from the corresponding start state and incoming messages.

²⁸Intuitively, φ_2 checks whether (1) P^{ξ} 's start state at round $\ell+1$ is equal to its end state at round ℓ , (2) P^{ν} 's incoming message from P^{μ} at round $\ell+1$ is equal to P^{μ} 's outgoing message to P^{ν} at round ℓ and vice versa, and (3) P^{ξ} 's output is 1.

- 2. Repeat the following for $\xi \in \{\mu, \nu\}, \ell \in [N_{\text{round}}]$.
 - (a) Obtain an assignment to the variables \mathbf{w}_{ℓ}^{ξ} by using $(\mathbf{st}_{\ell-1}^{\xi}, \mathbf{i}\text{-msgs}_{\ell}^{\xi})$, where \mathbf{w}_{ℓ}^{ξ} is viewed as the variables in $\varphi_{f,\Pi} = \mathsf{Aug}(\mathsf{Next}_{f,\Pi})$ (cf. Lemma D.3).
 - (b) Compute $(\mathsf{st}_{\ell}^{\xi}, \mathsf{o-msgs}_{\ell}^{\xi}) \coloneqq \mathsf{Next}_{f,\Pi}(\mathsf{st}_{\ell-1}^{\xi}, \mathsf{i-msgs}_{\ell}^{\xi}).$

Note that when an assignment is computed in this way, the assignment to the variables in $\mathbf{w}_{1,\text{in}}^{\xi}$ (i.e., those that are indexed by $\mathcal{I}_{1,\text{in}}^{\xi}$) is $(\mathbf{st}_{0}^{\xi}, \mathbf{i-msgs}_{1}^{\xi}) \in \{0, 1\}^{N_{\text{to}}}$, and the assignment to the variables in $\mathbf{w}_{1}^{\xi}, \cdots, \mathbf{w}_{N_{\text{round}}}^{\xi}$ (i.e., those that are indexed by \mathcal{I}^{ξ}) are computed from view alone for each $\xi \in \{\mu, \nu\}$. Also, note that if view and view are consistent and P^{μ} and P^{ν} output 1 in them, we obtain a satisfying assignment.

D.2.2 Prover PCP.P.

The prover PCP.P of our PCP system is given in Algorithm 2.

Algorithm 2 Prover PCP.P of our PCP system (PCP.P, PCP.V)

Given input of the form $(\mu, \nu, f, \text{view}^{\mu}, \text{view}^{\nu})$, PCP.P works as follows.

- 1. Obtain an assignment $x^{\mu:\nu}$ to the variables in $\varphi_f^{\mu:\nu}$ by using view, view as described in Remark 3.
- 2. Output $\pi^{\mu:\nu} = (X^{\mu:\nu}, ...) := \mathsf{PCP.P}_{\mathsf{KRR}}(\varphi_f^{\mu:\nu}, x^{\mu:\nu})$ as the PCP proof.

Regarding PCP.P, we introduce several notations and without-loss-of-generality assumptions. (Roughly, we introduce them so that given $X^{\mu:\nu}$, which is an LDE of $x^{\mu:\nu}$, the verifier can evaluate LDEs of several substrings of $x^{\mu:\nu}$.)

Notations. Let F, H, and m be the parameters of (PCP.P_{KRR}, PCP.V_{KRR}) used in PCP.P. Let $m_{\text{round}} := \log N_{\text{round}}/\log |H|$, $m_{\text{Aug}} := \log N_{\text{Aug}}/\log |H|$, and $m_{\text{to}} := \log N_{\text{to}}/\log |H|$ so that $N_{\text{round}} = |H|^{m_{\text{round}}}$, $N_{\text{Aug}} = |H|^{m_{\text{Aug}}}$, and $N_{\text{to}} = |H|^{m_{\text{to}}}$. (It is assumed that m_{round} , m_{Aug} , m_{to} are integers.) Given assignment $x^{\mu:\nu} : [M] \times [N_{\text{round}}] \times [N_{\text{Aug}}] \to \{0, 1\}$ to the variables in $\varphi_f^{\mu:\nu}$, let $x_{\ell,\text{in}}^{\xi} : [N_{\text{to}}] \to \{0, 1\}$ be its assignment to the variables indexed by $\mathcal{I}_{\ell,\text{in}}^{\xi}$ (i.e., $x_{\ell,\text{in}}^{\xi}$ is defined so that $x_{\ell,\text{in}}^{\xi} : [N_{\text{round}}] \times [N_{\text{Aug}}] \to \{0, 1\}$ be its assignment to the variables indexed by $\mathcal{I}_{\ell,\text{in}}^{\xi}$ (i.e., $x_{\ell,\text{in}}^{\xi}$ is defined so that $x_{\ell,\text{in}}^{\xi} : [N_{\text{round}}] \times [N_{\text{Aug}}] \to \{0, 1\}$ be its assignment to the variables indexed by $\mathcal{I}_{\ell,\text{in}}^{\xi}$ (i.e., $x_{\ell,\text{in}}^{\xi}$ is defined so that $x_{\ell,\text{in}}^{\xi} : [N_{\text{round}}], i \in [N_{\text{Aug}}],$ and $X_{\ell,\text{in}}^{\xi} : [N_{\text{round}}], i \in [N_{\text{Aug}}],$ and $X_{\ell,\text{in}}^{\xi} : [N_{\text{round}}], i \in [N_{\text{Aug}}],$ and $X_{\ell,\text{in}}^{\xi} : [N_{\text{round}}], i \in [N_{\text{round}}],$ if $X_{\ell,\text{in}}^{\xi} : [N_{\text{r$

WLOG assumptions. We assume that the parameters F, H, m of (PCP.P_{KRR}, PCP.V_{KRR}) and the LDE $X^{\mu:\nu}$ of $x^{\mu:\nu}$ satisfy the following.²⁹

- $H := \{0, \dots, \log(N_{\text{round}}N_{\text{Aug}}) 1\}$ and $m := m_{\text{round}} + m_{\text{Aug}} + 1$.
- For each $\xi \in \{\mu, \nu\}$, there exists $z_{\xi} \in H$ such that X^{ξ} can be evaluated on any point in F^{m-1} by evaluating $X^{\mu;\nu}$ on a point in $D(X^{\xi}) := \{(z_{\xi}, z) \mid z \in F^{m-1}\}$.
- For each $\xi \in \{\mu, \nu\}$ and $\ell \in [N_{\text{round}}]$, there exists $z_{\ell} \in \pmb{H}^{m_{\text{round}}}$ and $z_{\text{in}} \in \pmb{H}^{m_{\text{Aug}}-m_{\text{to}}}$ such that $X_{\ell,\text{in}}^{\xi}$ can be evaluated on any point in $\pmb{F}^{m_{\text{to}}}$ by evaluating $X^{\mu:\nu}$ on a point in $D(X_{\ell,\text{in}}^{\xi}) := \{(z_{\xi}, z_{\ell}, z_{\text{in}}, z) \mid z \in \pmb{F}^{m_{\text{to}}}\}$.

D.2.3 Verifier PCP.V = (PCP.Q, PCP.D).

The verifier PCP.V of our PCP system is given in Algorithm 3. (It becomes clear later in Appendix E why PCP.V additionally does various types of low-degree tests.)

It can be verified easily that PCP.V can be decomposed into a query algorithm PCP.Q and a decision algorithm PCP.D naturally, and that the query complexity of PCP.V is asymptotically the same as that of PCP.V_{KRR}, i.e., is at most $O(m|F|^2) \le \text{polylog}(N)$.

²⁹Concretely, we assume that $X^{\mu:\nu}$ is computed as follows. View $x^{\mu:\nu}$ as a function with domain $\{\mu, \nu\} \times [N_{\text{round}}] \times [N_{\text{Aug}}]$, and fix a mapping from $\{\mu, \nu\} \times [N_{\text{round}}] \times [N_{\text{Aug}}]$ to H^m by arbitrarily fixing a map from [M] to H, a map from $[N_{\text{round}}]$ to $H^{m_{\text{round}}}$, and a map from $[N_{\text{Aug}}]$ to $H^{m_{\text{Aug}}}$. Then, let $X^{\mu:\nu}$ be the LDE of $x^{\mu:\nu}$ w.r.t. F, H, m and the above mapping.

Algorithm 3 Verifier PCP.V of our PCP system (PCP.P, PCP.V)

Given input of the form (μ, ν, f) , the verifier V does the following tests.

- 1. Do the same tests as PCP.V_{KRR}($\varphi_f^{\mu;\nu}$).
- 2. $D(X_{1,\text{in}}^{\xi})$ -parallel Low-degree Test for $X^{\mu:\nu}$: Choose random points $\mathbf{r}_0 \in \mathbf{F}^m$ and $\mathbf{r}_1 \in \{0^{1+m_{\text{round}}+m_{\text{Aug}}-m_{\text{io}}}\} \times \mathbf{F}^{m_{\text{io}}}$, define a line $L: \mathbf{F} \to \mathbf{F}^m$ as $L(\alpha) = \mathbf{r}_0 + \alpha \cdot \mathbf{r}_1$, and query $X^{\mu:\nu}$ on all the points $\{L(t)\}_{t \in \mathbf{F}}$. Check that the univariate polynomial $X^{\mu:\nu} \circ L: \mathbf{F} \to \mathbf{F}$ has degree at most $m_{\text{io}}|\mathbf{H}|$.
- 3. $D(X^{\xi})$ -parallel Low-degree Test for $X^{\mu,\nu}$: Choose random points $r_0 \in F^m$ and $r_1 \in \{0\} \times F^{m-1}$, define a line $L: F \to F^m$ as $L(\alpha) = r_0 + \alpha \cdot r_1$, and query $X^{\mu,\nu}$ on all the points $\{L(t)\}_{t \in F}$. Check that the univariate polynomial $X^{\mu,\nu} \circ L: F \to F$ has degree at most (m-1)|H|.
- 4. Low-degree Test for $X^{\mu:\nu}$, conditioned on $L(0) \in D(X_{1,\text{in}}^{\xi})$: For every $\xi \in \{\mu, \nu\}$, do the following: Choose a random line $L: F \to F^m$ such that $L(0) \in D(X_{1,\text{in}}^{\xi})$, and query $X^{\mu:\nu}$ on all the points $\{L(t)\}_{t \in F}$; then, check that the univariate polynomial $X^{\mu:\nu} \circ L: F \to F$ has degree at most m|H|.
- 5. Low-degree Test for $X^{\mu:\nu}$, conditioned on $L(0) \in D(X^{\xi})$: For every $\xi \in \{\mu, \nu\}$, do the following: Choose a line $L: F \to F^m$ such that $L(0) \in D(X^{\xi})$, and query $X^{\mu:\nu}$ on all the points $\{L(t)\}_{t \in F}$; then, check that the univariate polynomial $X^{\mu:\nu} \circ L: F \to F$ has degree at most m|H|.
- 6. **Low-degree Test for** $X_{1,\text{in}}^{\xi}$: For every $\xi \in \{\mu, \nu\}$, do the following: Choose a line $L : F \to D(X_{1,\text{in}}^{\xi})$, and query $X^{\mu:\nu}$ on all the points $\{L(t)\}_{t \in F}$; then, check that the univariate polynomial $X^{\mu:\nu} \circ L : F \to F$ has degree at most $m_{\text{Io}}|H|$.
- 7. **Low-degree Test for** X^{ξ} : For every $\xi \in \{\mu, \nu\}$, do the following: Choose a line $L : F \to D(X^{\xi})$, and query $X^{\mu:\nu}$ on all the points $\{L(t)\}_{t \in F}$; then, check that the univariate polynomial $X^{\mu:\nu} \circ L : F \to F$ has degree at most (m-1)|H|.
- 8. Low-degree Test for $X^{\mu:\nu}$: Choose a line $L: F \to D(X)$, and query $X^{\mu:\nu}$ on all the points $\{L(t)\}_{t \in F}$. Check that the univariate polynomial $X^{\mu:\nu} \circ L: F \to F$ has degree at most m|H|.

D.2.4 Security.

(PCP.P, PCP.V) inherits security from (PCP.P_{KRR}, PCP.V_{KRR}). Specifically, since PCP.P just executes PCP.P_{KRR} on a specific form of a 3SAT instance, and PCP.V does all the tests that PCP.V_{KRR} does, we obtain the following lemma by combining Lemma D.1, Lemma D.2, and Lemma D.3.

Lemma D.4. The PCP system (PCP.P, PCP.V) in Algorithm 2 and Algorithm 3 satisfies the following soundness property. There exist polynomials κ_0 and κ_{\max} such that for every negligible function ϵ , every $\alpha, \beta \in [M]$, and every adaptive ($\kappa_0 \cdot \kappa_{\max}$)-CNS cheating prover PCP.P*, if it holds

$$\Pr \begin{bmatrix} b = 1 & (\mathsf{st}, Q) \leftarrow \mathsf{PCP}.\mathsf{Q}^{\otimes \lambda}(\alpha, \beta); \ (f, \pi) \leftarrow \mathsf{PCP}.\mathsf{P}^*(Q); \\ b \coloneqq \mathsf{PCP}.\mathsf{D}^{\geq \lambda - \zeta}(\mathsf{st}, f, \pi) \end{bmatrix} \geq 1 - \epsilon(\lambda),$$

for $\zeta = \omega(\log \lambda)$ for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of such λ), then the procedure SelfCorr in Algorithm 1 satisfies the following four claims.

Claim D.4. SelfCorr^{PCP.P*}_{m|H|,D(X)} is an adaptive κ_{\max} -local assignment generator for every sufficiently large $\lambda \in \Lambda$. Moreover, the distribution of f that is generated by $(f,A) \leftarrow \mathsf{SelfCorr}^{\mathsf{PCP.P^*}}_{m|H|,D(X)}(W)$ for any W is computationally indistinguishable from the distribution of f that is generated by $\mathsf{PCP.P^*}$.

Claim D.5. There exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$ and for $\forall \{(st_0^{\mu}, i\text{-msgs}_1^{\mu})\}_{\mu \in [M]} \in (\{0, 1\}^{N_{lo}})^M, \xi \in \{\alpha, \beta\}, \ell \in [N_{round}], \text{ it holds}$

$$\Pr\left[\begin{array}{c|c} \mathsf{Correct}(S) & S \coloneqq \{s_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ \textit{where} \ s_i \leftarrow I^{\xi}_{\ell, \mathsf{in}, \mathsf{LDE}} \\ & \land \neg \mathsf{Correct}(T) & T \coloneqq \{t_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ \textit{where} \ t_i \leftarrow I^{\xi}_{\ell, \mathsf{out}, \mathsf{LDE}} \\ & (f, A) \leftarrow \mathsf{SelfCorr}^{\mathsf{PCP}, \mathsf{P^*}}_{\mathit{MH}, D(X)}(S \cup T) \end{array}\right] \leq N_{\mathsf{Aug}} \cdot \mathsf{negl}(\lambda) \ ,$$

where the events $\mathsf{Correct}(S)$ and $\mathsf{Correct}(T)$ are defined as follows. Let the correct views $\{\mathsf{view}^\mu\}_{\mu \in [M]}$ be the views of the parties in the execution of Π on $(f, \{(\mathsf{st}_0^\mu, \mathsf{i-msgs}_1^\mu)\}_{\mu \in [M]})$, 31 and let the correct assignment A_{corr} (to the variables in $\varphi_f^{\alpha;\beta}$)

 $^{^{30}}$ Note that SelfCorr $^{\text{PCP.P}^*}_{m|H|,D(X)}$ outputs a function f rather than a 3CNF formula since we assume that PCP.P * outputs a function rather than a 3CNF formula. Here, we view SelfCorr $^{\text{PCP.P}^*}_{m|H|,D(X)}$ as an a local assignment generator for $\varphi^{\alpha;\beta}_f$.

³¹i.e., the execution of Π where the functionality to be computed is f, the initial states of the parties are $\{st_0^{\mu}\}_{\mu\in[M]}$, and the dummy incoming messages of the first round are $\{i\text{-msgs}_1^{\mu}\}_{\mu\in[M]}$.

be the assignment that is obtained from (view^{α} , view^{β}) as in Remark 3. Then, $\mathsf{Correct}(S)$ is the event that $A(s) = A_{\mathsf{corr}}(s)$ holds for $\forall s \in S$ and $\mathsf{Correct}(T)$ is the event that $A(t) = A_{\mathsf{corr}}(t)$ holds for $\forall t \in T$.

Claim D.6. There exists a negligible function negl such that for every sufficiently large $\lambda \in \Lambda$ and for $\forall \{(\mathsf{st}_0^\mu, \mathsf{i-msgs}_1^\mu)\}_{\mu \in [M]} \in (\{0,1\}^{N_{no}})^M, \, \xi \in \{\alpha,\beta\}, \, \ell \in [N_{\mathsf{round}}], \, and \, i^* \in \mathcal{I}_{\ell,\mathsf{out}}^\xi, \, it \, holds$

$$\Pr \left[\begin{array}{c|c} \mathsf{Correct}(T) & T \coloneqq \{t_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ where \ t_i \leftarrow I^{\xi}_{\ell, \mathsf{out}, \mathsf{LDE}} \\ \land A(i^*) \neq A_{\mathsf{corr}}(i^*) & (f, A) \leftarrow \mathsf{SelfCorr}^{\mathsf{PCP}, \mathsf{P}^*}_{m|H|, D(X)}(T \cup \{i^*\}) \end{array} \right] \leq \mathsf{negl}(\lambda) \enspace ,$$

where Correct(T) and A_{corr} are defined as in Claim D.5.

Claim D.7. There exist negligible functions negl, negl' such that for every sufficiently large $\lambda \in \Lambda$ and for $\forall \{(\mathsf{st}_0^\mu, \mathsf{i-msgs}_1^\mu)\}_{\mu \in [M]} \in (\{0,1\}^{N_{to}})^M$ and $\xi \in \{\alpha,\beta\}$, if it holds

$$\Pr\left[A(i^*) \neq A_{\operatorname{corr}}(i^*) \;\middle|\; (f,A) \leftarrow \mathsf{SelfCorr}^{\mathsf{PCP},\mathsf{P}^*}_{m|H\backslash D(X)}(\{i^*\})\right] \leq \mathsf{negl}(\lambda)$$

for every $i^* \in \mathcal{I}_{1,\mathsf{in}}^{\xi}$, it also holds

$$\Pr\left[\neg \mathsf{Correct}(S) \left| \begin{array}{c} S := \{s_i\}_{i \in [\log^2 \lambda]}, \ \textit{where} \ s_i \leftarrow \mathcal{I}^{\xi}_{1,\mathsf{in},\mathsf{LDE}} \\ (f,A) \leftarrow \mathsf{SelfCorr}^{\mathsf{PCP},\mathsf{P}^*}_{m|H,D(X)}(S) \end{array} \right] \leq \mathsf{negl'}(\lambda) \ ,$$

where Correct(S) and A_{corr} are defined as in Claim D.5. Similarly, for $\ell \in \{2, ..., N_{round}\}$, if it holds

$$\Pr\left[\begin{array}{c|c} \mathsf{Correct}(T) & T \coloneqq \{t_i\}_{i \in \lceil \log^2 \lambda \rceil}, \ where \ t_i \leftarrow \mathcal{I}^{\xi}_{\ell-1,\mathsf{out},\mathsf{LDE}} \\ \land A(i^*) \neq A_{\mathsf{corr}}(i^*) & (f,A) \leftarrow \mathsf{SelfCorr}^{\mathsf{PCP},\mathsf{P}^*}_{m|H|,D(X)}(T \cup \{i^*\}) \end{array} \right] \leq \mathsf{negl}(\lambda)$$

for every $i^* \in I_{\ell,in}^{\xi}$, it also holds

$$\Pr\left[\begin{array}{c|c} \mathsf{Correct}(T) & T \coloneqq \{t_i\}_{i \in [\log^2 \lambda]}, \ where \ t_i \leftarrow I^\xi_{\ell-1,\mathsf{out},\mathsf{LDE}} \\ \land \neg \mathsf{Correct}(S) & S \coloneqq \{s_i\}_{i \in [\log^2 \lambda]}, \ where \ s_i \leftarrow I^\xi_{1,\mathsf{in},\mathsf{LDE}} \\ (f,A) \leftarrow \mathsf{SelfCorr}^{\mathsf{PCP},\mathsf{P}^*}_{m|H|,D(X)}(S \cup T) \end{array}\right] \leq \mathsf{negl'}(\lambda) \ .$$

E Step 1: Non-WI Scheme with (1 – negl)-Soundness against Well-behaving CNS Provers

As the first step to our commit-and-prove protocol, we give a non-WI commit-and-prove protocol $\langle C_1, R_1 \rangle$ that is (1-negl)-sound against "well-behaving" CNS provers.

E.1 Protocol Description

The formal description of $\langle C_1, R_1 \rangle$ is given in Algorithm 4 and Algorithm 5, where $\zeta = \omega(\log \lambda)$ is a parameter for the use of Lemma D.4, and the subroutines SelfCorr and LD-Test are defined in Algorithm 1 and Algorithm 6 respectively. We remark that Time(f) is assumed to be known to the verifier in the commit phase so that the parameters F, H, m_{10} of (PCP.P, PCP.V) can be determined.

The communication complexity is polylogarithmic in $\mathsf{Time}(f)$ since the query complexity of (PCP.P, PCP.V) is polylogarithmic in the size of $\varphi_f^{\mu;\nu}$.

Remark 4 (On the open phase). At first sight, the open phase might seem to be unnecessarily too complex since the receiver does self-correction by using lines on $D(X_{1,\text{in}}^{\mu}) \subset F^m$ rather than those on $F^{m_{to}}$. Roughly speaking, the open phase is defined in this way since in the proof of soundness (where we construct an extractor that converts any successful prover into a successful decommitter), we consider an extractor that forwards the receiver's decommitment queries to the cheating prover by observing that queries in the open phase can be viewed as those in the prove phase. (Note that in the open phase, the receiver queries to $\tilde{X}^{\mu,\mu}: F^m \to F$, which has the same domains as the LDEs that the receiver queries to in the prove phase.)

E.2 Proof of Binding

We prove the binding property against cheating adversaries that are no-signaling and well-behaving in the following sense.

Definition 17 (No-signaling committer-decommitter). *A cheating committer-decommitter* $C_1^* = (C.Com_1^*, C.Dec_1^*)$ *against* $\langle C_1, R_1 \rangle$ *is* κ_{max} -CNS *if* $C.Dec_1^*(st_C, \cdot)$ *is* κ_{max} -CNS *for* $\forall (st_C, com) \leftarrow \langle C.Com_1^*, R.Com_1 \rangle$.

Algorithm 4 Commit Phase and Prove Phase of $\langle C_1, R_1 \rangle$

Commit Phase:

Round 1: Given x_{COM} as input, C.Com₁ does the following.

- 1. Sample random $x_{\text{MPC}}^1, \dots, x_{\text{MPC}}^M \in \{0, 1\}^n$ such that $x_{\text{MPC}}^1 \oplus \dots \oplus x_{\text{MPC}}^M = x_{\text{com}}$.
- 2. For each $\mu \in [M]$, define $X_{1,\text{in}}^{\mu}: \mathbf{\textit{F}}^{m_{\text{to}}} \to \mathbf{\textit{F}}$ as follows.
 - (a) Define $x_{1,\text{in}}^{\mu} \in \{0,1\}^{N_{\text{10}}}$ as follows: sample random $r_{\text{MPC}}^{\mu} \in \{0,1\}^{n_{\text{MPC}}}$ and let $\mathsf{st}_{0}^{\mu} \coloneqq x_{\text{MPC}}^{\mu} \parallel r_{\text{MPC}}^{\mu} \in \{0,1\}^{n_{\text{st}}}$, i-msgs $_{1}^{\mu} \coloneqq 0^{M} \in \{0,1\}^{M}$, $x_{1,\text{in}}^{\mu} \coloneqq \mathsf{st}_{0}^{\mu} \parallel \mathsf{i-msgs}_{1}^{\mu}$.
 - (b) Let $X_{1 \text{ in}}^{\mu}$ be the low-degree extension of $x_{1 \text{ in}}^{\mu}$ (w.r.t. F, H, m_{10}).
- 3. Output an empty string ε as the commitment, and $\{X_{1,\text{in}}^{\mu}\}_{\mu\in[M]}$ as the internal state.

......

Prove Phase:

Round 1: R.Prv.Q₁ runs $(st_V^{\mu,\nu}, Q^{\mu,\nu}) \leftarrow PCP.Q^{\otimes \lambda}(\mu,\nu)$ for every $\mu,\nu \in [M]$, and outputs $\{Q^{\mu,\nu}\}_{\mu,\nu \in [M]}$ as the query and $\{st_V^{\mu,\nu}\}_{\mu,\nu \in [M]}$ as the internal state.

Round 2: Given $(\operatorname{st}_C, f, \{Q^{\mu,\nu}\}_{\mu,\nu\in[M]})$ as input, C.Prv₁ does the following.

- 1. Run the MPC protocol Π in the head on $(f', \{(\mathsf{st}_0^\mu, \mathsf{i-msgs}_1^\mu)\}_{\mu \in [M]})$, where f' is defined as $f' : (y^1, \ldots, y^M) \mapsto f(y^1 \oplus \cdots \oplus y^M)$. (Here, $\{(\mathsf{st}_0^\mu, \mathsf{i-msgs}_1^\mu)\}_{\mu \in [M]}$ is recovered from $\mathsf{st}_C = \{X_{1,\mathsf{in}}^\mu\}_{\mu \in [M]}$.) Let $\{\mathsf{view}^\mu\}_{\mu \in [M]}$ be the view of the parties in this execution.
- 2. Run $\pi^{\mu:\nu} \leftarrow \mathsf{PCP.P}(\mu, \nu, f', \mathsf{view}^{\mu}, \mathsf{view}^{\nu})$ for every $\mu, \nu \in [M]$.
- 3. Output $\{\pi^{\mu:\nu}|_{Q^{\mu:\nu}}\}_{\mu,\nu\in[M]}$ as the proof.

Verification: Given $(\operatorname{st}_R, \operatorname{com}, f, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]})$ (where $\operatorname{st}_R = \{\operatorname{st}_V^{\mu,\nu}\}_{\mu,\nu\in[M]}$), R.Prv.D₁ does the following.

- 1. Run $b^{\mu:\nu} := \mathsf{PCP.D}^{\geq \lambda \zeta}(\mathsf{st}_V^{\mu:\nu}, f', \pi^{*\mu:\nu})$ for every $\mu, \nu \in [M]$.
- 2. Output 1 if and only if $b^{\mu:\nu} = 1$ for every $\mu, \nu \in [M]$.

Definition 18 (Well-behaving committer-decommitter). A cheating committer-decommitter $C_1^* = (C.Com_1^*, C.Dec_1^*)$ against $\langle C_1, R_1 \rangle$ is well-behaving if the following holds.

• Consistency on $D(X_{1,\text{in}}^{\mu})$: For $\forall \lambda \in \mathbb{N}$ and $\forall \{Q_0^{\mu}\}_{\mu \in [M]}, \{Q_1^{\mu}\}_{\mu \in [M]}$ such that $Q_0^{\mu}, Q_1^{\mu} \subseteq D(X_{1,\text{in}}^{\mu})$, it holds

$$\Pr\left[\begin{array}{c} \exists \mu \in [M], \, q \in \mathcal{Q}_0^\mu \cap \mathcal{Q}_1^\mu \, s.t. \\ \tilde{X}_0^{\mu,\mu}(q) \neq \bot \\ \wedge \tilde{X}_1^{*\mu,\mu}(q) \neq \bot \\ \wedge \tilde{X}_0^{*\mu,\mu}(q) \neq \tilde{X}_1^{*\mu,\mu}(q) \end{array} \right| \begin{array}{c} (\operatorname{st}_C, \operatorname{com}) \leftarrow \langle \operatorname{C.Com}_1^*, \operatorname{R.Com}_1 \rangle \\ \{\operatorname{dec}_b\}_{b \in \{0,1\}} \leftarrow \operatorname{C.Dec}_1^*(\operatorname{st}_C, \{\overline{\mathcal{Q}}_b\}_{b \in \{0,1\}}) \\ where \ \overline{\mathcal{Q}}_b = \{\mathcal{Q}_b^\mu\}_{\mu \in [M]} \ and \ \operatorname{dec}_b = \{\tilde{X}_b^{*\mu,\mu}\}_{\mu \in [M]} \end{array} \right] \leq \operatorname{negl}(\lambda) \ .$$

Note that well-behaving adversaries do not give different answers to the same query during the binding security experiment. The formal statement of the binding property of $\langle C_1, R_1 \rangle$ is given below. In the following, P-LD-Test is a parallel version of LD-Test; see Algorithm 7.

Lemma E.1. There exists a polynomial κ_{dec} such that the following holds. Let ϵ be any negligible function and $C_1^* = (C.Com_1^*, C.Dec_1^*)$ be any well-behaving κ_{dec} -CNS cheating committer-decommitter against $\langle C_1, R_1 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following binding condition holds with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com_1^*, R.Com_1 \rangle$.

• Binding Condition: If it holds

$$\Pr\left[b_{\scriptscriptstyle LD} = 1 \;\middle|\; b_{\scriptscriptstyle LD} \leftarrow \mathsf{P-LD-Test}^{\mathsf{C.Dec}^*_1(\mathsf{st}_{\scriptscriptstyle C}, \cdot)}_{d, \{D^\mu_b|_{b\in[0,1], \mu\in[M]}, 3\zeta}\right] \geq 1 - \epsilon(\lambda) \tag{E.1}$$

for $d := m_{io}|\mathbf{H}|$ and $D_b^{\mu} := D(X_{1,\text{in}}^{\mu})$, then for every $i \in [n]$ it holds $\Pr[b_{\text{BAD}} = 1] \leq \mathsf{negl}(\lambda)$ in the following probabilistic experiment.

- 1. For each $\forall b \in \{0, 1\}$, sample $\{Q_b^{\mu}\}_{\mu \in [M]}$ by $(\{Q_b^{\mu}\}_{\mu \in [M]}, \mathsf{st}_b) \leftarrow \mathsf{R.Dec.Q}_1(i)$.
- 2. $Run\{\tilde{X}_b^{\mu:\mu}\}_{b\in\{0,1\},\mu\in[M]} \leftarrow C.Dec_1^*(st_C,\{Q_b^{\mu}\}_{b\in\{0,1\},\mu\in[M]}).$
- 3. Let $b_{\text{BAD}} \coloneqq 1$ if and only if $x_0^* \neq \bot \land x_1^* \neq \bot \land x_0^* \neq x_1^*$ holds, where $x_b^* \coloneqq \mathsf{R.Dec.D_1}(\mathsf{st}_b, \mathsf{com}, \{\tilde{X}^*_b^{\mu;\mu}\}_{\mu \in [M]})$ for each $b \in \{0, 1\}$.

Algorithm 5 Open Phase of $\langle C_1, R_1 \rangle$

Open Phase:

Round 1: Given i as input, R.Dec.Q₁ does the following.

- 1. Define Q_0^{μ} , $Q_1^{\mu} \subset D(X_{1 \text{ in}}^{\mu})$ for every $\mu \in [M]$ as follows.
 - (Low-degree Test on $X_{1 \text{ in}}^{\mu}$) Run $(Q_0^{\mu}, \operatorname{st}_0^{\mu}) \leftarrow \mathsf{LD-Test.Q}_{D(X_{1 \text{ in}}^{\mu})}$.
 - (Self-correction) Run $(Q_1^{\mu}, \operatorname{st}_1^{\mu}) \leftarrow \operatorname{SelfCorr.Q}_{D(X_{1,\operatorname{in}}^{\mu})}(\{(\mu, 1, i)\})$, where $(\mu, 1, i) \in [M] \times [N_{\operatorname{round}}] \times [N_{\operatorname{Aug}}]$ is the index of a variable in the 3CNF formula $\varphi_f^{\mu,\nu}$ (concretely, the variable that corresponds to the i-th bit of P^{μ} 's internal state at the beginning of Round 1, or equivalently the i-th bit of P^{μ} 's input).

Then, let $Q^{\mu} := Q_0^{\mu} \cup Q_1^{\mu}$.

2. Output $\{Q^{\mu}\}_{{\mu}\in[M]}$ as the query, and $(i, \{\mathsf{st}_0^{\mu}, \mathsf{st}_1^{\mu}\}_{{\mu}\in[M]})$ as the internal state.

Round 2: Given $(\operatorname{st}_C, \{Q^{\mu}\}_{\mu \in [M]})$ as input (where $\operatorname{st}_C = \{X_{1 \text{ in}}^{\mu}\}_{\mu \in [M]})$, C.Dec₁ does the following.

- 1. For every $\mu \in [M]$, let $\tilde{X}^{\mu:\mu}: D(X) \to F$ be an arbitrary polynomial such that $\tilde{X}^{\mu:\mu}(z_{\mu}, z_{\ell}, z_{\text{in}}, z) = X_{1,\text{in}}^{\mu}(z)$ for every $z \in F^{m_{\text{to}}}$ (recall that $z_{\mu} \in H$, $z_{\ell} \in H^{m_{\text{round}}}$, and $z_{\text{in}} \in H^{m_{\text{Aug}}-m_{\text{to}}}$ are the points such that $D(X_{\ell,\text{in}}^{\mu}) = \{(z_{\mu}, z_{\ell}, z_{\text{in}}, z) \mid z \in F^{m_{\text{to}}}\}$; see Section D.2.2).
- 2. Output $\{\tilde{X}^{\mu:\mu}|_{O^{\mu}}\}_{\mu\in[M]}$ as the decommitment.

Verification: Given $(\mathsf{st}_R, \mathsf{com}, \{\tilde{X}^{*\mu:\mu}\}_{\mu \in [M]})$ as input (where $\mathsf{st}_R = (i, \{\mathsf{st}_0^\mu, \mathsf{st}_1^\mu\}_{\mu \in [M]}))$, R.Dec.D₁ does the following.

- 1. (Low-degree Test) For $\forall \mu \in [M]$, check that LD-Test. $D_{m_{io}|H|,3\zeta}(\operatorname{st}_{0}^{\mu}, \tilde{X}^{*\mu:\mu}) = 1$.
- 2. (**Self-correction**) For $\forall \mu \in [M]$, run $A^{\mu} := \mathsf{SelfCorr}.\mathsf{Rec}_{m_{io}|H|}(\mathsf{st}_{1}^{\mu}, \tilde{X}^{*}^{\mu;\mu})$, and check that $\tilde{x}_{i} := A^{\mu}(\mu, 1, i)$ is a binary value (i.e., $\tilde{x}_{i} = 0$ or $\tilde{x}_{i} = 1$).
- 3. Output $\tilde{x}_i := \tilde{x}_i^1 \oplus \cdots \oplus \tilde{x}_i^M$ as the decommitted value.

Intuition of the proof. At first sight, proving the binding property against well-behaving adversaries seems to be trivial. Specifically, since it is guaranteed that the adversary does not give different answers to the same query during the binding security experiment, it seems to be implied that the adversary cannot open a commitment to two different values.

A problem is that to open a commitment to two different values, the adversary does not need to give different answers to the same query. This is because the adversary only needs to let the receiver recover different values from SelfCorr, and SelfCorr can make different queries on the same input depending on the randomness.

We overcome this problem by relying on the other assumptions, namely that the adversary is CNS and that it passes the low-degree test (Equation (E.1)). First, we use the well-behaving assumption to argue that if a well-behaving CNS adversary can let the receiver recover two different values from SelfCorr in the real binding experiment (where two sets of queries for SelfCorr are included in two sets of decommitment queries), it can do so even in a hybrid binding experiment where two sets of queries for SelfCorr are included in a single set of decommitment queries. Next, we observe that from a result that is implicitly shown in [KRR14], it follows that if a CNS adversary passes the low-degree test, the adversary cannot let the receiver recover two different values from SelfCorr in the hybrid binding experiment.

Formal proof. We prove Lemma E.1 by showing the following stronger lemma (which we will reuse later in the proof of soundness (Lemma E.3)).

Lemma E.2. There exists a polynomial κ_{dec} such that the following holds. Let ϵ be any negligible function and $C_1^* = (C.Com_1^*, C.Dec_1^*)$ be any well-behaving κ_{dec} -CNS cheating committer-decommitter against $\langle C_1, R_1 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following binding condition holds with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com_1^*, R.Com_1 \rangle$.

• Binding Condition: If it holds

$$\Pr\left[b_{\scriptscriptstyle LD} = 1 \;\middle|\; b_{\scriptscriptstyle LD} \leftarrow \mathsf{P-LD-Test}^{\mathsf{C.Dec}^*_1(\mathsf{st}_{\scriptscriptstyle C},\cdot)}_{d,\{D^\mu_b\}_{b\in[0,1],\mu\in[M]},3\zeta}\right] \geq 1 - \epsilon(\lambda) \tag{E.2}$$

for $d:=m_{io}|\pmb{H}|$ and $D_b^\mu:=D(X_{1,\text{in}}^\mu)$, then for every $i\in[N_{io}]$ it holds $\Pr[b_{\scriptscriptstyle BAD}=1]\leq \mathsf{negl}(\lambda)$ in the following probabilistic experiment.

1. For $\forall b \in \{0, 1\}$, sample $\{Q_b^{\mu}\}_{\mu \in [M]}$ as follows.

Algorithm 6 Low-degree Test Procedure LD-Test $_{d,D,\mathcal{I}}^{\mathcal{A}}$

- 1. Run $(Q, st) \leftarrow LD$ -Test. Q_D .
- 2. Run (out, A) $\leftarrow \mathcal{A}(Q)$.
- 3. Output $b := \mathsf{LD}\text{-}\mathsf{Test}.\mathsf{D}_{d,\zeta}(\mathsf{st},A)$.

.....

Subroutine LD-Test. Q_D :

- 1. Choose λ random lines $L_1, \ldots, L_{\lambda} : \mathbf{F} \to D$.
- 2. Output (Q, st), where $Q = \{L_j(t)\}_{j \in [\lambda], t \in F}$ and $st := \{L_j\}_{j \in [\lambda]}$.

Subroutine LD-Test.D_{d,ζ}(st, A):

1. Output 1 if and only if

$$\left|\left\{j\in[\lambda]\;\middle|\;\mathrm{isLD}_d\left(\left\{A(L_j(t))\right\}_{t\in F}\right)=1\right\}\right|\geq\lambda-\zeta\ .$$

Subroutine isLD_d($\{z_i\}_{i \in F}$)

1. Output 1 if the function $f: i \mapsto z_i$ can be expressed as a degree-d polynomial, and output \bot in any other cases.

Algorithm 7 Parallel Low-degree Test Procedure P-LD-Test $_{d,(D_i)_{i\in[Y]},\mathcal{I}}^{\mathcal{A}}$

- 1. Run $(Q_i, \mathsf{st}_i) \leftarrow \mathsf{LD}\text{-}\mathsf{Test}.\mathsf{Q}_{D_i}$ for every $i \in [K]$.
- 2. Run (out, $\{A_i\}_{i \in [K]}$) $\leftarrow \mathcal{A}(\{Q_i\}_{i \in [K]})$.
- 3. Output 1 if and only if LD-Test.D_{d, ζ}(st_i, A_i) = 1 for $\forall i \in [K]$.
 - (a) $Run(Q_{b,0}^{\mu}, \mathsf{st}_{b,0}^{\mu}) \leftarrow \mathsf{LD}\text{-Test.Q}_{D(X_{1,\mathrm{in}}^{\mu})} for \, \forall \mu \in [M].$
 - $(b) \ \mathit{Run} \ (Q_{b,1}^{\mu}, \mathsf{st}_{b,1}^{\mu}) \leftarrow \mathsf{SelfCorr}. \mathsf{Q}_{D(X_{1:in}^{\mu})}(\{(\mu,1,i)\}) \ \mathit{for} \ \forall \mu \in [M].$
 - (c) Let $Q_b^{\mu} := Q_{b,0}^{\mu} \cup Q_{b,1}^{\mu}$.
 - 2. $Run\{\tilde{X}_{b}^{*,\mu:\mu}\}_{b\in\{0,1\},\mu\in[M]} \leftarrow C.Dec_{1}^{*}(st_{C},\{Q_{b}^{\mu}\}_{b\in\{0,1\},\mu\in[M]}).$
 - 3. Let $b_{BAD} := 1$ if and only if both of the following hold.
 - LD-Test. $D_{m_{lo}|H|,3\zeta}(\mathsf{st}_{b\,0}^{\mu}, \tilde{X}_{b}^{*\mu;\mu}) = 1 \ for \ \forall b \in \{0,1\}, \mu \in [M].$
 - $\ \exists \mu \in [M] \ \textit{s.t.} \ A_0^{\mu}(\mu, 1, i) \neq A_1^{\mu}(\mu, 1, i), \ \textit{where} \ A_b^{\mu} \coloneqq \mathsf{SelfCorr.Rec}_{m_{to}|H|}(\mathsf{st}_{b, 1}^{\mu}, \tilde{X}^{*}_b^{\mu; \mu}) \ \textit{for} \ \forall b \in \{0, 1\}.$

To see that Lemma E.2 indeed implies Lemma E.1, observe that if $b_{BAD} = 1$ holds in the experiment in Lemma E.1, $b_{BAD} = 1$ holds in the experiment in Lemma E.2 due to the definitions of R.Dec.Q₁ and R.Dec.D₁.

Proof of Lemma E.2. We let κ_{dec} be the polynomial κ_1 that is given in Section K.1. Fix any ϵ and C_1^* , and assume for contradiction that for infinitely many λ , with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_1^*, \mathsf{R.Com}_1 \rangle$, Equation (E.2) holds but $\Pr[b_{\text{BAD}} = 1] \geq 1/\mathsf{poly}(\lambda)$. Let $\Lambda \subseteq \mathbb{N}$ be the set of such λ . In what follows, we consider a sequence of claims to obtain a contradiction with Lemma K.2 (Consistency of SelfCorr) in Section K.1, which roughly says that if a CNS adversary passes the low-degree test, it cannot let two different values recovered from SelfCorr when two sets of queries are queried together.

Claim E.1. Let \mathcal{A}_1 be the following algorithm.

- *On input* (aux, $\{Q_b^{\mu}\}_{b \in \{0,1\}, \mu \in [M]}$):
 - 1. $Run\{\tilde{X}^{*\mu:\mu}_b\}_{b\in\{0,1\},\mu\in[M]} \leftarrow \text{C.Dec}_1^*(\text{aux},\{Q_b^\mu\}_{b\in\{0,1\},\mu\in[M]}).$
 - 2. Output $\{\tilde{X}_{b}^{*\mu:\mu}\}_{b\in\{0,1\},\mu\in[M]}$.

Then, for $\forall \lambda \in \Lambda$, there exists $\text{aux} \in \{0,1\}^{\text{poly}(\lambda)}$ such that all of the following hold, where we let $d := m_{io}|\mathbf{H}|$ and $D^{\mu} := D(X_{1:in}^{\mu})$ in the following.

1. For $\forall \{Q_h^{\mu}\}_{h\in\{0,1\},\mu\in[M]}$ such that $Q_h^{\mu}\subseteq D^{\mu}$, it holds

$$\Pr\left[\begin{array}{c} \exists \mu \in [M], \boldsymbol{q} \in \boldsymbol{Q}_{0}^{\mu} \cap \boldsymbol{Q}_{1}^{\mu} \\ s.t. \ \boldsymbol{A}_{0}^{\mu}(\boldsymbol{q}) \neq \bot \\ \wedge \boldsymbol{A}_{1}^{\mu}(\boldsymbol{q}) \neq \bot \\ \wedge \boldsymbol{A}_{0}^{\mu}(\boldsymbol{q}) \neq \boldsymbol{A}_{1}^{\mu}(\boldsymbol{q}) \end{array}\right] \left\{\boldsymbol{A}_{b}^{\mu}\right\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathcal{A}_{1}(\mathsf{aux}, \{\boldsymbol{Q}_{b}^{\mu}\}_{b \in \{0,1\}, \mu \in [M]})\right] \leq \mathsf{negl}(\lambda) \ . \tag{E.3}$$

2. It holds

$$\Pr\left[\begin{array}{c|c} \forall b \in \{0,1\}, \mu \in [M], & (Q_b^{\mu}, \operatorname{st}_b^{\mu}) \leftarrow \operatorname{LD-Test.Q}_{D^{\mu}} for \ \forall b \in \{0,1\}, \mu \in [M] \\ \operatorname{LD-Test.D}_{d,3\zeta}(\operatorname{st}_b^{\mu}, A_b^{\mu}) = 1 & \{A_b^{\mu}\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathcal{R}_1(\operatorname{aux}, \{Q_b^{\mu}\}_{b \in \{0,1\}, \mu \in [M]}) \end{array}\right] \geq 1 - \epsilon(\lambda) \ . \tag{E.4}$$

3. There exist $z^1 \in D^1, \dots, z^M \in D^M$ such that it holds

$$\Pr\left[\begin{array}{c} \exists \mu \in [M] \ s.t. \\ \tilde{A}_{0}^{\mu}(z^{\mu}) \neq \tilde{A}_{1}^{\mu}(z^{\mu}) \end{array} \right. \left. \begin{array}{c} (Q_{b}^{\mu}, \mathsf{st}_{b}^{\mu}) \leftarrow \mathsf{SelfCorr.Q}_{D^{\mu}}(\{z^{\mu}\}) \ for \ \forall b \in \{0,1\}, \mu \in [M] \\ \{A_{b}^{\mu}\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathcal{A}_{1}(\mathsf{aux}, \{Q_{b}^{\mu}\}_{b \in \{0,1\}, \mu \in [M]}) \\ \tilde{A}_{b}^{\mu} \coloneqq \mathsf{SelfCorr.Rec}_{d}(\mathsf{st}_{b}^{\mu}, A_{b}^{\mu}) \ for \ \forall b \in \{0,1\}, \mu \in [M] \end{array} \right] \geq 1/\mathsf{poly}(\lambda) \ . \tag{E.5}$$

Proof. Fix any $\lambda \in \Lambda$. When aux := st_C is chosen randomly as $(st_C, com) \leftarrow \langle C.Com_1^*, R.Com_1 \rangle$, the above three equations hold with non-negligible probability since C^* is well-behaving and CNS and it breaks the binding condition. Hence, from an average argument, there exists aux such that the above three equations hold.

Claim E.2. Let \mathcal{A}_1 be the algorithm that is defined in Claim E.1, and fix any λ , aux, and z^1, \ldots, z^M on which Equations (E.3), (E.4) and (E.5) hold. Then, it holds

$$\Pr\left[\begin{array}{l} \exists \mu \in [M] \ s.t. \\ \tilde{A}_{0}^{\mu}(z^{\mu}) \neq \tilde{B}_{1}^{\mu}(z^{\mu}) \end{array}\right. \left. \begin{array}{l} (Q_{b}^{\mu}, \operatorname{st}_{b}^{\mu}) \leftarrow \operatorname{SelfCorr}.Q_{D^{\mu}}(\{z^{\mu}\}) \ for \ \forall b \in \{0,1\}, \mu \in [M] \\ \{A_{b}^{\mu}\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathcal{A}_{1}(\operatorname{aux}, \{S_{b}^{\mu}\}_{b \in \{0,1\}, \mu \in [M]}), \\ where \ S_{0}^{\mu} \coloneqq Q_{0}^{\mu} \cup Q_{1}^{\mu} \ and \ S_{1}^{\mu} \coloneqq Q_{1}^{\mu} \\ \tilde{A}_{b}^{\mu} \coloneqq \operatorname{SelfCorr}.\operatorname{Rec}_{d}(\operatorname{st}_{b}^{\mu}, A_{b}^{\mu}) \ for \ \forall b \in \{0,1\}, \mu \in [M] \\ \tilde{B}_{1}^{\mu} \coloneqq \operatorname{SelfCorr}.\operatorname{Rec}_{d}(\operatorname{st}_{b}^{\mu}, A_{b}^{\mu}) \ for \ \forall \mu \in [M] \end{array}\right] \geq 1/\operatorname{poly}(\lambda) \ . \tag{E.6}$$

Proof. Since Equation (E.5) holds, the CNS of \mathcal{A}_1 (which is inherited from the CNS of C.Dec₁*) implies that we can obtain Equation (E.6) by showing

$$\Pr\left[\begin{array}{c} \forall \mu \in [M], \\ \tilde{A}_{1}^{\mu}(z^{\mu}) = \tilde{B}_{1}^{\mu}(z^{\mu}) \end{array}\right. \left(\begin{array}{c} (Q_{b}^{\mu}, \mathsf{st}_{b}^{\mu}) \leftarrow \mathsf{SelfCorr}. Q_{D^{\mu}}(\{z^{\mu}\}) \text{ for } \forall b \in \{0, 1\}, \mu \in [M] \\ \{A_{b}^{\mu}\}_{b \in [0, 1\}, \mu \in [M]} \leftarrow \mathcal{A}_{1}(\mathsf{aux}, \{S_{b}^{\mu}\}_{b \in [0, 1\}, \mu \in [M]}), \\ \text{where } S_{0}^{\mu} := Q_{0}^{\mu} \cup Q_{1}^{\mu} \text{ and } S_{1}^{\mu} := Q_{1}^{\mu} \\ \tilde{A}_{b}^{\mu} := \mathsf{SelfCorr}. \mathsf{Rec}_{d}(\mathsf{st}_{b}^{\mu}, A_{b}^{\mu}) \text{ for } \forall b \in \{0, 1\}, \mu \in [M] \\ \tilde{B}_{1}^{\mu} := \mathsf{SelfCorr}. \mathsf{Rec}_{d}(\mathsf{st}_{b}^{\mu}, A_{0}^{\mu}) \text{ for } \forall \mu \in [M] \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{E.7}$$

Hence, we focus on showing Equation (E.7).

First, we remark that from the construction of SelfCorr, the two values $\tilde{A}_1^{\mu}(z^{\mu})$ and $\tilde{B}_1^{\mu}(z^{\mu})$ in Equation (E.7) are sampled in the following manner.

- 1. For $\forall b \in \{0,1\}, \mu \in [M]$, choose λ random lines $L^{\mu}_{b,1}, \dots, L^{\mu}_{b,\lambda} : \mathbf{F} \to D^{\mu}$ such that each $L \in \{L^{\mu}_{b,1}, \dots, L^{\mu}_{b,\lambda}\}$ satisfies $L(0) = z^{\mu}$. Let $Q^{\mu}_b := \{L^{\mu}_{b,i}(t)\}_{j \in [\lambda], t \in \mathbf{F} \setminus \{0\}}$.
- 2. Run $\{A_b^{\mu}\}_{b\in\{0,1\},\mu\in[M]} \leftarrow \mathcal{A}_1(\mathsf{aux},\{S_b^{\mu}\}_{b\in\{0,1\},\mu\in[M]})$, where $S_0^{\mu} \coloneqq Q_0^{\mu} \cup Q_1^{\mu}$ and $S_1^{\mu} \coloneqq Q_1^{\mu}$.
- 3. For $\forall \mu \in [M]$, check that there exists $c_1^{\mu} \in F$ such that

$$\left|\left\{j\in[\lambda]\;\middle|\;\operatorname{Recon}_d\left(\left\{A_1^\mu(L_{1,j}^\mu(t))\right\}_{t\in F\backslash\{0\}}\right)=c_1^\mu\right\}\right|\geq 0.9\lambda\ .$$

Let $\tilde{A}_1^{\mu}(z^{\mu}) := c_1^{\mu}$ if such c_1^{μ} exists, and let $\tilde{A}_1^{\mu}(z^{\mu}) := \bot$ otherwise.

4. For $\forall \mu \in [M]$, check that there exists $d_1^{\mu} \in F$ such that

$$\left|\left\{j\in[\lambda]\;\middle|\; \mathsf{Recon}_d\left(\left\{A_0^\mu(L_{1,j}^\mu(t))\right\}_{t\in F\backslash\{0\}}\right)=d_1^\mu\right\}\right|\geq 0.9\lambda\ .$$

Let $\tilde{B}_1^{\mu}(z^{\mu}) := d_1^{\mu}$ if such d_1^{μ} exists, and let $\tilde{B}_1^{\mu}(z^{\mu}) := \bot$ otherwise.

In what follows, we always consider probability over this sampling.

Note that due to Equation (E.4) and Lemma K.1 (Correctness of SelfCorr) in Section K.1 (which roughly says that SelfCorr outputs \bot only with negligible probability for any adversary that passes the low-degree test), we have $\tilde{A}_1^{\mu}(z^{\mu}) \neq \bot$ and $\tilde{B}_1^{\mu}(z^{\mu}) \neq \bot$ for $\forall \mu \in [M]$ except with negligible probability. From the construction of Recon, this implies that except with negligible probability, we have

$$\left|\left\{j\in[\lambda]\;\middle|\;\bot\notin\left\{A_1^\mu(L_{1,j}^\mu(t))\right\}_{t\in F\setminus\{0\}}\right\}\right|\geq0.9\lambda$$

and

$$\left|\left\{j \in [\lambda] \;\middle|\; \bot \notin \left\{A_0^{\mu}(L_{1,j}^{\mu}(t))\right\}_{t \in F \setminus \{0\}}\right\}\right| \ge 0.9\lambda$$

for $\forall \mu \in [M]$. Combined with Equation (E.3), these two imply that except with negligible probability, we have

$$\left|\left\{j \in [\lambda] \mid A_0^{\mu}(L_{1,j}^{\mu}(t)) = A_1^{\mu}(L_{1,j}^{\mu}(t)) \neq \bot \text{ for } \forall t \in \mathbf{F} \setminus \{0\}\right\}\right| \ge 0.8\lambda \tag{E.8}$$

for $\forall \mu \in [M]$. Thus, from a union bound, we have $\tilde{A}_1^{\mu}(z^{\mu}) \neq \bot$, $\tilde{B}_1^{\mu}(z^{\mu}) \neq \bot$, and Equation (E.8) for $\forall \mu \in [M]$ except with negligible probability. From the definitions of $\tilde{A}_1^{\mu}(z^{\mu})$, $\tilde{B}_1^{\mu}(z^{\mu})$, this implies that we have $\tilde{A}_1^{\mu}(z^{\mu}) = \tilde{B}_1^{\mu}(z^{\mu})$ for $\forall \mu \in [M]$ except with negligible probability, as desired.

Claim E.3. Let \mathcal{A}_2 be the following algorithm.

- *On input* (aux, $\{Q^{\mu}\}_{\mu \in [M]}$):
 - 1. $Run\{A_b^{\mu}\}_{b\in\{0,1\},\mu\in[M]}\leftarrow \mathcal{A}_1(\mathsf{aux},\{Q_b^{\mu}\}_{b\in\{0,1\},\mu\in[M]}), where Q_0^{\mu}\coloneqq Q^{\mu} \ and \ Q_1^{\mu}\coloneqq\emptyset.$
 - 2. *Output* $\{A_0^{\mu}\}_{{\mu}\in[M]}$.

Then, for $\forall \lambda \in \Lambda$, there exist $\text{aux} \in \{0,1\}^{\text{poly}(\lambda)}$ and $z^1 \in D^1, \dots, z^M \in D^M$ such that

$$\Pr\left[\begin{array}{c|c} \forall \mu \in [M], & \left(Q^{\mu}, \operatorname{st}^{\mu}\right) \leftarrow \operatorname{LD-Test.Q}_{D^{\mu}} \textit{for } \mu \in [M] \\ \operatorname{LD-Test.D}_{d,3\zeta}(\operatorname{st}^{\mu}, A^{\mu}) = 1 & \left\{A^{\mu}\right\}_{\mu \in [M]} \leftarrow \mathcal{R}_{2}(\operatorname{aux}, \{Q^{\mu}\}_{\mu \in [M]}) \end{array}\right] \geq 1 - \operatorname{negl}(\lambda)$$

and

$$\Pr\left[\begin{array}{c} \exists \mu \in [M] \ s.t. \\ \tilde{A}_0^{\mu}(z^{\mu}) \neq \tilde{A}_1^{\mu}(z^{\mu}) \end{array} \right. \left. \begin{array}{c} (Q_b^{\mu}, \operatorname{st}_b^{\mu}) \leftarrow \operatorname{SelfCorr.Q}_{D^{\mu}}(\{z^{\mu}\}) \ for \ \forall b \in \{0,1\}, \mu \in [M] \\ \{A^{\mu}\}_{\mu \in [M]} \leftarrow \mathcal{A}_2(\operatorname{aux}, \{Q_0^{\mu} \cup Q_1^{\mu}\}_{\mu \in [M]}) \\ \tilde{A}_b^{\mu} \coloneqq \operatorname{SelfCorr.Rec}_d(\operatorname{st}_b^{\mu}, A^{\mu}) \ for \ \forall b \in \{0,1\}, \mu \in [M] \end{array} \right] \geq 1/\operatorname{poly}(\lambda) \ ,$$

where $d := m_{io}|\mathbf{H}|$ and $D^{\mu} := D(X_{1 \text{ in}}^{\mu})$.

Proof. This claim follows from Claim E.1, Claim E.2, and the assumption that C_1^* is CNS (which implies that \mathcal{A}_1 is CNS), since the latter implies that the output of $\mathcal{A}_2(\mathsf{aux}, \{Q_0^\mu \cup Q_1^\mu\}_{\mu \in [M]})$ is computationally indistinguishable from that of $\mathcal{A}_1(\mathsf{aux}, \{S_b^\mu\}_{b \in [0,1], \mu \in [M]})$, where $S_0^\mu \coloneqq Q_0^\mu \cup Q_1^\mu$ and $S_1^\mu \coloneqq Q_1^\mu$.

Clearly, Claim E.3 contradicts with (the straightforward parallel version of) Lemma K.2 (Consistency of SelfCorr). This concludes the proof of Lemma E.2.

E.3 Proof of Soundness

We prove the soundness against cheating adversaries that are no-signaling and well-behaving in the following sense.

Definition 19 (No-signaling committer-prover). A cheating committer-prover $C_1^* = (C.Com_1^*, C.Prv_1^*)$ against $\langle C_1, R_1 \rangle$ is κ_{max} -CNS if $C.Prv_1^*(st_C, \cdot)$ is κ_{max} -CNS for $\forall (st_C, com) \leftarrow \langle C.Com_1^*, R.Com_1 \rangle$.

Definition 20 (Well-behaving committer-prover). A cheating committer-prover $C_1^* = (C.Com_1^*, C.Prv_1^*)$ against $\langle C_1, R_1 \rangle$ is well-behaving if both of the following hold.

1. Consistency on $D(X_{1,\text{in}}^{\mu})$: For $\forall \lambda \in \mathbb{N}$, $\forall \{Q_0^{\mu:\nu}\}_{\mu,\nu \in [M]}, \{Q_1^{\mu:\nu}\}_{\mu,\nu \in [M]} \subseteq (D(X))^{M^2}$, $\forall \alpha,\beta,\gamma,\delta \in [M]$ such that $\exists \xi \in \{\alpha,\beta\} \cap \{\gamma,\delta\}$, and $\forall \mathbf{q} \in Q_0^{\alpha:\beta} \cap Q_1^{\gamma:\delta} \cap D(X_{1,\text{in}}^{\xi})$, it holds

$$\Pr \left[\begin{array}{c|c} \pi^{*\alpha:\beta}_{0}(\boldsymbol{q}) \neq \bot & (\operatorname{st}_{C}, \operatorname{com}) \leftarrow \langle \operatorname{C.Com}_{1}^{*}, \operatorname{R.Com}_{1} \rangle \\ \wedge \, \pi^{*\gamma:\delta}_{1}(\boldsymbol{q}) \neq \bot & (f, \{\pi^{*\mu:\nu}_{0}\}_{\mu,\nu \in [M]}) \leftarrow \operatorname{C.Prv}_{1}^{*}(\operatorname{st}_{C}, \{Q^{\mu:\nu}_{0}\}_{\mu,\nu \in [M]}) \\ \wedge \, \pi^{*\alpha:\beta}_{0}(\boldsymbol{q}) \neq \pi^{*\gamma:\delta}_{1}(\boldsymbol{q}) & (f, \{\pi^{*\mu:\nu}_{1}\}_{\mu,\nu \in [M]}) \leftarrow \operatorname{C.Prv}_{1}^{*}(\operatorname{st}_{C}, \{Q^{\mu:\nu}_{1}\}_{\mu,\nu \in [M]}) \end{array} \right] \leq \operatorname{negl}(\lambda) \ .$$

2. Consistency on $D(X^{\mu})$: For $\forall \lambda \in \mathbb{N}$, $\forall \{Q^{\mu:\nu}\}_{\mu,\nu \in [M]} \subseteq (D(X))^{M^2}$, $\forall \alpha,\beta,\gamma,\delta \in [M]$ such that $\exists \xi \in \{\alpha,\beta\} \cap \{\gamma,\delta\}$, and $\forall \mathbf{q} \in Q^{\alpha:\beta} \cap Q^{\gamma:\delta} \cap D(X^{\xi})$, it holds

$$\Pr \left[\begin{array}{c} \pi^{*\alpha:\beta}(\boldsymbol{q}) \neq \bot \\ \wedge \, \pi^{*\gamma:\delta}(\boldsymbol{q}) \neq \bot \\ \wedge \, \pi^{*\alpha:\beta}(\boldsymbol{q}) \neq \pi^{*\gamma:\delta}(\boldsymbol{q}) \end{array} \right| \quad (\operatorname{st}_{C}, \operatorname{com}) \leftarrow \langle \operatorname{C.Com}_{1}^{*}, \operatorname{R.Com}_{1} \rangle \\ (f, \{\pi^{*\mu:\gamma}\}_{\mu,\nu \in [M]}) \leftarrow \operatorname{C.Prv}_{1}^{*}(\operatorname{st}_{C}, \{Q^{\mu:\gamma}\}_{\mu,\nu \in [M]}) \end{array} \right] \leq \operatorname{negl}(\lambda) \ .$$

Remark 5 (Intuition of well-behaving adversaries). Roughly speaking, the above definition of well-behaving adversaries guarantees the following. Recall that in $\langle C_1, R_1 \rangle$, the (μ, ν) -th instance of (PCP.P, PCP.V) is used for proving consistency of a pair of views (view^{μ}, view^{ν}), and the receiver expects that it can query to an LDE $X^{\mu,\nu}$ of a satisfying assignment to the variables in $\varphi_f^{\mu,\nu}$. Also, recall that the receiver expects that through $X^{\mu,\nu}$, it can query to an LDE $X_{1,\text{in}}^{\mu}$ of an assignment to the variables indexed by $I_{1,\text{in}}^{\mu}$, and as remarked in Remark 3, the assignment to these variables should be the initial state of P^{μ} (which is committed in the commit phase). Also, recall that the receiver expects that through $X^{\mu,\nu}$, it can query to an LDE X^{μ} of an assignment to the variables indexed by I^{μ} , and as remarked in Remark 3, the assignment to these variables should depend only on view^{μ}, meaning that the same values should be assigned to these variables in the (μ, ν) -th instance of (PCP.P, PCP.V) and in the (μ, ξ) -th one. Now, the first condition of well-behaving adversaries guarantees that once the commit phase is completed, the adversary does not give different answers to the queries to $X_{1,\text{in}}^{\mu}$ in different invocations. Regarding the second condition, it guarantees that once the commit phase is completed, the adversary does not give different answers to the queries to X^{μ} in different instances of (PCP.P, PCP.V) in a single invocation.

The formal statement of the soundness of $\langle C_1, R_1 \rangle$ is given below.

Lemma E.3. There exists a polynomial κ_{prv} such that the following holds. Let ϵ_{SND} be any negligible function, E_1 be the extractor in Algorithm 8, and $C_1^* = (\text{C.Com}_1^*, \text{C.Prv}_1^*)$ be any well-behaving κ_{prv} -CNS cheating committer-prover against $\langle C_1, R_1 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following soundness condition holds with overwhelming probability over the choice of $(\text{st}_C, \text{com}) \leftarrow \langle \text{C.Com}_1^*, \text{R.Com}_1 \rangle$.

• Soundness Condition: If it holds

$$\Pr \left[b = 1 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R}.\mathsf{Prv}.\mathsf{Q}_1; \ (f, \pi^*) \leftarrow \mathsf{C}.\mathsf{Prv}_1^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R}.\mathsf{Prv}.\mathsf{D}_1(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \ge 1 - \epsilon_{\scriptscriptstyle SND}(\lambda) \ , \tag{E.9}$$

then there exists $x_{COM}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that

$$\forall i \in [n], \Pr\left[x_i = x_i^* \mid (\bot, x_i) \leftarrow \langle E_1^{\mathsf{C.Prv}_1^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_1(\mathsf{com}, i) \rangle\right] \geq 1 - \mathsf{negl}(\lambda) \tag{E.10}$$

and

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x^*_{\scriptscriptstyle COM})=0\end{array}\right| \left(\begin{array}{c}(Q,\operatorname{st}_{R})\leftarrow\operatorname{R.Prv.Q}_1;\ (f,\pi^*)\leftarrow\operatorname{C.Prv}_1^*(\operatorname{st}_{C},Q)\\ b\leftarrow\operatorname{R.Prv.D}_1(\operatorname{st}_{R},\operatorname{com},f,\pi^*)\end{array}\right] \leq \operatorname{negl}(\lambda)\ . \tag{E.11}$$

Algorithm 8 Extractor E_1 (against $\langle C_1, R_1 \rangle$)

Input: com, i, and $\{Q^{\mu}\}_{\mu\in[M]}\subset D(\overline{X^1_{1\text{ in}}})\times\cdots\times D(X^M_{1\text{ in}})$

- 1. Run $(f, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \text{C.Prv}_1^*(\text{st}_C, \{Q^{\mu:\nu}\}_{\mu,\nu\in[M]})$, where each $Q^{\mu:\nu}$ is defined as $Q^{\mu:\nu} := Q^{\mu}$ if $\mu = \nu$ and $Q^{\mu:\nu} := \emptyset$ otherwise
- 2. Output $\{\tilde{X}^{*\mu:\mu}\}_{\mu\in[M]}$ as the decommitment, where $\tilde{X}^{*\mu:\mu}\coloneqq\pi^{*\mu:\mu}$.

We directly go to the formal proof. An overview of the proof is given in Section 4.

Proof. We let $\kappa_{\text{prv}} = \kappa_0 \cdot \kappa_{\text{max}}$, where κ_0 , κ_{max} are the polynomials that are given in Lemma D.4. Fix any ϵ_{SND} and $C_1^* = (\text{C.Com}_1^*, \text{C.Prv}_1^*)$ such that for infinitely many $\lambda \in \mathbb{N}$, Equation (E.9) holds with non-negligible probability over the choice of (st_C, com) ← ⟨C.Com₁*, R.Com₁⟩ (if no such ϵ_{SND} and C_1^* exist, the lemma holds trivially), and let Λ be the set of such λ .

Step 0: showing consistency properties on self-corrected C.Prv*.

First, we make preliminary observations on SelfCorr. Specifically, we observe that the consistency conditions of well-behaving adversaries (Definition 20) hold even on the outputs of SelfCorr.

Claim E.4 (Consistency on $D(X_{1,\text{in}}^{\mu})$). For every $\lambda \in \Lambda$, the following holds with overwhelming probability over the choice of $(\mathsf{st}_C,\mathsf{com}) \leftarrow \langle \mathsf{C}.\mathsf{Com}_1^*,\mathsf{R}.\mathsf{Com}_1 \rangle$.

• If Equation (E.9) holds, then there exists $\{(x_1^{\mu}, \dots, x_{N_{io}}^{\mu})\}_{\mu \in [M]} \in (\{0, 1\}^{N_{io}})^M$ such that for $\forall \alpha, \beta \in [M], \forall \xi \in \{\alpha, \beta\},$ and $\forall i^* \in \mathcal{I}_{1 \text{ in}}^{\xi}$,

$$\Pr \begin{bmatrix} A^{\alpha:\beta}(i^*) = x_{i^*}^{\xi} & (Q, \operatorname{st}) \leftarrow \operatorname{SelfCorr.Q}_{D(X_{1,\operatorname{in}}^{\xi})}(\{i^*\}) \\ (f, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \operatorname{C.Prv}_1^*(\operatorname{st}_C, \{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}), \\ where \ Q^{\mu:\nu} \coloneqq \emptyset \ for \ \forall (\mu, \nu) \neq (\alpha, \beta) \ and \ Q^{\alpha:\beta} \coloneqq Q \\ A^{\alpha:\beta} \coloneqq \operatorname{SelfCorr.Rec}_{m_{o}\mid H}(\operatorname{st}, \pi^{*\alpha:\beta}) \end{bmatrix} \ge 1 - \operatorname{negl}(\lambda) \ . \tag{E.12}$$

Proof. Fix any $\lambda \in \Lambda$.

First, we observe that, by using the argument in the proof of Lemma E.2, we can show that with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com_1^*, R.Com_1 \rangle$, if Equation (E.9) holds, then there exists $\{(x_1^\mu, \dots, x_{N_{10}}^\mu)\}_{\mu \in [M]} \in (F^{N_{10}})^M$ such that for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in I_{1,in}^\xi$, we have Equation (E.12). To see this, first observe that since the proof verification includes the low-degree test for $D(X_{1,in}^\xi)$, Equation (E.9) implies $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$,

$$\Pr \left[b_{\text{LD}} = 1 \left| \begin{array}{l} (Q, \text{st}) \leftarrow \text{LD-Test.Q}_{D(X_{1,\text{in}}^{\xi})} \\ (f, \{\pi^{*\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \text{C.Prv}_1^*(\text{st}_C, \{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq \emptyset \text{ for } \forall (\mu,\nu) \neq (\alpha,\beta) \text{ and } Q^{\alpha:\beta} \coloneqq Q \\ b_{\text{LD}} \coloneqq \text{LD-Test.D}_{m_{\text{to}}|H|,\zeta}(\text{st}, \pi^{*\alpha:\beta}) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda) \enspace .$$

From this observation, it follows that given Equation (E.9), we can argue as in Lemma E.2 to show that for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in \mathcal{I}_{1, \text{in}}^{\xi}$, the value of $A^{\alpha; \beta}(i^*)$ is unique when computed as in Equation (E.12). Second, since the prove phase of $\langle C_1, R_1 \rangle$ is just parallel executions of (PCP.P, PCP.V), the everywhere local con-

Second, since the prove phase of $\langle C_1, R_1 \rangle$ is just parallel executions of (PCP.P, PCP.V), the everywhere local consistency of SelfCorr (Claim D.4) and Lemma K.3 in Section K.1 (which guarantees that the values recovered from SelfCorr_{$m_{io}|H|,D(X_{1,in}^{\xi})$} and SelfCorr_{$m_{io}|H|,D(X)$} are equal) imply that for any (st_C, com) $\leftarrow \langle \text{C.Com}_1^*, \text{R.Com}_1 \rangle$, if Equation (E.9) holds, then for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in \mathcal{I}_{1,in}^{\xi}$, we have $\Pr\left[A^{\alpha,\beta}(i^*) = 0 \lor A^{\alpha,\beta}(i^*) = 1\right] \ge 1 - \text{negl}(\lambda)$, where the probability is taken as in Equation (E.12).³²

By combining the above two, we obtain the claim.

Claim E.5 (Consistency on $D(X^{\mu})$). For every $\lambda \in \Lambda$, the following holds with overwhelming probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_1^*, \mathsf{R.Com}_1 \rangle$.

• If Equation (E.9) holds, then $\forall \alpha, \beta, \gamma, \delta \in [M]$ such that $\exists \xi \in \{\alpha, \beta\} \cap \{\gamma, \delta\}$, and $\forall i^* \in \mathcal{I}^{\xi}$,

$$\Pr\begin{bmatrix} A^{\alpha:\beta}(i^{*}) = A^{\gamma:\delta}(i^{*}) & (Q_{0}, \mathsf{st}_{0}) \leftarrow \mathsf{SelfCorr}.\mathsf{Q}_{D(X^{\xi})}(\{i^{*}\}) \\ (Q_{1}, \mathsf{st}_{1}) \leftarrow \mathsf{SelfCorr}.\mathsf{Q}_{D(X^{\xi})}(\{i^{*}\}) \\ (f, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \mathsf{C.Prv}_{1}^{*}(\mathsf{st}_{C}, \{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}), \\ where \ Q^{\mu:\nu} \coloneqq \emptyset \ for \ \forall (\mu,\nu) \notin \{(\alpha,\beta),(\gamma,\delta)\}, \\ Q^{\alpha:\beta} \coloneqq Q_{0}, \ and \ Q^{\gamma:\delta} \coloneqq Q_{1} \\ A^{\alpha:\beta} \coloneqq \mathsf{SelfCorr}.\mathsf{Rec}_{(m-1)|H|}(\mathsf{st}_{0}, \pi^{*\alpha:\beta}) \\ A^{\gamma:\delta} \coloneqq \mathsf{SelfCorr}.\mathsf{Rec}_{(m-1)|H|}(\mathsf{st}_{1}, \pi^{*\gamma:\delta}) \end{bmatrix} \ge 1 - \mathsf{negl}(\lambda) \ . \tag{E.13}$$

Proof. This claim can be proven similarly to Lemma E.2.

Next, we introduce a definition that we use in the rest of the proof.

Definition 21. For any $\lambda \in \Lambda$, we say that $(\operatorname{st}_C, \operatorname{com})$ is good if under the condition that $(\operatorname{st}_C, \operatorname{com})$ is output by $(\operatorname{C.Com}_1^*, \operatorname{R.Com}_1)$, the following hold.

- Equation (E.9) holds.
- There exists $\{(x_1^{\mu}, \dots, x_{N_{io}}^{\mu})\}_{\mu \in [M]} \in (\{0, 1\}^{N_{io}})^M$ such that Equation (E.12) holds for $\forall \alpha, \beta \in [M], \ \forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in \mathcal{I}_{1 \text{ in}}^{\xi}$.
- Equation (E.13) holds for $\forall \alpha, \beta, \gamma, \delta \in [M]$ such that $\exists \xi \in \{\alpha, \beta\} \cap \{\gamma, \delta\}$, and $\forall i^* \in \mathcal{I}^{\xi}$.

Furthermore, for any $\lambda \in \Lambda$ and any good (st_C, com), we say that $\{(\mathsf{st}_0^\mu, \mathsf{i-msgs}_1^\mu)\}_{\mu \in [M]} = \{(x_1^\mu, \dots, x_{N_{lo}}^\mu)\}_{\mu \in [M]} \in (\{0, 1\}^{N_{lo}})^M$ is the (unique) good MPC initial state if Equation (E.12) holds on $\{(x_1^\mu, \dots, x_{N_{lo}}^\mu)\}_{\mu \in [M]}$ for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in I_{1 \text{ in}}^\xi$.

From Claim E.4 and Claim E.5, it follows that for proving Lemma E.3, it suffices to show that for any good (st_C , com), there exists $x_{com}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that Equations (E.10) and (E.11) hold.

 $^{^{32}}$ The everywhere local consistency implies that if the three variables of a clause in $\varphi_f^{\alpha\beta}$ are assigned by SelfCorr, the values assigned to them are 0 or 1 with overwhelming probability. Then, CNS implies that even when each of the three variables is assigned by SelfCorr individually, the value assigned to it is still 0 or 1 with overwhelming probability.

Step 1: showing that E_1 succeeds with overwhelming probability.

Claim E.6. For every $\lambda \in \Lambda$ and good $(\mathsf{st}_C, \mathsf{com})$, Equation (E.10) holds for $x^*_{\mathsf{com}} = (x^*_1, \dots, x^*_n) \in \{0, 1\}^n$ that is defined as follows: let $\{(\mathsf{st}^\mu_0, \mathsf{i-msgs}^\mu_1)\}_{\mu \in [M]} = \{(x^\mu_1, \dots, x^\mu_{N_{io}})\}_{\mu \in [M]} \in (\{0, 1\}^{N_{io}})^M$ be the good MPC initial state; then, let $x^*_i := x^1_i \oplus \dots \oplus x^M_i$ for $\forall i \in [n]$.

Proof. Fix any $\lambda \in \Lambda$ and good (st_C , com), and let x^*_{com} be defined as in the claim statement. First, from the constructions of R.Dec₁ and E_1 , the CNS of C.Prv $_1^*$, and Equation (E.12), it follows that we have Equation (E.10) if we have

$$\Pr\left[\begin{array}{l} \forall \mu \in [M], \\ \mathsf{LD-Test.D}_{m_{\mathrm{to}}|H|,3\zeta}(\mathsf{st}^{\mu},\pi^{*\mu:\mu}) = 1 \end{array}\right| \left(\begin{array}{l} (Q^{\mu},\mathsf{st}^{\mu}) \leftarrow \mathsf{LD-Test.Q}_{D(X^{\mu}_{1,\mathrm{in}})} \text{ for } \mu \in [M] \\ (f,\{\pi^{*\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{C.Prv}^*_1(\mathsf{st}_C,\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq Q^{\mu} \text{ if } \mu = \nu \\ Q^{\mu:\nu} \coloneqq \emptyset \text{ otherwise} \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{E.14}$$

Second, from the construction of R.Prv₁ and the CNS of C.Prv₁*, it follows that we indeed have Equation (E.14) due to Equation (E.9) (which is guaranteed to hold since (st_C , com) is good).

Step 2: showing that C.Prv₁* fails to prove false statement.

Claim E.7. For every $\lambda \in \Lambda$ and good (st_C, com), Equation (E.11) holds for x_{com}^* that is defined as in Claim E.6. To prove this claim, we show the following claim.

Claim E.8. For every $\lambda \in \Lambda$ and good (st_C, com), we have

$$\Pr\bigg[f(x_{com}^*) = 1 \ \middle| \ \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R}.\mathsf{Prv}.\mathsf{Q}_1 \\ (f, \pi^*) \leftarrow \mathsf{C}.\mathsf{Prv}^*(\mathsf{st}_C, Q) \end{array}\bigg] \geq 1 - \mathsf{negl}(\lambda) \ , \tag{E.15}$$

where $x_{COM}^* \in \{0, 1\}^n$ is defined as in Claim E.6.

Note that Claim E.8 implies Claim E.7 since Equations (E.15) implies Equation (E.11).

Proof of Claim E.8. We show three subclaims. Let P-SelfCorr be a parallel version of SelfCorr; see Algorithm 9.

$\textbf{Algorithm 9} \ \text{Parallel Self-Correction Procedure P-SelfCorr}^{\text{C.Prv}_i^*(\text{st}_C,*)}_{d,D}$

Input: $\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}$.

- 1. Run $(\tilde{Q}^{\mu:\nu}, \mathsf{st}_Q^{\mu:\nu}) \leftarrow \mathsf{SelfCorr.Q}_D(Q^{\mu:\nu})$ for each $\mu, \nu \in [M]$.
- 2. Run $(f, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \text{C.Prv}_1^*(\text{st}_C, \{\tilde{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}).$
- 3. $\operatorname{Run} A^{\mu:\nu} \coloneqq \operatorname{SelfCorr.Rec}_d(\operatorname{St}_Q^{\mu:\nu}, \pi^{*\mu:\nu})$ for each $\mu, \nu \in [M]$.
- 4. Output $(f, \{A^{\mu:\nu}\}_{\mu,\nu\in[M]})$.

Sub-Claim E.1. There exists a negligible function ϵ_1 such that for every $\lambda \in \Lambda$ and every good (st_C , com), we have

$$\Pr\left[\mathsf{Correct}\left(\{S^{\mu}\}_{\mu \in [M]} \right) \middle| \begin{array}{l} S^{\mu} \coloneqq \{s_{i}^{\mu}\}_{i \in [\log^{2}\lambda]} \ for \ \forall \mu \in [M], \ where \ s_{i}^{\mu} \leftarrow \mathcal{I}_{1,\mathsf{in},\mathsf{LDE}}^{\mu} \\ (f, \{A^{\mu : \nu}\}_{\mu, \nu \in [M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H|,D(X)}^{\mathsf{C.Prv}_{1}^{\mu}(\mathsf{stc}, \cdot)}(\{Q^{\mu : \nu}\}_{\mu, \nu \in [M]}), \\ where \ Q^{\mu : \nu} \coloneqq S^{\mu} \cup S^{\nu} \ for \ \forall \mu, \nu \in [M] \end{array} \right] \geq 1 - \epsilon_{1}(\lambda) \ , \tag{E.16}$$

where the event $Correct(\{S^{\mu}\}_{\mu\in[M]})$ is defined as follows. Let the correct view $\{view^{\mu}\}_{\mu\in[M]}$ be the views of the parties in the execution of Π on $(f, \{(st_0^{\mu}, i\text{-msgs}_1^{\mu})\}_{\mu\in[M]})$, where $\{(st_0^{\mu}, i\text{-msgs}_1^{\mu})\}_{\mu\in[M]}$ is the good MPC initial states that is determined by (st_C, com) . For $\forall \mu, \nu \in [M]$, let the correct assignment $A_{corr}^{\mu,\nu}$ (to the variables in $\varphi_f^{\mu,\nu}$) be the assignment that is obtained from $(view^{\mu}, view^{\nu})$ as in Remark 3. Then, $Correct(\{S^{\mu}\}_{\mu\in[M]})$ is the event that $A^{\mu,\nu}(s) = A_{corr}^{\mu,\nu}(s)$ holds for $\forall \mu, \nu \in [M]$, $\forall \xi \in \{\mu, \nu\}, \forall s \in S^{\xi}$.

Proof. To show this subclaim, it suffices to show that there exists a negligible function ϵ such that for every $\lambda \in \Lambda$ and good (st_C , com), the following holds.

• For $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, $\forall i^* \in I_{1 \text{ in}}^{\xi}$, we have

$$\Pr\left[A^{\alpha:\beta}(i^*) = A_{\operatorname{corr}}^{\alpha:\beta}(i^*) \left| \begin{array}{c} (f,\{A^{\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H,D(X)}^{\mathsf{C.Prv}_1^*(\operatorname{st}_{C,\cdot})}(\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq \emptyset \text{ for } \forall (\mu,\nu) \neq (\alpha,\beta) \text{ and } Q^{\alpha:\beta} \coloneqq \{i^*\} \end{array} \right] \geq 1 - \epsilon(\lambda) \ . \tag{E.17}$$

(Indeed, given Equation (E.17), we can obtain Equation (E.16) for a negligible function ϵ_1 by using Claim D.7, the CNS of C.Prv₁, and a union bound.) Now, we conclude the proof by noticing that for every $\lambda \in \Lambda$ and every good (st_C, com), the above follows from Lemma K.3 in Section K.1 (which guarantees that the values recovered from SelfCorr $_{m_{10}|H|,D(X_{110}^{\xi})}$ and SelfCorr_{m|H|,D(X)} are equal) since $\{(st_0^{\mu}, i\text{-msgs}_1^{\mu})\}_{\mu \in [M]}$ is the good MPC initial states.

Sub-Claim E.2. There exists a negligible function ϵ_2 such that for every $\lambda \in \Lambda$, every good ($\operatorname{st}_C, \operatorname{com}$), and every $\ell \in \Lambda$ $[N_{\text{round}}]$, we have

$$\Pr\left[\begin{array}{l} S^{\mu} \coloneqq \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \text{Correct}\left(\{S^{\mu}\}_{\mu \in [M]}\right) \\ \wedge \neg \mathsf{Correct}\left(\{T^{\mu}\}_{\mu \in [M]}\right) \end{array} \right| \begin{array}{l} S^{\mu} \coloneqq \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ t_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ T^{\mu} \coloneqq \{t_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ t_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ (f, \{A^{\mu : \nu}\}_{\mu, \nu \in [M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H, D(X)}^{\mathsf{C.Prv}_1^*(\mathsf{st}_C, \cdot)} (\{Q^{\mu : \nu}\}_{\mu, \nu \in [M]}), \\ where \ Q^{\mu : \nu} \coloneqq S^{\mu} \cup S^{\nu} \cup T^{\mu} \cup T^{\nu} \ for \ \forall \mu, \nu \in [M] \end{array} \right] \leq N_{\mathsf{Aug}} \cdot \epsilon_2(\lambda) \ ,$$

where $Correct(\{S^{\mu}\}_{\mu\in[M]})$ and $Correct(\{T^{\mu}\}_{\mu\in[M]})$ are defined as in Sub-Claim E.1.

Proof. Since the prove phase of $\langle C_1, R_1 \rangle$ is just parallel executions of (PCP.P, PCP.V), this subclaim follows from Claim D.5.

Sub-Claim E.3. There exists a negligible function ϵ_3 such that for every $\lambda \in \Lambda$, every good ($\mathsf{st}_C, \mathsf{com}$), and every $\ell \in \Lambda$

$$\Pr\left[\begin{array}{l} \mathsf{T}^{\mu} := \{t_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ t_i^{\mu} \leftarrow I_{\ell-1, \mathsf{out}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ where \ s_i^{\mu} \leftarrow I_{\ell, \mathsf{in}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \forall \mu \in [M], \ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \mathsf{S}^{\mu} \in I_{\ell, \mathsf{In}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]} \ for \ \mathsf{S}^{\mu} \in I_{\ell, \mathsf{In}, \mathsf{LDE}}^{\mu} \\ \mathsf{S}^{\mu} := \{s_i^{\mu}\}_{i \in [\log^2 \lambda]$$

where $Correct(\{S^{\mu}\}_{\mu \in [M]})$ and $Correct(\{T^{\mu}\}_{\mu \in [M]})$ are defined as in Sub-Claim E.1.

Proof. As in the proof of Sub-Claim E.1, it suffices to show that there exists a negligible function ϵ such that for every $\lambda \in \Lambda$, every good (st_C, com), and every $\ell \in \{2, \dots, N_{\text{round}}\}\$, the following holds.

$$\Pr\left[\begin{array}{c} \operatorname{Correct}\left(\{T^{\mu}\}_{\mu\in[M]}\right) \\ \wedge A^{\alpha:\beta}(i^{*}) \neq A_{\operatorname{corr}}^{\alpha:\beta}(i^{*}) \end{array}\right| \left(\begin{array}{c} T^{\mu} \coloneqq \{t_{i}^{\mu}\}_{i\in[\log^{2}\lambda]} \text{ for } \forall \mu \in [M], \text{ where } t_{i}^{\mu} \leftarrow I_{\ell-1,\operatorname{out,LDE}}^{\mu} \\ (f,\{A^{\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \operatorname{P-SelfCorr}_{m|H,D(X)}^{\operatorname{C.Prv}_{i}^{*}(\operatorname{st}_{C},\cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}), \\ \operatorname{where } Q^{\mu:\nu} \coloneqq T^{\mu} \cup T^{\nu} \text{ for } \forall (\mu,\nu) \neq (\alpha,\beta) \text{ and } \\ Q^{\alpha:\beta} \coloneqq \{i^{*}\} \cup T^{\alpha} \cup T^{\beta} \end{array}\right] \leq \epsilon(\lambda) \ .$$

Fix any λ , (st_C, com), and ℓ as above, and for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, $\forall i^* \in \mathcal{I}_{\ell-1 \text{ out}}^{\xi} \cup \mathcal{I}_{\ell \text{ in}}^{\xi}$, let $p^{\alpha;\beta}(i^*)$ be the following probability.

$$p^{\alpha:\beta}(i^*) \coloneqq \Pr \left[\begin{array}{c} \mathsf{Correct}\left(\{T^\mu\}_{\mu \in [M]}\right) \\ \land A^{\alpha:\beta}(i^*) \neq A^{\alpha:\beta}_{\mathrm{corr}}(i^*) \end{array} \right. \\ \left. \begin{array}{c} T^\mu \coloneqq \{t_i^\mu\}_{i \in [\log^2 \lambda]} \text{ for } \forall \mu \in [M], \text{ where } t_i^\mu \leftarrow I_{\ell-1,\mathsf{out},\mathsf{LDE}}^\mu \\ (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H,D(X)}^{\mathsf{C.Prv}_1^*(\mathsf{st}_{\mathcal{C},\cdot})}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq T^\mu \cup T^\nu \text{ for } \forall (\mu,\nu) \neq (\alpha,\beta) \text{ and } \\ Q^{\alpha:\beta} \coloneqq \{i^*\} \cup T^\alpha \cup T^\beta \end{array} \right]$$

In this notation, our goal is to show that we have $p^{\alpha;\beta}(i^*) \leq \epsilon(\lambda)$ for $\forall \alpha,\beta \in [M], \ \forall \xi \in \{\alpha,\beta\}, \ \forall i^* \in I^{\xi}_{\ell,\mathrm{in}}$.

First, we observe that we have $p^{\alpha:\beta}(i^*) \leq \operatorname{negl}(\lambda)$ for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in \mathcal{I}^{\xi}_{\ell-1, \text{out}}$. Indeed, since the prove phase of $\langle C_1, R_1 \rangle$ is just parallel executions of (PCP.P, PCP.V), this bound follows from Claim D.6. Next, we observe that we have $p^{\alpha:\beta}(i^*) \leq \operatorname{negl}(\lambda)$ for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in \mathcal{I}^{\xi}_{\ell, \text{in}}$ such that

- either $i^* = (\xi, \ell, i)$ for some $i \in [n_{st}]$, i.e., i^* is the index of a variable in $\mathbf{w}_{\ell \text{ in}}^{\xi}$, which corresponds to P^{ξ} 's internal state at the beginning of Round ℓ ,
- or $i^* \in \{(\alpha, \ell, n_{st} + \beta), (\beta, \ell, n_{st} + \alpha)\}$, i.e., i^* is the index of the variable $\mathbf{w}_{\ell, \text{in}}^{\alpha}(n_{st} + \beta)$ or $\mathbf{w}_{\ell, \text{in}}^{\beta}(n_{st} + \alpha)$, where the former corresponds to P^{α} 's incoming message from P^{β} in Round ℓ and the latter corresponds to P^{β} 's incoming message from P^{α} in Round ℓ .

Fix any α, β, ξ as above.

Case 1. When $i^* = (\xi, \ell, i)$ for some $i \in [n_{st}]$, let $j^* \in I^{\xi}_{\ell-1, \text{out}}$ be the index of the variable $w^{\xi}_{\ell-1, \text{out}}(i)$, i.e., the variable that corresponds to the i-th bit of P^{ξ} 's internal state at the end of Round $\ell-1$. Since P-SelfCorr $^{\text{C.Prv}^*_{\ell}(\text{st}_{\ell}, \cdot)}_{m|H|,D(X)}$ is an adaptive local assignment generator (Claim D.4), we have

$$\Pr\begin{bmatrix} A^{\alpha:\beta}(i^*) \neq A^{\alpha:\beta}(j^*) & (f,\{A^{\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H|,D(X)}^{\mathsf{C.Prv}_1^*(\mathsf{st}_C,\cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq \emptyset \text{ for } \forall (\mu,\nu) \neq (\alpha,\beta) \text{ and } Q^{\alpha:\beta} \coloneqq \{i^*,j^*\} \end{bmatrix} \leq \mathsf{negl}(\lambda) \ . \tag{E.18}$$

Hence, we have

$$\begin{split} &p^{\alpha:\beta}(i^*) \\ &\leq \Pr \begin{bmatrix} &\operatorname{Correct}\left(\{T^\mu\}_{\mu \in [M]}\right) \\ &\wedge A^{\alpha:\beta}(i^*) \neq A_{\operatorname{corr}}^{\alpha:\beta}(i^*) \\ &\wedge A^{\alpha:\beta}(j^*) = A_{\operatorname{corr}}^{\alpha:\beta}(j^*) \end{bmatrix} & T^\mu \coloneqq \{t_i^\mu\}_{i \in [\log^2 \lambda]} \text{ for } \forall \mu \in [M], \text{ where } t_i^\mu \leftarrow \mathcal{I}_{\ell-1, \operatorname{out, LDE}}^\mu \\ &(f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \operatorname{P-SelfCorr}_{m|H|,D(X)}^{\operatorname{C.Prv}_i^*(\operatorname{stc}, \cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ &\text{where } Q^{\mu:\nu} \coloneqq T^\mu \cup T^\nu \text{ for } \forall (\mu,\nu) \neq (\alpha,\beta) \text{ and } \\ &Q^{\alpha:\beta} \coloneqq \{i^*,j^*\} \cup T^\alpha \cup T^\beta \end{bmatrix} \\ &\leq p^{\alpha:\beta}(j^*) + \operatorname{negl}(\lambda) \ , \end{split}$$

where in the last inequality, we use the fact that the event

$$\left(A^{\alpha:\beta}(i^*) \neq A_{\text{corr}}^{\alpha:\beta}(i^*)\right) \wedge \left(A^{\alpha:\beta}(j^*) = A_{\text{corr}}^{\alpha:\beta}(j^*)\right)$$

implies $A^{\alpha:\beta}(i^*) \neq A^{\alpha:\beta}(j^*)$ since we have $A^{\alpha:\beta}_{\mathrm{corr}}(i^*) = A^{\alpha:\beta}_{\mathrm{corr}}(j^*)$ from the definition of A_{corr} . Now, since we have $p^{\alpha:\beta}(j^*) \leq \mathsf{negl}(\lambda)$ from what is shown in the previous paragraph, we obtain $p^{\alpha:\beta}(i^*) \leq \mathsf{negl}(\lambda)$ as desired.

Case 2. When $i^* \in \{(\alpha, \ell, n_{\text{st}} + \beta), (\beta, \ell, n_{\text{st}} + \alpha)\}$, let us focus on the case of $i^* = (\alpha, \ell, n_{\text{st}} + \beta)$ for concreteness (the other case can be handled identically). Let $j^* \in \mathcal{I}^{\xi}_{\ell-1,\text{out}}$ be the index of the variable $w^{\beta}_{\ell-1,\text{out}}(n_{\text{st}} + \alpha)$, i.e., the variable that corresponds to P^{β} 's outgoing message to P^{α} in Round $\ell-1$. Since P-SelfCorr $_{m|H,D(X)}^{\text{C.Prv}_1(\text{stc},\cdot)}$ is an adaptive local assignment generator, we have Equation (E.18). Hence, as in Case 1, we obtain $p^{\alpha:\beta}(i^*) \leq \text{negl}(\lambda)$ as desired.

Finally, we observe that we have $p^{\alpha:\beta}(i^*) \leq \operatorname{negl}(\lambda)$ for $\forall \alpha, \beta \in [M]$, $\forall \xi \in \{\alpha, \beta\}$, and $\forall i^* \in I^{\xi}_{\ell, \text{in}}$. Fix any α, β, ξ as above, and let us focus for concreteness on the case of $\xi = \alpha$ (the case of $\xi = \beta$ can be handled identically). Given what is shown in the previous paragraph, it remains to consider the case that $i^* = (\alpha, \ell, n_{\text{st}} + \gamma) \in I^{\alpha}_{\ell, \text{in}}$ for some $\gamma \in [M] \setminus \{\beta\}$, i.e., i^* is the index of the variable $w^{\alpha}_{\ell, \text{in}}(n_{\text{st}} + \gamma)$, which corresponds to P^{α} 's incoming message from P^{γ} in Round ℓ . Now, for $\forall \gamma \in [M] \setminus \{\beta\}$, what is shown in the previous paragraph implies that $p^{\alpha:\gamma}(i^*) \leq \operatorname{negl}(\lambda)$. Also, Claim E.5 and Lemma K.3 imply that

$$\Pr\begin{bmatrix} A^{\alpha:\beta}(i^*) \neq A^{\alpha:\gamma}(i^*) & (f,\{A^{\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H|,D(X)}^{\mathsf{C.Prv}_1'(\mathsf{st}_{\mathbb{C}},\cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}), \\ & \text{where } Q^{\mu:\nu} \coloneqq \emptyset \text{ for } \forall (\mu,\nu) \notin \{(\alpha,\beta),(\alpha,\gamma)\}, \text{ and } \\ & Q^{\mu:\nu} \coloneqq \{t^*\} \text{ for } \forall (\mu,\nu) \in \{(\alpha,\beta),(\alpha,\gamma)\} \end{bmatrix} \leq \mathsf{negl}(\lambda) \enspace .$$

These two imply that

$$\begin{split} p^{\alpha:\beta}(i^*) \\ &\leq \Pr \begin{bmatrix} \operatorname{Correct}\left(\{T^{\mu}\}_{\mu \in [M]}\right) & T^{\mu} \coloneqq \{t_i^{\mu}\}_{i \in [\log^2 \lambda]} \text{ for } \forall \mu \in [M], \text{ where } t_i^{\mu} \leftarrow T_{\ell-1, \text{out,LDE}}^{\mu} \\ & \wedge A^{\alpha:\beta}(i^*) \neq A_{\operatorname{corr}}^{\alpha:\beta}(i^*) \\ & \wedge A^{\alpha:\gamma}(i^*) = A_{\operatorname{corr}}^{\alpha:\gamma}(i^*) \end{bmatrix} & T^{\mu} \coloneqq \{t_i^{\mu}\}_{i \in [\log^2 \lambda]} \text{ for } \forall \mu \in [M], \text{ where } t_i^{\mu} \leftarrow T_{\ell-1, \text{out,LDE}}^{\mu} \\ & (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \operatorname{P-SelfCorr}_{m|H,D(X)}^{\operatorname{C.Prv}_i^*(\operatorname{st}_C,\cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ & \text{where } Q^{\mu:\nu} \coloneqq T^{\mu} \cup T^{\nu} \text{ for } \forall (\mu,\nu) \notin \{(\alpha,\beta),(\alpha,\gamma)\} \end{aligned} \\ & + p^{\alpha:\gamma}(i^*) + \operatorname{negl}(\lambda) \\ & \leq \operatorname{negl}(\lambda) \ . \end{split}$$

where in the last inequality, we use the fact that the event

$$\left(A^{\alpha:\beta}(i^*) \neq A_{\text{corr}}^{\alpha:\beta}(i^*)\right) \wedge \left(A^{\alpha:\gamma}(i^*) = A_{\text{corr}}^{\alpha:\gamma}(i^*)\right)$$

implies $A^{\alpha:\beta}(i^*) \neq A^{\alpha:\gamma}(i^*)$ since we have $A_{\text{corr}}^{\alpha:\beta}(i^*) = A_{\text{corr}}^{\alpha:\gamma}(i^*)$ from the definition of A_{corr} . This completes the proof of Sub-Claim E.3.

Now, we are ready to prove Claim E.8. Fix any $\lambda \in \Lambda$ and good (st_C, com). For $\forall \ell \in [N_{\text{round}}]$, let $p_{\text{in}}(\ell)$ be the following probability.

$$p_{\mathsf{in}}(\ell) \coloneqq \Pr \begin{bmatrix} \mathsf{Correct} \left(\{S^{\mu}\}_{\mu \in [M]} \right) & S^{\mu} \coloneqq \{s_{i}^{\mu}\}_{i \in [\log^{2} \lambda]} \text{ for } \forall \mu \in [M], \text{ where } s_{i}^{\mu} \leftarrow \mathcal{I}_{\ell,\mathsf{in},\mathsf{LDE}}^{\mu} \\ (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H,D(X)}^{\mathsf{C.Prv}_{1}^{*}(\mathsf{st}_{C},\cdot)} (\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq S^{\mu} \cup S^{\nu} \text{ for } \forall \mu,\nu \in [M] \end{bmatrix}$$

Similarly, for $\forall \ell \in [N_{\text{round}}]$, let $p_{\text{out}}(\ell)$ be the following probability.

$$p_{\mathsf{out}}(\ell) \coloneqq \Pr \begin{bmatrix} \mathsf{Correct}\left(\{T^{\mu}\}_{\mu \in [M]}\right) & T^{\mu} \coloneqq \{t_{i}^{\mu}\}_{i \in [\log^{2}\lambda]} \text{ for } \forall \mu \in [M], \text{ where } t_{i}^{\mu} \leftarrow \mathcal{I}_{\ell,\mathsf{out},\mathsf{LDE}}^{\mu} \\ (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H,D(X)}^{\mathsf{C.Prv}_{1}^{\ast}(\mathsf{st}_{C},\cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq T^{\mu} \cup T^{\nu} \text{ for } \forall \mu,\nu \in [M] \end{bmatrix}$$

From Sub-Claim E.2 and Sub-Claim E.3, we have

$$p_{\text{out}}(\ell) \ge p_{\text{in}}(\ell) - N_{\text{Aug}} \epsilon_2(\lambda) - \text{negl}(\lambda)$$

and

$$p_{\text{in}}(\ell) \ge p_{\text{out}}(\ell-1) - \epsilon_3(\lambda) - \text{negl}(\lambda)$$
.

Hence, we have

$$p_{\text{out}}(N_{\text{round}}) \ge p_{\text{in}}(1) - \text{negl}(\lambda).$$

Since we have $p_{in}(1) \ge 1 - \text{negl}(\lambda)$ from Sub-Claim E.1, we have

$$p_{\text{out}}(N_{\text{round}}) \ge 1 - \text{negl}(\lambda)$$
.

Combining this with Claim D.6, for $\forall \alpha, \beta \in [M], \xi \in \{\alpha, \beta\}, i^* \in I_{N_{\text{count}}, \text{Out}}^{\xi}$, we have

$$\Pr \left[A^{\alpha:\beta}(i^*) = A_{\operatorname{corr}}^{\alpha:\beta}(i^*) \, \middle| \, \begin{array}{c} (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \operatorname{P-SelfCorr}_{m|H|,D(X)}^{\operatorname{C.Prv}_1^*(\operatorname{st}_C, \cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \operatorname{where} \ Q^{\mu:\nu} \coloneqq \emptyset \ \text{for} \ (\mu,\nu) \neq (\alpha,\beta) \ \text{and} \ Q^{\alpha:\beta} \coloneqq \{i^*\} \end{array} \right] \geq 1 - \operatorname{negl}(\lambda) \ .$$

Fix any $\alpha, \beta \in [M], \xi \in \{\alpha, \beta\}$, and let $i^* \in \mathcal{I}_{N_{\text{round}}, \text{out}}^{\xi}$ be the index of the variable $\mathbf{w}_{N_{\text{round}}, \text{out}}^{\xi}(1)$, which corresponds to the first bit of P^{ξ} 's internal state at the end of Round N_{round} (or equivalently the output of P^{ξ}). Since P-SelfCorr $_{m|H|,D(X)}^{\text{C.Prv}_1(\text{stc},\cdot)}$ is an adaptive local assignment generator, we have

$$\Pr \begin{bmatrix} A^{\alpha:\beta}(i^*) = 1 & (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H|,D(X)}^{\mathsf{C.Prv}_1^*(\mathsf{st}_C, \cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq \emptyset \text{ for } (\mu,\nu) \neq (\alpha,\beta) \text{ and } Q^{\alpha:\beta} \coloneqq \{i^*\} \end{bmatrix} \geq 1 - \mathsf{negl}(\lambda) \enspace .$$

Hence, from a union bound, we have

$$\Pr \begin{bmatrix} A_{\mathrm{corr}}^{\alpha:\beta}(i^*) = 1 & (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{P-SelfCorr}_{m|H|,D(X)}^{\mathsf{C.Prv}_1^*(\mathsf{st}_C,\cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ & \text{where } Q^{\mu:\nu} \coloneqq \emptyset \text{ for } (\mu,\nu) \neq (\alpha,\beta) \text{ and } Q^{\alpha:\beta} \coloneqq \{i^*\} \end{bmatrix} \geq 1 - \mathsf{negl}(\lambda) \enspace .$$

From the definition of $A_{\text{corr}}^{\alpha:\beta}$ and the correctness of Π , we have $A_{\text{corr}}^{\alpha:\beta}(i^*) = f(x_{\text{COM}}^*)$. Hence, we have

$$\Pr \left[f(\boldsymbol{x}^*_{\text{COM}}) = 1 \;\middle|\; \begin{array}{c} (f, \{A^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \text{P-SelfCorr}_{m|H|,D(X)}^{\text{C.Prv}_1^*(\text{st}_C, \cdot)}(\{Q^{\mu:\nu}\}_{\mu,\nu \in [M]}), \\ \text{where } Q^{\mu:\nu} \coloneqq \emptyset \text{ for } (\mu,\nu) \neq (\alpha,\beta) \text{ and } Q^{\alpha:\beta} \coloneqq \{i^*\} \end{array} \right] \geq 1 - \text{negl}(\lambda) \enspace .$$

Hence, from the "moreover" part of Claim D.4, we have

$$\Pr \left[f(x^*_{\text{\tiny COM}}) = 1 \; \middle| \; \begin{array}{l} (Q, \operatorname{St}_R) \leftarrow \operatorname{R.Prv.Q}_1 \\ (f, \pi^*) \leftarrow \operatorname{C.Prv}^*(\operatorname{st}_C, Q) \end{array} \right] \geq 1 - \operatorname{negl}(\lambda)$$

as desired. This completes the proof of Claim E.8.

This completes the proof of Lemma E.3.

Remark 6. By inspecting the proof of Claim E.6, one can easily see that Equation (E.10) holds even for a stronger version of R.Dec₁ that uses LD-Test.D_{m_{10}}|H|, ζ in the verification instead of LD-Test.D_{m_{10}}|H|, 3ζ . (This observation is used later in the proof of Lemma G.2).

F Step 2: Non-WI Scheme with (1 - negl)-Soundness against CNS Provers

As the second step to our commit-and-prove protocol, we give a non-WI commit-and-prove protocol $\langle C_2, R_2 \rangle$ that is (1 - negl)-sound against (not necessarily well-behaving) CNS provers.

In this step, we use a collision-resistant hash function family \mathcal{H} . For any $hf \in \mathcal{H}$, we denote by TreeHash_{hf} an algorithm that computes the Merkle tree-hash of the input.

Algorithm 10 Commit Phase and Open Phase of $\langle C_2, R_2 \rangle$

Commit Phase

Round 1: R.Com₂ sends a hash function $hf \in \mathcal{H}$ to C.Com₂.

Round 2: Given $(x_{\text{COM}}, \text{hf})$ as input, C.Com₂ runs $\{X_{1,\text{in}}^{\mu}\}_{\mu \in [M]} \leftarrow \text{C.Com}_1(x_{\text{COM}})$, and then outputs $\{\text{rt}_{1,\text{in}}^{\mu}\}_{\mu \in [M]}$ is the internal state.

Open Phase

Round 1: R.Dec.Q₂ works identically with R.Dec.Q₁. That is, given i as input, R.Dec.Q₂ runs $(\{Q^{\mu}\}_{\mu \in [M]}, \operatorname{st}_R) \leftarrow R.Dec.Q_1(i)$, and then outputs $\{Q^{\mu}\}_{\mu \in [M]}$ as the query and st_R as the internal state.

Round 2: Given $(\operatorname{st}_C, \{Q^{\mu}\}_{\mu \in [M]})$ as input (where $\operatorname{st}_C = (\operatorname{hf}, \{X^{\mu}_{1,\operatorname{in}}\}_{\mu \in [M]}))$, C.Dec₂ defines $\{\tilde{X}^{\mu:\mu}\}_{\mu \in [M]}$ just like C.Dec₁ does, and outputs $\{\tilde{Y}^{\mu:\mu}|_{O^{\mu}}\}_{\mu \in [M]}$ as the decommitment, where each $\tilde{Y}^{\mu:\mu}$ is defined as

$$\tilde{Y}^{\mu:\mu}(z) := \begin{cases} (\tilde{X}^{\mu:\mu}(z), \mathsf{cert}_{1,\mathsf{in}}^{\mu}(z)) & \text{if } z \in D(X_{1,\mathsf{in}}^{\mu}) \\ (\tilde{X}^{\mu:\mu}(z), \bot) & \text{otherwise} \end{cases},$$

where for each $z=(z_{\xi},z_{\ell},z_{\rm in},i)\in D(X_{\ell_{\rm in}}^{\xi})$ (cf. Section D.2.2), ${\rm cert}_{1,\rm in}^{\mu}(z)$ is the certificate of Merkle tree-hash for revealing the i-th bit of $X_{1,\rm in}^{\mu}$.

Verification: Given $(\operatorname{st}_R, \operatorname{com}, \{\tilde{Y}^{*\mu,\mu}\}_{\mu \in [M]})$ as input, R.Dec.D₂ outputs

$$\tilde{x}_i \coloneqq \mathsf{R.Dec.D}_1\left(\mathsf{st}_R,\mathsf{com}',\left\{\mathsf{Filter}^{\mu}(\tilde{Y^*}^{\mu:\mu})\right\}_{\mu\in[M]}\right)$$

as the decommitted value, where $com' = \varepsilon$ is an empty string, and Filter^{μ} is the following function: given input of the form $\{(x_z, cert_z)\}_{z \in Q}$, it outputs $\{\hat{x}_z\}_{z \in Q}$ such that $\hat{x}_z := x_z$ if $cert_z$ is a valid certificate w.r.t. $(rt^{\mu}_{1,in}, z, x_z)$ and $\hat{x}_z := \bot$ otherwise.

F.1 Protocol Description

The formal description of $\langle C_2, R_2 \rangle$ is given in Algorithm 10 and Algorithm 11.

F.2 Proof of Binding

Lemma F.1. Let ϵ be any negligible function, κ_{dec} be the polynomial that is given in Lemma E.1, and $C_2^* = (C.Com_2^*, C.Dec_2^*)$ be any κ_{dec} -CNS cheating committer-decommitter against $\langle C_2, R_2 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following binding condition holds with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com_2^*, R.Com_2 \rangle$.

• Binding Condition: If it holds

$$\Pr\left[\begin{array}{c} \forall b \in \{0,1\}, \mu \in [M], \\ \beta^{\mu}_b = 1 \end{array} \right. \left. \begin{array}{l} (\mathcal{Q}^{\mu}_b, \mathsf{st}^{\mu}_b) \leftarrow \mathsf{LD-Test.Q}_{D(X^{\mu}_{1,\mathrm{in}})}, for \ \forall b \in \{0,1\}, \mu \in [M] \\ \{\tilde{Y}^{*,\mu;\mu}_b\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathsf{C.Dec}^*_2(\mathsf{st}_C, \{\mathcal{Q}^{\mu}_b\}_{b \in \{0,1\}, \mu \in [M]}) \\ \beta^{\mu}_b \coloneqq \mathsf{LD-Test.D}_{m_{to}|H|,3\zeta}(\mathsf{st}^{\mu}_b, \mathsf{Filter}^{\mu}(\tilde{Y}^{*,\mu;\mu}_b)) \\ for \ \forall b \in \{0,1\}, \mu \in [M] \end{array} \right] \geq 1 - \epsilon(\lambda) \enspace ,$$

then for every $i \in [n]$ it holds $\Pr[b_{BAD} = 1] \leq \operatorname{negl}(\lambda)$ in the following probabilistic experiment $Exp_2^{\operatorname{bind}}(\mathsf{C}.\mathsf{Dec}_2^*,\mathsf{st}_C,\mathsf{com},i)$.

- 1. For each $\forall b \in \{0, 1\}$, sample $\{Q_b^{\mu}\}_{\mu \in [M]}$ by $\{Q_b^{\mu}\}_{\mu \in [M]}$, $\mathsf{st}_b\} \leftarrow \mathsf{R.Dec.Q}_2(i)$.
- 2. $Run\{\tilde{Y}_{h}^{*\mu:\mu}\}_{b\in\{0,1\},\mu\in[M]} \leftarrow C.Dec_{2}^{*}(st_{C},\{Q_{h}^{\mu}\}_{b\in\{0,1\},\mu\in[M]}).$
- 3. Let $b_{\text{BAD}} \coloneqq 1$ if and only if $x_0^* \neq \bot \land x_1^* \neq \bot \land x_0^* \neq x_1^*$ holds, where $x_b^* \coloneqq \mathsf{R.Dec.D}_2(\mathsf{st}_b, \mathsf{com}, \{\tilde{Y}_b^{*,\mu,\mu}\}_{\mu \in [M]})$ for each $b \in \{0, 1\}$.

Proof. Fix any ϵ and $C_2^* = (C.Com_2^*, C.Dec_2^*)$ as above, and assume for contradiction that for infinitely many $\lambda \in \mathbb{N}$, the binding condition does not hold with non-negligible probability over the choice of (st_C, com) ← (C.Com₂*, R.Com₂).

To obtain a contradiction with the binding property of $\langle C_1, R_1 \rangle$ (Lemma E.1), we consider the following cheating committer-decommitter $C_1^* = (C.Com_1^*, C.Dec_1^*)$ against $\langle C_1, R_1 \rangle$.

- Committer. C.Com₁* runs (st_C, com) ← ⟨C.Com₂*, R.Com₂⟩ internally, sends an empty string to R.Com₁ as the commitment, and stores (com, st_C) as the internal state.
- **Decommitter.** Given (com, st_C) and $\{Q^{\mu}\}_{\mu \in [M]}$ as input, C.Dec₁* runs $\{\tilde{Y}^{*\mu:\mu}\}_{\mu \in [M]} \leftarrow \text{C.Dec}_2^*(\text{st}_C, \{Q^{\mu}\}_{\mu \in [M]})$ internally, and sends $\{\text{Filter}^{\mu}(\tilde{Y}^{*\mu:\mu})\}_{\mu \in [M]}$ to R.Dec₁ as the decommitment, where the function $\text{Filter}^{\mu}(\tilde{Y}^{*\mu:\mu})$ is defined as in $\langle C_2, R_2 \rangle$.

Algorithm 11 Prove Phase of $\langle C_2, R_2 \rangle$

Prove Phase

Round 1: R.Prv.Q₂ works identically with R.Prv.Q₁. That is, R.Prv.Q₂ runs $(\{Q^{\mu,\nu}\}_{\mu,\nu\in[M]}, \{st_V^{\mu,\nu}\}_{\mu,\nu\in[M]}) \leftarrow \text{R.Prv.Q}_1$, and outputs $\{Q^{\mu,\nu}\}_{\mu,\nu\in[M]}$ as the query and $\{st_V^{\mu,\nu}\}_{\mu,\nu\in[M]}$ as the internal state.

Round 2: Given $(\operatorname{st}_C, f, \{Q^{\mu,\nu}\}_{\mu,\nu\in[M]})$ as input, C.Prv₂ does the following.

- 1. Obtain $\{\text{view}^{\mu}\}_{\mu \in [M]}$ just like C.Prv₁ does.
- 2. Run $(\mathsf{rt}^{\mu}, \mathsf{rt}^{\nu}, \pi^{\mu:\nu}) \leftarrow \mathsf{PCP.P'}(\mu, \nu, f, \mathsf{view}^{\mu}, \mathsf{view}^{\nu}, \mathsf{hf})$ for every $\mu, \nu \in [M]$, where $\mathsf{PCP.P'}$ is identical with $\mathsf{PCP.P'}$ except for the following.
 - · A hash function hf is given as an additional input.
 - The hash of X^{μ} and X^{ν} , denoted by $\mathsf{rt}^{\mu} \coloneqq \mathsf{TreeHash}_{\mathsf{hf}}(X^{\mu})$ and $\mathsf{rt}^{\nu} \coloneqq \mathsf{TreeHash}_{\mathsf{hf}}(X^{\nu})$, are computed as additional outputs.
 - The proof $\pi^{\mu:\nu}$ given by PCP.P' is $\pi^{\mu:\nu} = (Y^{\mu:\nu}, \ldots)$ instead of $\pi^{\mu:\nu} = (X^{\mu:\nu}, \ldots)$, where $Y^{\mu:\nu}$ is defined as

$$Y^{\mu:\nu}(z) \coloneqq \begin{cases} (X^{\mu:\nu}(z), \mathsf{cert}^{\xi}_{1,\mathsf{in}}(z)) & \text{if } \exists \xi \in \{\mu,\nu\} \text{ s.t. } z \in D(X^{\xi}_{1,\mathsf{in}}) \\ (X^{\mu:\nu}(z), \mathsf{cert}^{\xi}(z)) & \text{if } \exists \xi \in \{\mu,\nu\} \text{ s.t. } z \in D(X^{\xi}) \setminus D(X^{\xi}_{1,\mathsf{in}}) \\ (X^{\mu:\nu}(z), \bot) & \text{otherwise} \end{cases},$$

where $\operatorname{cert}_{1,\operatorname{in}}^{\xi}(z)$ is defined as in the open phase, and $\operatorname{cert}^{\xi}(z)$ is defined as follows: for each $z=(z_{\xi},i)\in D(X^{\xi})$ (cf. Section D.2.2), $\operatorname{cert}^{\xi}(z)$ is the certificate of Merkle tree-hash for revealing the *i*-th bit of X^{ξ} .

3. Output $(\{rt^{\mu}\}_{\mu\in[M]}, \{\pi^{\mu:\nu}|_{Q^{\mu:\nu}}\}_{\mu,\nu\in[M]})$ as the proof. (Note: the value of rt^{μ} that is computed by PCP.P' (μ,ν,\ldots) and the value of it that is computed by PCP.P' (μ,ξ,\ldots) are identical; see Remark 3.)

Verification: Given $(\operatorname{st}_R, \operatorname{com}, f, \{\operatorname{rt}^\mu\}_{\mu \in [M]}, \{\pi^{*\mu:\nu}\}_{\mu,\nu \in [M]})$ as input (where $\operatorname{st}_R = \{\operatorname{st}_V^{\mu,\nu}\}_{\mu,\nu \in [M]}$ and $\operatorname{com} = \{\operatorname{rt}_{1,\operatorname{in}}^\mu\}_{\mu \in [M]}$), R.Prv.D₂ does the following.

- 1. Run $b^{\mu:\nu} \leftarrow \mathsf{PCP.D}'^{\geq \lambda \zeta}(\mathsf{st}_V^{\mu:\nu}, f', \mathsf{rt}_{1,\mathsf{in}}^{\mu}, \mathsf{rt}_{1,\mathsf{in}}^{\nu}, \mathsf{rt}^{\nu}, \pi^{*\mu:\nu})$ for every $\mu, \nu \in [M]$, where $\mathsf{PCP.D}'$ is identical with $\mathsf{PCP.D}$ except for the following.
 - The TreeHash_hf roots $\mathsf{rt}_{1,\mathsf{in}}^\mu,\mathsf{rt}_{1,\mathsf{in}}^\nu,\mathsf{rt}^\mu,\mathsf{rt}^\nu$ are given as additional inputs.
 - Each test for $X^{\mu:\nu}$ is made in the following manner.
 - (a) Make the query to $Y^{\mu:\nu}$ instead of to $X^{\mu:\nu}$. Let $\{(x_z, \mathsf{cert}_z)\}_{z\in Q}$ be the response from $Y^{\mu:\nu}$.
 - (b) Verify the test by considering $\mathsf{Filter}^{\mu:\nu}(\{(x_z,\mathsf{cert}_z)\}_{z\in Q})$ to be the response from $X^{\mu:\nu}$, where $\mathsf{Filter}^{\mu:\nu}$ is the following function: given input $\{(x_z,\mathsf{cert}_z)\}_{z\in Q}$, it outputs $\{\hat{x}_z^{\mu:\nu}\}_{z\in Q}$ such that

if
$$\exists \xi \in \{\mu, \nu\}$$
 s.t. $z \in D(X_{1, \text{in}}^{\xi})$ and cert_z is not a valid certificate w.r.t. $(\text{rt}_{1, \text{in}}^{\xi}, z, x_z)$, or $\exists \xi \in \{\mu, \nu\}$ s.t. $z \in D(X^{\xi}) \setminus D(X_{1, \text{in}}^{\xi})$ and cert_z is not a valid certificate certificate w.r.t. $(\text{rt}^{\xi}, z, x_z)$ otherwise

2. Output 1 if and only if $b^{\mu:\nu} = 1$ for every $\mu, \nu \in [M]$.

In what follows, we observe that (1) the binding condition of $\langle C_1, R_1 \rangle$ does not hold with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_1^*, \mathsf{R.Com}_1 \rangle$, and (2) C_1^* is well-behaving and CNS.

First, the binding condition of $\langle C_1, R_1 \rangle$ does not hold with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C}.\mathsf{Com}_1^*, \mathsf{R}.\mathsf{Com}_1 \rangle$ since C_1^* perfectly emulates (R.Com₂, R.Dec₂) for the internally emulated C_2^* .

Second, C_1^* is a well-behaving CNS committer-decommitter since (1) the CNS property of C.Dec₁* follows from that of C.Dec₂*, and (2) the consistency on $D(X_{1,in}^{\mu})$ follows from the binding property of TreeHash_{hf}.

Hence, we obtain a contradiction.

F.3 Proof of Soundness

Lemma F.2. Let ϵ_{SND} be any negligible function, κ_{prv} be the polynomial that is given in Lemma E.3, E_2 be the extractor in Algorithm 12, and $C_2^* = (\text{C.Com}_2^*, \text{C.Prv}_2^*)$ be any κ_{prv} -CNS cheating committer-prover against $\langle C_2, R_2 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following soundness condition holds with overwhelming probability over the choice of $(\text{st}_C, \text{com}) \leftarrow \langle \text{C.Com}_2^*, \text{R.Com}_2 \rangle$.

• Soundness Condition: If it holds

$$\Pr \left[b = 1 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_2; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_2^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D}_2(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \ge 1 - \epsilon_{\scriptscriptstyle SND}(\lambda) \ , \tag{F.1}$$

then there exists $x_{COM}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that

$$\forall i \in [n], \Pr\left[x_i = x_i^* \mid (\bot, x_i) \leftarrow \langle E_2^{\mathsf{C.Prv}_2^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_2(\mathsf{com}, i) \rangle \right] \geq 1 - \mathsf{negl}(\lambda) \tag{F.2}$$

and

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x_{\scriptscriptstyle COM}^*)=0\end{array}\right] \left(\begin{array}{c}(Q,\operatorname{st}_R)\leftarrow\operatorname{R.Prv.Q_2};\ (f,\pi^*)\leftarrow\operatorname{C.Prv}_2^*(\operatorname{st}_C,Q)\\ b\leftarrow\operatorname{R.Prv.D_2}(\operatorname{st}_R,\operatorname{com},f,\pi^*)\end{array}\right] \leq \operatorname{negl}(\lambda)\ . \tag{F.3}$$

Algorithm 12 Extractor E_2 (against $\langle C_2, R_2 \rangle$)

Input: com, i, and $\{Q^{\mu}\}_{\mu \in [M]} \subset D(X_{1,\mathsf{in}}^1) \times \cdots \times D(X_{1,\mathsf{in}}^M)$

- 1. Run $(f, \{\mathsf{rt}^{\mu}\}_{\mu \in [M]}, \{\pi^{*\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{C.Prv}_2^*(\mathsf{st}_C, \{Q^{\mu:\nu}\}_{\mu,\nu \in [M]})$, where each $Q^{\mu:\nu}$ is defined as $Q^{\mu:\nu} := Q^{\mu}$ if $\mu = \nu$ and $Q^{\mu:\nu} := \emptyset$ otherwise.
- 2. Output $\{\tilde{Y}^{*\mu:\mu}\}_{\mu\in[M]}$ as the decommitment, where $\tilde{Y}^{*\mu:\mu}:=\pi^{*\mu:\mu}$.

Proof. Fix any ϵ_{SND} and $C_2^* = (\text{C.Com}_2^*, \text{C.Prv}_2^*)$ as in the lemma statement, and assume for contradiction that for infinitely many $\lambda \in \mathbb{N}$, the soundness condition does not hold with non-negligible probability over the choice of (st_C, com) ← (C.Com₂*, R.Com₂).

To obtain a contradiction with the soundness of $\langle C_1, R_1 \rangle$ (Lemma E.3), we consider the following cheating committer-prover $C_1^* = (\mathsf{C.Com}_1^*, \mathsf{C.Prv}_1^*)$ against $\langle C_1, R_1 \rangle$.

- Committer. C.Com₁* runs (st_C , com) $\leftarrow \langle \mathsf{C.Com}_2^*, \mathsf{R.Com}_2 \rangle$ internally, sends an empty string to $\mathsf{R.Com}_1$ as the commitment, and stores (com , st_C) as the internal state.
- **Prover.** Given $(\mathsf{com}, \mathsf{st}_C)$ and $\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}$ as input, $\mathsf{C.Prv}_1^*$ first runs $(f, \{\mathsf{rt}^\mu\}_{\mu\in[M]}, \{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]})$ \leftarrow $\mathsf{C.Prv}_2^*(\mathsf{st}_C, \{Q^{\mu:\nu}\}_{\mu,\nu\in[M]})$. For each $\mu, \nu \in [M]$, let $\hat{\pi^*}^{\mu:\nu}: Q^{\mu:\nu} \to F \cup \{\bot\}$ be defined by

$$\forall z \in Q^{\mu : \nu} : \hat{\pi^{*}}^{\mu : \nu}(z) \coloneqq \begin{cases} \hat{x}_z^{\mu : \nu} & \text{if } z \in Q^{\mu : \nu} \cap D(X) \\ \pi^{*\mu : \nu}(z) & \text{otherwise} \end{cases},$$

where $\{\hat{\chi}_{z}^{\mu,\nu}\}_{z\in Q^{\mu\nu}\cap D(X)}:= \mathsf{Filter}^{\mu;\nu}(\pi^{*\mu;\nu}|_{Q^{\mu\nu}\cap D(X)})$ is defined as in $\langle C_2,R_2\rangle$. Then, $\mathsf{C.Prv}_1^*$ sends $(f,\{\hat{\pi^*}^{\mu,\nu}\}_{\mu,\nu\in[M]})$ to $\mathsf{R.Prv}_1$ as the proof.

In what follows, we observe that (1) the soundness condition of $\langle C_1, R_1 \rangle$ does not hold with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_1^*, \mathsf{R.Com}_1 \rangle$, and (2) C_1^* is well-behaving CNS.

First, the soundness condition of $\langle C_1, R_1 \rangle$ does not hold with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C}.\mathsf{Com}_1^*, \mathsf{R}.\mathsf{Com}_1 \rangle$ since C_1^* perfectly emulates (R.Com₂, R.Dec₂, R.Prv₂) and E_2 for the internally emulated C_2^* .

Second, C_1^* is a well-behaving committer-prover since (1) the CNS property of C.Prv₁* follows from that of C.Prv₂*, and (2) the consistencies on $D(X_{1 \text{ in}}^{\mu})$ and on $D(X_{1 \text{ in}}^{\mu})$ follow from the binding property of TreeHash_{hf}.

Hence, we obtain a contradiction.

Remark 7. Given the observation in Remark 6, one can easily see that Equation (F.2) holds even for a stronger version of R.Dec₂ that uses LD-Test.D_{$m_{10}|H|,\zeta$} in the verification instead of LD-Test.D_{$m_{10}|H|,3\zeta$}. (This observation is used later in the proof of Lemma G.2).

G Step 3: Non-WI Scheme with negl-Soundness against CNS Provers

As the third step to our commit-and-prove protocol, we give a non-WI commit-and-prove protocol $\langle C_3, R_3 \rangle$ that is negl-sound against CNS provers.

In this step, we use a slightly extended version of the soundness amplification lemma of Brakerski et al. [BHK17], which is given as Lemma K.4 in Section K.2.

G.1 Protocol Description

The formal description of $\langle C_3, R_3 \rangle$ is given in Algorithm 13.

Algorithm 13 Commit Phase, Open Phase, and Prove Phase of $\langle C_3, R_3 \rangle$

Commit Phase

The commit phase of $\langle C_3, R_3 \rangle$ is identical with that of $\langle C_2, R_2 \rangle$.

.....

Open Phase

The open phase of $\langle C_3, R_3 \rangle$ is identical with that of $\langle C_2, R_2 \rangle$ except that in the verification, R.Dec.D₃ uses LD-Test.D_{$m_{10}|H|,\zeta$} instead of LD-Test.D_{$m_{10}|H|,3\zeta$} for testing each $\tilde{X}^{*\mu;\mu}$.

.....

Prove Phase

The prove phase of $\langle C_3, R_3 \rangle$ is identical with that of $\langle C_2, R_2 \rangle$ except that in the verification, R.Prv.D₃ uses PCP.D'^{$\otimes \lambda$} instead of PCP.D'^{$\geq \lambda - \zeta$} for verifying each proof $\pi^{\mu:\nu}$.

G.2 Proof of Binding

Lemma G.1. Let κ_v be the query complexity of (PCP.P, PCP.V), κ_{dec} be the polynomial that is given in Lemma E.1, and $C_3^* = (\text{C.Com}_3^*, \text{C.Dec}_3^*)$ be any $(\kappa_{\text{dec}} + \lambda \kappa_v)$ -CNS cheating committer-decommitter against $\langle C_3, R_3 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following binding condition holds with overwhelming probability over the choice of $(\text{st}_C, \text{com}) \leftarrow \langle \text{C.Com}_3^*, \text{R.Com}_3 \rangle$.

- Binding Condition: For every $i \in [n]$, it holds $\Pr[b_{BAD} = 1] \leq \operatorname{negl}(\lambda)$ in the following probabilistic experiment $ExP_3^{\operatorname{bind}}(\mathsf{C.Dec}_3^*, \mathsf{st}_C, \mathsf{com}, i)$.
 - 1. For each $\forall b \in \{0,1\}$, sample $\{Q_b^{\mu}\}_{\mu \in [M]}$ by $(\{Q_b^{\mu}\}_{\mu \in [M]}, \mathsf{st}_b) \leftarrow \mathsf{R.Dec.Q}_3(i)$.
 - 2. $Run\{\tilde{Y}_{b}^{\mu:\mu}\}_{b\in\{0,1\},\mu\in[M]} \leftarrow C.Dec_{3}^{*}(st_{C},\{Q_{b}^{\mu}\}_{b\in\{0,1\},\mu\in[M]}).$
 - 3. Let $b_{\text{BAD}} \coloneqq 1$ if and only if $x_0^* \neq \bot \land x_1^* \neq \bot \land x_0^* \neq x_1^*$ holds, where $x_b^* \coloneqq \mathsf{R.Dec.D}_3(\mathsf{st}_b, \mathsf{com}, \{\tilde{Y}^*_b^{\mu\mu}\}_{\mu \in [M]})$ for each $b \in \{0, 1\}$.

Proof. Fix any $C_3^* = (\mathsf{C.Com}_3^*, \mathsf{C.Dec}_3^*)$ as above, and assume for contradiction that for infinitely many $\lambda \in \mathbb{N}$, with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_3^*, \mathsf{R.Com}_3 \rangle$, there exists $i \in [n]$ such that we have $\Pr[b_{\mathsf{BAD}} = 1] \geq 1/\mathsf{poly}(\lambda)$ in the experiment $\operatorname{Exp}_3^{\mathsf{bind}}(\mathsf{C.Dec}_3^*, \mathsf{st}_C, \mathsf{com}, i)$.

To obtain a contradiction, we first define a cheating committer-decommitter $C_2^* = (\text{C.Com}_2^*, \text{C.Dec}_2^*)$ against $\langle C_2, R_2 \rangle$ by using C_3^* , and then show that C_2^* breaks the binding property of $\langle C_2, R_2 \rangle$.

Let us first give some preliminaries. First, from the definitions of R.Dec.Q₃ and R.Dec.D₃, the probabilistic experiment $\operatorname{Exp}_3^{\operatorname{bind}}(\mathsf{C.Dec}^*,\operatorname{st}_C,\operatorname{com},i)$ in the lemma statement can also be written as follows.

- 1. For $\forall b \in \{0, 1\}$, sample $\{Q_b^{\mu}\}_{\mu \in [M]}$ as follows.
 - (a) Run $(Q_{b,0}^{\mu}, \mathsf{st}_{b,0}^{\mu}) \leftarrow \mathsf{LD}\text{-Test.Q}_{D(X_{1:b}^{\mu})}$ for $\forall \mu \in [M]$.
 - (b) Run $(Q_{h_1}^{\mu}, \mathsf{St}_{h_1}^{\mu}) \leftarrow \mathsf{SelfCorr.Q}_{D(X_{i,in}^{\mu})}(\{(\mu, 1, i)\})$ for $\forall \mu \in [M]$.
 - (c) Let $Q_h^{\mu} := Q_{h,0}^{\mu} \cup Q_{h,1}^{\mu}$.
- $\begin{aligned} \text{2. Run } & \{\tilde{Y}^{*\mu:\mu}_b\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \text{C.Dec}^*(\text{st}_C, \{Q^{\mu}_b\}_{b \in \{0,1\}, \mu \in [M]}). \\ & \text{For each } b \in \{0,1\}, \mu \in [M], \text{ let } \tilde{Y}^{*\mu:\mu}_{b,0} \coloneqq \tilde{Y}^{*\mu:\mu}_b|_{Q^{\mu}_{b,0}} \text{ and } \tilde{Y}^{*\mu:\mu}_{b,1} \coloneqq \tilde{Y}^{*\mu:\mu}_b|_{Q^{\mu}_{b,1}} \end{aligned}$
- 3. Let $b_{BAD} := 1$ if and only if both of the following events hold.

- Event₀: LD-Test.D_{$m_0|H|,\zeta$}(st^{μ}_{b,0}, Filter^{μ}($\tilde{Y}^{*\mu;\mu}_{b,0}$)) = 1 holds for $\forall b \in \{0,1\}, \mu \in [M]$.
- Event₁: $x_0^* \neq \bot \land x_1^* \neq \bot \land x_0^* \neq x_1^*$ holds for x_0^*, x_1^* that are defined as follows.
 - (a) Let $\tilde{x}_b^{\mu} := A_b^{\mu}(\mu, 1, i)$, where $A_b^{\mu} := \mathsf{SelfCorr}.\mathsf{Rec}_{m_{10}|H|}(\mathsf{st}_{b,1}^{\mu}, \mathsf{Filter}^{\mu}(\tilde{Y}^{*\mu;\mu}_{b,1}))$.
 - (b) Let $x_b^* := \bot$ if $\exists \mu \in [M]$ such that $\tilde{x}_b^{\mu} \notin \{0, 1\}$, and let $x_b^* := \tilde{x}_b^1 \oplus \cdots \oplus \tilde{x}_b^M$ otherwise.

(Note that Event_0 depends only on $\{\tilde{Y}^{*\mu:\mu}_{b,0}\}_{b\in\{0,1\},\mu\in[M]}$ while Event_1 depends only on $\{\tilde{Y}^{*\mu:\mu}_{b,1}\}_{b\in\{0,1\},\mu\in[M]}$.)

Second, the probabilistic experiment $\operatorname{Exp_2^{bind}}(\mathsf{C.Dec}^*, \operatorname{st}_C, \operatorname{com}, i)$ in Lemma F.1 can be written similarly, where the only difference is that $\operatorname{Event_0}$ is replaced with the following event.

• Event'₀: LD-Test.D_{m_0} $|_{H \setminus 3\zeta}$ (st'_{b, 0}, Filter^{μ}($\tilde{Y}_{b}^{*\mu;\mu}$)) = 1 holds for $\forall b \in \{0,1\}, \mu \in [M]$.

Now, we define $C_2^* = (\mathsf{C.Com}_2^*, \mathsf{C.Dec}_2^*)$. Recall that we assume, for contradiction, that for infinitely many λ , with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_3^*, \mathsf{R.Com}_3 \rangle$, there exists $i \in [n]$ such that we have $\Pr[b_{\mathsf{BAD}} = 1] \geq 1/\mathsf{poly}(\lambda)$ in $\operatorname{Exp}_3^{\mathsf{bind}}(\mathsf{C.Dec}_3^*, \mathsf{st}_C, \mathsf{com}, i)$. Let us call such λ , $(\mathsf{st}_C, \mathsf{com})$, and i be good. Then, the CNS of $\mathsf{C.Dec}_3^*$ guarantees that there exists a constant $c \in \mathbb{N}$ such that for every $\mathsf{good}\ \lambda$ and $(\mathsf{st}_C, \mathsf{com})$, we have

$$\Pr \Bigg[\mathsf{Event}_0 \, \Bigg| \, \begin{array}{l} (Q^\mu_{b,0}, \mathsf{st}^\mu_{b,0}) \leftarrow \mathsf{LD-Test.Q}_{D(X^\mu_{1,\mathsf{in}})} \text{ for } \forall b \in \{0,1\}, \mu \in [M] \\ \{\tilde{Y}^{*\mu,\mu}_{b,0}\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathsf{C.Dec}^*_3(\mathsf{st}_C, \{Q^\mu_{b,0}\}_{b \in \{0,1\}, \mu \in [M]}) \end{array} \Bigg] \geq \frac{1}{\lambda^c}$$

(where Event₀ is defined as in Exp_3^{bind}). Hence, we can use Lemma K.4 (Soundness Amplification Lemma) to obtain a PPT oracle algorithm Amplify, such that for every good λ and (st_C , com), we have

$$\Pr\left[\mathsf{Event}_0' \middle| \begin{array}{l} (Q_{b,0}^{\mu}, \mathsf{st}_{b,0}^{\mu}) \leftarrow \mathsf{LD-Test.Q}_{D(X_{1,n}^{\mu})} \text{ for } \forall b \in \{0,1\}, \mu \in [M] \\ \{\tilde{Y}_{b,0}^{*,\mu;\mu}\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathsf{Amplify}_c^{\mathsf{C.Dee}_3^*(\mathsf{st}_C, \cdot)}(\{Q_{b,0}^{\mu}\}_{b \in \{0,1\}, \mu \in [M]}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \tag{G.1}$$

(where Event'₀ is defined as in Exp_2^{bind}). Given $Assign_c$, we define $C_2^* = (C.Com_2^*, C.Dec_2^*)$ as follows.

- Committer. C.Com₂* works identically with C.Com₃*.
- **Decommitter.** Given st_C and $\{Q^\mu\}_{\mu\in[M]}$ as input, $\operatorname{C.Dec}_2^*$ works identically with $\operatorname{Amplify}_c^{\operatorname{C.Dec}_3^*(\operatorname{st}_C,\cdot)}(\{Q^\mu\}_{\mu\in[M]})$.

Now, our goal is to show that C_2^* breaks the binding property of $\langle C_2, R_2 \rangle$. Since the CNS of C_2^* follows from Lemma K.4,³³ we focus on showing that for infinitely many λ , the binding condition of $\langle C_2, R_2 \rangle$ in Lemma F.1 does not hold with non-negligible probability over the choice of $(\operatorname{st}_C, \operatorname{com}) \leftarrow \langle \operatorname{C.Com}_2^*, \operatorname{R.Com}_2 \rangle$. Toward this end, it suffices to show that for every good λ and $(\operatorname{st}_C, \operatorname{com})$, we have both

$$\Pr \left[\mathsf{Event}_0' \, \left| \begin{array}{l} (Q_b^\mu, \mathsf{st}_b^\mu) \leftarrow \mathsf{LD\text{-}Test.} Q_{D(X_{1,\mathsf{in}}^\mu)} \text{ for } \forall b \in \{0,1\}, \mu \in [M] \\ \{\tilde{Y}_b^{*,\mu:\mu}\}_{b \in \{0,1\}, \mu \in [M]} \leftarrow \mathsf{C.Dec}_2^*(\mathsf{st}_C, \{Q_b^\mu\}_{b \in \{0,1\}, \mu \in [M]}) \end{array} \right] \geq 1 - \epsilon(\lambda) \right. \tag{G.2}$$

and $\Pr[b_{BAD} = 1] \ge 1/\text{poly}(\lambda)$ in $\exp^{\text{bind}}_2(\text{C.Dec}_2^*, \text{st}_C, \text{com}, i)$. First, since C.Dec_2^* is identical with $\text{Amplify}_c^{\text{C.Dec}_3^*}$ by definition, Equation (G.2) follows from Equation (G.1). Second, we obtain $\Pr[b_{BAD} = 1] \ge 1/\text{poly}(\lambda)$ in $\exp^{\text{bind}}_2(\text{C.Dec}_2^*, \text{st}_C, \text{com}, i)$ as follows.

• On the one hand, since we have

$$\Pr[\mathsf{Event}_1 \mid \mathsf{Event}_0] \ge \Pr[\mathsf{Event}_1 \land \mathsf{Event}_0] = \Pr[b_{\text{BAD}} = 1] \ge \frac{1}{\mathsf{poly}(\lambda)}$$

in $\operatorname{Exp}_3^{\operatorname{bind}}(\mathsf{C.Dec}_3^*, \mathsf{st}_C, \mathsf{com}, i)$, the "furthermore" part of Lemma K.4 guarantees that we have $\operatorname{Pr}[\mathsf{Event}_1] \geq 1/\mathsf{poly}(\lambda)$ in $\operatorname{Exp}_2^{\operatorname{bind}}(\mathsf{C.Dec}_2^*, \mathsf{st}_C, \mathsf{com}, i)$.

- On the other hand, Equation (G.2) and the CNS of C.Dec₂* guarantees that we have $\Pr\left[\mathsf{Event}_0'\right] \geq 1 \mathsf{negl}(\lambda)$ in $\mathsf{ExP}_2^\mathsf{bind}(\mathsf{C.Dec}_2^*, \mathsf{st}_C, \mathsf{com}, i)$.
- Hence, from a union bound, we have

$$\Pr\left[b_{\text{\tiny BAD}} = 1\right] = \Pr\left[\mathsf{Event}_0' \land \mathsf{Event}_1\right] \ge \frac{1}{\mathsf{poly}(\lambda)} - \mathsf{negl}(\lambda) \ge \frac{1}{\mathsf{poly}(\lambda)}$$

in $Exp_2^{bind}(C.Dec_2^*, st_C, com, i)$ as desired.

This completes the proof of Lemma G.1.

 $^{^{33}}$ We use $\kappa_{\rm v}$ as an upper bound on the query complexity of each test in LD-Test.

G.3 Proof of Soundness

We first remark that, due to Lemma K.4 and the definition of C.Prv₃*, for every constant $c \in \mathbb{N}$, there is a PPT oracle algorithm Amplify_c such that for every κ_{max} -CNS cheating prover $C_3^* = (\text{C.Com}_3^*, \text{C.Prv}_3^*)$ and every $(\text{st}_C, \text{com}) \leftarrow (\text{C.Com}_3^*, \text{R.Com}_3)$, if it holds

$$\Pr \left[b = 1 \, \middle| \, \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_3; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_3^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D}_3(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \geq \frac{1}{\lambda^c}$$

for infinitely many λ (let Λ be the set of such λ), then Amplify $^{C.Prv_3^*(st_C,\cdot)}$ is an adaptive $(\kappa_{max} - \lambda \kappa_v)$ -CNS cheating prover such that there is a negligible function negl such that for every $\lambda \in \Lambda$,

$$\Pr \left[b = 1 \, \middle| \, \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_2; \ (f, \pi^*) \leftarrow \mathsf{Amplify}_c^{\mathsf{C.Prv}_3^*(\mathsf{st}_C, \cdot)}(Q) \\ b \leftarrow \mathsf{R.Prv.D}_2(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

Now, as the soundness of $\langle C_3, R_3 \rangle$, we prove the following lemma

Lemma G.2. Let $c \in \mathbb{N}$ be any constant, κ_v be the query complexity of (PCP.P, PCP.V), κ_{prv} be the polynomial that is given in Lemma E.3, and let E_3 be the extractor in Algorithm 14. Then, for any $(\kappa_{prv} + \lambda \kappa_v)$ -CNS cheating committerprover $C_3^* = (\text{C.Com}_3^*, \text{C.Prv}_3^*)$ against $\langle C_3, R_3 \rangle$, the following holds with overwhelming probability over the choice of $(\text{st}_C, \text{com}) \leftarrow \langle \text{C.Com}_3^*, \text{R.Com}_3 \rangle$.

• Soundness Condition: If it holds

$$\Pr \left[b = 1 \, \middle| \, \begin{array}{l} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_3; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_3^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D}_3(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \geq \frac{1}{\lambda^c} \ , \tag{G.3}$$

then there exists $x_{COM}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that

$$\forall i \in [n], \Pr\left[x_i = x_i^* \mid (\bot, x_i) \leftarrow \langle E_3^{\mathsf{C.Prv}_3^*(\mathsf{st}_{C,\cdot})}(\mathsf{com}, i), \mathsf{R.Dec}_3(\mathsf{com}, i) \rangle\right] \geq 1 - \mathsf{negl}(\lambda) \tag{G.4}$$

and

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x^*_{\scriptscriptstyle COM})=0\end{array}\right] \left(\begin{array}{c}(Q,\operatorname{st}_R)\leftarrow \operatorname{R.Prv.Q_3};\ (f,\pi^*)\leftarrow \operatorname{C.Prv}_3^*(\operatorname{st}_C,Q)\\ b\leftarrow \operatorname{R.Prv.D_3}(\operatorname{st}_R,\operatorname{com},f,\pi^*)\end{array}\right] \leq \operatorname{negl}(\lambda)\ . \tag{G.5}$$

Algorithm 14 Extractor E_3 (against $\langle C_3, R_3 \rangle$)

Input: com, i, and $\{Q^{\mu}\}_{\mu\in[M]}\subset D(X^1_{1,\mathsf{in}})\times\cdots\times D(X^M_{1,\mathsf{in}})$

- $1. \ \operatorname{Run} \ \{\tilde{Y}^{\mu}\}_{\mu \in [M]} \leftarrow E_2^{\operatorname{C.Prv}_2^*(\operatorname{st}_C, *)}(\operatorname{com}, i, \{Q^{\mu}\}_{\mu \in [M]}), \ \text{where } \operatorname{C.Prv}_2^*(\operatorname{st}_C, \cdot) \coloneqq \operatorname{Amplify}_c^{\operatorname{C.Prv}_3^*(\operatorname{st}_C, \cdot)}(\cdot).$
- 2. Output $\{\tilde{Y}^{\mu}\}_{\mu\in[M]}$ as the decommitment.

Proof. Fix any c and $C_3^* = (C.Com_3^*, C.Prv_3^*)$ as in the lemma statement, and assume for contradiction that for infinitely many $\lambda \in \mathbb{N}$, the soundness condition does not hold with non-negligible probability over the choice of (st_C, com) ← (C.Com₃*, R.Com₃). Let us say that any λ and (st_C, com) are *good* (for C_3^*) if the soundness condition does not hold on λ and (st_C, com). Then, from the binding property of (C_3 , C_3), it follows that for every good λ and (st_C, com), we have Equation (G.3) and also have

• either there exists $i \in [n]$ such that

$$\Pr\left[x_i = \bot \mid (\bot, x_i) \leftarrow \langle E_3^{\mathsf{C.Prv}_3^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_3(\mathsf{com}, i) \rangle\right] \ge \frac{1}{\mathsf{poly}(\lambda)} \ , \tag{G.6}$$

• or there exists $x_{\text{com}}^* = (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that we have Equation (G.4), but we also have

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x_{\text{COM}}^*)=0\end{array}\right| \left(\begin{array}{c}(Q,\operatorname{st}_R)\leftarrow \operatorname{R.Prv.Q_3};\ (f,\pi^*)\leftarrow \operatorname{C.Prv}_3^*(\operatorname{st}_C,Q)\\ b\leftarrow \operatorname{R.Prv.D_3}(\operatorname{st}_R,\operatorname{com},f,\pi^*)\end{array}\right] \geq \frac{1}{\operatorname{poly}(\lambda)}\ . \tag{G.7}$$

To obtain a contradiction, we consider the following cheating committer-prover $C_2^* = (\mathsf{C.Com}_2^*, \mathsf{C.Prv}_2^*)$ against $\langle C_2, R_2 \rangle$.

- C.Com₂* is identical with C.Com₃*.
- C.Prv₂*(st_C, ·) is identical with Amplify_CC.Prv₃*(st_C, ·)(·).

Note that, due to Lemma K.4, C_2^* is κ_{prv} -CNS.

We now show that C^* breaks the soundness property of $\langle C_2, R_2 \rangle$. Fix any good λ and (st_C, com) . First, since we have Equation (G.3), we have

$$\Pr \left[b = 1 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_2; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_2^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D}_2(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right]$$

$$= \Pr \left[b = 1 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_2; \ (f, \pi^*) \leftarrow \mathsf{Amplify}_c^{\mathsf{C.Prv}_3^*(\mathsf{st}_C, \cdot)}(Q) \\ b \leftarrow \mathsf{R.Prv.D}_2(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right]$$

$$\geq 1 - \mathsf{negl}(\lambda) \qquad (\mathsf{from Lemma K.4}) \ . \tag{G.8}$$

Next, consider the following two cases.

• Case 1. Assume that there exists $i \in [n]$ such that we have Equation (G.6). Then, from the constructions of R.Dec₃, C.Prv₂, E_2 , and E_3 , we have

$$\begin{split} & \Pr\left[x_i = \bot \ \middle| \ (\bot, x_i) \leftarrow \langle E_2^{\mathsf{C.Prv}_2^*(\mathsf{st}_{C, \cdot})}(\mathsf{com}, i), \mathsf{R.Dec}_3(\mathsf{com}, i) \rangle \right] \\ & = \Pr\left[x_i = \bot \ \middle| \ (\bot, x_i) \leftarrow \langle E_3^{\mathsf{C.Prv}_3^*(\mathsf{st}_{C, \cdot})}(\mathsf{com}, i), \mathsf{R.Dec}_3(\mathsf{com}, i) \rangle \right] \\ & \geq \frac{1}{\mathsf{poly}(\lambda)} \ . \end{split}$$

This contradicts with the observation that is made in Remark 7.

• Case 2. Assume that there exists $x_{\text{COM}}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that we have Equation (G.4), but we also have Equation (G.7). Then, from the furthermore part of Lemma K.4, we have

$$\begin{split} &\Pr\Big[f(x_{\text{COM}}^*) = 0 \; \Big| \quad (Q, \text{st}_R) \leftarrow \text{R.Prv.Q}_2; \; (f, \pi^*) \leftarrow \text{C.Prv}_2^*(\text{st}_C, Q) \; \Big] \\ &= \Pr\Big[f(x_{\text{COM}}^*) = 0 \; \Big| \quad (Q, \text{st}_R) \leftarrow \text{R.Prv.Q}_2; \; (f, \pi^*) \leftarrow \text{Amplify}_c^{\text{C.Prv}_3^*(\text{st}_C, \cdot)}(Q) \; \Big] \\ &= \Pr\Big[f(x_{\text{COM}}^*) = 0 \; \Big| \quad (Q, \text{st}_R) \leftarrow \text{R.Prv.Q}_3; \; (f, \pi^*) \leftarrow \text{Amplify}_c^{\text{C.Prv}_3^*(\text{st}_C, \cdot)}(Q) \; \Big] \\ &\geq \Pr\Big[\begin{array}{c} b = 1 \\ \wedge \; f(x_{\text{COM}}^*) = 0 \; \Big| \; \; (Q, \text{st}_R) \leftarrow \text{R.Prv.Q}_3; \; (f, \pi^*) \leftarrow \text{C.Prv}_3^*(\text{st}_C, Q) \\ b \leftarrow \text{R.Prv.D}_3(\text{st}_R, \text{com}, f, \pi^*) \\ \end{array} \Big] - \operatorname{negl}(\lambda) \\ &\geq \frac{1}{\operatorname{poly}(\lambda)} \quad (\operatorname{from Equation} \; (G.7)) \; . \end{split}$$

From a union bound with Equation (G.8), we obtain

$$\Pr\begin{bmatrix}b = 1 \land f(x_{\text{com}}^*) = 0 & (Q, \text{st}_R) \leftarrow \text{R.Prv.Q}_2\\ (f, \pi^*) \leftarrow \text{C.Prv}_2^*(\text{st}_C, Q)\\ b \leftarrow \text{R.Prv.D}_2(\text{st}_R, \text{com}, f, \pi^*)\end{bmatrix} \ge \frac{1}{\text{poly}(\lambda)}$$

and thus C_2^* breaks soundness of $\langle C_2, R_2 \rangle$.

By combining what are shown above, we conclude that C_2^* breaks the soundness of $\langle C_2, R_2 \rangle$. By combining all the above, we obtain a contradiction. This concludes the proof of Lemma G.2.

H Step 4: Non-WI Scheme with Standard negl-Soundness

As the fourth step to our commit-and-prove protocol, we give a non-WI commit-and-prove protocol $\langle C_4, R_4 \rangle$ that is negl-sound against (not necessarily CNS) provers. Additionally, we also show that $\langle C_4, R_4 \rangle$ satisfies "2-privacy" in a similar sense to the MPC protocol II. In this step, we use a 2-round PIR protocol PIR = (PIR.Enc, PIR.Res, PIR.Dec).

H.1 Preliminaries

In $\langle C_4, R_4 \rangle$, we use PIR for the receiver to encrypt its PCP queries as in Kalai et al. [KRR14]. To simplify the exposition of $\langle C_4, R_4 \rangle$, we introduce a triple of algorithms, (PIR.EncSet, PIR.ResSet, PIR.DecSet), that use PIR for this purpose. Let κ be a polynomial and $Q \subseteq [N]$ be a subset of the indices of a database $DB \in \Sigma^N$.

- $(\mathbb{Q}, \mathsf{st}_{\mathsf{PIR}}) \leftarrow \mathsf{PIR}.\mathsf{EncSet}_{\kappa(\lambda)}(1^{\lambda}, Q, N)$:
 - 1. Choose a random injective function $\tau: Q \to [\kappa(\lambda)]$.
 - 2. For each $i \in [\kappa(\lambda)]$, run

$$(\textbf{Q}_i, \textbf{st}_{\text{PIR}}^i) \leftarrow \begin{cases} \mathsf{PIR}.\mathsf{Enc}(1^\lambda, q, N) & \text{if } \exists q \in Q \text{ s.t. } \tau(q) = i \\ \mathsf{PIR}.\mathsf{Enc}(1^\lambda, 1, N) & \text{otherwise} \end{cases}$$

- 3. Output $\mathbb{Q} := \{ \mathbb{Q}_i \}_{i \in [\kappa(\lambda)]}$ and $\mathsf{st}_{PIR} := \{ \mathsf{st}_{PIR}^i \}_{i \in [\kappa(\lambda)]}$.
- $\mathbb{X} \leftarrow \mathsf{PIR}.\mathsf{ResSet}(1^{\lambda}, \mathbb{Q}, DB)$
 - 1. For each $i \in [\kappa(\lambda)]$, run $x_i \leftarrow \mathsf{PIR.Res}(1^{\lambda}, q_i, DB)$
 - 2. Output $\mathbb{X} := \{ \mathbb{X}_i \}_{i \in [\kappa(\lambda)]}$.
- $X \leftarrow \mathsf{PIR.DecSet}(\mathsf{st}_{\mathtt{PIR}}, \mathbb{X})$
 - 1. For each $q \in Q$, run $x_{\tau(q)} \leftarrow \mathsf{PIR.Dec}(\mathsf{st}_{\mathsf{PIR}}^{\tau(q)}, \mathsf{x}_{\tau(q)})$.
 - 2. Output $X := \{x_{\tau(a)}\}_{a \in O}$.

H.2 Protocol Description

The formal description of $\langle C_4, R_4 \rangle$ is given in Algorithm 15, where polynomials κ'_{dec} , κ'_{prv} are defined as $\kappa'_{dec} = \kappa_{dec} + \lambda \kappa_{v}$ and $\kappa'_{prv} = \kappa_{prv} + \lambda \kappa_{v}$, where κ_{dec} is the polynomial that is given in Lemma E.1, κ_{prv} is the polynomial that is given in Lemma E.3, and κ_{v} is the query complexity of (PCP.P, PCP.V).

The communication complexity of $\langle C_4, R_4 \rangle$ can be bounded by a polynomial in κ_{dec} , κ_{prv} , and κ_{v} , all of which can be bounded by a polynomial in λ and log(Time(f)) (cf. Footnote 22, Footnote 24, Lemma D.2).

H.3 Proof of Binding

Lemma H.1. Let $C_4^* = (C.Com_4^*, C.Dec_4^*)$ be any PPT cheating committer-decommitter against $\langle C_4, R_4 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following binding condition holds with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com_4^*, R.Com_4 \rangle$.

- Binding Condition: For every $i \in [n]$, it holds $\Pr[b_{BAD} = 1] \leq \operatorname{negl}(\lambda)$ in the following probabilistic experiment $Exp_{\Delta}^{\operatorname{bind}}(\mathsf{C.Dec}_4^*, \mathsf{st}_C, \mathsf{com}, i)$.
 - 1. For each $\forall b \in \{0,1\}$, sample $\{\mathbb{Q}_b^{\mu}\}_{\mu \in [M]}$ by $(\{\mathbb{Q}_b^{\mu}\}_{\mu \in [M]}, \mathsf{st}_b) \leftarrow \mathsf{R.Dec.Q}_4(i)$.
 - 2. $Run\{\tilde{\mathbb{Y}}_{b}^{\mu:\mu}\}_{b\in\{0,1\},\mu\in[M]}\leftarrow\mathsf{C.Dec}_{4}^{*}(\mathsf{st}_{C},\{\mathbb{Q}_{b}^{\mu}\}_{b\in\{0,1\},\mu\in[M]}).$
 - 3. Let $b_{\text{BAD}} \coloneqq 1$ if and only if $x_0^* \neq \bot \land x_1^* \neq \bot \land x_0^* \neq x_1^*$ holds, where $x_b^* \coloneqq \mathsf{R.Dec.D_4}(\mathsf{st}_b, \mathsf{com}, \{\tilde{\mathbb{Y}}_b^{*,\mu:\mu}\}_{\mu \in [M]})$ for each $b \in \{0, 1\}$.

Proof. Fix any $C_4^* = (\mathsf{C.Com}_4^*, \mathsf{C.Dec}_4^*)$ as above, and assume for contradiction that for infinitely many $\lambda \in \mathbb{N}$, with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_4^*, \mathsf{R.Com}_4 \rangle$, there exists $i \in [n]$ such that we have $\Pr[b_{\mathsf{BAD}} = 1] \geq 1/\mathsf{poly}(\lambda)$ in the experiment $\exp^{\mathsf{bind}}_4(\mathsf{C.Dec}_4^*, \mathsf{st}_C, \mathsf{com}, i)$.

To obtain a contradiction, we define a cheating committer-decommitter $C_3^* = (\text{C.Com}_3^*, \text{C.Dec}_3^*)$ against $\langle C_3, R_3 \rangle$ by using C_4^* , and show that C_3^* breaks the binding property of $\langle C_3, R_3 \rangle$. Specifically, consider the following $C_3^* = (\text{C.Com}_3^*, \text{C.Dec}_3^*)$.

- Committer: C.Com₃ is identical with C.Com₄.
- **Decommitter:** Given st_C and $\{Q^{\mu}\}_{\mu\in[M]}$ as input, $\operatorname{C.Dec}_3^*$ does the following.
 - 1. Run $(\mathbb{Q}^{\mu}, \operatorname{st}^{\mu}_{\operatorname{PIR}}) \leftarrow \operatorname{PIR.EncSet}_{\kappa'_{\operatorname{dec}}}(Q^{\mu}, |\pi^{\mu:\nu}|)$ for each $\mu \in [M]$.
 - $2. \ \operatorname{Run} \, \widetilde{\mathbb{Y}_*}^{\mu:\mu} \leftarrow \operatorname{C.Dec}_4^*(\operatorname{st}_C, \{\mathbb{Q}^\mu\}_{\mu \in [M]}).$
 - 3. Run $\tilde{Y}^{*\mu,\mu} \leftarrow \mathsf{PIR.DecSet}(\mathsf{st}^{\mu}_{\mathsf{PIR}}, \tilde{\mathbb{Y}}^{*\mu,\mu})$ for each $\mu \in [M]$.
 - 4. Output $\{\tilde{Y}^{\mu:\mu}_*\}_{\mu\in[M]}$ as the decommitment.

It is straightforward to see that for every λ , (st_C, com), and i such that we have $\Pr[b_{\text{BAD}} = 1] \ge 1/\text{poly}(\lambda)$ in the experiment $\text{Exp}_4^{\text{bind}}(\text{C.Dec}_4^*, \text{st}_C)$, we have $\Pr[b_{\text{BAD}} = 1] \ge 1/\text{poly}(\lambda)$ in the experiment $\text{Exp}_3^{\text{bind}}(\text{C.Dec}_3^*, \text{st}_C)$. (To see this, observe that C.Dec_3^* perfectly emulates R.Dec_4 for the internally emulated C.Dec_4^*). Furthermore, it is relatively easy to see that the security of PIR implies the CNS of C.Dec_4^* . (The proof of this fact is virtually the same as the proof of, e.g., [KRR13, Theorem 12] and [BHK16, Claim 7].) Hence, C_3^* breaks the binding property of $\langle C_3, R_3 \rangle$, and we obtain a contradiction. \square

Algorithm 15 Commit Phase, Open Phase, and Prove Phase of $\langle C_4, R_4 \rangle$

Commit Phase

The commit phase of $\langle C_4, R_4 \rangle$ is identical with that of $\langle C_3, R_3 \rangle$.

Open Phase

Round 1: Given *i* as input, R.Dec.Q₄ does the following.

- 1. Run $(\{Q^{\mu}\}_{\mu \in [M]}, \mathsf{st}_{R}') \leftarrow \mathsf{R.Dec.Q}_{3}(i)$.
- 2. Run $(\mathbb{Q}^{\mu}, \mathsf{st}^{\mu}_{PIR}) \leftarrow \mathsf{PIR}.\mathsf{EncSet}_{\kappa'_{dec}}(Q^{\mu}, |\pi^{\mu:\nu}|)$ for each $\mu \in [M]$. (Here, $|\pi^{\mu:\nu}|$ denotes the length of the PCP proofs that are computed by the prover in the prove phase).
- 3. Output $\{\mathbb{Q}^{\mu}\}_{\mu\in[M]}$ as the query and $(\mathsf{st}'_{R},\{\mathsf{st}^{\mu}_{\mathsf{PIR}}\}_{\mu\in[M]})$ as the internal state.

Round 2: Given $(\operatorname{st}_C, \{\mathbb{Q}^{\mu}\}_{\mu \in [M]})$ as input, C.Dec₄ does the following.

- 1. Define $\tilde{Y}^{\mu:\mu}$ just like C.Dec₃ does.
- 2. Run $\tilde{\mathbb{Y}}^{\mu:\mu} \leftarrow \mathsf{PIR.ResSet}(\mathbb{Q}^{\mu}, \tilde{Y}^{\mu:\mu})$ for each $\mu \in [M]$.
- 3. Output $\{\tilde{\mathbb{Y}}^{\mu:\mu}\}_{\mu\in[M]}$ as the decommitment.

Verification: Given $(\mathsf{st}_R,\mathsf{com},\{\tilde{\mathbb{Y}}^{\mu:\mu}\}_{\mu\in[M]})$ as input (where $\mathsf{st}_R=(\mathsf{st}_R',\{\mathsf{st}_{\mathtt{Pir}}^{\mu}\}_{\mu\in[M]})$, R.Dec₄ does the following.

- 1. Run $\tilde{Y}^{\mu:\mu} \leftarrow \mathsf{PIR.DecSet}(\mathsf{st}^{\mu}_{\mathsf{PIR}}, \tilde{\mathbb{Y}}^{\mu:\mu})$ for each $\mu \in [M]$.
- 2. Output $\tilde{x}_i := \mathsf{R.Dec.D}_3(\mathsf{st}'_R, \mathsf{com}, \{\tilde{Y}^{\mu:\mu}\}_{\mu \in [M]})$ as the decommitted value.

Prove Phase

Round 1: R.Prv₄ does the following.

- 1. Run $(\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]},\operatorname{st}_R')\leftarrow \mathsf{R.Prv.Q}_3.$
- 2. Run $(\mathbb{Q}^{\mu:\nu}, \mathsf{st}_{\mathtt{PIR}}^{\mu:\nu}) \leftarrow \mathsf{PIR}.\mathsf{EncSet}_{\kappa'_{\mathtt{PIV}} + \kappa'_{\mathtt{Aloc}}}(Q^{\mu:\nu}, |\pi^{\mu:\nu}|)$ for each $\mu, \nu \in [M]$.
- 3. Output $\{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}$ as the query and $(\mathsf{st}_R',\{\mathsf{st}_{\mathsf{PIR}}^{\mu:\nu}\}_{\mu,\nu\in[M]})$ as the internal state.

Round 2: Given $(\operatorname{st}_C, f, \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]})$ as input, C.Prv₄ does the following.

- 1. Obtain $\{\mathsf{rt}^{\mu}\}_{\mu\in[M]}$ and $\{\pi^{\mu:\nu}\}_{\mu,\nu\in[M]}$ just like C.Prv₃ does.
- 2. Run $\mathbb{T}^{\mu:\nu} \leftarrow \mathsf{PIR.ResSet}(\mathbb{Q}^{\mu:\nu}, \pi^{\mu:\nu})$ for each $\mu, \nu \in [M]$.
- 3. Output $(\{\mathsf{rt}^{\mu}\}_{\mu\in[M]}, \{\mathbb{T}^{\mu:\nu}\}_{\mu,\nu\in[M]})$ as the proof.

Verification: Given $(\operatorname{st}_R, \operatorname{com}, f, \{\operatorname{rt}^\mu\}_{\mu \in [M]}, \{\operatorname{\mathbb{T}}^{\mu:\nu}\}_{\mu,\nu \in [M]})$ as input (where $\operatorname{st}_R = (\operatorname{st}_R', \{\operatorname{st}_{\operatorname{PIR}}'^{\mu:\nu}\}_{\mu,\nu \in [M]})$), R.Prv₄ does the following.

- 1. Run $\pi^{*\mu:\nu} \leftarrow \mathsf{PIR.DecSet}(\mathsf{st}^{\mu:\nu}_{\mathsf{PIR}}, \pi^{*\mu:\nu})$ for each $\mu, \nu \in [M]$.
- 2. Run $b \leftarrow \mathsf{R.Prv.D}_3(\mathsf{st}_R',\mathsf{com},f,\{\mathsf{rt}^\mu\}_{\mu\in[M]},\{\pi^{*\mu:\nu}\}_{\mu,\nu\in[M]})$. That is, do the following.
 - (a) Run $b^{\mu:\nu} \leftarrow \mathsf{PCP.D'}^{\otimes \lambda}(\mathsf{st}_{V}^{\mu:\nu}, f', \mathsf{rt}_{\mathsf{l} \; \mathsf{in}}^{\mu}, \mathsf{rt}_{\mathsf{l} \; \mathsf{in}}^{\nu}, \mathsf{rt}_{\mathsf{l} \; \mathsf{in}}^{\nu})$ for every $\mu, \nu \in [M]$
 - (b) Let b := 1 if and only if $b^{\mu:\nu} = 1$ for every $\mu, \nu \in [M]$.
- 3. Output *b*.

H.4 Proof of 2-Privacy

Lemma H.2. For any PPT cheating receiver R^* , there exists a PPT simulator S_4 such that for $\forall \lambda \in \mathbb{N}$, $\forall x_{COM} \in \{0, 1\}^n$ (where $n = \mathsf{poly}(\lambda)$), $\forall \alpha, \beta \in [M]$, and $\forall z \in \{0, 1\}^*$, the outputs of the following two experiments are identically distributed.

Real Experiment:

- 1. $(\mathsf{hf}, \mathsf{st}_R) \leftarrow R^*(x_{com}, z)$.
- 2. $(\{\mathsf{rt}_{1 \text{ in}}^{\mu}\}_{\mu \in [M]}, \mathsf{st}_C) \leftarrow \mathsf{C.Com}_4(x_{com}, \mathsf{hf}).$
- 3. $(f, {\mathbb{Q}^{\mu:\nu}}_{\mu,\nu\in[M]}, \operatorname{st}'_R) \leftarrow R^*(\operatorname{st}_R)$. If $f(x_{com}) \neq 1$, the experiment aborts with output \perp .
- 4. $(\{\mathsf{rt}^{\mu}\}_{\mu\in[M]}, \{\mathbb{T}^{\mu:\nu}\}_{\mu,\nu\in[M]}) \leftarrow \mathsf{C.Prv}_4(\mathsf{st}_C, f, \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}).$
- 5. Output (st'_R , $\mathsf{rt}^\alpha_{\mathsf{l},\mathsf{in}}$, $\mathsf{rt}^\beta_{\mathsf{l},\mathsf{in}}$, rt^α , rt^β , $\pi^{\alpha:\beta}$).

Ideal Experiment:

- 1. (hf, st_R) $\leftarrow R^*(x_{COM}, z)$.
- 2. $(f, \{\mathbb{Q}^{\mu;\nu}\}_{\mu,\nu\in[M]}, \mathsf{st}_R') \leftarrow R^*(\mathsf{st}_R)$. If $f(x_{com}) \neq 1$, the experiment aborts with output \perp .
- 3. $(\mathsf{rt}_{1\,\mathsf{in}}^{\alpha},\mathsf{rt}_{1\,\mathsf{in}}^{\beta},\mathsf{rt}^{\alpha},\mathsf{rt}^{\beta},\mathsf{m}^{\alpha:\beta}) \leftarrow \mathcal{S}_4(\alpha,\beta,f,\mathsf{hf},\mathbb{Q}^{\alpha:\beta}).$
- 4. Output $(\mathsf{st}_R', \mathsf{rt}_{\mathsf{l.in}}^\alpha, \mathsf{rt}_{\mathsf{l.in}}^\beta, \mathsf{rt}^\alpha, \mathsf{rt}^\beta, \mathsf{\pi}^{\alpha:\beta})$.

Proof. First, we remark that from the constructions of $C.Com_4$ and $C.Prv_4$, Step 2 and Step 4 of the real experiment can be written as follows.

- 2. Compute $(\{\mathsf{rt}_{1 \text{ in}}^{\mu}\}_{\mu \in [M]}, \mathsf{st}_C)$, where $\mathsf{st}_C = (\mathsf{hf}, \{X_{1 \text{ in}}^{\mu}\}_{\mu \in [M]})$, as follows.
 - (a) Sample random $x_{\text{MPC}}^1, \dots, x_{\text{MPC}}^M \in \{0, 1\}^n$ such that $x_{\text{MPC}}^1 \oplus \dots \oplus x_{\text{MPC}}^M = x_{\text{COM}}$.
 - (b) Compute $X_{1,\text{in}}^{\mu}$ and $\operatorname{rt}_{1,\text{in}}^{\mu}$ for $\forall \mu \in [M]$ as follows: sample random $r_{\text{MPC}}^{\mu} \in \{0,1\}^{n_{\text{MPC}}}$, let $\operatorname{st}_0^{\mu} \coloneqq x_{\text{MPC}}^{\mu} \parallel r_{\text{MPC}}^{\mu}$, let $\operatorname{i-msgs}_1^{\mu} \coloneqq 0^M$, let $x_{1,\text{in}}^{\mu} \coloneqq \operatorname{st}_0^{\mu} \parallel \operatorname{i-msgs}_1^{\mu}$, let $X_{1,\text{in}}^{\mu}$ be the low-degree extension of $x_{1,\text{in}}^{\mu}$, and let $\operatorname{rt}_{1,\text{in}}^{\mu} \coloneqq \operatorname{TreeHash}_{\operatorname{hf}}(X_{1,\text{in}}^{\mu})$.
- 4. Compute $(\{\mathsf{rt}^\mu\}_{\mu\in[M]}, \{\mathbb{T}^{\mu:\nu}\}_{\mu,\nu\in[M]})$ as follows.
 - (a) Run the MPC protocol Π on $(f', \{st_0^{\mu}, i\text{-msgs}_1^{\mu}\}_{\mu \in [M]})$, and let $\{view^{\mu}\}_{\mu \in [M]}$ be the view of the parties in this execution.
 - (b) Run $(\mathsf{rt}^{\mu}, \mathsf{rt}^{\nu}, \pi^{\mu:\nu}) \leftarrow \mathsf{PCP.P'}(\mu, \nu, f', \mathsf{view}^{\mu}, \mathsf{view}^{\nu}, \mathsf{hf})$ for every $\mu, \nu \in [M]$.
 - (c) Run $\pi^{\mu:\nu} \leftarrow \mathsf{PIR.ResSet}(\mathbb{Q}^{\mu:\nu}, \pi^{\mu:\nu})$ for every $\mu, \nu \in [M]$.

Given the above description of the real experiment in mind, we consider the simulator S_4 in Algorithm 16 (where we use the simulator S_{MPC} of Π that is guaranteed to exist due to its 2-privacy).

Algorithm 16 Simulator S_4 (against $\langle C_4, R_4 \rangle$)

- 1. Run {view^{ξ}} $_{\xi \in \{\alpha, \beta\}} \leftarrow \mathcal{S}_{\text{MPC}}(\{\alpha, \beta\}, \{x_{\text{MPC}}^{\xi}\}_{\xi \in \{\alpha, \beta\}}, 1)$ for random $x_{\text{MPC}}^{\alpha}, x_{\text{MPC}}^{\beta} \in \{0, 1\}^{n}$. Let r_{MPC}^{ξ} , i-msgs $_{0}^{\xi}$ be the randomness and the dummy incoming messages of the first round that are recorded in view^{ξ} for $\forall \xi \in \{\alpha, \beta\}$.
- 2. Compute $\operatorname{rt}_{1,\operatorname{in}}^{\xi}$ for $\xi \in \{\alpha,\beta\}$ as follows: let $x_{1,\operatorname{in}}^{\xi} \coloneqq x_{\operatorname{MPC}}^{\xi} \| r_{\operatorname{MPC}}^{\xi} \| i \operatorname{msgs}_{1}^{\xi}$, let $X_{1,\operatorname{in}}^{\xi}$ be the low-degree extension of $x_{1,\operatorname{in}}^{\xi}$, and let $\operatorname{rt}_{1,\operatorname{in}}^{\xi} \coloneqq \operatorname{TreeHash}_{\operatorname{hf}}(X_{1,\operatorname{in}}^{\xi})$.
- 3. Run $(\mathsf{rt}^{\alpha}, \mathsf{rt}^{\beta}, \pi^{\alpha:\beta}) \leftarrow \mathsf{PCP.P'}(\alpha, \beta, f', \mathsf{view}^{\alpha}, \mathsf{view}^{\beta}, \mathsf{hf})$.
- 4. Run $\pi^{\alpha:\beta} \leftarrow \mathsf{PIR}.\mathsf{ResSet}(\mathbb{Q}^{\alpha:\beta}, \pi^{\alpha:\beta})$.
- 5. Output $(\mathsf{rt}_{1,\mathsf{in}}^{\alpha},\mathsf{rt}_{1,\mathsf{in}}^{\beta},\mathsf{rt}^{\alpha},\mathsf{rt}^{\beta},\pi^{\alpha:\beta})$.

Now, it is easy to see that the output of the ideal experiment is identically distributed with that of the real experiment. \Box

H.5 Proof of Soundness

Lemma H.3. Fix any constant $c \in \mathbb{N}$, and let E_4 be the extractor in Algorithm 17. Then, for any PPT cheating committer-prover $C_4^* = (C.Com_4^*, C.Prv_4^*)$ against $\langle C_4, R_4 \rangle$, the following condition holds with overwhelming probability over the choice of $(st_C, com) \leftarrow \langle C.Com_4^*, R.Com_4 \rangle$.

• Soundness Condition: If it holds

$$\Pr \left[b = 1 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q_4}; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_4^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D_4}(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \ge \frac{1}{\lambda^c} \ , \tag{H.1}$$

then there exists $x_{com}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that

$$\forall i \in [n], \Pr\left[x_i = x_i^* \mid (\bot, x_i) \leftarrow \langle E_4^{\mathsf{C.Prv}_4^*(\mathsf{st}_{C,\cdot})}(\mathsf{com}, i), \mathsf{R.Dec}_4(\mathsf{com}, i) \rangle\right] \geq 1 - \mathsf{negl}(\lambda) \tag{H.2}$$

and

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x^*_{\scriptscriptstyle COM})=0\end{array}\right| \left(\begin{array}{c}(Q,\operatorname{st}_R)\leftarrow \operatorname{R.Prv.Q_4};\ (f,\pi^*)\leftarrow \operatorname{C.Prv}_4^*(\operatorname{st}_C,Q)\\ b\leftarrow \operatorname{R.Prv.D_4}(\operatorname{st}_R,\operatorname{com},f,\pi^*)\end{array}\right] \leq \operatorname{negl}(\lambda)\ . \tag{H.3}$$

Algorithm 17 Extractor E_4 (against $\langle C_4, R_4 \rangle$)

Input: com, i, and $\{\mathbb{Q}^{\mu}\}_{\mu\in[M]}$.

- 1. Define $\{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}$ as follows: let $\mathbb{Q}^{\mu:\mu}:=\mathbb{Q}^{\mu}$ for $\forall\mu\in[M]$, and let $\mathbb{Q}^{\mu:\nu}$ be sampled by $(\mathbb{Q}^{\mu:\nu},\mathsf{st}^{\mu:\nu}_{\mathsf{PIR}})\leftarrow\mathsf{PIR}.\mathsf{EncSet}_{\kappa'_{\mathsf{Abc}}}(\emptyset,|\pi^{\mu:\nu}|)$ for $\forall\mu,\nu\in[M]$ such that $\mu\neq\nu$.
- 2. For each $i \in [\lambda^{c+1}]$, do the following.
 - (a) Run $(\{\mathbb{Q}_i^{\mu:\nu}\}_{\mu,\nu\in[M]},\mathsf{st}_i)\leftarrow\mathsf{R.Prv.Q}_4'$, where $\mathsf{R.Prv.Q}_4'$ is identical with $\mathsf{R.Prv.Q}_4$ except that $\mathsf{PIR.EncSet}_{\kappa'_{\mathsf{prv}}+\kappa'_{\mathsf{dec}}}$.
 - (b) For each $\mu, \nu \in [M]$, choose a random permutation $\tau : [\kappa'_{\text{prv}} + \kappa'_{\text{dec}}] \to [\kappa'_{\text{prv}} + \kappa'_{\text{dec}}]$ and define $\mathbb{S}_i^{\mu:\nu} = \{s_j\}_{j \in [\kappa'_{\text{prv}} + \kappa'_{\text{dec}}]}$ as follows: parse $\mathbb{Q}^{\mu:\nu}$ as $\{\mathbb{Q}_j^{\mu:\nu}\}_{j \in [\kappa'_{\text{dec}}]}$ and parse $\mathbb{Q}_i^{\mu:\nu}$ as $\{\mathbb{Q}_{i,j}^{\mu:\nu}\}_{j \in [\kappa'_{\text{prv}}]}$; then, each s_j is defined as

$$s_j \coloneqq \begin{cases} \mathbb{q}_{i,k}^{\mu:\nu} & \text{if } k \coloneqq \tau^{-1}(j) \text{ satisfies } k \in \{1,\dots,\kappa'_{\text{prv}}\} \\ \mathbb{q}_k^{\mu:\nu} & \text{if } k \coloneqq \tau^{-1}(j) \text{ satisfies } k \in \{\kappa'_{\text{prv}} + 1,\dots,\kappa'_{\text{prv}} + \kappa'_{\text{dec}}\} \end{cases}.$$

- $\text{(c) } \operatorname{Run} \ (f_i, \{\mathsf{rt}_i^\mu\}_{\mu \in [M]}, \{\wp_i^{*\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \operatorname{C.Prv}_4^*(\mathsf{st}_C, \{\mathbb{S}_i^{\mu:\nu}\}_{\mu,\nu \in [M]}), \text{ and parse each } \wp_i^{*\mu:\nu} \text{ as } \{\wp_{i,j}^{*\mu:\nu}\}_{j \in [\kappa_{\max}]}.$
- $\text{(d) For each } \mu,\nu\in[M]\text{, let }\mathbb{Y}_{i}^{*\mu:\nu}\coloneqq\{\mathbb{p}_{i,\tau(k)}^{*\mu:\nu}\}_{k\in[\kappa_{\text{prv}}+1,\dots,\kappa_{\text{prv}}'+\kappa_{\text{dec}}']}\text{ and }\mathbb{\pi}_{i}^{*\mu:\nu}\coloneqq\{\mathbb{p}_{i,\tau(k)}^{*\mu:\nu}\}_{k\in\{1,\dots,\kappa_{\text{prv}}'\}}.$
- 3. Find the first $i^* \in [\lambda^{c+1}]$ such that

R.Prv.D₄(st_{i*}, com,
$$f_{i*}$$
, {rt ^{μ} _{i*}} _{$\mu \in [M]$} , { $\pi^{*\mu:\nu}_{i*}$ } _{$\mu,\nu \in [M]$}) = 1.

If such i^* does not exist, output \perp . Otherwise, send $\{\mathbb{Y}^{*\mu:\mu}_{i^*}\}_{\mu\in[M]}$ to R.Dec₄.

Remark 8 (Intuition of E_4). Essentially, E_4 emulates an execution of E_3 while (1) inlining executions of Amplify_c (Algorithm 22 in Section K.2) and E_2 , and (2) encrypting the queries to the cheating prover by PIR.

Proof. Fix any c and $C_4^* = (C.Com_4^*, C.Prv_4^*)$ as above, and assume for contradiction that for infinitely many $\lambda \in \mathbb{N}$, with non-negligible probability over the choice of $(st_C, com) \leftarrow (C.Com_4^*, R.Com_4)$, the soundness condition does not hold.

To obtain a contradiction, we define a cheating committer-prover $C_3^* = (\mathsf{C.Com}_3^*, \mathsf{C.Prv}_3^*)$ against $\langle C_3, R_3 \rangle$ by using C_4^* , and show that C_3^* breaks the soundness of $\langle C_3, R_3 \rangle$. Specifically, consider the following $C_3^* = (\mathsf{C.Com}_3^*, \mathsf{C.Prv}_3^*)$.

- Committer: C.Com₃ is identical with C.Com₄.
- **Prover:** Given st_C and $\{Q^{\mu:\nu}\}_{\mu,\nu\in[M]}$ as input, $\operatorname{C.Prv}_3^*$ does the following.
 - 1. Run $(\mathbb{Q}^{\mu:\nu}, \mathsf{st}_{\mathtt{PIR}}^{\mu:\nu}) \leftarrow \mathsf{PIR}.\mathsf{EncSet}_{\kappa'_{\mathtt{DIY}} + \kappa'_{\mathtt{dec}}}(Q^{\mu:\nu}, |\pi^{\mu:\nu}|)$ for each $\mu, \nu \in [M]$.
 - 2. Run $(f, \{\mathsf{rt}^{\mu}\}_{\mu \in [M]}, \{\pi^{*\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{C.Prv}_{4}^{*}(\mathsf{st}_{C}, \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu \in [M]}).$
 - 3. Run $\pi^{*\mu:\nu} \leftarrow \mathsf{PIR.DecSet}(\mathsf{st}^{\mu:\nu}_{\scriptscriptstyle \mathsf{PIR}}, \pi^{*\mu:\nu})$ for each $\mu, \nu \in [M]$.

4. Output $(f, \{rt^{\mu}\}_{\mu \in [M]}, \{\pi^{*\mu:\nu}\}_{\mu,\nu \in [M]})$ as the proof.

By carefully comparing $E_4^{\text{C.Prv}_4^*(\text{st}_C,\cdot)}$ and $E_3^{\text{C.Prv}_3^*(\text{st}_C,\cdot)}$, one can see that for every λ and (st_C,com) for which the soundness condition of $\langle C_4,R_4\rangle$ does not hold against C_4^* , the soundness condition of $\langle C_3,R_3\rangle$ does not hold against C_3^* either. (To see this, observe that the execution of $E_4^{\text{C.Prv}_4^*(\text{st}_C,\cdot)}$ is perfectly emulated during the execution of $E_3^{\text{C.Prv}_3^*(\text{st}_C,\cdot)}$). Furthermore, it is relatively easy to see that the security of PIR implies the CNS of C.Prv $_4^*$. (The proof of this fact is virtually the same as the proof of, e.g., [KRR13, Theorem 12] and [BHK16, Claim 7].) Hence, C_3^* breaks the soundness of $\langle C_3,R_3\rangle$, and we obtain a contradiction.

I Step 5: WI Scheme with Standard O(1)-Soundness

Finally, we give our WI commit-and-prove protocol $\langle C_5, R_5 \rangle$. In this section, we use the following additional building blocks.

- Non-interactive statistically binding commitment scheme SBCom. (A non-interactive one is used for simplicity, and actually a 2-round one is sufficient. It is known that a 2-round statistically binding commitment scheme can be obtained from a one-way function, which in turn can be obtained from a collision-resistant hash function.)
- 2-round statistically hiding commitment scheme SHCom such that the first-round message from the receiver is a hash function. (Such a statistically hiding commitment scheme can be obtained from a collision-resistant hash function.)
- 2-round 1-out-of-M² OT protocol OT.

Also, we assume that all the building blocks (the above ones and the ones that are used in $\langle C_4, R_4 \rangle$) are sub-exponentially secure, so that we can assume that (1) the committed value of a SBCom commitment can be extracted by brute force in a quasi-polynomial time T_{SB} , and (2) the security of SHCom, OT, and $\langle C_4, R_4 \rangle$ holds against poly(T_{SB})-time adversaries.

I.1 Protocol Description

The formal description of $\langle C_5, R_5 \rangle$ is given in Algorithm 18. It is easy to see that the communication complexity of $\langle C_5, R_5 \rangle$ is bounded by a polynomial in λ and the communication complexity of $\langle C_4, R_4 \rangle$, and hence is bounded by a polynomial in λ and log(Time(f)).

I.2 Proof of Binding

Lemma I.1. Let $C_5^* = (\mathsf{C.Com}_5^*, \mathsf{C.Dec}_5^*)$ be any $\mathsf{poly}(T_{s_B})$ -time cheating committer-decommitter against $\langle C_5, R_5 \rangle$. Then, for every $\lambda \in \mathbb{N}$, the following binding condition holds with overwhelming probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_5^*, \mathsf{R.Com}_5 \rangle$.

- Binding Condition: For every $i \in [n]$, it holds $\Pr[b_{BAD} = 1] \leq \operatorname{negl}(\lambda)$ in the following probabilistic experiment $Exp_5^{\text{bind}}(\text{C.Dec}_5^*, \text{st}_C, \text{com}, i)$.
 - 1. For each $\forall b \in \{0, 1\}$, sample $\{\mathbb{Q}_b^{\mu}\}_{\mu \in [M]}$ by $(\{\mathbb{Q}_b^{\mu}\}_{\mu \in [M]}, \mathsf{st}_b) \leftarrow \mathsf{R.Dec.Q}_5(i)$.
 - 2. $Run \{ dec_b \}_{b \in \{0,1\}} \leftarrow C.Dec_5^*(st_C, \{\mathbb{Q}_b^{\mu}\}_{b \in \{0,1\}, \mu \in [M]}).$
 - 3. Let $b_{\text{\tiny BAD}} \coloneqq 1$ if and only if $x_0^* \neq \bot \land x_1^* \neq \bot \land x_0^* \neq x_1^*$ holds, where $x_b^* \coloneqq \mathsf{R.Dec.D}_5(\mathsf{st}_b, \mathsf{com}, \mathsf{dec}_b)$ for each $b \in \{0, 1\}$.

Proof. Fix any $C_5^* = (\mathsf{C.Com}_5^*, \mathsf{C.Dec}_5^*)$ as above, and assume for contradiction that for infinitely many $\lambda \in \mathbb{N}$, with non-negligible probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_5^*, \mathsf{R.Com}_5 \rangle$, there exists $i \in [n]$ such that we have $\Pr[b_{\mathsf{BAD}} = 1] \geq 1/\mathsf{poly}(\lambda)$ in the experiment $\operatorname{Exp}_5^{\mathsf{bind}}(\mathsf{C.Dec}_5^*, \mathsf{st}_C, \mathsf{com}, i)$.

To obtain a contradiction, we define a cheating committer-decommitter $C_4^* = (\mathsf{C.Com}_4^*, \mathsf{C.Dec}_4^*)$ against $\langle C_4, R_4 \rangle$ by using C_5^* , and show that C_4^* breaks the binding property of $\langle C_4, R_4 \rangle$.

Concretely, we define $C_4^* = (C.Com_4^*, C.Dec_4^*)$ as follows.

- Committer: Given hf as input, C.Com₄ does the following.
 - 1. Run (com', st'_C) \leftarrow C.Com₅*(hf), and parse com' as {com_{SH}^{\(\mu\)}} $_{\mu\in[M]}$.
 - 2. Define com and st_C as follows.
 - (a) Sample $\{\mathbb{Q}_b^{\mu}\}_{\mu \in [M]}$ by $(\{\mathbb{Q}_b^{\mu}\}_{\mu \in [M]}, \mathsf{st}_b) \leftarrow \mathsf{R.Dec.Q}_5(i)$ for $\forall b \in \{0, 1\}$.
 - $\text{(b) } \text{Run } \{ \mathsf{dec}_b \}_{b \in \{0,1\}} \leftarrow \mathsf{C.Dec}_5^*(\mathsf{st}_C, \{\mathbb{Q}_b^{\mu}\}_{b \in \{0,1\}, \mu \in [M]}), \text{ and parse } \mathsf{dec}_0 \text{ as } (\{\tilde{\mathbb{Y}}^{*^{H'H}}\}_{\mu \in [M]}, \{\mathsf{rt}_{1,\mathsf{in}}^{\mu}, \mathsf{dec}_{\mathsf{sh}}^{\mu}\}_{\mu \in [M]}).$
 - (c) Let com := $\{\mathsf{rt}^{\mu}_{1,\mathsf{in}}\}_{\mu\in[M]}$ and $\mathsf{st}_C \coloneqq (\mathsf{com},\mathsf{com}',\mathsf{st}'_C)$ if $\mathsf{dec}^{\mu}_{\mathsf{sH}}$ is a valid decommitment for opening $\mathsf{com}^{\mu}_{\mathsf{sH}}$ to $\mathsf{rt}^{\mu}_{1,\mathsf{in}}$ for $\forall \mu \in [M]$, and let $\mathsf{com} \coloneqq \bot$ and $\mathsf{st}_C \coloneqq \bot$ otherwise.

Algorithm 18 Commit Phase, Open Phase, and Prove Phase of $\langle C_5, R_5 \rangle$

Commit Phase

Round 1: R.Com₅ works identically with R.Com₄, that is, sends a hash function $hf \in \mathcal{H}$ to C.Com₅.

Round 2: Given (x_{com}, hf) as input, C.Com₅ does the following.

- 1. Run $(\{\mathsf{rt}_{1 \text{ in}}^{\mu}\}_{\mu \in [M]}, \mathsf{st}_C') \leftarrow \mathsf{C}.\mathsf{Com}_4(x_{\mathsf{COM}}, \mathsf{hf}).$
- 2. Run $com_{sH}^{\mu} \leftarrow SHCom_{hf}(rt_{1.in}^{\mu})$ for every $\mu \in [M]$. Let dec_{sH}^{μ} be the decommitment for opening com_{sH}^{μ} to $rt_{1.in}^{\mu}$.
- 3. Output $\{\mathsf{com}_{\mathtt{SH}}^{\mu}\}_{\mu\in[M]}$ as the commitment and $(\mathsf{st}_C', \{\mathsf{dec}_{\mathtt{SH}}^{\mu}\}_{\mu\in[M]})$ as the internal state.

.....

Open Phase

Round 1: Given i as input, R.Dec.Q₅ runs $(\{\mathbb{Q}^{\mu}\}_{\mu \in [M]}, \operatorname{st}_R) \leftarrow \operatorname{R.Dec.Q}_4(i)$, and then outputs $\{\mathbb{Q}^{\mu}\}_{\mu \in [M]}$ as the query and st_R as the internal state.

Round 2: Given $(\operatorname{st}_C, \{\mathbb{Q}^\mu\}_{\mu \in [M]})$ as input (where $\operatorname{st}_C = (\operatorname{st}'_C, \{\operatorname{dec}^\mu_{\operatorname{sH}}\}_{\mu \in [M]})$), C.Dec₅ runs $\{\tilde{\mathbb{Y}}^{\mu : \mu}\}_{\mu \in [M]} \leftarrow C.Dec_4(\operatorname{st}'_C, \{\mathbb{Q}^\mu\}_{\mu \in [M]})$ and then outputs $(\{\tilde{\mathbb{Y}}^{\mu : \mu}\}_{\mu \in [M]}, \{\operatorname{rt}^\mu_{1,\operatorname{in}}, \operatorname{dec}^\mu_{\operatorname{sH}}\}_{\mu \in [M]})$ as the decommitment.

Verification: Given $(\operatorname{st}_R, \operatorname{com}, \{\widetilde{\mathbb{Y}}^{*^{\mu,\mu}}\}_{\mu \in [M]}, \{\operatorname{rt}^{\mu}_{1,\operatorname{in}}, \operatorname{dec}^{\mu}_{\operatorname{sH}}\}_{\mu \in [M]})$ as input (where $\operatorname{com} = \{\operatorname{com}^{\mu}_{\operatorname{sH}}\}_{\mu \in [M]}$), R.Dec.D₅ does the following.

- 1. Check that each $\operatorname{dec}_{\operatorname{sh}}^{\mu}$ is a valid decommitment for opening $\operatorname{com}_{\operatorname{sh}}^{\mu}$ to $\operatorname{rt}_{\operatorname{lin}}^{\mu}$.
- 2. Output $\tilde{x}_i := \mathsf{R.Dec.D_4}(\mathsf{st}_R, \mathsf{com'}, \{\tilde{\mathbb{Y}}^{*,\mu^{1,\mu}}\}_{u \in [M]})$ as the decommitted value, where $\mathsf{com'} := \{\mathsf{rt}^{\mu}_{1, \mathsf{in}}\}_{u \in [M]}$.

Prove Phase

Round 1: R.Prv.Q₅ does the following.

- 1. Run $(\{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]},\mathsf{st}_R')\leftarrow\mathsf{R.Prv.Q}_4.$
- 2. Choose random $\alpha, \beta \in [M]$, and run $(\mathsf{ot}_1, \mathsf{st}_{\mathsf{or}}) \leftarrow \mathsf{OT}_1(1^\lambda, (\alpha, \beta))$.
- 3. Output $(\{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]},\mathsf{ot}_1)$ as the query and $(\mathsf{st}_R',\alpha,\beta,\mathsf{st}_{ot})$ as the internal state.

Round 2: Given $(\operatorname{st}_C, f, \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}, \operatorname{ot}_1)$ as input (where $\operatorname{st}_C = (\operatorname{st}'_C, \{\operatorname{dec}^{\mu}_{\operatorname{sh}}\}_{\mu\in[M]})$), C.Prv₅ does the following.

- 1. Run $(\{\mathsf{rt}^{\mu}\}_{\mu \in [M]}, \{\mathbb{T}^{\mu:\nu}\}_{\mu,\nu \in [M]}) \leftarrow \mathsf{C.Prv}_4(\mathsf{st}'_C, f, \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu \in [M]}).$
- 2. Run $com_{s_B}^{\mu} \leftarrow SBCom(rt_{1,in}^{\mu} \| dec_{s_H}^{\mu} \| rt^{\mu})$ for each $\mu \in [M]$. Let $dec_{s_B}^{\mu}$ be the decommitment for opening $com_{s_B}^{\mu}$ to $rt_{1,in}^{\mu} \| dec_{s_H}^{\mu} \| rt^{\mu}$.
- 3. Run $com_{sB}^{\mu:\nu} \leftarrow SBCom(\mathbb{T}^{\mu:\nu})$ for each $\mu, \nu \in [M]$. Let $dec_{sB}^{\mu:\nu}$ be the decommitment for opening $com_{sB}^{\mu:\nu}$ to $\mathbb{T}^{\mu:\nu}$.
- 4. Run ot₂ \leftarrow OT₂(ot₁, {msg}^{\mu:\nu}}_{\mu,\nu\in[M]}), where msg}^{\mu:\nu} := (rt^{\mu}_{1 \text{ in}}, dec^{\mu}_{sH}, rt^{\mu}, dec^{\mu}_{sB}, rt^{\nu}_{1, \text{in}}, dec^{\nu}_{sH}, rt^{\nu}, dec^{\nu}_{sB}, \pi^{\mu:\nu}, dec^{\mu:\nu}_{sB}).
- 5. Output $(\{com_{sB}^{\mu}\}_{\mu \in [M]}, \{com_{sB}^{\mu : \nu}\}_{\mu,\nu \in [M]}, ot_2)$ as the proof.

Verification Given $(\mathsf{st}_R,\mathsf{com},f,\{\mathsf{com}_{\mathtt{SB}}^\mu\}_{\mu\in[M]},\{\mathsf{com}_{\mathtt{SB}}^{\mu,\nu}\}_{\mu,\nu\in[M]},\mathsf{ot}_2)$ as input (where $\mathsf{st}_R=(\mathsf{st}_R',\alpha,\beta,\mathsf{st}_{\mathtt{OT}})$ and $\mathsf{com}=\{\mathsf{com}_{\mathtt{SH}}^\mu\}_{\mu\in[M]}$), R.Prv.D₅ does the following.

- $1. \ \text{Run msg}^{\alpha:\beta} \coloneqq \text{OT}_3(\text{st}_{\text{ot}}, \text{ot}_2), \text{ where msg}^{\alpha:\beta} \coloneqq (\text{rt}_{1,\text{in}}^{\alpha}, \text{dec}_{\text{sh}}^{\alpha}, \text{rt}^{\alpha}, \text{dec}_{\text{sh}}^{\alpha}, \text{rt}^{\beta}, \text{dec}_{\text{sh}}^{\alpha}, \text{rt}^{\beta}, \text{dec}_{\text{sh}}^{\alpha:\beta}).$
- 2. Check that
 - $\bullet \ \operatorname{dec}_{\operatorname{SH}}^{\xi} \text{ is a valid decommitment for opening } \operatorname{com}_{\operatorname{SH}}^{\xi} \text{ to } \operatorname{rt}_{1,\operatorname{in}}^{\xi} \text{ for } \forall \xi \in \{\alpha,\beta\},$
 - $\mathsf{dec}_{\mathsf{SB}}^{\xi}$ is a valid decommitment for opening $\mathsf{com}_{\mathsf{SB}}^{\xi}$ to $\mathsf{rt}_{\mathsf{Lin}}^{\xi} \| \mathsf{dec}_{\mathsf{SH}}^{\xi} \| \mathsf{rt}^{\xi}$ for $\forall \xi \in \{\alpha, \beta\}$, and
 - $dec_{SB}^{\alpha:\beta}$ is a valid decommitment for opening $com_{SB}^{\alpha:\beta}$ to $\pi^{\alpha:\beta}$.
- 3. Run $\pi^{\alpha:\beta} \leftarrow \mathsf{PIR.DecSet}(\mathsf{st}^{\alpha:\beta}_{\mathsf{PIR}}, \pi^{\alpha:\beta})$.
- 4. Output $b^{\alpha:\beta} \leftarrow \mathsf{PCP.D'}^{\otimes \lambda}(\mathsf{st}_V^{\alpha:\beta}, f', \mathsf{rt}_{\mathsf{Lin}}^\alpha, \mathsf{rt}_{\mathsf{Lin}}^\beta, \mathsf{rt}^\alpha, \mathsf{rt}^\beta, \pi^{*\alpha:\beta}).$

- 3. Output com as the commitment, and st_C as the internal state.
- **Decommitter:** Given st_C and $\{\mathbb{Q}^\mu_b\}_{b\in\{0,1\},\mu\in[M]}$ as input, $\operatorname{C.Dec}_4^*$ aborts if $\operatorname{st}_C=\bot$, and does the following otherwise.
 - 1. Parse st_C as (com, com', st'_C), and parse com' as $\{\operatorname{com}^{\mu}_{\operatorname{SH}}\}_{\mu \in [M]}$.
 - 2. Run $\{ dec'_b \}_{b \in \{0,1\}} \leftarrow C. Dec_5^*(st'_C, \{ \mathbb{Q}_b^{\mu} \}_{b \in \{0,1\}, \mu \in [M]}).$
 - 3. For each $b \in \{0, 1\}$, define dec_b as follows.
 - (a) Parse dec_b' as $(\{\tilde{\mathbb{Y}}_*^{\mu:\mu}\}_{\mu\in[M]}, \{\text{rt}_{1,\text{in}}^{\mu}, \text{dec}_{\text{sH}}^{\mu}\}_{\mu\in[M]}).$
 - (b) Let $\operatorname{dec}_b := \bot$ if (1) $\operatorname{dec}_{\operatorname{sH}}^{\mu}$ is a valid decommitment for opening $\operatorname{com}_{\operatorname{sH}}^{\mu}$ to $\operatorname{rt}_{1,\operatorname{in}}^{\mu}$ for $\forall \mu \in [M]$ but (2) $\{\operatorname{rt}_{1,\operatorname{in}}^{\mu}\}_{\mu \in [M]} \neq \operatorname{com}$. Let $\operatorname{dec}_b := \{\widetilde{\mathbb{Y}}^{*^{\mu,\mu}}\}_{\mu \in [M]}$ otherwise.
 - 4. Output $\{ dec_b \}_{b \in \{0,1\}}$ as the decommitments.

Now, we show that C_4^* breaks the binding property of $\langle C_4, R_4 \rangle$. First, note that the binding property of SHCom implies that for $\forall \lambda \in \mathbb{N}$, with overwhelming probability over the choice of $(\mathsf{st}_C, \mathsf{com}) \leftarrow \langle \mathsf{C.Com}_4^*, \mathsf{R.Com}_4 \rangle$, we have that for $\forall i \in [n]$,

$$\Pr[\mathsf{st}_C \neq \bot \land (\exists b \in \{0,1\} \text{ s.t. } \mathsf{dec}_b = \bot)] \leq \mathsf{negl}(\lambda)$$

in $\operatorname{Exp}_4^{\operatorname{bind}}(\operatorname{C.Dec}_4^*,\operatorname{st}_C,\operatorname{com},i)$. Given this, one can easily see that from the assumption that C_5^* breaks the binding property of $\langle C_5,R_5\rangle$, it follows that for $\forall \lambda \in \mathbb{N}$, with non-negligible probability over the choice of $(\operatorname{st}_C,\operatorname{com}) \leftarrow \langle \operatorname{C.Com}_4^*,\operatorname{R.Com}_4\rangle$, there exists $i \in [n]$ such that we have $\operatorname{Pr}[b_{\operatorname{BAD}}=1] \geq 1/\operatorname{poly}(\lambda)$ in the experiment $\operatorname{Exp}_4^{\operatorname{bind}}(\operatorname{C.Dec}_4^*,\operatorname{st}_C,\operatorname{com},i)$. Hence, C_4^* breaks the binding property of $\langle C_4,R_4\rangle$, and we obtain a contradiction.

I.3 Proof of Witness Indistinguishability

Lemma I.2. $\langle C_5, R_5 \rangle$ is witness indistinguishable.

Proof. We show the witness indistinguishability by showing that there exists a super-polynomial-time simulator that can simulate the receiver's view in Experiment b (Definition 4) for $\forall b \in \{0, 1\}$ without knowing the value of b. Fix any cheating receiver R^* , and consider the simulator S_5 in Algorithm 19 (where we use the simulator S_4 of $\langle C_4, R_4 \rangle$ that is guaranteed to exist due to its 2-privacy).

Algorithm 19 Simulator S_5

Commit Phase. Given hf from R^* , do the following.

- 1. Simulate $\{com_{sH}^{\mu}\}_{{\mu}\in[M]}$ by committing to all-zero strings by using SHCom.
- 2. Send $\{\operatorname{com}_{\operatorname{SH}}^{\mu}\}_{\mu\in[M]}$ to R^* .

Prove Phase. Given $(f, {\mathbb{Q}^{\mu:\nu}}_{\mu,\nu\in[M]}, \text{ot}_1)$ from R^* , does the following.

- 1. Extract the receiver's choice (α, β) by $(\alpha, \beta) := \mathsf{Ext}_{\mathsf{OT}}(\mathsf{ot}_1)$. (Note: This step requires super-polynomial time; cf. Definition 13.)
- 2. Simulate com_{SB}^{α} , com_{SB}^{β} , and $com_{SB}^{\alpha;\beta}$ as follows.
 - (a) Run $(\mathsf{rt}_{1,\mathsf{in}}^{\alpha},\mathsf{rt}_{1,\mathsf{in}}^{\beta},\mathsf{rt}^{\alpha},\mathsf{rt}^{\beta},\pi^{\alpha:\beta}) \leftarrow \mathcal{S}_4(\alpha,\beta,f,\mathsf{hf},\mathbb{Q}^{\alpha:\beta}).$
 - (b) For each $\xi \in \{\alpha, \beta\}$, compute (by brute force) a decommitment $\operatorname{dec}_{\operatorname{sh}}^{\xi}$ for opening $\operatorname{com}_{\operatorname{sh}}^{\xi}$ to $\operatorname{rt}_{\operatorname{Lin}}^{\xi}$.
 - (c) For each $\xi \in \{\alpha, \beta\}$, obtain $\mathsf{com}_{\mathtt{SB}}^{\xi}$ by $\mathsf{com}_{\mathtt{SB}}^{\xi} \leftarrow \mathsf{SBCom}(\mathsf{rt}_{\mathtt{Lin}}^{\xi} \| \mathsf{dec}_{\mathtt{SH}}^{\xi} \| \mathsf{rt}^{\xi})$.
 - (d) Obtain $com_{SR}^{\alpha:\beta}$ by $com_{SR}^{\alpha:\beta} \leftarrow SBCom(\pi^{\alpha:\beta})$.
- 3. Simulate com_{SB}^{μ} , com_{SB}^{ν} , and $com_{SB}^{\mu:\nu}$ for $\forall (\mu, \nu) \neq (\alpha, \beta)$ by committing to all-zero strings by using SBCom.
- 4. Simulate ot₂ by ot₂ \leftarrow OT₂(ot₁, {msg}^{μ : ν}} $_{\mu,\nu\in[M]}$), where

$$\mathsf{msg}^{\alpha:\beta} \coloneqq (\mathsf{rt}^\alpha_{1.\mathsf{in}}, \mathsf{dec}^\alpha_{\mathsf{sH}}, \mathsf{rt}^\alpha, \mathsf{dec}^\alpha_{\mathsf{sB}}, \mathsf{rt}^\beta_{1.\mathsf{in}}, \mathsf{dec}^\beta_{\mathsf{sH}}, \mathsf{rt}^\beta, \mathsf{dec}^\beta_{\mathsf{sB}}, \mathbb{T}^{\alpha:\beta}, \mathsf{dec}^{\alpha:\beta}_{\mathsf{sB}}) \enspace ,$$

and $\mathsf{msg}^{\mu:\nu}$ for $\forall (\mu, \nu) \neq (\alpha, \beta)$ is an all-zero string.

5. Output $(f, \{com_{sB}^{\mu}\}_{\mu \in [M]}, \{com_{sB}^{\mu : \nu}\}_{\mu, \nu \in [M]}, ot_2)$ as the proof.

Now, we consider a sequence of hybrid experiments.

- Hybrid H_0 is identical with the real experiment.
- Hybrid H_1 is identical with H_0 except that in the prove phase, (1) the receiver's choice (α, β) is extracted by $(\alpha, \beta) := \mathsf{Ext}_{\mathsf{OT}}(\mathsf{ot}_1)$, and (2) $\mathsf{msg}^{\mu,\nu}$ for $\forall (\mu, \nu) \neq (\alpha, \beta)$ is an all-zero string.
- Hybrid H_2 is identical with H_1 except that in the prove phase, com_{sB}^{μ} , com_{sB}^{ν} , and $com_{sB}^{\mu:\nu}$ for $\forall (\mu, \nu) \neq (\alpha, \beta)$ are generated by committing to all-zero strings by using SBCom.
- Hybrid H_3 is identical with H_2 except that (1) in the commit phase, $\{\mathsf{com}_{\mathtt{SH}}^{\mu}\}_{\mu \in [M]}$ is generate by committing to all-zero strings by using SHCom, and (2) in the prove phase, $\mathsf{com}_{\mathtt{SB}}^{\alpha}$ and $\mathsf{com}_{\mathtt{SB}}^{\beta}$ are generated as follows.
 - 1. For each $\xi \in \{\alpha, \beta\}$, compute (by brute force) a decommitment $\operatorname{dec}_{\operatorname{sh}}^{\xi}$ for opening $\operatorname{com}_{\operatorname{sh}}^{\mu}$ to $\operatorname{rt}_{\operatorname{l.in.}}^{\xi}$.
 - 2. For each $\xi \in \{\alpha, \beta\}$, obtain $\mathsf{com}_{\mathtt{SB}}^{\xi}$ by $\mathsf{com}_{\mathtt{SB}}^{\xi} \leftarrow \mathsf{SBCom}(\mathsf{rt}_{\mathtt{Lin}}^{\xi} \| \mathsf{dec}_{\mathtt{SH}}^{\xi} \| \mathsf{rt}^{\xi})$.
- Hybrid H_4 is identical with H_3 except that $\mathsf{rt}_{1,\mathsf{in}}^\alpha, \mathsf{rt}_{1,\mathsf{in}}^\beta, \mathsf{rt}^\alpha, \mathsf{rt}^\beta, \mathsf{and} \ \mathbb{\pi}^{\alpha;\beta}$ are generated by $(\mathsf{rt}_{1,\mathsf{in}}^\alpha, \mathsf{rt}^\alpha, \mathsf{rt}^\alpha, \mathsf{rt}^\beta, \mathbb{\pi}^{\alpha;\beta}) \leftarrow \mathcal{S}_4(\alpha, \beta, f, \mathsf{hf}, \mathbb{Q}^{\alpha;\beta}).$
- Hybrid H_5 is identical with the ideal execution.

From a hybrid argument, it suffices to show that the output of each hybrid is indistinguishable from that of the preceding one

Claim I.1. The output of H_1 is computationally indistinguishable from that of H_0 .

Proof. Assume for contradiction that for infinitely many λ , the output of H_0 and that of H_1 are distinguishable. Fix any such λ . From an average argument, it follows that the output of H_0 and that of H_1 are distinguishable even when the transcript of their executions are fixed up until Round 1 of the prove phase. Since the transcript up until Round 1 of the prove phase determines the value of (α, β) , and Round 2 of the prove phase of H_1 can be emulated in polynomial time given (α, β) , we can now break the sender security of OT, and hence obtain a contradiction.

Claim I.2. The output of H_2 is computationally indistinguishable from that of H_1 .

Proof. This claim can be proven similarly to Claim I.1; the difference is that we rely on the hiding property of SBCom instead of the sender security of OT. \Box

Claim I.3. The output of H_3 is statistically indistinguishable from that of H_2 .

Proof . This claim follows from the statistical hiding property of SHCom.

Claim I.4. The output of H_4 is identically distributed with that of H_3 .

Proof. This claim follows from the 2-privacy of $\langle C_4, R_4 \rangle$.

Claim I.5. The output of H_5 is identically distributed with that of H_4 .

Proof. This claim can be proven trivially since, by inspection, one can see that the messages that are sent to R^* in H_4 are generated identically with those in the ideal execution.

This concludes the proof of Lemma I.2.

I.4 Proof of Soundness

Lemma I.3. Fix any constant $c \in \mathbb{N}$, and let E_5 be the extractor in Algorithm 20. Then, for any poly(T_{sB})-time cheating committer-prover $C_5^* = (C.Com_5^*, C.Prv_5^*)$ against $\langle C_5, R_5 \rangle$, the following condition holds with overwhelming probability over the choice of (st_C , com) $\leftarrow \langle C.Com_5^*, R.Com_5 \rangle$.

• Soundness Condition: If it holds

$$\Pr\left[b = 1 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_5; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_5^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D}_5(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \ge 1 - \frac{1}{M^2} + \frac{1}{\lambda^c} \ , \tag{I.1}$$

then there exists $x_{COM}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that

$$\forall i \in [n], \Pr\left[x_i = x_i^* \mid (\bot, x_i) \leftarrow \langle E_5^{\mathsf{C.Prv}_5^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_5(\mathsf{com}, i) \rangle \right] \geq 1 - \mathsf{negl}(\lambda) \tag{I.2}$$

and

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x^*_{\scriptscriptstyle COM})=0\end{array}\right] \quad \begin{array}{c|c}(Q,\operatorname{st}_R) \leftarrow \operatorname{R.Prv.Q}_5; \ (f,\pi^*) \leftarrow \operatorname{C.Prv}_5^*(\operatorname{st}_C,Q)\\ b \leftarrow \operatorname{R.Prv.D}_5(\operatorname{st}_R,\operatorname{com},f,\pi^*)\end{array}\right] \leq 1-\frac{1}{M^2} + \operatorname{negl}(\lambda) \ .$$

Algorithm 20 Extractor E_5 (against $\langle C_5, R_5 \rangle$)

Input: com, i, and $\{\mathbb{Q}^{\mu}\}_{\mu\in[M]}$, where com = $\{\operatorname{com}_{\operatorname{SH}}^{\mu}\}_{\mu\in[M]}$.

- **Step 1.** Repeat the following at most T_{sB} times to obtain $\{\mathsf{rt}_{1,\mathsf{in}}^{\mu}, \mathsf{dec}_{sH}^{\mu}\}_{\mu \in [M]}$ such that each dec_{sH}^{μ} is a valid decommitment for opening com_{sH}^{μ} to $\mathsf{rt}_{1,\mathsf{in}}^{\mu}$.
 - 1. Run $(\{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}, \mathsf{ot}_1, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_5$.
 - 2. Run $(f, \{com_{sB}^{\mu}\}_{\mu \in [M]}, \{com_{sB}^{\mu;\nu}\}_{\mu,\nu \in [M]}, ot_2) \leftarrow C.Prv_5^*(st_C, \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu \in [M]}, ot_1).$
 - 3. Extract the committed values of $\{com_{sB}^{\mu}\}_{\mu \in [M]}$ by brute force. Let $\{rt_{1 \text{ in}}^{\mu} || dec_{sH}^{\mu} || rt^{\mu}\}_{\mu \in [M]}$ be the extracted values.

Abort if such $\{rt_{1 \text{ in}}^{\mu}, dec_{\text{SH}}^{\mu}\}_{\mu \in [M]}$ does not obtained after repeating the above for T_{SB} times.

- Step 2. Run $\{\mathbb{Y}^{*\mu:\mu}\}_{\mu\in[M]} \leftarrow E_4^{\mathcal{A}}(\mathsf{com}',i,\{\mathbb{Q}^{\mu}\}_{\mu\in[M]})$, where com' is defined as $\mathsf{com}' \coloneqq \{\mathsf{rt}_{1,\mathsf{in}}^{\mu}\}_{\mu\in[M]}$, and \mathcal{A} is the cheating prover that works as follows on input any $\{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}$:
 - 1. Run $(f, \{\mathsf{com}_{\mathsf{sB}}^{\mu}\}_{\mu \in [M]}, \{\mathsf{com}_{\mathsf{sB}}^{\mu : \nu}\}_{\mu,\nu \in [M]}, \mathsf{ot}_2) \leftarrow \mathsf{C.Prv}_5^*(\mathsf{st}_C, \{\mathbb{Q}^{\mu : \nu}\}_{\mu,\nu \in [M]}, \mathsf{ot}_1), \text{ where ot}_1 \text{ is sampled by } (\mathsf{ot}_1, \mathsf{st}_{\mathsf{or}}) \leftarrow \mathsf{OT}_1(1^{\lambda}, (\alpha, \beta)) \text{ for random } \alpha, \beta \in [M].$
 - 2. Define proof as follows.
 - (a) Extract the committed values of $\{\mathsf{com}_{\mathtt{SB}}^{\mu}\}_{\mu \in [M]}, \{\mathsf{com}_{\mathtt{SB}}^{\mu : \nu}\}_{\mu, \nu \in [M]}$ by brute force. Let $\{\mathsf{rt}_{1,\mathsf{in}}^{\mu} \parallel \mathsf{dec}_{\mathtt{SH}}^{\mu} \parallel \mathsf{rt}_{1,\mathsf{in}}^{\mu}\}_{\mu \in [M]}, \{\mathbb{T}^{\mu : \nu}\}_{\mu, \nu \in [M]}$ be the extracted values.
 - (b) Let proof := \bot if (1) $\operatorname{dec}_{\operatorname{sh}}^{\mu}$ is a valid decommitment for opening $\operatorname{com}_{\operatorname{sh}}^{\mu}$ to $\operatorname{rt}_{1,\operatorname{in}}^{\mu}$ for $\forall \mu \in [M]$ but (2) $\{\operatorname{rt}_{1,\operatorname{in}}^{\mu}\}_{\mu \in [M]}$ is not equal to the one that is obtained in Step 1. Let proof := $(\{\operatorname{rt}^{\mu}\}_{\mu \in [M]}, \{\operatorname{\mathbb{T}}^{\mu:\nu}\}_{\mu,\nu \in [M]})$ otherwise.
 - 3. Output (f, proof) as the proof.

Step 3. Send $\{\{Y^{*\mu:\mu}\}_{\mu\in[M]}, \{rt^{\mu}_{1 \text{ in}}, dec^{\mu}_{SH}\}_{\mu\in[M]}\}$ to R.Dec₅.

Proof. Fix any c and $C_5^* = (C.Com_5^*, C.Prv_5^*)$ as above, and assume for contradiction that for infinitely many λ , with non-negligible probability over the choice of (st_C, com) ← $(C.Com_5^*, R.Com_5)$, the soundness condition does not hold.

To obtain a contradiction, we define a cheating committer-prover $C_4^* = (\mathsf{C.Com}_4^*, \mathsf{C.Prv}_4^*)$ against $\langle C_4, R_4 \rangle$ by using C_5^* , and show that C_4^* breaks the soundness of $\langle C_4, R_4 \rangle$.

Concretely, we consider the following $C_4^* = (C.Com_4^*, C.Prv_4^*)$.

- Committer: Given hf as input, C.Com₄ does the following.
 - 1. Run (com', st'_C) \leftarrow C.Com₅*(hf), and parse com' as {com_{SH}}_{$\mu \in [M]$}.
 - 2. Repeat the following at most T_{SB} times to obtain $\{\text{rt}_{1,\text{in}}^{\mu}, \text{dec}_{\text{SH}}^{\mu}\}_{\mu \in [M]}$ such that each $\text{dec}_{\text{SH}}^{\mu}$ is a valid decommitment for opening $\text{com}_{\text{SH}}^{\mu}$ to $\text{rt}_{1,\text{in}}^{\mu}$.
 - (a) Run $(\{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]}, \mathsf{ot}_1, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_5$.
 - (b) $\operatorname{Run}(f, \{\operatorname{\mathsf{com}}^{\mu}_{\operatorname{SB}}\}_{\mu \in [M]}, \{\operatorname{\mathsf{com}}^{\mu : \nu}_{\operatorname{SB}}\}_{\mu, \nu \in [M]}, \operatorname{\mathsf{ot}}_2) \leftarrow \operatorname{\mathsf{C.Prv}}^*_5(\operatorname{\mathsf{st}}_C, \{\mathbb{Q}^{\mu : \nu}\}_{\mu, \nu \in [M]}, \operatorname{\mathsf{ot}}_1).$
 - (c) Extract the committed values of $\{com_{sB}^{\mu}\}_{\mu\in[M]}$ by brute force. Let $\{rt_{1,in}^{\mu}\|dec_{sH}^{\mu}\|rt^{\mu}\}_{\mu\in[M]}$ be the extracted values.

Let $b_{\text{BAD}} := 1$ if such $\{\mathsf{rt}_{1,\text{in}}^{\mu}, \mathsf{dec}_{\text{SH}}^{\mu}\}_{\mu \in [M]}$ does not obtained after repeating the above T_{SB} times, and let $b_{\text{BAD}} := 0$ otherwise.

- 3. If $b_{\text{BAD}} = 0$, output com := $\{\text{rt}_{1,\text{in}}^{\mu}\}_{\mu \in [M]}$ as the commitment and $\text{st}_C := (\text{com}, \text{com}', \text{st}_C')$ as the internal state. If $b_{\text{BAD}} = 1$, output com := \bot as the commitment and $\text{st}_C := (\bot, \text{com}', \text{st}_C')$ as the internal state.
- **Prover:** Given $(\operatorname{st}_C, \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu\in[M]})$ as input, $\operatorname{C.Prv}_4^*$ does the following.
 - 1. Parse st_C as $(\operatorname{com}, \operatorname{com}', \operatorname{st}'_C)$, and parse com' as $\{\operatorname{com}^{\mu}_{\operatorname{SH}}\}_{\mu \in [M]}$. Abort if $\operatorname{com} = \bot$.
 - 2. Run $(f, \{\mathsf{com}^{\mu}_{\mathtt{SB}}\}_{\mu \in [M]}, \{\mathsf{com}^{\mu:\nu}_{\mathtt{SB}}\}_{\mu,\nu \in [M]}, \mathsf{ot}_2) \leftarrow \mathsf{C.Prv}_5^*(\mathsf{st}_C', \{\mathbb{Q}^{\mu:\nu}\}_{\mu,\nu \in [M]}, \mathsf{ot}_1), \text{ where ot}_1 \text{ is sampled by } (\mathsf{ot}_1, \mathsf{st}_{\mathtt{OT}}) \leftarrow \mathsf{OT}_1(1^{\lambda}, (\alpha, \beta)) \text{ for random } \alpha, \beta \in [M].$
 - 3. Define proof as follows.
 - (a) Extract the committed values of $\{com_{sB}^{\mu}\}_{\mu\in[M]}, \{com_{sB}^{\mu;\nu}\}_{\mu,\nu\in[M]}$ by brute force. Let $\{rt_{1,in}^{\mu} \parallel dec_{sH}^{\mu} \parallel rt^{\mu}\}_{\mu\in[M]}, \{rt^{\mu;\nu}\}_{\mu,\nu\in[M]}$ be the extracted values.
 - (b) Let proof := \bot if (1) $\operatorname{dec}_{\operatorname{sh}}^{\mu}$ is a valid decommitment for opening $\operatorname{com}_{\operatorname{sh}}^{\mu}$ to $\operatorname{rt}_{1,\operatorname{in}}^{\mu}$ for $\forall \mu \in [M]$ but (2) $\{\operatorname{rt}_{1,\operatorname{in}}^{\mu}\}_{\mu \in [M]} \neq \operatorname{com}$. Let proof := $(\{\operatorname{rt}^{\mu}\}_{\mu \in [M]}, \{\mathbb{T}^{\mu : \nu}\}_{\mu,\nu \in [M]})$ otherwise.
 - 4. Output (f, proof) as the proof.

Now, we show that C_4^* breaks the soundness of $\langle C_4, R_4 \rangle$. Recall that we assume for contradiction that C_5^* breaks the soundness of $\langle C_5, R_5 \rangle$. Combined with the the binding property of $\langle C_5, R_5 \rangle$, this assumption implies that for infinitely many $\lambda \in \mathbb{N}$, with non-negligible probability over the choice of $(\operatorname{st}_C, \operatorname{com}) \leftarrow \langle \operatorname{C.Com}_5^*, \operatorname{R.Com}_5 \rangle$, we have both Equation (I.1) and the following:

• either there exists $i \in [n]$ such that

$$\Pr\left[x_i = \bot \mid (\bot, x_i) \leftarrow \langle E_5^{\mathsf{C.Prv}_5^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_5(\mathsf{com}, i) \rangle\right] \ge \frac{1}{\mathsf{poly}(\lambda)} , \tag{I.3}$$

• or there exists $x_{\text{com}}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that we have Equation (I.2), but we also have

$$\Pr\left[b = 1 \land f(x_{\text{com}}^*) = 0 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q}_5; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_5^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D}_5(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \ge 1 - \frac{1}{M^2} + \frac{1}{\mathsf{poly}(\lambda)} \ . \tag{I.4}$$

Then, we consider two cases.

Case 1. First, we consider the case that for infinitely many λ , with non-negligible probability over the choice of $(st_C, com) \leftarrow \langle C.Com_5^*, R.Com_5 \rangle$, we have Equations (I.1) and (I.3). Let Λ be the set of such λ . In this case, our goal is to obtain a contradiction by showing that for $\forall \lambda \in \Lambda$, with non-negligible probability over the choice of $(st_C, com) \leftarrow \langle C.Com_4^*, R.Com_4 \rangle$, we have

$$\Pr \left[b = 1 \middle| \begin{array}{c} (Q, \mathsf{st}_R) \leftarrow \mathsf{R.Prv.Q_4}; \ (f, \pi^*) \leftarrow \mathsf{C.Prv}_4^*(\mathsf{st}_C, Q) \\ b \leftarrow \mathsf{R.Prv.D_4}(\mathsf{st}_R, \mathsf{com}, f, \pi^*) \end{array} \right] \ge \frac{1}{\lambda^c} \ , \tag{I.5}$$

and

$$\Pr\left[x_i = \bot \mid (\bot, x_i) \leftarrow \langle E_4^{\mathsf{C.Prv}_4^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_4(\mathsf{com}, i) \rangle\right] \ge \frac{1}{\mathsf{poly}(\lambda)} \ . \tag{I.6}$$

For any $\lambda \in \Lambda$ and $(\operatorname{st}_C, \operatorname{com}) \leftarrow \langle \operatorname{C.Com}_5^*, \operatorname{R.Com}_5 \rangle$, let us say that $(\operatorname{st}_C, \operatorname{com})$ is good for C_5^* if Equations (I.1) and (I.3) hold w.r.t. $(\operatorname{st}_C, \operatorname{com})$. Similarly, for any $\lambda \in \Lambda$ and $(\operatorname{st}_C, \operatorname{com}) \leftarrow \langle \operatorname{C.Com}_4^*, \operatorname{R.Com}_4 \rangle$, let us say that $(\operatorname{st}_C, \operatorname{com})$ is good for C_4^* if st_C can be parsed as $(\operatorname{com}, \operatorname{com}', \operatorname{st}'_C)$ such that $(\operatorname{st}'_C, \operatorname{com}')$ is good for C_5^* . From the assumption of this case, it follows that for $\forall \lambda \in \Lambda$, an execution of $\langle \operatorname{C.Com}_5^*, \operatorname{R.Com}_5 \rangle$ produces good $(\operatorname{st}_C, \operatorname{com})$ with non-negligible probability. Hence, from the construction of C_4^* , it follows that for $\forall \lambda \in \Lambda$, an execution of $\langle \operatorname{C.Com}_4^*, \operatorname{R.Com}_4 \rangle$ produces good $(\operatorname{st}_C, \operatorname{com})$ with non-negligible probability.

We first observe that for $\forall \lambda \in \Lambda$ and an overwhelming fraction of good (st_C , com) for C_4^* , we have Equation (I.5). Toward this end, we make a sequence of observations.

- 1. First, for $\forall \lambda \in \Lambda$ and an overwhelming fraction of good (st_C , com) for C_4^* , we have that st_C can be parsed as (com , com' , st'_C) such that $\mathsf{com} \neq \bot$. This is because under that condition that an execution of $\langle \mathsf{C.Com}_4^*, \mathsf{R.Com}_4 \rangle$ produces good (st_C , com), we have $\Pr[b_{\mathsf{BAD}} = 0] \leq \mathsf{negl}(\lambda)$ during the execution of $\langle \mathsf{C.Com}_4^*, \mathsf{R.Com}_4 \rangle$ due to the receiver security of OT (Definition 12) and Equation (I.1), 34 which is guaranteed by the definition of good (st_C , com).
- 2. Next, for $\forall \lambda \in \Lambda$ and an overwhelming fraction of good (st_C , com) for C_4^* , we have both of the following.
 - st_C can be parsed as (com, com', st'_C) such that com $\neq \bot$.
 - The probability that $C.Prv_4^*(st_C)$ outputs (f, proof) such that $proof = \bot$ during an execution of $(C.Prv_4^*(st_C), R.Prv_4(com))$ is negligible.

This is because of the above observation and the binding property of SHCom.

3. Finally, for $\forall \lambda \in \Lambda$ and an overwhelming fraction of good (st_C , com) for C_4^* , we have Equation (I.5). This is because of the above observation and the receiver security of OT. Indeed, if Equation (I.5) does not hold for non-negligible fraction of good (st_C , com) for C_4^* , one can break the receiver security of OT (Definition 12) by emulating an execution of $\langle \mathsf{C.Prv}_4^*(\mathsf{st}_C), \mathsf{R.Prv}_4(\mathsf{com}) \rangle$ since the internally emulated $\mathsf{C.Prv}_5^*(\mathsf{st}_C)$ satisfies Equation (I.1).

We next observe that for $\forall \lambda \in \Lambda$ and a non-negligible fraction of good (st_C , com) for C_4^* , we have Equation (I.6). This can be observed by combining the fact that an execution of

³⁴Specifically, the receiver security of OT and Equation (I.1) imply that during the execution of $\langle C.Com_4^*, R.Com_4 \rangle$, each trial for obtaining $\{rt_{1,in}^{\mu}, dec_{si}^{\mu}\}_{\mu \in [M]}$ succeeds with probability at least $1/\lambda^c$. Thus, from Marcov's inequality, the probability that all the T_{sb} trials fail is negligible.

- 1. $(st_C, com) \leftarrow \langle C.Com_4^*, R.Com_4 \rangle$;
- 2. $(\bot, x_i) \leftarrow \langle E_4^{\mathsf{C.Prv}_4^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_4(\mathsf{com}, i) \rangle$

perfectly emulates an execution of

- 1. $(st_C, com) \leftarrow \langle C.Com_5^*, R.Com_5 \rangle$;
- 2. $(\bot, x_i) \leftarrow \langle E_5^{\mathsf{C.Prv}_5^*(\mathsf{st}_C, \cdot)}(\mathsf{com}, i), \mathsf{R.Dec}_5(\mathsf{com}, i) \rangle$

(where the first step of E_5 in the latter execution is emulated in the commit phase in the former execution), and the fact that for $\forall \lambda \in \Lambda$ and any good (st_C , com) for C_5^* , we have Equation (I.3).

By combining what is observed in the above two paragraphs, we conclude that for $\forall \lambda \in \Lambda$, with non-negligible probability over the choice of $(st_C, com) \leftarrow \langle C.Com_4^*, R.Com_4 \rangle$, we have Equations (I.5) and (I.6) as desired.

Case 2. We next consider the case that for infinitely many $\lambda \in \mathbb{N}$, with non-negligible probability over the choice of $(st_C, com) \leftarrow (C.Com_5^*, R.Com_5)$, we have Equation (I.1) and the following.

• There exists $x_{\text{COM}}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that we have Equation (I.2), but we also have Equation (I.4).

Let Λ be the set of such λ . In this case, our goal is to obtain a contradiction by showing that for $\forall \lambda \in \Lambda$, with non-negligible probability over the choice of $(st_C, com) \leftarrow \langle C.Com_4^*, R.Com_4 \rangle$, we have Equations (I.5) and the following.

• There exists $x_{\text{COM}}^* := (x_1^*, \dots, x_n^*) \in \{0, 1\}^n$ such that we have Equation (H.2), but we also have

$$\Pr\left[\begin{array}{c|c}b=1\\ \land f(x_{\text{COM}}^*)=0\end{array}\right| \left(\begin{array}{c}(Q,\mathsf{st}_R)\leftarrow\mathsf{R.Prv.Q_4};\ (f,\pi^*)\leftarrow\mathsf{C.Prv}_4^*(\mathsf{st}_C,Q)\\ b\leftarrow\mathsf{R.Prv.D_4}(\mathsf{st}_R,\mathsf{com},f,\pi^*)\end{array}\right]\geq \frac{1}{\mathsf{poly}(\lambda)}\ . \tag{I.7}$$

The proof for this case is similar to the proof for Case 1. Specifically, we can show that Equation (I.1) implies Equation (I.5) as in Case 1, and we can also show that Equation (I.4) implies Equation (I.7) very similarly to Case 1.

By combining the analysis of the above two cases, we complete the proof of Lemma I.3.

J ZK Scheme with Standard negl-Soundness

As mentioned in Section 1.1, our constant-sound WI commit-and-prove protocol $\langle C_5, R_5 \rangle$ in Appendix I can be transformed into a one that is zero-knowledge and negl-sound by using a variant of a transformation of Khurana et al. [KOS18]. For completeness, we briefly explain how $\langle C_5, R_5 \rangle$ is transformed by it. (An overview of the transformation is given in [KOS18].)

J.1 Protocol Description

The transformation uses the following building blocks.

- A three-round commitment scheme two-com such that the committer commits to two values \hat{s}_0 , \hat{s}_1 in the first round and then reveals them in the third round, and it is guaranteed that (1) one of the two commitments is binding while the other is equivocal, and (2) the receiver cannot tell which commitment is equivocal. Black-box constructions of such a commitment scheme are given in [ORS15, KOS18].
- A finite field G such that |G| is exponentially large.

We describe the transformed protocol $\langle C_6, R_6 \rangle$ in Algorithm 21. (The only difference between the transformation that we use and the one given in [KOS18] is the definition of a in Round 2 of the prove phase, where in [KOS18], it is defined as $a := x\alpha + r$ by viewing the committed value x as an element in G.

J.2 Proof Sketch of Binding

The binding property follows from that of $\langle C_5, R_5 \rangle$.

³⁵Roughly speaking, we can still use the same security proof as Khurana et al. [KOS18] since, essentially, the only property that they use is that for any (x, r) and (x', r') such that $x \neq x'$, we have $x\alpha + r \neq x'\alpha + r'$ with high probability when α is chosen uniformly randomly.

Algorithm 21 Commit Phase, Open Phase, and Prove Phase of $\langle C_6, R_6 \rangle$

Commit Phase

Round 1: R.Com₆ runs $hf_j \leftarrow R.Com_5$ for $\forall j \in [\lambda]$ and then sends $\{hf_j\}_{j \in [\lambda]}$ and the first-round message of two-com (\hat{s}_0, \hat{s}_1) for randomly chosen $\hat{s}_0, \hat{s}_1 \in \{0, 1\}^{\lambda}$.

Round 2: C.Com₆ does the following.

- 1. Choose random $r \in G$ and $s \in \{0, 1\}^{\lambda}$, and let $x'_{COM} := x_{COM} ||r|| s$.
- 2. Run $(com_j, st_C^j) \leftarrow C.Com_5(x'_{com}, hf)$ for $\forall j \in [\lambda]$.
- 3. Send $\{com_i\}_{i \in [\lambda]}$ and the second-round message of two-com as the commitment.

.....

Prove Phase

Round 1: R.Prv.Q₆ runs $(Q_j, \operatorname{st}_R^j) \leftarrow \operatorname{R.Prv.Q}_5$ for $j \in [\lambda]$, and then sends $\{Q_j\}_{j \in [\lambda]}$, randomly chosen $\alpha \in G$, and the third-round message of two-com (which reveals \hat{s}_0 and \hat{s}_1) as the query.

Round 2: C.Prv₆ does the following.

- 1. Compute $a = x_n \alpha^n + \dots + x_1 \alpha + r$ (in the field G), where x_i is the i-th bit of x_{com} .
- 2. Run $\pi_j \leftarrow \text{C.Prv}_5(\text{st}_C^j, \hat{f}, Q_j)$ for $\forall j \in [\lambda]$, where the function \hat{f} checks, given input $x'_{\text{COM}} = x_{\text{COM}} \| r \| s$, whether it holds

$$(f(x) = 1 \land x_n \alpha^n + \dots + x_1 \alpha + r = a) \lor (s = \hat{s}_0 \lor s = \hat{s}_1) .$$

3. Send a and $\{\pi_j\}_{j\in[\lambda]}$ as the proof.

Verification: R.Prv.D₆ outputs 1 if and only if R.Prv.D₅(st_R^j , com_j , \hat{f} , π_j) = 1 for $\forall j \in [\lambda]$.

.....

Open Phase

Round 1: R.Dec.Q₆ runs $(Q_j, \operatorname{st}_R^j) \leftarrow \operatorname{R.Dec.Q}_5(i)$ for $\forall j \in [\lambda]$ and sends $\{Q_j\}_{j \in [\lambda]}$ and the third-round message of two-com as the query.

Round 2: C.Dec₆ runs $\text{dec}_j \leftarrow \text{C.Dec}_5(\text{st}_C^j, Q_j)$ for $\forall j \in [\lambda]$ and sends $\{\text{dec}_j\}_{j \in [\lambda]}$ as the decommitment. **Verification:** R.Dec.D₆ does the following.

- 1. Check whether there exists x_i such that it holds R.Dec.D₅(st_R^j , com_i , dec_i) = x_i for more than $\lambda/2$ values of $j \in [\lambda]$.
- 2. Output x_i as the decommitted value if such x_i exists, and output \perp otherwise.

J.3 Proof Sketch of Zero-knowledge

The proof of zero-knowledge proceeds identically with that given in [KOS18]. Specifically, it suffices to consider a simulator that first obtains the values \hat{s}_0 , \hat{s}_1 in Round 1 of the prove phase and then uses them to simulate the receiver's view in the commit and prove phases by relying on rewinding techniques.

J.4 Proof Sketch of Soundness

The proof of soundness proceeds essentially identically with that given in [KOS18]. Specifically, we prove it by considering an extractor that applies the extractor E_5 of $\langle C_5, R_5 \rangle$ for each instance of $\langle C_5, R_5 \rangle$ in the prove phase. Given this extractor, it suffices to show the following.

- 1. The extraction succeeds in most instances of $\langle C_5, R_5 \rangle$.
- 2. Any cheating committer cannot prove false statements on the extracted value in most instances of $\langle C_5, R_5 \rangle$.
- 3. The values extracted from most instances of $\langle C_5, R_5 \rangle$ are equal.

The first two can be shown by following the analysis of E_5 (Lemma I.3). Specifically, the receiver security of OT guarantees that in at least $\lambda - \omega(\log \lambda)$ instances of $\langle C_5, R_5 \rangle$, all the values that are committed by SBCom (e.g., decommitments to the hash of the initial MPC states) are correctly generated with non-negligible probability, and thus we can reuse the analysis of E_5 . The last one can be shown by noticing that if different values, $x_{\text{com}} || r || s$ and $x'_{\text{com}} || r' || s'$ such that $x_{\text{com}} \neq x'_{\text{com}}$, are extracted, then with overwhelming probability over the choice of $\alpha \in G$, it holds $x_n \alpha^n + \dots + x_1 \alpha + r \neq x'_n \alpha^n + \dots + x'_1 \alpha + r'$ (which implies that we have either $x_n \alpha^n + \dots + x_1 \alpha + r \neq a$ or $x'_n \alpha^n + \dots + x'_1 \alpha + r' \neq a$) and hence the cheating committer is required to prove a false statement in an instance of $\langle C_5, R_5 \rangle$ unless it can break the security of two-com.

K Lemmas from Kalai et al. [KRR14] and Subsequent Works

We give several lemmas that are based on those given in Kalai et al. [KRR14] and subsequent works. All the lemmas in this section can be trivially extended for the parallel setting that we consider in Appendix E (i.e., the setting where \mathcal{A} takes multiple sets of queries and outputs multiple functions).

K.1 Lemmas on SelfCorr

For each $\lambda \in \mathbb{N}$, let F, H, m, m_{10} be the parameters of our PCP system (PCP.P, PCP.V), and let D(X), $D(X^{\xi})$, $D(X^{\xi}_{1,in})$ be defined as in Section D.2.2. Let $\zeta = \omega(\log \lambda)$. Let LD-Test and SelfCorr be the algorithms that are defined in Algorithm 6 and Algorithm 1.

Then, there exists a polynomial $\kappa_1 = O(\lambda |F|^2)$ such that for any κ_1 -CNS adversary \mathcal{A} , the following lemmas hold. In the following, unless otherwise specified, we use (d, D) to denote any of $(m|H|, D(X)), ((m-1)|H|, D(X^{\xi})),$ and $(m_{10}|H|, D(X^{\xi}_{1 \text{ in}})).$

Lemma K.1 (Correctness of SelfCorr). If it holds

$$\Pr\left[b = 1 \mid b \leftarrow \mathsf{LD-Test}_{d,D,\zeta}^{\mathcal{A}}\right] \ge 1 - \mathsf{negl}(\lambda)$$

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of such λ), then there exists a negligible function ϵ such that for every sufficiently large $\lambda \in \Lambda$ and $\forall q \in D$, it holds

$$\Pr\left[\tilde{A}(q) = \bot \;\middle|\; (\mathsf{out}, \tilde{A}) \leftarrow \mathsf{SelfCorr}_{d,D}^{\mathcal{A}}(\{q\})\right] \leq \epsilon(\lambda) \;\; .$$

Proof. See [KRR13, Theorem 7.27] or [BHK16, Lemma 8].

Lemma K.2 (Consistency of SelfCorr). If it holds

$$\Pr\left[b=1 \mid b \leftarrow \mathsf{LD-Test}_{d,D,\zeta}^{\mathcal{A}}\right] \ge 1 - \mathsf{negl}(\lambda)$$
,

for infinitely many $\lambda \in \mathbb{N}$ (let Λ be the set of such λ), then there exists a negligible function ϵ such that for every sufficiently large $\lambda \in \Lambda$ and $\forall q \in D$, it holds

$$\Pr \begin{bmatrix} \tilde{A}_0(q) \neq \tilde{A}_1(q) & (Q_b, \operatorname{st}_b) \leftarrow \operatorname{SelfCorr}. \mathsf{Q}_D(\{q\}) \ \textit{for} \ \forall b \in \{0, 1\} \\ (\operatorname{out}, A) \leftarrow \mathcal{A}(Q_0 \cup Q_1) \\ \tilde{A}_b \coloneqq \operatorname{SelfCorr}. \operatorname{Rec}_d(\operatorname{st}_b, A) \ \textit{for} \ \forall b \in \{0, 1\} \end{bmatrix} \leq \epsilon(\lambda)$$

Proof sketch. This can be proven by slightly modifying the proofs of [KRR13, Theorem 7.27] and [BHK16, Lemma 8]. Specifically, by inspection, one can see that these proofs consider, very roughly speaking, a modified version of SelfCorr that chooses 2λ lines $L_{0,1}, \ldots, L_{0,\lambda}, L_{1,1}, \ldots, L_{1,\lambda}$ for each query (instead of choosing λ lines), and then show that

- 1. the same value c_i is recovered by self-correction from $L_{0,i}$ and $L_{1,i}$ for most $i \in [\lambda]$, and
- 2. there exists a single value c such that it holds $c_i = c$ for most $i \in [\lambda]$.

Hence, it follows that the same value is recovered by self-correction from most of the 2λ lines, and thus, when those 2λ lines are considered to be chosen from two instances of SelfCorr, the values that are recovered from them are the same. (For a formal argument, see [Kiy18, Claim 3].)

Lemma K.3. For each $\lambda \in \mathbb{N}$, let $(m_0, d_0, D_0) := (m, m|\boldsymbol{H}|, D(X))$ and let (m_1, d_1, D_1) be either $(m - 1, (m - 1)|\boldsymbol{H}|, D(X^{\xi}))$ or $(m_{to}, m_{to}|\boldsymbol{H}|, D(X^{\xi}_{1,in}))$.

Assume that the following hold for infinitely many $\lambda \in \mathbb{N}$ and let Λ be the set of such λ .

• Low-degree Test on D_b ($b \in \{0, 1\}$): For $b \in \{0, 1\}$, it holds

$$\Pr\left[b = 1 \mid b \leftarrow \mathsf{LD-Test}_{d_b, D_b, \mathcal{L}}^{\mathcal{A}}\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

- Low-degree Test, conditioned on $L(0) \in D_1$: It holds b = 1 with probability at least $1 \text{negl}(\lambda)$ in the following probabilistic experiment.
 - 1. Choose λ random lines $L_1, \ldots, L_{\lambda} : \mathbf{F} \to D_0$ such that each $L \in \{L_1, \ldots, L_{\lambda}\}$ satisfies $L(0) \in D_1$.
 - 2. $Run(out, A) \leftarrow \mathcal{A}(Q)$, where $Q = \{L_i(t)\}_{i \in [\lambda], t \in F}$.

³⁶Note that κ_1 is a polynomial in λ and |F| = polylogN. Note that when N is expressed as a polynomial in λ , κ_1 is a polynomial in λ ; cf. Footnote 22, Footnote 24.

3. Let b = 1 if and only if

$$\left|\left\{j \in [\lambda] \mid \mathsf{isLD}_{d_0}\left(\left\{A(L_j(t))\right\}_{t \in F}\right) = 1\right\}\right| \ge \lambda - \zeta \ .$$

- D_1 -parallel Low-degree Test: It holds b = 1 with probability at least $1 \text{negl}(\lambda)$ in the following probabilistic experiment.
 - 1. Choose λ random lines $L_1, \ldots, L_{\lambda} : \mathbf{F} \to D_0$ as follows: for each $j \in [\lambda]$, choose random points $\mathbf{r} \in D_0 = \mathbf{F}^{m_0}$ and $\mathbf{r}' \in \{(0, \cdots, 0, v_{m_0 m_1 + 1}, \ldots, v_{m_0}) \mid \forall (v_{m_0 m_1 + 1}, \ldots, v_{m_0}) \in \mathbf{F}^{m_1}\} \in D_0 = \mathbf{F}^{m_0}$, and define $L_j : \mathbf{F} \to D$ as $L_j(\alpha) = \mathbf{r} + \alpha \cdot \mathbf{r}'$.
 - 2. Run (out, A) $\leftarrow \mathcal{A}(Q)$, where $Q = \{L_i(t)\}_{i \in [\lambda], t \in F}$.
 - 3. Let b = 1 if and only if

$$\left|\left\{j\in[\lambda]\;\middle|\;\mathrm{isLD}_{d_1}\left(\left\{A(L_j(t))\right\}_{t\in F}\right)=1\right\}\right|\geq\lambda-\zeta\ .$$

Then, there exists a negligible function ϵ such that for every sufficiently large $\lambda \in \Lambda$, and for $\forall q \in D_1$, it holds

$$\Pr \begin{bmatrix} \tilde{A}_0(q) \neq \tilde{A}_1(q) & (Q_b, \operatorname{st}_b) \leftarrow \operatorname{SelfCorr}. \mathsf{Q}_{D_b}(\{q\}) \ \textit{for} \ \forall b \in \{0, 1\} \\ (\operatorname{out}, A) \leftarrow \mathcal{A}(Q_0 \cup Q_1) \\ \tilde{A}_b \coloneqq \operatorname{SelfCorr}. \operatorname{Rec}_d(\operatorname{st}_b, A) \ \textit{for} \ \forall b \in \{0, 1\} \end{bmatrix} \leq \epsilon(\lambda) \enspace .$$

Proof sketch. This lemma can be proven in a similar spirit to [HR18, Proposition 10.12]. We give a proof sketch for completeness.

From the CNS of \mathcal{A} , it suffices to show $\Pr[\tilde{A}_0(q) \neq \tilde{A}_1(q)] \leq \epsilon(\lambda)$ in the following probabilistic experiment.

- 1. Choose λ random planes $M_1, \ldots, M_{\lambda} : F^2 \to D_0$ such that each $M \in \{M_1, \ldots, M_{\lambda}\}$ satisfies the following: (1) M(0,0) = q; (2) the line $M(\cdot,0)$ is fully contained in D_1 .
- 2. Run (out, A) $\leftarrow \mathcal{A}(Q)$, where $Q = \{M_j(t, t')\}_{j \in [\lambda], t, t' \in F}$.
- 3. Let $\tilde{A}_b := \mathsf{SelfCorr}.\mathsf{Rec}_d(\mathsf{st}_b, A)$ for $\forall b \in \{0, 1\}$, where $\mathsf{st}_0 := (\{q\}, \{M_i(0, \cdot)\}_{i \in [\lambda]})$ and $\mathsf{st}_1 := (\{q\}, \{M_i(\cdot, 0)\}_{i \in [\lambda]})$.

(To see that it indeed suffices to bound the probability in the above experiment, observe that the lines $M_1(0,\cdot),\ldots,M_{\lambda}(0,\cdot)$ are random lines on D_0 such that each $L\in\{M_1(0,\cdot),\ldots,M_{\lambda}(0,\cdot)\}$ satisfies L(0)=q, and the lines $M_1(\cdot,0),\ldots,M_{\lambda}(\cdot,0)$ are random lines on D_1 such that each $L\in\{M_1(\cdot,0),\ldots,M_{\lambda}(\cdot,0)\}$ satisfies L(0)=q.)

Hence, we focus on showing $\Pr[\tilde{A}_0(q) \neq \tilde{A}_1(q)] \leq \epsilon(\lambda)$ in the above experiment. First, from "Low-degree Test, conditioned on $L(0) \in D_1$," it follows that for each $\alpha \in F \setminus \{0\}$, we have $\Pr[|J_{0,\alpha}| \geq \lambda - \zeta] \geq 1 - \mathsf{negl}(\lambda)$, where

$$J_{0,\alpha} := \left\{ j \in [\lambda] \; \middle|\; \mathsf{isLD}_{d_0}\left(\left\{A(M_j(\alpha,t))\right\}_{t \in F}\right) = 1 \right\} \;\; .$$

Second, from " D_1 -parallel Low-degree Test," it follows that for each $\beta \in \mathbf{F} \setminus \{0\}$, we have $\Pr\left[|J_{1,\beta}| \geq \lambda - \zeta\right] \geq 1 - \mathsf{negl}(\lambda)$, where

$$J_{1,\beta} \coloneqq \left\{ j \in [\lambda] \mid \mathsf{isLD}_{d_1}\left(\left\{A(M_j(t,\beta))\right\}_{t \in F}\right) = 1 \right\} \ .$$

Thus, using a union bound, we obtain $\Pr[|J_{\text{good}}| \ge \lambda - 2|F|\zeta] \ge 1 - \text{negl}(\lambda)$, where

$$J_{\text{good}} := \bigcap_{b \in \{0,1\}, t \in F \setminus \{0\}} J_{b,t}$$

Also, it is easy to observe that for every $j \in J_{good}$, there exists c_i such that

$$\mathsf{Recon}_{d_0}\left(\left\{A(M_j(0,t))\right\}_{t\in F\setminus\{0\}}\right) = \mathsf{Recon}_{d_1}\left(\left\{A(M_j(t,0))\right\}_{t\in F\setminus\{0\}}\right) = c_j \ .$$

(See, e.g., [KRR13, Proposition 7.23].) Now, note that Lemma K.1 implies that with overwhelming probability, there exist c'_0, c'_1 such that

$$\left|\left\{j \in [\lambda] \;\middle|\; \mathsf{Recon}_{d_0}\left(\left\{A(M_j(0,t))\right\}_{t \in F\setminus\{0\}}\right) = c_0'\right\}\right| \geq 0.9\lambda$$

and

$$\left|\left\{j \in [\lambda] \;\middle|\; \mathsf{Recon}_{d_1}\left(\left\{A(M_j(t,0))\right\}_{t \in F \setminus \{0\}}\right) = c_1'\right\}\right| \geq 0.9\lambda \ .$$

Using a union bound and the fact that $0.8\lambda - 2|F|\zeta > 0$ for sufficiently large λ , we obtain that with overwhelming probability, there exists j such that $c_0' = c_j = c_1'$. From the construction of SelfCorr, this implies that $\Pr[\tilde{A}_0(q) \neq \tilde{A}_1(q)] \leq \epsilon(\lambda)$, as desired.

K.2 Soundness Amplification Lemma

We give a slightly extended version of the soundness amplification lemma of Brakerski et al. [BHK17].

Lemma K.4 (Soundness Amplification Lemma). For any polynomials κ , κ_{max} , any κ -query verifier V = (Q, D), any c > 0, and the PPT oracle algorithm Amplify_c in Algorithm 22, the following holds. For any $\zeta(\lambda) = \omega(\log \lambda)$ and $k \in \{0, 1, ..., \}$, if there exists an adaptive κ_{max} -CNS adversary \mathcal{A} such that it holds

$$\Pr\left[\mathsf{D}^{\geq \lambda - k\zeta}(\mathsf{st},\mathsf{out},A) = 1 \,\middle| \, \begin{array}{c} (Q,\mathsf{st}) \leftarrow \mathsf{Q}^{\otimes \lambda}(1^{\lambda}) \\ (\mathsf{out},A) \leftarrow \mathcal{A}(Q) \end{array} \right] \geq \frac{1}{\lambda^c} \tag{K.1}$$

for infinitely many λ (let Λ be the set of such λ), then Amplify $^{\mathcal{A}}(1^{\lambda},\cdot)$ is an adaptive $(\kappa_{\max} - \lambda \kappa)$ -CNS cheating prover such that there is a negligible function negl such that for every $\lambda \in \Lambda$,

$$\Pr\left[\mathsf{D}^{\geq \lambda - (2k+1)\zeta}(\mathsf{st},\mathsf{out},A) = 1 \, \middle| \, \begin{array}{c} (Q,\mathsf{st}) \leftarrow \mathsf{Q}^{\otimes \lambda}(1^\lambda) \\ (\mathsf{out},A) \leftarrow \mathsf{Amplify}_c^{\mathcal{A}}(1^\lambda,Q) \end{array} \right] \geq 1 - \mathsf{negl}(\lambda) \ . \tag{K.2}$$

Furthermore, for any sequence of queries $\{S_{\lambda}\}_{{\lambda}\in\mathbb{N}}$ such that $|S_{\lambda}| \leq \kappa_{\max} - 2\lambda \kappa$, the following two distributions on $(\text{out}, A|_{S_{\lambda}})$ are computationally indistinguishable.

1. (out, $A|_{S_\lambda}$) is sampled through the conditional distribution

$$\begin{array}{c|c} (Q,\mathsf{st}) \leftarrow \mathsf{Q}^{\otimes \lambda}(1^{\lambda}) \\ (\mathsf{out},A) \leftarrow \mathcal{A}(Q \cup S_{\lambda}) \end{array} \middle| \mathsf{D}^{\geq \lambda - k\zeta}(\mathsf{st},\mathsf{out},A) = 1 \ . \tag{K.3}$$

2. (out, $A|_{S_A}$) is sampled through the distribution

$$\begin{aligned} &(Q,\mathsf{st}) \leftarrow \mathsf{Q}^{\otimes \lambda}(1^{\lambda}) \\ &(\mathsf{out},A) \leftarrow \mathsf{Amplify}_c^{\mathcal{A}}(1^{\lambda},Q \cup S_{\lambda}) \ . \end{aligned}$$
 (K.4)

Algorithm 22 Amplify $^{\mathcal{A}}(1^{\lambda}, Q)$

- 1. Run $(O_i, st_i) \leftarrow \mathbb{Q}^{\otimes \lambda}$ and $(out_i, A_i) \leftarrow \mathcal{A}(O \cup O_i)$ for each $i \in [\lambda^{c+1}]$.
- 2. Find the first $i^* \in [\lambda^{c+1}]$ such that $\mathsf{D}^{\geq \lambda k\zeta}(\mathsf{st}_{i^*}, \mathsf{out}, A_{i^*}|_{Q_{i^*}}) = 1$, and output $(\mathsf{out}_{i^*}, A_{i^*}|_Q)$ if such i^* exists, and output \bot otherwise.

The above version is extended from the one by Brakerski et al. [BHK17] in that (1) we consider soundness with $D^{\geq \lambda - k\zeta}$ and $D^{\geq \lambda - (2k+1)\zeta}$ rather than soundness with $D^{\otimes \lambda}$ and $D^{\geq \lambda - \zeta}$, and (2) in the "furthermore" part, we consider the distribution on (out, $A|_{S_{\lambda}}$) rather than on out.

Proof sketch. Since the proof is a straightforward extension of the one by Brakerski et al. [BHK17], we only give a proof sketch.

Toward proving this lemma, we consider three claims.

Claim K.1. Amplify $^{\mathcal{H}}(1^{\lambda}, \cdot)$ is an adaptive $(\kappa_{\text{max}} - \lambda \kappa)$ -CNS cheating prover.

Proof. This can be proven in exactly the same way as in Brakerski et al. [BHK17].

Claim K.2. Equation (K.2) holds for every $\lambda \in \Lambda$.

Proof. We first note that in exactly the same way as Brakerski et al. [BHK17], we can show that the probability that $Assign_c$ outputs \bot is negligible.

Thus, we focus showing Equation (K.2) under the condition that Assign_c does not output \bot , i.e., under the condition that a "good" i^* exists in the execution of Assign_c . In this case, it suffices to show that the conditional probability

$$\Pr\left[\mathsf{D}^{\lambda-(2k+1)\zeta}(\mathsf{st},\mathsf{out}_{i^*},A_{i^*}|_{\mathcal{Q}}) = 0 \;\middle|\; \mathsf{D}^{\lambda-k\zeta}(\mathsf{st}_{i^*},\mathsf{out}_{i^*},A_{i^*}|_{\mathcal{Q}_{i^*}}) = 1\right] \tag{K.5}$$

is negligible, where the probability is taken over $(Q, st) \leftarrow Q^{\otimes \lambda}$, $(Q_{i^*}, st_{i^*}) \leftarrow Q^{\otimes \lambda}$, and $(out_{i^*}, A_{i^*}) \leftarrow \mathcal{A}(Q \cup Q_{i^*})$. First, from the CNS of \mathcal{A} and Equation (K.1), we have

$$\Pr\left[\mathsf{D}^{\lambda-k\zeta}(\mathsf{st}_{i^*},\mathsf{out}_{i^*},A_{i^*}|_{\mathcal{Q}_{i^*}})=1\right] \geq \frac{1}{\lambda^c} - \mathsf{negl}(\lambda) \ . \tag{K.6}$$

Next, we have

$$\Pr\left[\mathsf{D}^{\lambda-(2k+1)\zeta}(\mathsf{st},\mathsf{out}_{i^*},A_{i^*}|_{\mathcal{Q}}) = 0 \land \mathsf{D}^{\lambda-k\zeta}(\mathsf{st}_{i^*},\mathsf{out}_{i^*},A_{i^*}|_{\mathcal{Q}_{i^*}}) = 1\right] \le \mathsf{negl}(\lambda) \ . \tag{K.7}$$

This can be obtained from the following observations.

- We consider a mental experiment where, instead of sampling Q and Q_{i^*} individually, we first sample \hat{Q} of size $|Q| + |Q_{i^*}|$, next separate it into λ pairs $\{(\hat{Q}_{j,0}, \hat{Q}_{j,1})\}_{j \in [\lambda]}$, and then for each $j \in [\lambda]$, pick one of $\hat{Q}_{j,0}$ and $\hat{Q}_{j,1}$ for Q and the other for Q_{i^*} . Fix \hat{Q} and $\{(\hat{Q}_{j,0}, \hat{Q}_{j,1})\}_{j \in [\lambda]}$ (but not Q or Q_{i^*} individually), and further fix \mathcal{A} 's response on \hat{Q} . Let N be the number of \hat{J} 's such that \mathcal{A} gives a rejecting answer to at least one of $\hat{Q}_{j,0}$ and $\hat{Q}_{j,1}$.
- Case 1. $N \ge (2k+1)\zeta$: In this case, we have $\mathsf{D}^{\lambda-k\zeta}(\mathsf{st}_{i^*},\mathsf{out}_{i^*},A_{i^*}|_{Q_{i^*}})=1$ only when at most $k\zeta$ rejecting queries are picked for Q_{i^*} in the N bad pairs, but Chernoff Bound implies that the probability that this occurs is at most

$$\exp\left(-\frac{1}{2}\delta^2 E\right) = \exp\left(-\frac{1}{8(k+\frac{1}{2})}\zeta\right) = \mathsf{negl}(\lambda)$$
,

where $\delta = \frac{1}{2}/(k+\frac{1}{2})$ and $E = (k+\frac{1}{2})\zeta$.

• Case 2. $N < (2k+1)\zeta$: In this case, we never have $\mathsf{D}^{\lambda-(2k+1)\zeta}(\mathsf{st},\mathsf{out}_{i^*},A_{i^*}|_O) = 0$.

Combining Equation (K.6) and (K.7), we have that the probability in Equation (K.5) is negligible, as desired.

Claim K.3. The "furthermore" part of the lemma holds.

Proof. For any $\{S_{\lambda}\}_{\lambda \in \mathbb{N}}$, consider a modified version of Amplify_c, denoted by Amplify'_c, that samples each (out_i, A_i) from $\mathcal{A}(S_{\lambda} \cup Q_i)$ instead of from $\mathcal{A}(Q \cup S_{\lambda} \cup Q_i)$. Now, the distribution on $(\text{out}, A|_{S_{\lambda}})$ that is sampled through the distribution (K.3) is identically distributed with the one that is sampled through the conditional distribution

$$\begin{array}{c|c} (Q,\operatorname{st}) \leftarrow \mathsf{Q}^{\otimes \lambda}(1^{\lambda}) \\ (\operatorname{out},A) \leftarrow \operatorname{Amplify'}_c^{\mathcal{A}}(1^{\lambda},Q \cup S_{\lambda}) \end{array} \ \text{ Amplify'}_c^{\mathcal{A}} \ \operatorname{does \ not \ output} \ \bot \ \ .$$

Furthermore, since $\mathsf{Amplify'}^{\mathcal{A}}_c$ outputs \bot only with negligible probability, this distribution is statistically close to the one that is sampled through

$$\begin{split} &(Q,\mathsf{st}) \leftarrow \mathsf{Q}^{\otimes \lambda}(1^{\lambda}) \\ &(\mathsf{out},A) \leftarrow \mathsf{Amplify'}^{\mathcal{A}}_{c}(1^{\lambda},Q \cup S_{\lambda}) \end{split} \ .$$

Finally, due to the CNS of \mathcal{A} , this distribution is computationally indistinguishable from the one that is sampled through the distribution (K.4)

This completes the proof sketch of Lemma K.4.