Concurrent Non-Malleable Commitments from One-way Functions

Huijia Lin* Rafael Pass† Muthuramakrishnan Venkitasubramaniam‡

December 4, 2007

Abstract

We show the existence of concurrent non-malleable commitments based on the existence one-way functions. Our proof of security only requires the use of black-box techniques, and additionally provides an arguably simplified proof of the existence of even stand-alone secure non-malleable commitments.

*Cornell University, E-Mail: huijia@cs.cornell.edu
†Cornell University, E-Mail: rafael@cs.cornell.edu
‡Cornell University, E-Mail: vmuthu@cs.cornell.edu

1 Introduction

Often described as the "digital" analogue of sealed envelopes, commitment schemes enable a *sender* to commit itself to a value while keeping it secret from the *receiver*. For some applications, however, the most basic security guarantees of commitments are not sufficient. For instance, the basic definition of commitments does rule out an attack where an adversary, upon seeing a commitment to a specific value v, is able to commit to a related value (say, $v-1$), even though it does not know the actual value of v. This kind of attack might have devastating consequences if the underlying application relies on the *independence* of committed values (e.g., consider a case in which the commitment scheme is used for securely implementing a contract bidding mechanism). The state of affairs is even worsened by the fact that many of the known commitment schemes are actually susceptible to this kind of attack. In order to address the above concerns, Dolev, Dwork and Naor (DDN) introduced the concept of *non-malleable commitments* [6]. Loosely speaking, a commitment scheme is said to be non-malleable if it is infeasible for an adversary to "maul" a commitment to a value v into a commitment of a related value $\tilde{v}$.

The first non-malleable commitment protocol was constructed by Dolev, Dwork and Naor [6]. The security of their protocol relies on the existence of one-way functions and requires $O(\log n)$ rounds of interaction, where $n \in N$ is the length of party identifiers (or alternatively, a security parameter). A more recent result by Barak presents a constant-round protocol for non-malleable commitments whose security relies on the existence of trapdoor permutations and hash functions that are collision-resistant against sub-exponential sized circuits [2]. Even more recently, Pass and Rosen present a constant-round protocol, assuming only collision resistant hash function secure against polynomial sized circuits [13].

1.1 Concurrent Non-Malleable Commitments

The basic definition of non-malleable commitments only considers a scenario in which two executions take place at the same time. A natural extension of this scenario (already suggested in [6]) is one in which more than two invocations of the commitment protocol take place concurrently. In the concurrent scenario, the adversary is receiving commitments to multiple values $v_1, \ldots, v_m$, while attempting to commit to related values $\tilde{v}_1, \ldots, \tilde{v}_m$. As argued in [6], non-malleability with respect to two executions can be shown to guarantee *individual* independence of any $\tilde{v}_i$ from any v_j. However, it does not rule out the possibility of an adversary creating *joint* dependencies between more than a single individual pair (see [6], Section 3.4.1 for an example in the context of non-malleable encryption). Resolving this issue has been stated as a major open problem in [6].

Partially addressing this issue, Pass demonstrated the existence of commitment schemes that remain non-malleable under *bounded concurrent* composition [10]. That is, for any (predetermined) polynomial $p(\cdot)$, there exists a non-malleable commitment that remains secure as long as it is not executed more than $p(n)$ times, where $n \in N$ is a security parameter. More recently, Pass and Rosen [13] constructed of a commitment scheme that remains non-malleable also under an unbounded number of concurrent executions. Their construction uses only a constant number of rounds and is based on the existence of (certified) claw-free permutations. The protocol—which is a variant of the protocol of of [12]—relies on the message-length technique of [10], which in turn relies on the non-black box zero-knowledge protocol of Barak [1]. As such, it seems that practical implementations of this approach currently are not within reach.

In contrast, the original construction of Dolev, Dwork and Naor (which is only stand-alone secure) relied on the minimal assumption of one-way functions and had a black-box security proof. Natural questions left open are thus:

Can concurrent non-malleable commitments be based solely on the existence of one-way functions?

Does there exist concurrent non-malleable commitments with black-box proofs of security?

A partial answer to the second question was provided by Pass and Vaikuntanathan[**?**], demonstrating the existence of concurrent non-malleable commitments with black-box security proofs; their construction, however, relies on a new (and non-standard) hardness assumption with a strong non-malleability flavor.[1]

1.2 Our Results

In this work, we fully resolve both of the above questions. Namely, we show the following theorem using only black-box techniques.

Main Theorem *If one-way functions exist, then there exists a statistically-binding commitment scheme that is concurrently non malleable.*

Our protocol, which is a variant of the protocol of [6] (and in particular relies on the same scheduling techniques as in [6]), uses $O(n)$ number of communication rounds.[2] Moreover, it seems that by relying on specific (number theoretic) hardness assumptions (and appropriate Σ-protocols [4]), one can obtain an "implementable" instantiation of our protocol (without going through Cook's reductions).

Additional results All previous constructions of non-malleable commitments require complex and subtle proofs. As an additional contribution, our protocol and its proof provide the arguably simplest proof of existence of non-malleable commitments (let alone the question of concurrency); more precisely, it provides a new (and arguably simpler) proof of the feasibility result of [6].

Furthermore, by relying on the concurrent security of our protocol, we also obtain a simple (and self-contained) proof of the existence of $\log n$-round (stand-alone secure) non-malleable commitments based on only the existence of one-way functions. As far as we know, a complete proof of this statement (which appeared only with a proof sketch in [6]) has never appeared before.

Finally, we mention that our protocols satisfy a notion of non-malleability called *strict* (as opposed to *liberal*) non-malleability—this notion, which was defined (but not achieved) in [6], requires simulation to be performed by a strict polynomial-time machine (as opposed to an expected polynomial-time machine). Our results provide the first construction of strictly non-malleable commitments based on one-way functions, or using a black-box security proof.

2 Definitions

2.1 Commitment Schemes

Commitment schemes are used to enable a party, known as the *sender*, to commit itself to a value while keeping it secret from the *receiver* (this property is called hiding). Furthermore, the

[1] More precisely, they assume the existence of, so called, *adaptive one-way permutations*—namely permutations which remain one-way even when the adversary has access to an inversion oracle.

[2] Although we haven't checked the details, it seems likely that our proof could also be applied to show that the n-round protocol of [6] is concurrently non-malleable.

commitment is binding, and thus in a later stage when the commitment is opened, it is guaranteed that the "opening" can yield only a single value determined in the committing phase. In this work, we consider commitment schemes that are statistically-binding, namely while the hiding property only holds against computationally bounded (non-uniform) adversaries, the binding property is required to hold against unbounded adversaries. More precisely, a pair of PPT machines (C, R) is said to be a commitment scheme if the following two properties hold.

Computationally hiding: For every (expected) PPT machine R^*, it holds that, the following ensembles are computationally indistinguishable over $n \in N$.

- $\{\mathsf{sta}^{R^*}_{(C,R)}(v_1, z)\}_{n\in N, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*}$
- $\{\mathsf{sta}^{R^*}_{(C,R)}(v_2, z)\}_{n\in N, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*}$

where $\mathsf{sta}^{R^*}_{\mathsf{com}}(v, z)$ denotes the random variable describing the output of R^* after receiving a commitment to v.

Statistically binding: Informally, the statistical-binding property asserts that, with overwhelming probability over the coin-tosses of the receiver R, the transcript of the interaction fully determines the value committed to by the sender. We refer [8] for more details.

2.2 Concurrent Non-Malleable Commitments

Our definition of concurrent non-malleable commitments is very similar to that of [11], but different in two aspects: first, our definition of non-malleability is w.r.t identities (in analogy with DDN [6])[3]; second, our definition considers not only the values the adversary commits to, but also the view of the adversary[4]. Let $\langle C, R\rangle$ be a commitment scheme, and let $n \in N$ be a security parameter. Consider man-in-the-middle adversaries that are participating in left and right interactions in which $m = \text{poly}(n)$ commitments take place. We compare between a *man-in-the-middle* and a *simulated* execution. In the man-in-the-middle execution, the adversary A is simultaneously participating in m left and right interactions. In the left interactions the man-in-the-middle adversary A interacts with C receiving commitments to values $v_1, \ldots, v_m$, using identities $\mathsf{id}_1, \ldots, \mathsf{id}_m$ of its choice. In the right interaction A interacts with R attempting to commit to a sequence of related values $\tilde{v}_1, \ldots, \tilde{v}_m$, again using identities of its choice $\tilde{\mathsf{id}}_1, \ldots, \tilde{\mathsf{id}}_m$. If any of the right commitments are invalid, or undefined, its value is set to $\perp$. For any i such that $\tilde{\mathsf{id}}_i = \mathsf{id}_j$ for some j, set $\tilde{v}_i = \perp$—i.e., any commitment where the adversary uses the same identity as one of the honest committers is considered invalid. Let $\mathsf{mim}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)$ denote a random variable that describes the values $\tilde{v}_1, \ldots, \tilde{v}_m$ and the view of A, in the above experiment.

In the simulated execution a simulator S directly interacts with R. Let $\mathsf{sim}^S_{\langle C,R\rangle}(1^n, z)$ denote the random variable describing the values $\tilde{v}_1, \ldots, \tilde{v}_m$ committed to by S, and the output *view* of S; again, whenever *view* contains a right interaction i where the identity is the same as any of the left interactions, $\tilde{v}_i$ is set to $\perp$.

[3] That it, we disallow even copying of commitment as long as the adversary uses a different identity (than all the committers he receives commitments from). In contrast, [11] defined non-malleability w.r.t content; i.e., the adversary allowed copy commitments. This difference is inconsequential as any commitment non-malleable w.r.t content can be turned into one that is non-malleable w.r.t identities, and vice versa.

[4] The second point is particularly important for showing that one-many non-malleability implies many-many non-malleability. See proposition 2.2.

Definition 1. *A commitment scheme $\langle C, R\rangle$ is said to be* concurrent non-malleable (with respect to commitment) *if for every polynomial $p(\cdot)$, and every probabilistic polynomial-time man-in-the-middle adversary A that participates in at most $m = p(n)$ concurrent executions, there exists a probabilistic polynomial time simulator S such that the following ensembles are computationally indistinguishable over $n \in N$:*

$$\left\{\mathsf{mim}^A_{\mathsf{com}}(v_1, \ldots, v_m, z)\right\}_{n\in N, v_1,\ldots,v_m\in\{0,1\}^n, z\in\{0,1\}^*}$$

$$\left\{\mathsf{sim}^S_{\langle C,R\rangle}(1^n, z)\right\}_{n\in N, v_1,\ldots,v_m\in\{0,1\}^n, z\in\{0,1\}^*}$$

We also consider relaxed notions of concurrent non-malleability: one-many, many-one and one-one secure non-malleable commitments. In a one-one (i.e., a stand-alone secure) non-malleable commitment, we consider only adversaries A that participate in one left and one right interaction; in one-many, A participates in one left and many right, and in many-one, A participates in many left and one right.

Dolev, Dwork and Naor [6] argued that one-one commitments are also many-one secure. Pass and Rosen [11] additionally showed that one-many non-malleability implies (many-many) concurrent non-malleability. [, if the commitment protocol is "natural". Slightly different from the definition by Pass and Rosen, under our definition, any protocol that is one-many non-malleability is also concurrently non-malleable.] Namely, R***

Proposition. *Let $\langle C, R\rangle$ be a one-many concurrent non-malleable commitment. Then, $\langle C, R\rangle$ is also a concurrent non-malleable commitment.*

The proof follows using a standard hybrid argument and is reproduced for completeness in the Appendix.[5]

2.3 Other primitives

We informally define the other primitives we use in the construction of our protocols.

Witness-indistinguishable proofs An interactive proof is said to be *witness indistinguishable* ($\mathcal{WI}$) if the Verifier's view is "computationally independent" of the witness used by the Prover for proving the statement—i.e. the view of the Verifier in the interaction with a Prover using witness w_1 or w_2 for two different witnesses are indistinguishable.

Special-sound proofs: A 3-round public-coin interactive proof for the language $L \in \mathcal{NP}$ with witness relation R_L is special-sound with respect to R_L, if for any two transcripts (α, β, γ) and $(\alpha', \beta', \gamma')$ such that the initial messages α, α' are the same but the challenges β, β' are different, there is a deterministic procedure to extract the witness from the two transcripts that runs in polynomial time. Special-sound $\mathcal{WI}$ proofs for languages in $\mathcal{NP}$ can be based on the existence of non-interactive commitment schemes, which in turn can be based on one-way permutations. Assuming only one-way functions, 4-round special-sound $\mathcal{WI}$ proofs for NP exists.[6] For simplicity, we use 3-round special-sound proofs in our protocol though our proof works also with 4-round proofs.

[5] Our proof is actually a bit different than the proof of [11] due to the difference in the definition of non-malleability.

[6] A 4-round protocol is special sound if a witness can be extracted from any two transcripts $(\tau, \alpha, \beta, \gamma)$ and $(\tau', \alpha', \beta', \gamma')$ such that $\tau = \tau$,$\alpha = \alpha'$ and $\beta \neq \beta'$.

3 The Protocol

Our protocol is based on Feige-Shamir's zero-knowledge protocol [7] while relying on the message scheduling technique of Dolev, Dwork and Naor [6]. For simplicity of exposition, our description below relies on the existence of one-way functions with efficiently recognizable range, but the protocol can be easily modified to work with any arbitrary one-way function (by simply providing a witness hiding proof that an element is in the range of the one-way function). The protocol proceeds in the following three stages on common input the identity $\mathsf{id} \in \{0,1\}^l$ of the the committer, and security parameter n.

1. In Stage 1, the Receiver picks a random string $r \in \{0,1\}^k$, and sends its image $s = f(r)$ through a one-way function f with an efficiently recognizable range to the Committer. The Committer checks that s is in the range of f and aborts otherwise.

2. In Stage 2, the Committer sends $c = \mathsf{com}(v)$, where $\mathsf{com}(\cdot)$ is any commitment scheme that is statistically-binding.

3. In Stage 3, the Committer proves that c is a valid commitment for v or s is in the image set of f. This is proved by $4l$ invocations of a special-sound $\mathcal{WI}$ proof where the messages are scheduled based on the id (very similar to the scheduling presented in [6]). More precisely, there are l rounds, where in round i, the schedule $\mathsf{design}_{\mathsf{id}_i}$ is followed by $\mathsf{design}_{1-\mathsf{id}_i}$ (See Table 1).

Let com be a statistically-binding commitment scheme

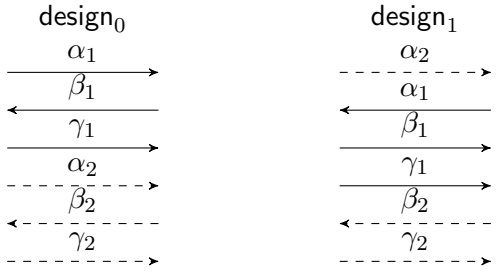

Table 1: Description of the schedules used in Stage 3 of the protocol

Claim 1. *$\langle C, R\rangle$ is a statistically-binding commitment scheme.*

Proof. We show that the $\langle C, R\rangle$ scheme satisfies the binding and hiding properties.

Binding: The binding property follows directly from the binding property of com.

Hiding: The hiding property essentially follows from the hiding property of com and the fact that Stage 3 of the protocol is $\mathcal{WI}$ (since $\mathcal{WI}$ proofs are closed under concurrent composition[7]). For completeness, we give the proof of hiding below.

We show that any adversary R^* that violates the hiding property of $\langle C, R\rangle$ can be used to violate the hiding property of com. More precisely, given any adversary R^* (without loss of generality, deterministic), we construct a machine R' which on auxiliary-input a "fake"-witness r such that $s = f(r)$, where s is the first message sent by R^* (Note that since R^* is deterministic, the string s is fixed), internally incorporates R^*, forwards the external

Protocol ConcNMCom

Common Input: An identifier $\mathsf{id} \in \{0,1\}^l$.

Auxiliary Input for Committer: A string $v \in \{0,1\}^n$.

Stage 1:

R uniformly chooses $r \in \{0,1\}^k$.

R $\rightarrow$ C: $s = f(r)$.

R aborts if s not in the range of f.

Stage 2:

C uniformly chooses $r' \in \{0,1\}^{poly(n)}$.

C $\rightarrow$ R: $c = \mathsf{com}(v, r')$.

Stage 3:

C $\rightarrow$ R: $4l$ special-sound $\mathcal{WI}$ proofs of the statement

either there exists values v, r' s.t $c = \mathsf{com}(v, r')$

or there exists a value r s.t $s = f(r)$

with verifier query of length $2n$, in the following schedule:

For $j = 1$ to l do: Execute $\mathsf{design}_{\mathsf{id}_j}$ followed by Execute $\mathsf{design}_{1-\mathsf{id}_j}$

Figure 1: Non-Malleable String Commitment Scheme $\langle C, R \rangle$

commitment to R^* and next simulates the rest of the protocol using "fake witness" r. R' finally outputs what R^* outputs. From the $\mathcal{WI}$ property of Stage 3, it follows that R' distinguishes the commitment made using com if R^* distinguishes the commitment made using $\langle C, R \rangle$.

□

4 Proof of Security

Theorem 1. *$\langle C, R \rangle$ is one-many concurrent non-malleable.*

Proof: Let A be a man-in-the-middle adversary that participates in one execution in the left and many executions in the right. We construct a simulator S such that the following ensembles are computationally indistinguishable over $n \in N$.

$$\left\{ \mathsf{mim}^A_{\langle C,R \rangle}(v, z) \right\}_{n \in N, v \in \{0,1\}^n, z \in \{0,1\}^*}$$

$$\left\{ \mathsf{sim}^S_{\langle C,R \rangle}(1^n, z) \right\}_{n \in N, v \in \{0,1\}^n, z \in \{0,1\}^*}$$

The simulator S on input $(1^n, z)$ proceeds as follows. S incorporates $A(z)$ and internally emulates the left interaction by *honestly* committing to the string 0^n. Messages in the right interactions are instead forwarded externally. Finally, S outputs the view of A. We show that the values that S commits to combined with the output view are indistinguishable from the values that A commits to combined with its view. Since S emulates the left interaction by *honestly* committing to 0^n, this

is equivalent to showing that

$$\left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(v,z)\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*} \approx \left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(0^n,z)\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*}$$

We note that in both the experiments ($\mathsf{mim}^{A}_{\langle C,R\rangle}(v,z)$ and $\mathsf{mim}^{A}_{\langle C,R\rangle}(0^n,z)$), the joint view of A and the receivers in the right are *identically* distributed up until the point where $A(z)$ sends its first message in the left interaction. Let $\Gamma(A,z)$ denote the set of all possible joint views τ of A and the receivers in the right, such that $A(z)$ sends its first message in the left interaction directly after receiving the messages in τ.

Therefore, it suffices to show that

$$\left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(v,z) \mid \tau\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*,\tau\in\Gamma(A,z)} \approx \left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(0^n,z) \mid \tau\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*,\tau\in\Gamma(A,z)}$$

where $\left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(v,z) \mid \tau\right\}$ is a probability distribution describing the output of $\mathsf{mim}^{A}_{\langle C,R\rangle}(v,z)$ conditioned on the joint view τ being fed to A and the receivers in the right.

Towards this goal, we define a new commitment scheme $\langle \hat{C}, \hat{R}\rangle$ (much like the adaptor scheme in DDN [6]), which is a variant of $\langle C, R\rangle$ where the receiver can ask for an arbitrary number of special-sound $\mathcal{WI}$ designs in Stage 3. Furthermore, the receiver is allowed to choose the scheduling in each iteration; it sends bit i to choose schedule design_i. Note that any receiver using $\langle \hat{C}, \hat{R}\rangle$ can emulate $\langle C, R\rangle$ by requesting the appropriate designs. Using the same proof as in Claim 1, we get:

Lemma 1. *For every (expected) PPT machine M,*

$$\left\{\mathsf{sta}^{M}_{\langle \hat{C},\hat{R}\rangle}(v,z)\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*} \approx \left\{\mathsf{sta}^{M}_{\langle \hat{C},\hat{R}\rangle}(0^n,z)\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*}$$

We show in Lemma 2 below that for every adversary A, there exists an expected PPT machine R^*, such that for all z, $\tau \in \Gamma(A,z)$, there exists a $z' \in \{0,1\}^*$ such that, $R^*(z')$, on receiving a commitment to v (using $\langle \hat{C}, \hat{R}\rangle$) outputs values *indistinguishable* from the actual values *committed* to by $A(z)$ when receiving commitment to v (using $\langle C, R\rangle$) conditioned on τ. Combining Lemma 1 and Lemma 2, we obtain that the outputs of the two experiments $\{\mathsf{mim}^{A}_{\langle C,R\rangle}(v,z)|\tau\}$ and $\{\mathsf{mim}^{A}_{\langle C,R\rangle}(0^n,z)|\tau\}$ are indistinguishable and this concludes the proof of Theorem 1

Lemma 2. *For every PPT adversary A, there exists an expected PPT adversary R^* and a function $\mathcal{Z} : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that the following ensembles*

- $\left\{z' \leftarrow \mathcal{Z}(z,\tau) : \mathsf{sta}^{R^*}_{\langle \hat{C},\hat{R}\rangle}(v,z')\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*,\tau\in\Gamma(A,z)}$
- $\left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(v,z) \mid \tau\right\}_{n\in N,v\in\{0,1\}^n,z\in\{0,1\}^*,\tau\in\Gamma(A,z)}$

are indistinguishable over $n \in N$.

Proof. We define the function $\mathcal{Z}$ on input z and τ as follows: Let $\tilde{v}_1 \ldots \tilde{v}_\ell$ be the values committed to by $A(z)$ in the right interactions in the view τ. Let $Z(z,\tau) = z\|\tau\|\tilde{v}_1\| \ldots \|\tilde{v}_\ell$. R^* on auxiliary input z', internally incorporates $A(z)$ and proceeds in three phases:

- In the Main Execution Phase, it starts by feeding the joint view τ to A. It emulates $\langle C, R\rangle$ in the left interaction for A using $\langle \hat{C}, \hat{R}\rangle$ by requesting the appropriate design every step. More precisely, if id is the identifier for the left interaction, in round i it requests $\mathsf{design}_{\mathsf{id}_i}$ followed by $\mathsf{design}_{1-\mathsf{id}_i}$ and forwards all messages A sends to the left committer. On the other hand, messages from the right interactions are internally emulated by running the honest receiver strategy R starting from the state in τ.

- In the Rewinding phase, once the Main Execution Phase completes, R^* extracts the committed values in the right interactions. Since R^* knows the values committed to in the joint view τ (given as auxiliary input), R^* needs only to extract the values committed after τ. We show that there are certain *safe-points* for each such right-interaction where R^* can rewind A in the right interaction without "affecting" the left interaction. R^* re-executes A from the *safe-points* until it obtains a second proof transcript (i.e. a challenge β and response γ) for one of the proofs in Stage 3 of the protocol; using the special-sound property R^* extracts the witness. In the unlikely event that the same proof transcript is received twice, R^* halts and outputs fail. Additionally, if the witnesses extracted are not valid decommitments, R^* again halts and outputs fail.
- In the Output Phase, R^* output the committed values.

Furthermore, to simplify our analysis the procedure is cut-off if it runs "too long" (2^n steps) and R^* halts and outputs fail. The formal description of the machine R^* can be found in Figure 4. Below we give the formal definition of safe-points.

Definition 2. *A prefix ρ of a transcript Δ is a* safe-point *for a particular right interaction, if there exists an accepting proof $(\alpha_r, \beta_r, \gamma_r)$ in that interaction in Δ, such that:*

- *α_r occurs in ρ, while β_r (and γ_r) occur after ρ in the transcript Δ.*
- *if any proof $(\alpha_l, \beta_l, \gamma_l)$ in the left interaction in Δ is such that only α_l occurs in ρ, then β_l occurs after γ_r.*

If ρ is a safe-point, let $(\alpha_\rho, \beta_\rho, \gamma_\rho)$ denote the canonical "safe" right proof.

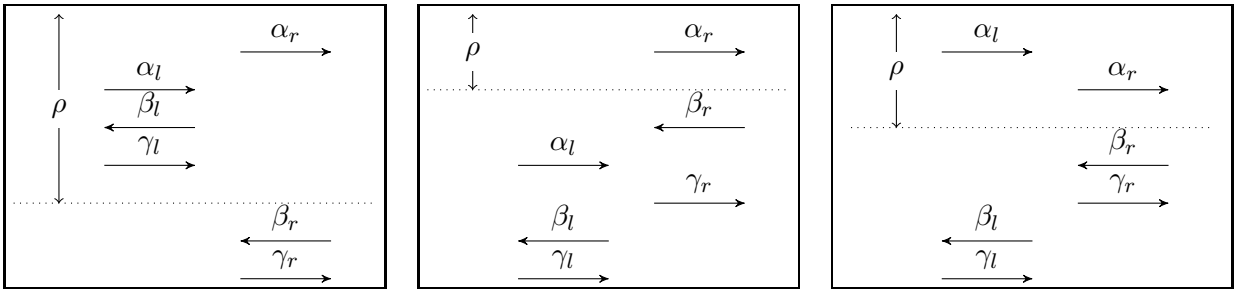

Figure 2: Three characteristic safe-points.

Description of R^*

Input: R^* receives auxiliary input $z' = z\|\tau\|\tilde{v}_1\| \ldots \|\tilde{v}_\ell$.

Procedure: R^* interacts externally as a receiver using $\langle \hat{C}, \hat{R} \rangle$. Internally it incorporates $A(z)$ and emulates a one-many man-in-the-middle execution by simulating all right receivers and emulating the left $\langle C, R \rangle$ interaction by requesting the appropriate designs expected by $A(z)$ using $\langle \hat{C}, \hat{R} \rangle$ from outside.

Main Execution Phase: Feed the view in τ to A and all right receivers. Emulate all the interactions from τ and complete the execution with A. Let Δ be the transcript of messages obtained.

Rewinding Phase: For $k = \ell$ to m, if interaction k is convincing, do:

- In Δ, find the first point ρ that is a safe-point for interaction k; let the associated proof be $(\alpha_\rho, \beta_\rho, \gamma_\rho)$.
- Repeat until a second-proof transcript $(\alpha_\rho, \beta'_\rho, \gamma'_\rho)$ is obtained:
  Emulate the right interaction as in the Main-Execution Phase. For the left interaction:
  - If A expects to get a new proof from the external committer (case (ii) in Figure 2): Emulate the proof, by requesting for design_0 from outside committer. Forward one of the two proofs internally.
  - If A sends a challenge for a proof whose first message occurs in ρ (Figure C): Cancel the execution, rewind to ρ and continue.
- If $\beta_\rho \neq \beta'_\rho$ extract witness w from $(\alpha_\rho, \beta_\rho, \gamma_\rho)$ and $(\alpha_\rho, \beta'_\rho, \gamma'_\rho)$. Otherwise halt and output fail.
- If $w = (v, r)$ is valid commitment for interaction k, i.e. $com(v, r) = c_k$ then set $\hat{v}_k = v$. Otherwise halt and output fail.

Note that, since we start the execution from τ, none of the rewinding can make A request a new commitment from the external committer.

Output Phase: For every interaction k that is not convincing, set $\hat{v}_k = \perp$. Output $(\hat{v}_1, \ldots, \hat{v}_m)$ and the view from the Main Execution Phase.

Finally, if it runs for more than 2^n steps, halt and output fail.

Figure 3: The construction of R^*

We proceed to show that output of R^* is "correctly" distributed and bound its running-time.

Running-time analysis of R^*: We show that R^* is expected PPT. Note that the time spent by R^* in the Main Execution Phase is $poly(n)$ (where n is the security parameter), since A is a strict polynomial time machine. We show below that expected time spent by R^* in the Rewinding Phase is $poly(n)$. We shall assume, just for the analysis that R^* does not check the fail conditions and bound its running time (since this can only increase R^*'s running time).

Recall that in the Rewinding Phase, R^* rewinds A from all safe points. Let $T_k(i)$ be the random variable that describes the time spent in rewinding for interaction k after i messages have been exchanged. We show that $E[T_k(i)] \leq poly(n)$ and then by linearity of expectation, we conclude

that the expected time spent by R^* in the Rewinding phase is

$$\sum_{k=1}^{m}\sum_{i} E[T_k(i)] \leq \sum_{k=1}^{m}\sum_{i} poly(n) \leq poly(n),$$

since at most $poly(n)$ messages are exchanged.

We now proceed to show that $E[T_k(i)]$ is bounded by some $poly(n)$. Given a (partial) transcript of messages ρ, let $\Pr[\rho]$ denote the probability that ρ occurs in the Main Execution phase. Furthermore, let p_ρ denote the probability ρ is a safe-point that will be rewound—i.e. p_ρ is the probability that the right interaction k is convincing and ρ is a safe-point for interaction k.

Recall that R^* rewinds until it finds another transcript for the proof $(\alpha_\rho, \beta_\rho, \gamma_\rho)$ associated with ρ. It cancels every rewinding for which A requests the second message of a proof in the left-interaction whose first message occurs in ρ. We claim that, the probability of cancelling a rewinding from ρ, is at most $1-p_\rho$. This is because every view from ρ that occurs in the Main Execution phase has the same probability of occurring in the rewinding too (the receiver picks uniformly random messages in Stage 3 of the protocol), and ρ is not a safe-point for every rewinding from ρ that is cancelled. Thus, the expected number of rewindings is at most $\frac{1}{p_\rho}$.

Therefore, the expected number of rewindings from ρ is at most $p_\rho \cdot \dfrac{1}{p_\rho} = 1$ and each rewinding takes at most $poly(n)$ steps. Thus,

$$E[T_k(i)] = \sum_{\rho \text{ of length } i} E[T_k(i)|\rho] \Pr[\rho] \leq poly(n) \times \sum_{\rho \text{ of length } i} \Pr[\rho] \leq poly(n)$$

□

Output distribution of R^* is correct: We proceed to show that output distribution of R^* is correct. This follows from the following two claims:

Claim 2. *Assume that R^* does not output* fail*, then except with negligible probability, its output is identical to the values committed to by A in the right interactions combined with its view.*

Proof. We first note that since in the Main Execution Phase, R^* feeds A messages according to the correct distribution, the view of A in the simulation by R^* is identical to the view of A in a real interaction. We show in Lemma 3 that there is a safe point for every right interaction that has an identifier different from the left interaction. Hence, for every convincing right interaction $k \geq \ell$ that has a different identifier, R^* rewinds that interaction and eventually will either output fail or a witness is extracted from the rewinding phase of R^*. Conditioned on R^* not outputting fail, by the statistically-binding property of com, except with negligible probability the witness extracted by R^* are the values committed to by A.

Lemma 3 (Safe-point Lemma)**.** *In any one-many man-in-the-middle execution with m right interactions, for any right interaction k, $1 \leq k \leq m$, such that it has a different identifier from the identifier of the left interaction, there exists a safe point for interaction k.*

Proof. Essentially using the same proof as in [6] we can prove this claim. For completeness, we give the proof in the Appendix. □

Claim 3. *R^* outputs* fail *with negligible probability.*

Proof. Recall that R^* outputs fail only when one of the following cases happen:

R^* **runs for more than** 2^n **steps:** We know that the expected running time of R^* is $poly(n)$. Using Markov inequality, we conclude that the probability that R^* runs more than 2^n steps is at most $\frac{poly(n)}{2^n}$.

The same proof transcript is obtained from some safe-point**:** This happens if R^* picks some challenge β in the Rewinding Phase that appeared as a challenge in the Main Execution Phase. As R^* runs for at most 2^n steps, it picks at most 2^n challenges. Furthermore, the length of each challenge is $2n$. By applying the union bound, we obtain that the probability that one β is picked twice is at most $\frac{2^n}{2^{2n}}$. Since there are at most polynomially many challenges picked in the Main Execution Phase; using the union bound again, we conclude that the probability that it outputs fail in this case is negligible.

The witness extracted is not a valid decommitment: Suppose, the witness extracted is not the decommitment information, then by the special-sound property it follows that it must be a value r such that $f(r) = s$. We show that if this happens with non-negligible probability, then we can invert the one-way function f. More precisely, given R^*, z', v we construct A^* that inverts f; A^* on input $f(r)$, internally incorporates $\hat{C}(v)$, proceeds just as $R^*(z')$ with the following exception: it uniformly picks a right interaction and feeds $f(r)$ to $R^*(z')$ as the first message from the receiver in that right interaction. A^* finally outputs the witness extracted by $R^*(z')$ in the chosen right interaction if the witness it not a valid decommitment, and otherwise outputs $\perp$.

Since each of the above cases occur with negligible probability, using the union bound, we conclude that R^* outputs fail with negligible probability. □

□

References

[1] B. Barak. How to go Beyond the Black-Box Simulation Barrier. In *42nd FOCS*, pages 106–115, 2001.

[2] B. Barak. Constant-Round Coin-Tossing or Realizing the Shared-Random String Model. In *43rd FOCS*, pages 345-355, 2002.

[3] G. Brassard, D. Chaum and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *JCSS*, Vol. 37, No. 2, pages 156–189, 1988. Preliminary version by Brassard and Crépeau in *27th FOCS*, 1986.

[4] R. Cramer, I. Damgård and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Crypto94*, Springer LNCS 839, pages. 174–187, 1994.

[5] G. di Crescenzo, G. Persiano and I. Visconti Constant-Round Resettable Zero Knowledge with Concurrent Soundness in the Bare Public-Key Model. In *Crypto04*, Springer LNCS 3152, pages. 237–253, 2004.

[6] D. Dolev, C. Dwork and M. Naor. Non-Malleable Cryptography. *SIAM Jour. on Computing*, Vol. 30(2), pages 391–437, 2000.

[7] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Crypto86*, Springer LNCS 263, pages 181–187, 1987

[8] O. Goldreich. *Foundations of Cryptography – Basic Tools*. Cambridge University Press, 2001.

[9] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Jour. on Computing*, Vol. 18(1), pp. 186–208, 1989.

[10] R. Pass. Bounded-Concurrent Secure Multi-Party Computation with a Dishonest Majority. In *36th STOC*, pages 232–241, 2004

[11] R. Pass and A. Rosen. Bounded-Concurrent Two-Party Computation in Constant Number of Rounds. In *44th FOCS*, pages 404–413, 2003

[12] R. Pass and A. Rosen. Bounded-Concurrent Secure Two-Party Computation in a Constant Number of Rounds. In *44th FOCS*, 2003.

[13] R. Pass and A. Rosen. New and Improved Constructions of Non-Malleable Cryptographic Protocols. In *37th STOC*, pages 533–542, 2005.

A General notation

We let N denote the set of all integers. For any integer $m \in N$, denote by $[m]$ the set $\{1, 2, \ldots, m\}$. For any $x \in \{0,1\}^*$, we let $|x|$ denote the size of x (i.e., the number of bits used in order to write it). For two machines M, A, we let $M^A(x)$ denote the output of machine M on input x and given oracle access to A. The term negligible is used for denoting functions that are (asymptotically) smaller than one over any polynomial. More precisely, a function $\nu(\cdot)$ from non-negative integers to reals is called negligible if for every constant $c > 0$ and all sufficiently large n, it holds that $\nu(n) < n^{-c}$.

A.1 Witness Relations

We recall the definition of a witness relation for an $\mathcal{NP}$ language [8].

Definition 3 (Witness relation)**.** *A* witness relation *for a language $L \in \mathcal{NP}$ is a binary relation R_L that is polynomially bounded, polynomial time recognizable and characterizes L by*

$$L = \{x : \exists y \, s.t. \, (x, y) \in R_L\}$$

We say that y is a witness for the membership $x \in L$ if $(x, y) \in R_L$. We will also let $R_L(x)$ denote the set of witnesses for the membership $x \in L$, i.e.,

$$_L(x) = \{y : (x, y) \in L\}$$

In the following, we assume a fixed witness relation R_L for each language $L \in \mathcal{NP}$.

A.2 Interactive Proofs

We use the standard definitions of interactive proofs (and interactive Turing machines) [9] and arguments (a.k.a computationally-sound proofs) [3]. Given a pair of interactive Turing machines, P and V, we denote by $\langle P, V \rangle(x)$ the random variable representing the (local) output of V when interacting with machine P on common input x, when the random input to each machine is uniformly and independently chosen.

Definition 4 (Interactive Proof System)**.** *A pair of interactive machines $\langle P, V \rangle$ is called an interactive proof system for a language L if for every PPT machine V there is a negligible function $\nu(\cdot)$ such that the following two conditions hold :*

- Completeness: *For every $x \in L$,*

$$\Pr\left[\langle P, V \rangle(x) = 1\right] = 1$$

- Soundness: *For every $x \notin L$, and every interactive machine B,*

$$\Pr\left[\langle B, V \rangle(x) = 1\right] \leq \frac{1}{\nu(|x|)}$$

In case that the soundness condition is required to hold only with respect to a computationally bounded prover, the pair $\langle P, V \rangle$ is called an interactive argument *system.*

A.3 Indistinguishability

Definition 5 ((Computational) Indistinguishability)**.** *Let X and Y be countable sets. Two ensembles $\{A_{x,y}\}_{x\in X,y\in Y}$ and $\{B_{x,y}\}_{x\in X,y\in Y}$ are said to be* computationally indistinguishable over X*, if for every probabilistic "distinguishing" machine D whose running time is polynomial in its first input, there exists a negligible function $\nu(\cdot)$ so that for every $x \in X, y \in Y$:*

$$|\Pr[D(A_{x,y}) = 1] - \Pr[D(B_{x,y}) = 1]| < \nu(|x|)$$

A.4 Witness Indistinguishability

An interactive proof is said to be *witness indistinguishable* ($\mathcal{WI}$) if the verifier's view is "computationally independent" of the witness used by the prover for proving the statement. In this context, we focus on languages $L \in \mathcal{NP}$ with a corresponding witness relation R_L. Namely, we consider interactions in which on common input x the prover is given a witness in $R_L(x)$. By saying that the view is computationally independent of the witness, we mean that for any two possible $\mathcal{NP}$-witnesses that could be used by the prover to prove the statement $x \in L$, the corresponding views are computationally indistinguishable.

Definition 6 (Witness-indistinguishability)**.** *Let $\langle P, V\rangle$ be an interactive proof system for a language $L \in \mathcal{NP}$. We say that $\langle P, V\rangle$ is* witness-indistinguishable *for R_L, if for every probabilistic polynomial-time interactive machine V^* and for every two sequences $\{w_x^1\}_{x\in L}$ and $\{w_x^2\}_{x\in L}$, such that $w_x^1, w_x^2 \in R_L(x)$ for every $x \in L$, the probability ensembles $\{\mathrm{VIEW}_2[P(x, w_x^1) \leftrightarrow V^*(x, z)]\}_{x\in L,z\in\{0,1\}^*}$ and $\{\mathrm{VIEW}_2[P(x, w_x^2) \leftrightarrow V^*(x, z)]\}_{x\in L,z\in\{0,1\}^*}$ are computationally indistinguishable over $x \in L$.*

B One-many implies many-many

Below we give the proof of Proposition 2.2.

Proof. Let A be a man-in-the-middle adversary that participates in at most $m = p(n)$ concurrent executions. We provide a simulator S for A. S proceeds as follows on input 1^n and z. S incorporates $A(z)$ and internally emulates all the left interactions for A by simply honestly committing to the string 0^n. Messages from the right interactions are instead forwarded externally. Finally S outputs the view of A.

We show that the values that S commits to are indistinguishable from the values that A commits to. Suppose, for contradiction, that this is not the case. That is, there exists a polynomial-time distinguisher D and a polynomial $p(n)$ such that for infinitely many n, there exist strings $v_1, \ldots, v_m \in \{0,1\}^n, z \in \{0,1\}^*$ such that D distinguishes $\mathsf{mim}^A_{\mathsf{com}}(v_1, \ldots, v_m, z)$ and $\mathsf{sta}^S_{\mathsf{com}}(1^n, z)$ with probability $\frac{1}{p(n)}$. Fix a generic n for which this happens. Consider the hybrid simulator S_i that on input $1^n, z' = v_1, \ldots, v_m, z$, proceeds just as S, with the exception that in left interactions $j \le i$, it instead commits to v_j. It directly follows that $\mathsf{mim}^A_{\mathsf{com}}(v_1, \ldots, v_m, z) = \mathsf{sta}^{S_m}_{\mathsf{com}}(1^n, z')$ and $\mathsf{sta}^S_{\mathsf{com}}(1^n, z) = \mathsf{sta}^{S_0}_{\mathsf{com}}(1^n, z')$. By a standard hybrid argument there exists an $i \in [m]$ such that

$$\Pr\Big[D(\mathsf{sta}^{S_{i-1}}_{\mathsf{com}}(1^n, z') = 1\Big] - \Pr\Big[D(\mathsf{sta}^{S_i}_{\mathsf{com}}(1^n, z') = 1\Big] \ge \frac{1}{p(n)m}$$

Note that the only difference between the executions by $S_{i-1}(1^n, z')$ and $S_i(1^n, z')$ is that in the former A receives a commitment to 0^n in session i, whereas in the latter it receives a commitment to v_i. Consider the one-many adversary $\tilde{A}$ that on input $\tilde{z} = z', n, i$ executes $S_{i-1}(1^n, z')$ with the

exception that the i'th left interaction is forwarded externally. Consider, the function $\mathsf{reconstruct}$ that on input values $v'_1, \ldots, v'_m$, and the view of $\tilde{A}$, reconstructs the view *view* of A in the emulation by $\tilde{A}$, and set $\tilde{v}_i = v'_1$ if A did not copy the identity of any of the left interactions, and $\perp$ otherwise, and finally outputs $\tilde{v}_1, \ldots, \tilde{v}_m, view$. By construction, it follows that $\mathsf{reconstruct}(\mathsf{mim}^{\tilde{A}}_{\mathsf{com}}(0^n, \tilde{z})) = \mathsf{sta}^{S_{i-1}}_{\mathsf{com}}(1^n, z')$ and $\mathsf{reconstruct}(\mathsf{mim}^{\tilde{A}}_{\mathsf{com}}(v_i, \tilde{z})) = \mathsf{sta}^{S_i}_{\mathsf{com}}(1^n, z')$. Since $\mathsf{reconstruct}$ is polynomial-time computable, this contradicts the one-many non-malleability of (C, R). □

C Safe-point Lemma

Lemma 3 (restated)**.** *In any one-many man-in-the-middle execution with m right interactions, for any right interaction k, $1 \leq k \leq m$, such that it has a different identifier from the identifier of the left interaction, there exists a safe point for interaction k.*

Proof of Lemma 3. Consider a one-many man-in-the-middle execution Δ, where the identifiers in the left and right interaction are different. Assume for contradiction, that there is some right interaction k which does not have a $\mathsf{safe\text{-}point}$, i.e. every prefix of Δ is not a $\mathsf{safe\text{-}point}$ for interaction k.

Consider any proof $(\alpha_r, \beta_r, \gamma_r)$ in the right interaction k. Let ρ be the prefix ending just before β_r. By assumption, ρ is not a $\mathsf{safe\text{-}point}$. This means there is some proof $(\alpha_l, \beta_l, \gamma_l)$ in the left interaction, such that α_l occurs before ρ, β_l occurs after ρ and before γ_r, as depicted in figure C. Therefore, there is one such proof in the left for every proof in the right. Furthermore, there is a one to one correspondence between the proofs. Hence, if there is no $\mathsf{safe\text{-}point}$, the only possible arrangement of the $4l$ proofs in the left and right is where the ith proof in the left is matched with the ith proof in the right (*i.e.* the second message of the proof on the left is in between the challenge-reply of the proof in the right).

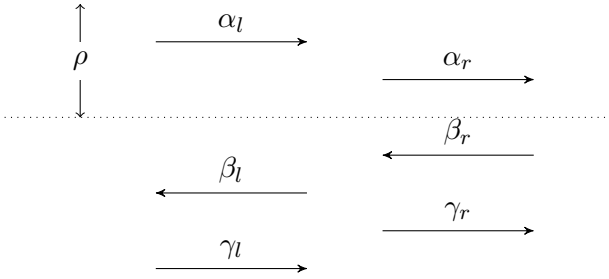

Figure 4: Interaction k does not have a safe point.

Since the identifiers in the left and right interactions are different, there must be a position j they differ in. Let the jth bit in the left be b and the right $1 - b$. Recall that, in the j^{th} round of Stage 3 of the protocol, the messages in the left interaction are arranged as design_b followed by a design_{1-b} and vice-versa in the right interaction. Since all the proofs are matched up one to one, it must be the case that there is a design_0 arrangement in the left that is matched with a design_1 arrangement in the right, as depicted in figure 5. Let $(\alpha^l_i, \beta^l_i, \gamma^l_i)$ be the two proofs in design_0, and $(\alpha^r_i, \beta^r_i, \gamma^r_i)$ be the ones on the right in design_1, $i = 0, 1$. However, in this case, consider ρ to be the prefix that includes all the message up until the message β^l_1. Consider the second proof $(\alpha^r_2, \beta^r_2, \gamma^r_2)$; there is no proof on the left having its first message before ρ and its challenge before γ^r_2 at the same time. Hence, we arrive at a contradiction to our assumption that there is no $\mathsf{safe\text{-}point}$ for that right interaction.

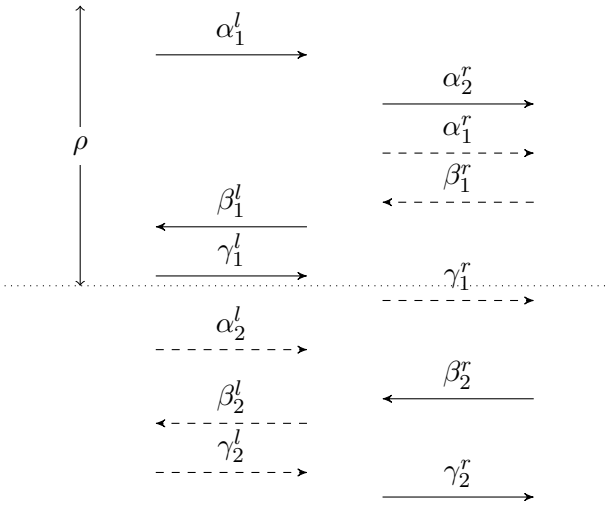

Figure 5: A design_0 matches up with design_1.

□

D A $\log n$-round non-malleable commitment scheme

The scheme $\langle C, R\rangle$ described in the previous section uses $O(n)$ rounds. We construct a commitment scheme that uses only $O(\log(n))$ rounds, but is only one-one concurrent non-malleable.

Description of the Protocol $\langle \tilde{C}, \tilde{R}\rangle$: This protocol is identical to $O(\log n)$-round protocol in [6]. To commit to value $v \in \{0,1\}^n$, choose random shares $r_1, \ldots, r_n \in \{0,1\}^n$, such that $v = r_1 \oplus \ldots \oplus r_n$. If id is the identifier of the $\langle \tilde{C}, \tilde{R}\rangle$ interaction, then for each i, commit to r_i (in parallel) using $\langle C, R\rangle$ with identifier $(i, \mathsf{id}_\mathsf{i})$, where id_i is the ith bit of id.

Claim 4. *$\langle \tilde{C}, \tilde{R}\rangle$ is one-one concurrent non-malleable*

Proof. We describe a simulator S that on input $(1^n, z)$ proceeds as follows. S incorporates $A(z)$ and internally emulates the left interaction by *honestly* committing to 0^n. Messages from the right interaction are instead forwarded externally. Finally, S outputs the view of A. Let $\mathsf{sha}^A_{\langle \tilde{C}, \tilde{R}\rangle}(v, z)$ be the random variable describing the random shares, $\{\tilde{r}_1, \ldots, \tilde{r}_n\}$, that A commits to in the right interaction. To show that the output of $\mathsf{mim}^A_{\langle \tilde{C}, \tilde{R}\rangle}(v, z)$ and $\mathsf{sta}^S_{\langle \tilde{C}, \tilde{R}\rangle}(1^n, z)$ are indistinguishable, it is sufficient to show that the following ensembles are indistinguishable over $n \in N$:

$$\left\{\mathsf{sha}^A_{\langle \tilde{C}, \tilde{R}\rangle}(v, z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{sha}^A_{\langle \tilde{C}, \tilde{R}\rangle}(0^n, z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*}$$

We show that the values that S commits to are indistinguishable from the values that A commits to. Suppose, for contradiction, that this is not the case. That is, there exists a polynomial-time distinguisher D and a polynomial $p(n)$ such that for infinitely many n, there exist strings $v \in \{0,1\}^n, z \in \{0,1\}^*$ such that D distinguishes $\mathsf{sha}^A_{\langle \tilde{C}, \tilde{R}\rangle}(v, z)$ and $\mathsf{sha}^A_{\langle \tilde{C}, \tilde{R}\rangle}(0^n, z)$ with probability $\frac{1}{p(n)}$. We construct an adversary A' and distinguisher D' that violates the one-many non-malleable property of $\langle C, R\rangle$ using A.

In the man-in-the-middle environment of $\langle \tilde{C}, \tilde{R} \rangle$, if the identifiers are the same for the left and right interaction, then the value A commits to in the right is $\perp$ (by definition). Otherwise, if the identifiers are different, namely id^{l} and id^{r} for the left and right interactions respectively, there exists k such that (k, id^l_k) is different from (j, id^r_j) for all j; we say that an execution with A is k-good if this happens. Let K be such that conditioned on the execution being K-good, D distinguishes with probability at least $\frac{1}{np(n)}$ (such a K must exist). Furthermore, there must be shares $n-1$ shares $r_1, \ldots, r_{K-1}, r_{K+1}, \ldots, r_n$ such that, conditioned on r_i being committed to in the ith parallel execution in the left for all $i \neq K$, D distinguishes the outputs of the two experiments.

We proceed to construct a one-many adversary A' for $\langle C, R \rangle$. A' on auxiliary input $z' = z \| r_1 \| \ldots \| r_{K-1} \| r_{K+1} \| \ldots \| r_{n-1}$, proceeds as follows: Externally, $A'(z')$ participates in a one-many concurrent execution of $\langle C, R \rangle$; internally, $A'(z')$ emulates a one-one environment of $\langle \tilde{C}, \tilde{R} \rangle$ with $A(z)$. For every ith parallel interaction in the left of A, such that $i \neq K$, it honestly committing to r_i and for the Kth interaction, it externally forwards the messages from A

It follows that, if D distinguishes the values committed to by A when it receives a commitment to v and 0^n in the left, then there exists a machine D' that also distinguishes the value committed to by A' combined with the view.

Consider, the function $\mathsf{reconstruct}$ that on input values $v'_1, \ldots, v'_m$, and the view of A', reconstructs the view *view* of A in the emulation by A', and set $\tilde{v}_i = v'_1$ if A did not copy the identity of any of the left interactions, and $\perp$ otherwise, and finally outputs $\tilde{v}_1, \ldots, \tilde{v}_m, view$. D' on input view of A' and the values committed to by A', runs *reconstruct* to obtain the values committed to by A combined with its view. Then D' runs D on this view if it is K-good and guesses otherwise. By construction, it follows that $D'(reconstruct(\cdot))$ distinguishes the output of $\mathsf{mim}^{A'}_{\langle \tilde{C}, \tilde{R} \rangle}(v_1, \tilde{z})$ and $\mathsf{mim}^{A'}_{\langle \tilde{C}, \tilde{R} \rangle}(v_2, \tilde{z})$ where $v_1 = (\oplus_{i \neq K} r_i) \oplus v$ and $v_2 = (\oplus_{i \neq K} r_i) \oplus 0^n$. Since $\mathsf{reconstruct}$ is polynomial-time computable, this contradicts the one-many non-malleability of $\langle C, R \rangle$. $\square$