Concurrent Non-malleable Commitments from any One-way Function

Margarita Vald

Tel-Aviv University
Outline

Non-Malleable Commitments

Problem Presentation
Overview
DDN - First NMC Protocol

Concurrent Non-Malleable Commitments

Overview
Definitions
The conNMCcom Protocol
Security Proof

Summary
Outline

Non-Malleable Commitments

Problem Presentation
  Overview
  DDN - First NMC Protocol

Concurrent Non-Malleable Commitments

Overview
Definitions
The conNMCcom Protocol
Security Proof

Summary
Commitments

- String commitment protocol consists of two stages
  - **Commit stage**: $C$ and $R$ exchange messages. At the end of this stage $R$ has some information that represents $v$ ($R$ should gain no information on the value of $v$)
  - **The reveal stage**: at the end of this stage $R$ knows $v$ (There should be only one string that $C$ can reveal).
Commitment Schemes I

The requirements of string commitment protocol:

- **Secrecy - Computational hiding**
  - For every (expected) PPT machine $R^*$ the following ensembles are computationally indistinguishable over $\{0, 1\}^n$

\[
\begin{align*}
\left\{ \text{sta}_{\langle C, R \rangle}^R(v_1, z) \right\} & & v_1, v_2 \in \{0, 1\}^n, n \in \mathbb{N}, z \in \{0, 1\}^* \\
\left\{ \text{sta}_{\langle C, R \rangle}^R(v_2, z) \right\} & & v_1, v_2 \in \{0, 1\}^n, n \in \mathbb{N}, z \in \{0, 1\}^*
\end{align*}
\]

- Where $\text{sta}_{\langle C, R \rangle}^R(v, z)$ denote random variable describing the output of $R^*$ after receiving a commitment to $v$ using $\langle C, R \rangle$. 

Commitment Schemes II

- **Binding - 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.

- Instead we can require a computationally binding and statistically hiding commitment scheme

- It is a known result that statistically binding and statistically hiding commitment schemes don’t exist

- We consider commitment schemes that are statistically-binding and computationally-hiding
Outline

Non-Malleable Commitments
  Problem Presentation
  Overview
  DDN - First NMC Protocol

Concurrent Non-Malleable Commitments
  Overview
  Definitions
  The conNMCcom Protocol
  Security Proof

Summary
Non-Malleable Commitments I

- Non-Malleability is the case where the adversary sees a commitment to a specific value and doesn’t succeeds to commit to a related value.
- Intuitively we tend to think about commitments as sealed envelopes.
  - It appears to captures the notion of non-malleability.
  - In reality when we replace the envelopes with real commitments we see our intuition is faulty.
- The basic definition of commitment schemes doesn’t capture non-malleability.
Non-Malleable Commitments II

- Consider the following commitment scheme for \( m \in [1...p] \):
- Let \( G \) be a group of prime order \( p \approx 2^n \) and let \( g \) be a generator of \( G \).

\[
\text{Commit: } g^{r_1}, g^{r_2}, g^{r_1r_2} \cdot m
\]

\[
\text{Reveal: } m, r_1, r_2
\]

- This scheme is computationally hiding and perfectly binding as desired.
  - Recall the DDH assumption
Non-Malleable Commitments III

- A commitment that was made using this scheme can be malleable.
- Consider an adversary that upon receiving a commitment \((a, b, c)\) proceeds as follows:
  - Choose \(r'_1, r'_2 \in \mathbb{Z}_p\).
  - Compute \(a' = (g^{r'_1} \cdot a),\ b' = (g^{r'_2} \cdot b)\) and \(c' = (g^{r'_1 \cdot r'_2} \cdot c)\).
  - Commit to \((a', b', c')\).
- Note: The adversary commits to
  \[
  (g^{r'_1 \cdot r_1}, g^{r'_2 \cdot r_2}, g^{r'_2 \cdot r_2 \cdot r'_1 \cdot r_1} \cdot m)
  \]
The first to raise this notion were Dolev, Dvork and Naor in their paper “Non-Malleable Cryptography” (1991).

- Presented a non-malleable commitment scheme with $O(\log n)$ rounds.
- The paper provides a very complex and subtle proof.
  - Based on the existence of OWF.
  - Uses ZKP as a building block.
NMC (cont.)

- Intuitively a commitment scheme is non-malleable if a man-in-the-middle adversary that receives commitment to $v$ will not be able to “successfully” commit to related value.

\[
\text{C} \xrightarrow{\text{com}(v, i)} A(z) \xrightarrow{\text{com}(v', i)} R
\]

- We compare between a Man-in-the-middle execution and an ideal execution.

- **Notation:** Let $\langle C, R \rangle$ be a commitment scheme, $R \subseteq \{0, 1\}^n \times \{0, 1\}^n$ be a polynomial-time computable non-reflexive relation (i.e., $R(v, v) = 0$), and let $n \in \mathbb{N}$ be a security parameter.
Man-in-the-middle Execution

The success of $A$ is defined by:

- $\text{mim}_A^{C,R} (R, v, z) = 1$ iff $A$ produces a valid commitment to $\bar{v}$ such that $R(v, \bar{v}) = 1$. 

The success of $S$ is defined by:

- $\text{sim}^S_{<C,R>} (R, \nu, z) = 1$ if and only if $S$ produces a valid commitment to $\bar{\nu}$ such that $R(\nu, \bar{\nu}) = 1$. 

Ideal Execution
Non-malleable Commitments

Definition (Non-Malleable Commitment Scheme)

A commitment scheme $\langle C, R \rangle$ is said to be liberal non-malleable if for every probabilistic polynomial-time man-in-the-middle adversary $A$, there exists a probabilistic expected polynomial time adversary $S$, such that for every non-reflexive polynomial-time computable relation $R \subseteq \{0, 1\}^n \times \{0, 1\}^n$, every $v \in \{0, 1\}^n$, and every $z \in \{0, 1\}^*$, it holds that:

$$\left| \Pr [ \text{mim}^A_{\langle C, R \rangle} (R, v, z) = 1] - \Pr [ \text{sim}^S_{\langle C, R \rangle} (R, v, z) = 1] \right| < \nu (n)$$
Outline

Non-Malleable Commitments
  Problem Presentation
  Overview
  DDN - First NMC Protocol

Concurrent Non-Malleable Commitments
  Overview
  Definitions
  The conNMCcom Protocol
  Security Proof

Summary
POK - First Attempt

- We wish to convert a malleable commitment scheme to a non-malleable one
- Our first attempt would be to use a 3 rounds POK
  - To ensure the committer knows the value he committed to
- 3 rounds POK consists of
  - First step: Commitment
  - Second step: Challenge
  - Third step: Response
Malleable Protocol

- POK may be malleable
  - Consider the following scenario:

```
C  A  R
Commit  Commit  
Challenge  Challenge  
Reply  Reply  
POK
```

- Goal: ensure that for each execution there will be a “triple” for which no other triple is useful
Outline

Non-Malleable Commitments
  Problem Presentation
  Overview
  DDN - First NMC Protocol

Concurrent Non-Malleable Commitments
  Overview
  Definitions
  The conNMCcom Protocol
  Security Proof

Summary
Concurrent Non-Malleable commitments

- A natural extension of Non-Malleability is one in which more than two invocations of the commitment protocol take place concurrently
  - DDN scheme does not rule out joint dependencies between more than one individual pair
- Still two major questions remained open:
  - 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?
Concurrent NMC from any OWF

- These questions were answered fully in the recent paper “Concurrent Non-Malleable Commitments from any One-Way Function” [Lin Pass Venkitasubramaniam 2008]
  - Proves the existence of statistically-binding commitment scheme that is concurrent non-malleable based solely on the existence of OWF and using only black-box techniques.
- Variant of the DDN protocol
- Provides stronger definition of non-malleability
- The commitment scheme uses $O(n)$ communication rounds
- Provides simple proof of security
Concurrent NMC (Cont.)

- 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 Man-in-the-middle execution and the ideal execution.
- **Notation:** Let $\langle C, R \rangle$ be a commitment scheme, $n \in \mathbb{N}$ be a security parameter.
  - Let $id_i$ be the identifier of committer $C_i$.
  - Let $\bar{id}_i$ the identifier used by the adversary in committing to $\bar{v}_i$. 

Man-in-the-middle Execution

Let \( \text{mim}^A_{<C,R>} (v_1, \ldots, v_m, z) \) denote a random variable that describes the values \( \tilde{v}_1, \ldots, \tilde{v}_m \) and the view of \( A \), in the above experiment.
Ideal Execution

Let \( \{ \text{sim}^S_{\langle C, R \rangle}(1^n, z) \} \) denote the random variable describing the values \( \bar{v}_1, \ldots, \bar{v}_m \) committed to by \( S \), and the output view of
Identities

- We make extensive use of identities
- There are two ways to model the identity of the committer
  - Assuming there is a name-space in which a committer is represented by a unique string
  - Concatenation of the committer identity to the commitment
Outline

Non-Malleable Commitments
- Problem Presentation
- Overview
- DDN - First NMC Protocol

Concurrent Non-Malleable Commitments
- Overview
- Definitions
- The conNMCcom Protocol
- Security Proof

Summary
**Concurrent Non-Malleable Scheme I**

**Definition (Concurrent Non-Malleable Commitment Scheme)**

A commitment scheme \( \langle C, R \rangle \) is said to be concurrent non-malleable if every probabilistic polynomial-time man-in-the-middle adversary \( A \) that participates in at most \( m \) concurrent executions, there exists a probabilistic polynomial time adversary \( S \) such that the following ensembles are computationally indistinguishable over \( \{0, 1\}^n \):

\[
\left\{ \text{mim}_A^{\langle C, R \rangle} \left( v_1, \ldots, v_m, z \right) \right\}_{v_1, \ldots, v_m \in \{0, 1\}^n, n \in \mathbb{N}, z \in \{0, 1\}^*} = \left\{ \text{sim}_S^{\langle C, R \rangle} \left( 1^n, z \right) \right\}_{v_1, \ldots, v_m \in \{0, 1\}^n, n \in \mathbb{N}, z \in \{0, 1\}^*}
\]
This is a stronger definition that considers not only the values the adversary commits to, but also the view of the adversary!

For $m = 1$ any protocol that satisfies this definition also satisfies the definition of DDN.

This is because the existence of a polynomial time computable relation $R$ that violates the original definition of non-malleability could be used to distinguish between the values of $\text{mim}_{\langle C, R \rangle}^A (v, z)$ and $\text{sim}_{\langle C, R \rangle}^S (1^n, z)$. 
Relaxed Notion

relaxed notions of concurrent non-malleability:

- one-one non-malleable commitment: we consider only adversaries $A$ that participate in one left and one right interaction.
- one-many: $A$ participates in one left and many right.
- many-one: $A$ participates in many left and one right.
One-many implies many-many I

**Proposition 1.** 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.

**Proof**

- Let $A$ be a man-in-the-middle adversary that participates in at most $m = p(n)$ concurrent executions.
- Simulator $S$ on input $(1^n, z)$
  - $S$ 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 $A$. 
One-many implies many-many II

- Suppose, for contradiction, there exists a polynomial-time distinguisher \( D \) that distinguishes \( \text{mim}^{A}_{C,R}(v_1, \ldots, v_m, z) \) and \( \text{sim}^{S}_{C,R}(1^n, z) \) w.p \( \frac{1}{p(n)} \).

- 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 \( j \leq i \), it instead commits to \( v_j \).

- By a standard hybrid argument there exists an \( i \in [m] \) such that

\[
\left| \Pr_{a \leftarrow \text{sim}^{S_i}_{C,R}(1^n, z')} \left[ D \left( 1^n, z', a \right) = 1 \right] \right| - \Pr_{b \leftarrow \text{sim}^{S_i}_{C,R}(1^n, z')} \left[ D \left( 1^n, z', b \right) = 1 \right] \geq \frac{1}{p(n)m}
\]
One-many implies many-many III

- 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 reconstruct: on input $\text{mim}_{\text{com}}^{\tilde{A}}(0^n, \tilde{z})$
  - reconstructs the view $\text{view}$ of $A$ in the emulation by $\tilde{A}$, and sets $\tilde{v}_i = v'_i$ if $A$ did not copy the identity of any of the left interactions, and $\perp$ otherwise,
  - finally outputs $\tilde{v}_i, \ldots, \tilde{v}_m$, $\text{view}$.

- By construction, it follows that
  - $\text{reconstruct}(\text{mim}_{\langle C, R \rangle}^{\tilde{A}}(0^n, \tilde{z})) = \text{sim}_{\langle C, R \rangle}^{S_{i-1}}(1^n, z')$
  - $\text{reconstruct}(\text{mim}_{\langle C, R \rangle}^{\tilde{A}}(v_i, \tilde{z})) = \text{sim}_{\langle C, R \rangle}^{S_i}(1^n, z')$

Since reconstruct is polynomial-time computable, this contradicts the one-many non-malleability of $\langle C, R \rangle$. 
Special-sound proofs

Definition (Special-sound proofs)
A 3-round public-coin interactive proof for the language $L \in NP$ with witness relation $R_L$ is special-sound with respect to $R_L$, if for any two transcripts $(\alpha, \beta, \gamma)$ and $(\alpha', \beta', \gamma')$ such that the initial messages $\alpha, \alpha'$ are the same but the challenges $\beta, \beta'$ are different, there is a deterministic procedure to extract the witness from the two transcripts and runs in polynomial time.
Witness Indistinguishability

Definition (Witness-indistinguishability)

Let $\langle P, V \rangle$ be an interactive proof system for a language $L \in NP$. 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 $\{w^1_x\}_{x \in L}$ and $\{w^2_x\}_{x \in L}$, such that $w^1_x, w^2_x \in R_L(x)$ for every $x \in L$,

- $\{\langle P(w^1_x), V(z) \rangle(x) \}_{x \in L, z \in \{0,1\}^*}$
- $\{\langle P(w^2_x), V(z) \rangle(x) \}_{x \in L, z \in \{0,1\}^*}$

are computationally indistinguishable over $x \in L$. 
Outline

Non-Malleable Commitments
  Problem Presentation
  Overview
  DDN - First NMC Protocol

Concurrent Non-Malleable Commitments
  Overview
  Definitions
  The conNMCcom Protocol
  Security Proof

Summary
Protocol conNMCcom 1

**Common Input:** An identifier \( id \in \{0, 1\}^l \) (The identity of the committer).

**Private Input for Committer:** A string \( v \in \{0, 1\}^n \).

**Stage 1:**
- R uniformly chooses \( r \in \{0, 1\}^n \).
- \( R \rightarrow C: s = f(r) \) where \( f \) is a OWF.
- C aborts if \( s \) not in the range of \( f \).

**Stage 2:**
- \( C \rightarrow R: c = \text{com}(v, r') \) where \( r' \in_R \{0, 1\}^{\text{poly}(n)} \).
Protocol conNMCcom II

Stage 3:
C→R: 4l special-sound WI proofs of the statement either there exists values \( v, r' \) s.t \( c = \text{com}(v, r') \) 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 \( \text{design}_{id_j} \) followed by Execute \( \text{design}_{1-id_j} \)
One-Way Functions

- The protocol relies on the existence of one-way functions with efficiently recognizable range.
- 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.
Designs

- Description of the schedules used in Stage 3.
- The scheduling identical to DDN massage scheduling technique.
Statistically-binding Commitment Scheme

Lemma

\[ \langle C, R \rangle \text{ is a statistically-binding commitment scheme.} \]

- **Binding**: Follows directly from the binding property of com.
- **Hiding**: Follows from the hiding property of com and the fact that Stage 3 of the protocol is \( WI \)
Outline

Non-Malleable Commitments
- Problem Presentation
- Overview
- DDN - First NMC Protocol

Concurrent Non-Malleable Commitments
- Overview
- Definitions
- The conNMCcom Protocol
- Security Proof

Summary
Proof of Security

Theorem
\( \langle C, R \rangle \) is one-many concurrent non-malleable.

- Let \( A \) be an adversary.
- Simulator \( S \): on input \((1^n, z)\)
  - 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, outputs the view of \( A \).
- Goal: Show \( \{ \text{mim}^A_{\langle C, R \rangle}(v, z) \}_{v \in \{0,1\}^n, n \in \mathbb{N}, z \in \{0,1\}^*} \) 
  \( \approx \{ \text{mim}^A_{\langle C, R \rangle}(0^n, z) \}_{v \in \{0,1\}^n, n \in \mathbb{N}, z \in \{0,1\}^*} \)
Modified Scheme

Let $\langle \hat{C}, \hat{R} \rangle$ be a commitment scheme as follow:

- Variant of $\langle C, R \rangle$ Protocol
- Modified Stage 3:
  - The receiver can ask for an arbitrary number of special-sound \(\mathcal{W}\)I designs
  - The receiver chooses which design to execute in each iteration
    - By sending bit \(i\)
- Note: $\langle \hat{C}, \hat{R} \rangle$ is computationally hiding
  - Same proof as for $\langle C, R \rangle$
Proof (Cont.)

- Let $\Gamma(A, z)$ denote the distribution of all joint views $\tau$ of $A$ and the receivers in the right, such that $A$ sends its first message in the left interaction directly after receiving the messages in $\tau$.
- Let $Z(z, \tau) = z \parallel \tau \parallel \tilde{v}_1 \parallel ... \parallel \tilde{v}_\ell$ where $\ell \in [m]$ and $\tilde{v}_1, ..., \tilde{v}_\ell$ are the values committed to by $A(z)$ in $\tau$.

Lemma

For every PPT adversary $A$, there exists an expected PPT adversary $R^*$ such that the following ensembles are indistinguishable over $\{0, 1\}^*$:

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

$$\left\{ \text{mim}^A_{\langle C, R \rangle} (v, Z) \right\}_{v \in \{0, 1\}^n, n \in \mathbb{N}, z \in \{0, 1\}^*}$$
Why does it help?

- According to the hiding property of $\langle \hat{C}, \hat{R} \rangle$ the following ensembles are indistinguishable

$$
\begin{align*}
\left\{ \text{sta}^{R^*}_{\langle \hat{C}, \hat{R} \rangle} (v, Z_A(z)) \right\}_{v \in \{0,1\}^n, n \in N, z, \tau, z' \in \{0,1\}^*} \\
\left\{ \text{sta}^{R^*}_{\langle \hat{C}, \hat{R} \rangle} (0^n, Z_A(z)) \right\}_{v \in \{0,1\}^n, n \in N, z, \tau, z' \in \{0,1\}^*}
\end{align*}
$$

- Using the Lemma, we conclude that the following ensembles are indistinguishable:

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

- This concludes the proof of theorem.
Proving the Lemma

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.
Adversary $R^*$

- $R^*$ interacts externally as a receiver using $\langle \hat{C}, \hat{R} \rangle$.
- $R^*$ simulating all right receivers and emulating the left $\langle C, R \rangle$ interaction.
• **Main Execution Phase:** Feed the view in $\tau$ to $A$ and all right receivers. Emulate all the interaction from $\tau$ and complete the execution with $A$. Let $\Delta$ be the transcript of messages obtained.

• **Rewinding Phase:** attempt to extract all the values committed by $A$.

• **Output phase:** For every interaction $k$ that is not convincing or if the identity of the right interaction is the same as the left interaction, 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**.
Rewinding Phase

- Goal: Ensure that there exist some point (called safe-point) where we can rewind A on the right interaction, without rewinding on the left.

- This is possible in two cases:
  - If rewinding on the right does not cause A to request any new messages on the left.
  - If rewinding on the right causes A to only request a new special-sound proof—in this case $R^*$ can perfectly emulate this new proof by simply requesting another design from $\langle \hat{C}, \hat{R} \rangle$. 
Safe-points

Definition (Safe-point)
A prefix $\rho$ of a transcript $\Delta$ is called a safe-point for interaction $k$, if there exists an accepting proof $(\alpha_r, \beta_r, \gamma_r)$ in the $k^{th}$ right interaction, such that:

1. $\alpha_r$ occurs in $\rho$, but not $\beta_r$ (and $\gamma_r$).
2. for any proof $(\alpha_\ell, \beta_\ell, \gamma_\ell)$ in the left interaction, if $\alpha_\ell$ occurs in $\rho$ and $\beta_\ell$ not in $\rho$, then $\beta_\ell$ occurs after $\gamma_r$. 
Safe-point examples

- When rewinding from $\rho$, $R^*$ can emulate the left proof by requesting a new design from $\langle \hat{C}, \hat{R} \rangle$. 
Safe-point examples (Cont.)

- \( R^* \) can simply re-send the third message of the left proof
  - since it is determined by the first two messages in the proof
Safe-point examples (Cont.)

- No new message is requested by $A$, so the left interaction can be “trivially” emulated by doing nothing.
Non Safe-point example

- Note: The only case a right-interaction proof does not have a safe-point is if it is "aligned" with a left execution proof
  - A can forward messages between the left and the right interactions.
Rewinding Phase Procedure I

**Rewinding Phase:** For $k = \ell + 1$ to $m$, if interaction $k$ is convincing and its identity is different from the left interaction do:

- In $\Delta$, find the first point $\rho$ that is a **safe-point** for the 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 left interaction as in the Main-Execution Phase.
  - For the left interaction:
    - If $A$ expects to get a new proof from the external committer:
      - Emulate the proof by requesting for design$_0$ from outside committer.
      - Forward one of the two proofs internally.
Rewinding Phase Procedure II

- If $A$ sends a challenge for a proof whose first message occurs in $\rho$:
  - Cancel the execution, rewind to $\rho$ 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 $\text{fail}$.
- If $w = (v, r)$ is a valid commitment for interaction $k$, then set $\hat{v}_k = v$. Otherwise halt and output $\text{fail}$.

**Note:** since right interaction $\ell + 1$ to $m$ all have their Stage 2 and 3 occurring after $\tau$, none of the rewinding can make $A$ request a new commitment from the external committer.
Safe-point Lemma I

Lemma

In any one-many man-in-the-middle execution with $m$ right interactions, for any right interaction $k \in [m]$, such that it has a different identity from the identity of the left interaction, there exists a safe-point for interaction $k$.

- Consider a one-many man-in-the-middle execution $\Delta$, where the identities in the left and right interaction are different.
- Assume for contradiction, that there is some right interaction $k$ which does not have a safe-point
  - Every prefix of $\Delta$ is not a safe-point for interaction $k$.
- Consider any proof $(\alpha_r, \beta_r, \gamma_r)$ in the right interaction $k$. Let $\rho$ be the prefix after which $\beta_r$ is sent immediately.
Safe-point Lemma II

- By assumption, there exists a proof \((\alpha_\ell, \beta_\ell, \gamma_\ell)\) in the left interaction, such that \(\alpha_\ell\) occurs before \(\rho\), \(\beta_\ell\) occurs after \(\rho\) and before \(\gamma_\ell\) \((\beta_\ell\) occurs in between \(\beta_r\), and \(\gamma_r\)).
  - In this case a left proof is associated with a right proof.
  - **Note:** Each left proof can be associated with at most one right proof.

- Since interaction \(k\) doesn't have a safe-point, the proofs in the left and right interactions must match up each other one by one.

- There is a position \(j\) where the identities in the left and right interactions are different.
Safe-point Lemma III

- Let the $j^{th}$ bit in the left be $b$ and that in the right be $1-b$.
  - In the $j^{th}$ round of Stage 3 of the protocol, the left interaction
    has $\text{design}_b$ followed by $\text{design}_{1-b}$; and the right interaction
    has $\text{design}_{1-b}$ followed by $\text{design}_b$.
  - Since all the proofs are “matched up”, there must be a $\text{design}_0$
    on the left that is matched with a $\text{design}_1$ on the right,
Consider $\rho$ to be the prefix that includes all the message up until $\beta_1^\ell$.

- Consider the second proof $(\alpha_2^r, \beta_2^r, \gamma_2^r)$;
Safe-point Lemma V

- there is no proof on the left having its first message before $\rho$ and its challenge before $\gamma^r_2$ at the same time.

It is a contradiction to the assumption that there is no safe-point for that right interaction.
Output distribution 1

**Claim:** 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.

- 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.
Output distribution II

- For every convincing right interaction $k > \ell$ that has a different identity, $R^*$ 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 statistical-binding property, except with negligible probability the witnesses extracted by $R^*$ are the values committed to by $A$. 
Claim: $R^*$ outputs fail with negligible probability.

$R^*$ outputs fail only in the following cases:

- **R runs for more than $2n$ steps:** The expected running time of $R^*$ is $\text{poly}(n)$. Therefore, the probability that $R^*$ runs more than $2^n$ steps is at most $\frac{\text{poly}(n)}{2^n}$ (Markov).
The same proof transcript is obtained from some safe-point: This case occurs if $R^*$ picks some challenge $\beta$ 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 $2n$. Therefore, the probability that a $\beta$ is picked twice is at most $\frac{2^n}{2^{2n}}$ (Union bound).
- Since there are at most polynomially many challenges picked in the Main Execution Phase, the probability that it outputs fail in this case is negligible.
Failure Probability III

- **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 $f(r) = s$.

  - If this happens with non-negligible probability, then we can invert the one-way function $f$.
  - Given $A$, $z$ and $v$; $A^*$ on input $y$, picks uniformly at random from $\Gamma(A, z)$ and proceeds identically as $R^*$ with inputs $\tau, z' = z \| \tau \| \bot \| ...$ with the exception that it picks a random right interaction, say $k$, and feeds $y$ as the Stage 1 message in that interaction.
  - On the left interaction it honestly commits to the string $v$ using $\langle \hat{C}, \hat{R} \rangle$.
  - Finally, if $f(r') = y$ then $A^*$ outputs $r'$. 
Questions?

Thank you for listening!