On Round-Efficient Non-Malleable Protocols

Michele Ciampi
DIEM, Università di Salerno
ITALY
mciampi@unisa.it

Rafail Ostrovsky
UCLA
Los Angeles
rafail@cs.ucla.edu

Luisa Siniscalchi
DIEM, Università di Salerno
ITALY
lsiniscalchi@unisa.it

Ivan Visconti
DIEM, Università di Salerno
ITALY
visconti@unisa.it

Abstract

The round complexity of non-malleable commitments and non-malleable zero knowledge arguments has been an open question for long time. Very recent results of Pass [TCC 2013] and of Goyal et al. [FOCS 2014, STOC 2016], gave almost definitive answers.

In this work we show how to construct round-efficient non-malleable protocols via compilers. Starting from protocols enjoying limited non-malleability features, our compilers obtain full-fledged non-malleability without penalizing the round complexity.

By instantiating our compilers with known candidate constructions, the resulting schemes improve the current state of the art in light of subtleties that revisit the analysis of previous work. Additionally, our compilers give a non-malleable zero-knowledge argument of knowledge that features delayed-input completeness. This property is satisfied by the proof of knowledge of Lapidot and Shamir [CRYPTO 1990] and has recently been used to improve the round complexity of several cryptographic protocols.

Contents

1 Introduction

Commitment schemes and zero-knowledge argument systems are fundamental primitives in Cryptography. Here we consider the intriguing question of constructing round-efficient schemes that remain secure even against man-in-the-middle (MiM) attacks: non-malleable (NM) commitments and NM zero-knowledge (NMZK) argument systems [DDN91].

Non-malleable commitments. The round complexity of commitment schemes in the stand-alone setting is well understood. Non-interactive commitments exist assuming one-to-one OWFs [GL89], and 2-round commitments exist assuming one-way functions (OWFs) only. Moreover non-interactive commitments do not exist if one relies on the black-box use of OWFs only [MP12].

Instead, the round complexity of NM commitments[1] after 25 years of research remains a fascinating open question, in particular when taking into account the required computational assumptions. The original construction of [DDN91] required a logarithmic number of rounds and the sole use of OWFs. Then, through a long sequence of very exciting positive results [Bar02, PR03, PR05b, PR05a, PR08b, PR08a, LPV08, PW10, Wee10, LP11b, LP15, Goy11, GLOV12], the above open question has been in part addressed obtaining a constant[2]-round (even concurrent) NM commitment scheme by using any OWF in a black-box fashion. On the negative side, Pass recently proved that NM commitments require at least 3 rounds [Pas13][3] when security is proved through a black-box reduction to falsifiable (polynomial or subexponential time) hardness assumptions.

Among the various attempts to solve the above challenging open problem a major progress has been done by Goyal et al. [GRRV14] that showed a 4-round NM commitment scheme based on OWFs only. Their scheme admits a very efficient implementation [BGR+15] and is secure also with respect to adversaries mounting concurrent MiM attacks. In such attacks, there are multiple interleaved executions of commitments from senders to receivers.

A recent breakthrough of Goyal et al. [GPR16] exploited the use of the NM codes in the split-state model of Aggarwal et al. [ADL14] to show (surprisingly) a 3-round NM commitment scheme based on the black-box use of any one-to-one OWF. Their construction is proved one-one NM only[4]. A new result of Ciampi et al. [COSV16] obtains concurrent non-malleability in 3 rounds through a compiler that requires any 3-round one-one NM commitment scheme[5] and assumes one-way permutations (OWPs) secure against subexponential-time adversaries[6]. Interestingly, the negative result of Pass [Pas13] applies also to such stronger OWPs, and therefore when properly instantiated, the compiler of [COSV16] gives a round-optimal concurrent NM commitment scheme.

[1]In this paper we will consider only NM commitments w.r.t. commitments. Difficulties on additionally achieving the notion of NM w.r.t. decommitments were discussed in [OPV09, CVZ10].

[2]According to [GRRV14] the construction of [GLOV12] can be squeezed to 6 rounds.

[3]If instead one relies on non-standard assumptions or trusted setups (e.g., using trusted parameters, working in the random oracle model, relying on the existence of NM OWFs) then there exist non-interactive NM commitments [DG03, PPV08].

[4]One-one NM means that there exists only one sender and one receiver with the MiM playing one session with each of them. One-many instead allows polynomially many receivers.

[5]The scheme must also enjoy some extractability property.

[6]Hardness assumptions against subexponential-time adversaries have been used in the past [PR03, PW10, Wee10] to improve the round-complexity of NM commitments.

Non-malleable zero knowledge. The progress on NMZK arguments has tightly followed advances on NM commitments[7]. The construction of [GRRV14] has closed this line of research since they showed a 4-round NMZK argument of knowledge (AoK) that relies on the existence of OWFs only. This result is clearly optimal both in round complexity and in computational assumptions.

Interestingly, several years after the 4-round zero knowledge (ZK) argument system from OWFs of [BJY97], the same optimal round complexity and optimal complexity assumptions have been shown sufficient for NMZK [GRRV14] and resettably sound ZK [COP+14].

Delayed-input protocols. In [LS90] Lapidot and Shamir showed a 3-round witness-indistinguishable (WI) proof of knowledge (PoK) for $\mathcal{NP}$ where the instance (except its length) and the witness are not needed before playing the last round. This "delayed-input" form of completeness has been critically used in the past (e.g., [KO04, DPV04a, YZ07, Wee10]) and very recently (e.g., [CPS+16a, CPS+16b, GMPP16, COSV16, HV16, MV16]), since it often helps in improving the round complexity of an external protocol.

1.1 Our Results

In this work we first point out subtleties in the security proofs of [GRRV14][8] that re-open the questions of constructing 4-round concurrent NM commitments and 4-round (i.e., round optimal) NMZK arguments. The subtleties concern a MiM that completes the commitment phase when playing as a sender, even though the committed message is invalid (i.e., it corresponds to $\perp$). We stress that such subtleties appear only in the "parallelized" 4-round version of the main construction of [GRRV14] (which required a larger constant number of rounds). These subtleties affect in turn the security of the one-one NM commitment scheme of [BGR+15] since this protocol (and proof) has essentially the same structure of the one of [GRRV14].

Then we show compilers that starting with basic non-malleability features obtain full-fledged non-malleability. By instantiating our compilers with candidate constructions proposed in previous work one obtains 4-round concurrent NM commitments and 4-round delayed-input NMZK arguments of knowledge assuming the existence of collision-resistant hash functions (CRHFs).

Our approach: non-malleability upgrades. Constructions for NM commitments and NMZK arguments are usually complicated and start from basic and extremely malleable tools likes regular commitment schemes and zero-knowledge proofs. Obtaining non-malleability from such basic building blocks is a well known complicated task and achieving it in a round-efficient way has always been a major challenge. Given the above indisputable difficulties, security proofs are usually very non-trivial and can be difficult to study and re-use. Here we take a different approach that consists of starting with a very basic and limited form of non-malleability, to then upgrade it to the desired notions.

[7]For NMZK we will omit the case of concurrent MiM attacks since there is a logarithmic lower bound on the round complexity of concurrent NMZK [CKPR01] when relying on black-box simulation. While this lower bound has been matched by Barak et al. [BPS06], no non-black-box construction is known with sub-logarithmic rounds.

[8]We have informed the authors of [GRRV14] about the above subtleties describing specific one-one and one-many adversaries. They 1) confirmed in [GRRV16a] that for the 4-round protocols given in [GRRV14] there is currently no security proof; 2) communicated to us [GRRV16b] that modified protocols and proofs for 4-round concurrent NM commitments from OWFs and one-one NM zero knowledge from OWFs can be found in [GRRV16c].

Informally, we say that a commitment scheme is weak non-malleable if it is non-malleable w.r.t. adversaries that never commit to $\perp$ when receiving honestly computed commitments. Moreover, we say that an adversary is synchronous if all (i.e., both left and right) sessions are played in parallel. Clearly the design of a synchronous weak one-one NM commitment scheme can be an easier task and schemes with such limited non-malleability guarantees might exist with improved round complexity, efficiency and complexity assumptions compared to previous work achieving full-fledged non-malleability. Last but not least, the security proof of a synchronous weak NM protocol is potentially much simpler to write in a robust and easy to read way than security proofs for full-fledged non-malleability.

On the other hand, two of the three compilers that we propose require the last round of the receiver of the underlying (limited) NM commitment scheme to be simulatable without having the private coins used to compute the previous message of the receiver. This constraint clearly disappears when considering a 3-round protocol, and is clearly satisfied by any public-coin protocol. Recent work on NM commitments includes 3-round and 4-round protocols from OWFs that are also public coin as we will discuss below.

More specifically, in this work we provide the following compilers[9].

1. Given a 4-round public-coin synchronous weak one-one NM commitment scheme Π, our 1st compiler gives a 4-round one-one NM commitment scheme Π' assuming CRHFs. If Π is also weak one-many NM then Π' is a concurrent NM commitment scheme. The preliminary protocol Π_3 (when implemented in 4 rounds it needs OWFs only) of [GPR16][10], as input to our 1st compiler gives a 4-round one-one NM commitment scheme from CRHFs. The basic scheme Π_4 of [GRRV14] (i.e., their construction without the ZKAoK, still implementable with OWFs only in 4 rounds), is a candidate public-coin weak one-many NM commitment scheme that used in our 1st compiler would give 4-round concurrent NM commitments from CRHFs.

2. Given a 4-round public-coin one-one NM extractable commitment scheme Π, our 2nd compiler gives a 4-round delayed-input NMZK AoK assuming CRHFs. By considering the final protocol Π'_3 of [GPR16] (in 4 rounds it needs OWFs only) our 2nd compiler gives a 4-round delayed-input NMZK AoK from CRHFs.

3. Given a 3-round synchronous weak one-one NM commitment scheme Π, our 3rd compiler gives a 3-round extractable one-one NM commitment scheme Π' assuming OWPs secure against subexponential-time adversaries. By considering again either Π_3 or Π_4 (both are based on OWPs when implemented in 3 rounds) our 3rd compiler gives a 3-round extractable one-one NM commitment scheme Π', and the compiler of [COSV16] on input Π' gives 3-round concurrent NM commitments from subexponentially strong OWPs.

Remark 1: on the need of public-coin protocols. We require the protocols as input to two of our three compilers to be public coin because in a reduction we will have to simulate the last round of the receiver without knowing the randomness he used to compute the previous round. Of course the public-coin property satisfies the above requirement and moreover the candidate

[9]When considering a 4-round commitment scheme given as input to our compilers we always assume that the sender plays the 4-th round.

[10]In [GPR16] this preliminary public-coin commitment scheme is proved to be synchronous one-one non-malleable, and is then extended with another subprotocol in order to deal with non-synchronized adversaries obtaining one-one non-malleability.

constructions [GRRV14, GPR16] that can be used to instantiate our compilers are public-coin protocols. It is straightforward to notice that any 3-round protocol would also let us conclude successfully the reduction since there is no previous round played by the receiver. Just for simplicity we state our theorems requiring the public-coin property.

1.2 Technical Overview

We start discussing subtleties in security proofs of 4-round NM commitments and NMZK arguments given in [GRRV14]. Then we will describe our compilers.

Computational zero knowledge in parallel with (weak) NM commitments. The constructions of 4-round (one-one and concurrent) NM commitments and of 4-round NMZK arguments from OWFs given in [GRRV14] are both based on the paradigm of running in parallel two subprotocols. For simplicity and w.l.o.g. we consider now a simpler variant of [GRRV14] based on OWPs.

For concurrent NM commitments, the first subprotocol is a 3-round synchronous weak NM commitment while the second subprotocol is a delayed-input ZKAoK based on OWPs (it consists of the ZKAoK of [FS90] instantiating the underlying WIPoK given by the sender/prover with the one of [LS90]). Instead, for NMZK the first subprotocol is a NM commitment while the second subprotocol is the above delayed-input ZKAoK based on OWPs [LS90].

The security proofs of both schemes do not address the case of a synchronous MiM that mauls the witness used in the last round of the ZKAoK played in the left session. More specifically, whenever the witness used in the ZKAoK played in the left session corresponds to a legitimate witness that can be used by the honest sender in the real game, then the right session is played by the MiM correctly, committing for instance to the same message (therefore mauling). Instead, when the witness used in the left session corresponds to a trapdoor (i.e., an information that is never used as witness by a honest sender) then the right session is played by the MiM wrongly, completing the synchronous NM subprotocols with a wrong last round (still computationally indistinguishable from a valid last round) and completing the ZKAoK by using the trapdoor.

Notice that the MiM is successful in the real game, and therefore the security proof should reach a contradiction. The MiM is clearly attacking the computational WI of the execution of [LS90] inside [FS90] but unfortunately a reduction seems to be problematic. Indeed, in order to construct a reduction to the WI of [LS90] an extraction on the right session is required. However the challenger of the WI experiment can not be rewound and therefore it is not clear how to complete the left session after the rewind, in order to then extract the committed message from the right session.

Compilers for non-malleability upgrades. We take a different approach to non-malleability. Our goal is to start with a commitment scheme Π that enjoys some partial non-malleability features only. For instance, we assume that the initial scheme is non-malleable in case the adversary never commits to $\perp$ when receiving a well formed commitment. Also we consider a limitation on the scheduling of the messages, requiring that the adversary be synchronous. We notice that this type of commitment scheme corresponds to the initial subprotocol given in [GRRV14] as well as the first subprotocol given in [GPR16]. We also require some subprotocols to be public coin (see Remark 1 for further details).

Next we give compilers that upgrade these limited forms of non-malleability to the desired non-malleability. The common idea in all compilers is that during the experiment we should have a subprotocol to get a trapdoor to later on fool the adversary. We implement this step in our compilers by just assuming OWFs when 3 rounds are available to get the trapdoor (we will extract two signatures from the adversary under the same public key, following previous ideas of [DPV04b, GJO$^+$13, CPS13, COP$^+$14]), and by assuming subexponentially strong OWPs (similarly to [COSV16]) otherwise (indeed in this case there are only 2 rounds, therefore rewinds are useless and instead we will invert through brute-force search an element in the range of a OWP sent by the adversary).

Our compilers will use a delayed-input WIPoK where the prover proves knowledge of either a well formed (message, randomness) pair certifying the correctness of the execution of Π (in the case of NMZK this PoK also proves that the message is a witness) or of signatures of messages. In order to avoid the same difficulties in the security proof of [GRRV14], we require the WI to hold against subexponential-time adversaries when only 3 rounds are available, or to be statistical when 4 rounds are available. This last case can be implemented by running the construction of [LS90] using statistically hiding commitments, therefore obtaining an argument of knowledge (AoK) instead of a PoK. In the former case we avoid the above issues because statistical WI guarantees that the adversary can not notice a switch from a legitimate witness to a trapdoor. In the latter case we can still complete the reduction to the computational WI because by making use of complexity leveraging we can extract in straight-line the messages committed on the right in order to break the WI of the proof received from a challenger that is plugged in the left session.

We give two compilers for NM commitments that differ on the number of rounds and the complexity assumptions required (3 rounds and subexponentially strong OWPs in one case, 4 rounds and CRHFs in the other case). We also give a compiler for NMZK that follows the one for NM commitments under CRHFs.

Delayed-input NMZK. The above discussion is not sufficient for *delayed-input* NMZK. The reason is that the commitment scheme Π could require the message to commit before the last round. We address this point by instead relying on using Π to commit to a random string s_0 and in sending in the last round a string s_1 such that $w = s_0 \oplus s_1$ is a witness. This technique was introduced in [COSV16] to obtain delayed-input NM commitments.

Efficiency. The above description of our constructions seems to indicate that our results are interesting only from a theoretical point of view, mainly because of the $\mathcal{NP}$ reductions required by the delayed-input WIPoK [LS90]. However we point out that depending on the existence of an efficient Σ-protocol for the weak/synchronous NM commitment schemes used in our compiler, the new OR-composition technique of Σ-protocols of [CPS$^+$16a] could replace [LS90] avoiding $\mathcal{NP}$ reductions. Tweaking and instantiating properly our compilers for a practical scheme can be the subject of future work.

2 Notation and Non-Malleability Definitions

We denote the security parameter by λ and use "|" as concatenation operator (i.e., if a and b are two strings then by $a|b$ we denote the concatenation of a and b). For a finite set Q, $x \leftarrow Q$ sampling of x from Q with uniform distribution. We use the abbreviation PPT that stays for probabilistic polynomial time. We use $\mathsf{poly}(\cdot)$ to indicate a generic polynomial function.

A *polynomial-time relation* Rel (or *polynomial relation*, in short) is a subset of $\{0,1\}^* \times \{0,1\}^*$ such that membership of (x,w) in Rel can be decided in time polynomial in $|x|$. For $(x,w) \in \mathsf{Rel}$, we call x the *instance* and w a *witness* for x. For a polynomial-time relation Rel, we define the $\mathcal{NP}$-language L_{Rel} as $L_{\mathsf{Rel}} = \{x | \exists w : (x,w) \in \mathsf{Rel}\}$. Analogously, unless otherwise specified, for an $\mathcal{NP}$-language L we denote by $\mathsf{Rel_L}$ the corresponding polynomial-time relation (that is, $\mathsf{Rel_L}$ is such that $L = L_{\mathsf{Rel_L}}$).

Let A and B be two interactive probabilistic algorithms. We denote by $\langle A(\alpha), B(\beta)\rangle(\gamma)$ the distribution of B's output after running on private input β with A using private input α, both running on common input γ. Typically, one of the two algorithms receives 1^λ as input. A *transcript* of $\langle A(\alpha), B(\beta)\rangle(\gamma)$ consists of the messages exchanged during an execution where A receives a private input α, B receives a private input β and both A and B receive a common input γ. Moreover, we will refer to the *view* of A (resp. B) as the messages it received during the execution of $\langle A(\alpha), B(\beta)\rangle(\gamma)$, along with its randomness and its input. We denote by A_r an algorithm A that receives as randomness r. We say that a protocol (A, B) is public coin if B sends to A random bits only.

Standard definitions and their variants w.r.t. subexponential-time adversaries can be found in App. A. We will say that a complexity assumption is $\tilde{T}$-breakable if it can be broken with overwhelming probability by running in time $\tilde{T}$.

2.1 Non-Malleable Commitments and Zero Knowledge

Here we follow [LPV08]. Let $\Pi = (\mathsf{Sen}, \mathsf{Rec})$ be a statistically binding commitment scheme. Consider MiM adversaries that are participating in left and right sessions in which $\mathsf{poly}(\lambda)$ commitments take place. We compare between a MiM and a simulated execution. In the MiM execution the adversary $\mathcal{A}$, with auxiliary information z, is simultaneously participating in $\mathsf{poly}(\lambda)$ left and right sessions. In the left sessions the MiM adversary $\mathcal{A}$ interacts with Sen receiving commitments to values $m_1, \ldots, m_{\mathsf{poly}(\lambda)}$ using identities $\mathtt{id}_1, \ldots, \mathtt{id}_{\mathsf{poly}(\lambda)}$ of its choice. In the right session $\mathcal{A}$ interacts with Rec attempting to commit to a sequence of related values $\tilde{m}_1, \ldots, \tilde{m}_{\mathsf{poly}(\lambda)}$ again using identities of its choice $\tilde{\mathtt{id}}_1, \ldots, \tilde{\mathtt{id}}_{\mathsf{poly}}(\lambda)$. If any of the right commitments is invalid, or undefined, its value is set to $\perp$. For any i such that $\tilde{\mathtt{id}}_i = \mathtt{id}_j$ for some j, set $\tilde{m}_i = \perp$ (i.e., any commitment where the adversary uses the same identity of one of the honest senders is considered invalid). Let $\mathsf{mim}_{\Pi}^{\mathcal{A}, m_1, \ldots, m_{\mathsf{poly}(\lambda)}}(z)$ denote a random variable that describes the values $\tilde{m}_1, \ldots, \tilde{m}_{\mathsf{poly}(\lambda)}$ and the view of $\mathcal{A}$, in the above experiment. In the simulated execution, an efficient simulator S directly interacts with Rec. Let $\mathsf{sim}_{\Pi}^{S}(1^\lambda, z)$ denote the random variable describing the values $\tilde{m}_1, \ldots, \tilde{m}_{\mathsf{poly}(\lambda)}$ committed by S, and the output view of S; whenever the view contains in the i-th right session the same identity of any of the identities of the left session, then m_i is set to $\perp$.

In all the paper we denote by $\tilde{\delta}$ a value associated with the right session (where the adversary $\mathcal{A}$ plays with a receiver Rec) where δ is the corresponding value in the left session. For example, the sender commits to v in the left session while $\mathcal{A}$ commits to $\tilde{v}$ in the right session.

Definition 1 (Concurrent NM commitment scheme [LPV08])**.** *A commitment scheme is* concurrent NM with respect to commitment *(or a many-many NM commitment scheme) if, for every* PPT *concurrent MiM adversary $\mathcal{A}$, there exists a* PPT *simulator S such that for all $m_i \in \{0,1\}^{\mathsf{poly}(\lambda)}$ for $i = \{1, \ldots, \mathsf{poly}(\lambda)\}$ the following ensembles are computationally indistinguishable:*
$\{\mathsf{mim}_{\Pi}^{\mathcal{A}, m_1, \ldots, m_{\mathsf{poly}(\lambda)}}(z)\}_{z \in \{0,1\}^\star} \approx \{\mathsf{sim}_{\Pi}^{S}(1^\lambda, z)\}_{z \in \{0,1\}^\star}$.

As in [LPV08] we also consider relaxed notions of concurrent non-malleability: one-many and one-one NM commitment schemes. In a one-many NM commitment scheme, $\mathcal{A}$ participates in one left and polynomially many right sessions. In a one-one (i.e., a stand-alone secure) NM commitment scheme, we consider only adversaries $\mathcal{A}$ that participate in one left and one right session. We will make use of the following proposition of [LPV08].

Proposition 1. *Let* $(\mathsf{Sen}, \mathsf{Rec})$ *be a one-many NM commitment scheme. Then,* $(\mathsf{Sen}, \mathsf{Rec})$ *is also a concurrent (i.e., many-many) NM commitment scheme.*

We say that a commitment is valid or well formed if it can be decommitted to a message $m \neq \perp$. Following [LP11b] we say that a MiM is *synchronous* if it "aligns" the left and the right sessions; that is, whenever it receives message i on the left, it directly sends message i on the right, and vice versa.

Definition 2 (*synchronous* NM commitment scheme)**.** *A commitment scheme is* synchronous *one-one (resp., one-many) non-malleable if it is one-one (resp., one-many) NM with respect to synchronous MiM adversaries.*

Definition 3 (*weak* NM commitment scheme)**.** *A commitment scheme is* weak *one-one (resp., one-many) non-malleable if it is a one-one (resp., one-many) NM commitment scheme with respect to MiM adversaries that when receiving a well formed commitment in the left session, except with negligible probability computes well formed commitments (i.e.,* $\neq \perp$*) in the right sessions.*

We also consider the definition of a NM commitment scheme secure against a MIM $\mathcal{A}$ running in time bounded by $T = 2^{\lambda^\alpha}$ for some positive constant $\alpha < 1$. In this case we will say that a commitment scheme is T-non-malleable.

As already pointed out in previous work, when the identity is selected by the sender then the above id-based definitions guarantee non-malleability without ids as long as the MiM does not behave like a proxy (an unavoidable attack). Indeed the sender can pick as $\mathtt{id}$ the public key of a strong signature scheme signing the transcript. The MiM will have to use a different $\mathtt{id}$ or to break the signature scheme.

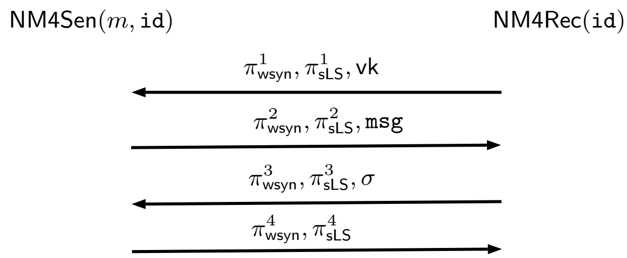
Delayed-input non-malleable zero knowledge. Following [LP11a] we give a definition that gives to the adversary the power of adaptive-input selection[11].

Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a delayed-input interactive argument system for a $\mathcal{NP}$-language L with witness relation $\mathsf{Rel_L}$. Consider a PPT MiM adversary $\mathcal{A}$ that is simultaneously participating in one left session and one right session. Before the execution starts, both $\mathcal{P}, \mathcal{V}$ and $\mathcal{A}$ receive as a common input the security parameter in unary 1^λ, and $\mathcal{A}$ receives as auxiliary input $z \in \{0,1\}^\star$.

In the left session $\mathcal{A}$ interacts with $\mathcal{P}$ using identity $\mathtt{id}$ of his choice. In the right session, $\mathcal{A}$ interacts with $\mathcal{V}$, using identity $\tilde{\mathtt{id}}$ of his choice.

Furthermore, in the left session $\mathcal{A}$, before the last round of Π, adaptively selects the statement x to be proved and the witness w, s.t $(x, w) \in \mathsf{Rel_L}$, and sends them to $\mathcal{P}$. Also, in the right session $\mathcal{A}$, during the last round of Π, adaptively selects the statement $\tilde{x}$ to be proved and sends it to $\mathcal{V}$. Let $\mathsf{View}^{\mathcal{A}}(1^\lambda, z)$ denote a random variable that describes the view of $\mathcal{A}$ in the above experiment.

[11] In [LP11a] the adversary selects the instance and a Turing machines outputs the witness in exponential time. Here we slightly deviate by 1) requiring the adversary to output also the witness (similarly to [SCO+01]) and 2) allowing the adversary to make this choice at the last round.



- vk is a verification key of a signature scheme.
- $\tau = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}})$ is the transcript of $\langle \mathsf{Sen}_{\mathsf{wsyn}}(m), \mathsf{Rec}_{\mathsf{wsyn}} \rangle(\mathtt{id})$.
- $(\pi^1_{\mathsf{sLS}}, \pi^2_{\mathsf{sLS}}, \pi^3_{\mathsf{sLS}}, \pi^4_{\mathsf{sLS}})$ is the transcript of sLS proving knowledge of either the decommitment of τ or of two signatures of two different messages w.r.t vk.

Figure 1: Informal description of our 4-round NM commitment scheme Π_{NM4Com}.

Definition 4 (Delayed-input NMZK)**.** *A delayed-input argument system $\Pi = (\mathcal{P}, \mathcal{V})$ for a $\mathcal{NP}$-language L with witness relation $\mathsf{Rel_L}$ is NM Zero Knowledge (NMZK) if for any MiM adversary $\mathcal{A}$ that participates in one left session and one right session, there exists a* PPT *machine $S(1^\lambda, z)$ such that:*

1. *The probability ensembles $\{S^1(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^\star}$ and $\{\mathsf{View}^{\mathcal{A}}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^\star}$ are computationally indistinguishable over λ, where $S^1(1^\lambda, z)$ denotes the first output of $S(1^\lambda, z)$.*
2. *Let $z \in \{0,1\}^\star$, and let $(\mathsf{View}, \tilde{w})$ denote the output of $S(1^\lambda, z)$. Let $\tilde{x}$ be the right-session statement appearing in View and let $\mathtt{id}$ and $\tilde{\mathtt{id}}$ be the identities of the left and right sessions appearing in View. If the right session is accepting and $\mathtt{id} \neq \tilde{\mathtt{id}}$, then $\mathsf{Rel_L}(\tilde{x}, \tilde{w}) = 1$.*

The above definition of NMZK allows the adversary to select statements adaptively at the last round both on left and right sessions, therefore any argument system that is NMZK according to the above definition enjoys also adaptive-input argument of knowledge and adaptive-input zero knowledge.

3 4-Round NM Commitment Scheme from CRHFs

Our construction is based on a compiler that takes as input a 4-round public-coin synchronous weak one-one NM commitment scheme $\Pi_{\mathsf{wsyn}} = (\mathsf{Sen}_{\mathsf{wsyn}}, \mathsf{Rec}_{\mathsf{wsyn}})$, a delayed-input adaptive-input statistical WI adaptive-input AoK $\mathsf{sLS} = (\mathcal{P}, \mathcal{V})$ (see Sec. A) a signature scheme, and outputs a 4-round one-one NM commitment scheme $\Pi_{\mathsf{NM4Com}} = (\mathsf{NM4Sen}, \mathsf{NM4Rec})$.

Let m be the message that $\mathsf{NM4Sen}$ wants to commit. The high-level idea of our compiler is depicted below. In the 1st round the receiver $\mathsf{NM4Rec}$ computes and sends the 1st round π^1_{sLS} of sLS and the 1st round π^1_{wsyn} of Π_{wsyn}. Also he computes a pair of signature and verification keys $(\mathsf{sk}, \mathsf{vk})$ and sends the verification key vk to the sender $\mathsf{NM4Sen}$. The sender $\mathsf{NM4Sen}$, on input the session-id $\mathtt{id}$ and the message m computes and sends the 2nd round π^2_{wsyn} of Π_{wsyn} to commit to the message m using $\mathtt{id}$ as session-id. Moreover he computes the 2nd round π^2_{sLS} of sLS and sends a random message $\mathtt{msg}$. In the 3rd round the receiver $\mathsf{NM4Rec}$ sends the 3rd round π^3_{wsyn} of Π_{wsyn}, the 3rd round of sLS and a signature σ (computed using sk) of the message $\mathtt{msg}$. In the last round $\mathsf{NM4Sen}$ verifies whether or not σ is a valid signature for $\mathtt{msg}$. If σ is a valid signature, then $\mathsf{NM4Sen}$ computes the last round π^4_{wsyn} of Π_{wsyn}, the 4th round π^4_{sLS} of sLS and sends $(\pi^4_{\mathsf{wsyn}}, \pi^4_{\mathsf{sLS}})$ to the receiver $\mathsf{NM4Rec}$. The delayed-input adaptive-input statistical WI adaptive-input AoK protocol sLS is used by $\mathsf{NM4Sen}$ to prove either knowledge of a message and randomness consistent with the transcript computed using Π_{wsyn} or knowledge of signatures of two different messages w.r.t vk.

Our compiler needs the following tools:

1. a 4-round public-coin synchronous weak one-one NM commitment scheme $\Pi_{\mathsf{wsyn}} = (\mathsf{Sen}_{\mathsf{wsyn}}, \mathsf{Rec}_{\mathsf{wsyn}})$;
2. a signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$;
3. a delayed-input statistical WIAoK protocol $\mathsf{sLS} = (\mathcal{P}, \mathcal{V})$ for the language

$$L = \big\{\big(\tau = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}), \mathtt{id}, \mathsf{vk}\big) : \exists\ (m, \mathtt{dec}, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2) \text{ s.t.}$$
$$\big(\mathsf{Rec}_{\mathsf{wsyn}} \text{ on input } (\tau, m, \mathtt{dec}, \mathtt{id}) \text{ accepts } m \text{ as a decommitment of } \tau \ \mathtt{OR}$$
$$\big(\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma_1) = 1 \ \mathtt{AND}\ \mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_2, \sigma_2) = 1 \ \mathtt{AND}\ \mathtt{msg}_1 \neq \mathtt{msg}_2\big)\big)\big\}$$

that is adaptive-input statistical WI and an adaptive AoK for the corresponding relation $\mathsf{Rel_L}$. We remark that to execute sLS the instance x is not needed until the last round but the instance length is required from the onset of the protocol. We will refer to the instance length as $\ell = |x|$. Fig. 2 describes in details our 4-round one-one NM.

Theorem 1. *Suppose Π_{wsyn} is a 4-round public-coin synchronous weak one-one NM commitment scheme and CRHFs exist then Π_{NM4Com} is a one-one NM commitment scheme.*

Proof. The security proof is divided in two parts. In the 1st part we prove that Π_{NM4Com} is indeed a commitment scheme. Then we prove that Π_{NM4Com} is a non-malleable commitment scheme.

Lemma 1. Π_{NM4Com} *is a statistically-binding computationally-hiding commitment scheme.*

Proof. **Correctness.** The correctness follows directly from the completeness of sLS, the correctness of Π_{wsyn} and from the validity of the signature scheme Σ.

Statistical Binding. Observe that the message given in output in the decommitment phase of Π_{NM4Com} is the message committed using Π_{wsyn}. Moreover the decommitment of Π_{NM4Com} coincides with the decommitment of Π_{wsyn}. Since Π_{wsyn} is statistically binding then so is Π_{NM4Com}.

Hiding. Following Def. 11 to prove the hiding of Π_{NM4Com} we have to show that an experiment $\mathsf{ExpHiding}^0_{\mathcal{A},\Pi_{\mathsf{NM4Com}}}(\lambda)$ where $\mathsf{NM4Sen}$ commits to a message m_0 is computationally indistinguishable from the experiment $\mathsf{ExpHiding}^1_{\mathcal{A},\Pi_{\mathsf{NM4Com}}}(\lambda)$ where $\mathsf{NM4Sen}$ commits to a message m_1. Therefore, in order to prove hiding, we consider the following hybrid experiments.

- The 1st hybrid experiment $\mathcal{H}^0(\lambda)$ differs from $\mathsf{ExpHiding}^0_{\mathcal{A},\Pi_{\mathsf{NM4Com}}}(\lambda)$ in the witness used to compute the messages of sLS. In more details in $\mathcal{H}^0(\lambda)$ $\mathsf{NM4Sen}$ extracts, by rewinding from the 3rd round to the 2nd round, two signatures of two different messages[12]. These two signatures are used as a witness to compute messages of sLS. The adaptive-input statistical WI of sLS guarantees that $\mathcal{H}^0(\lambda)$ and $\mathsf{ExpHiding}^0_{\mathcal{A},\Pi_{\mathsf{NM4Com}}}(\lambda)$ are statistically close.

[12] In the proof of Lemma 6 we show that the extraction fails with negligible probability. The same analysis applies here for the proof of hiding.

Common input: security parameter λ, instance length ℓ, $\mathsf{NM4Sen}$'s identity $\mathtt{id} \in \{0,1\}^\lambda$.
Input to $\mathsf{NM4Sen}$**:** $m \in \{0,1\}^{\mathsf{poly}\{\lambda\}}$.
Commitment phase:

1. $\mathsf{NM4Rec} \rightarrow \mathsf{NM4Sen}$
   1. Run $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$.
   2. Run $\mathcal{V}$ on input 1^λ and ℓ thus obtaining the 1st round π^1_{sLS} of sLS.
   3. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $1^\lambda, \mathtt{id}$ thus obtaining the 1st round π^1_{wsyn} of Π_{wsyn}.
   4. Send $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{wsyn}})$ to $\mathsf{NM4Sen}$.
2. $\mathsf{NM4Sen} \rightarrow \mathsf{NM4Rec}$
   1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^\lambda, \mathtt{id}$, π^1_{wsyn} and m thus obtaining the 2nd round π^2_{wsyn} of Π_{wsyn}.
   2. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the 2nd round π^2_{sLS} of sLS.
   3. Pick a message $\mathtt{msg} \leftarrow \{0,1\}^\lambda$.
   4. Send $(\pi^2_{\mathsf{wsyn}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg})$ to $\mathsf{NM4Rec}$.
3. $\mathsf{NM4Rec} \rightarrow \mathsf{NM4Sen}$
   1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input π^2_{wsyn} thus obtaining the 3rd round π^3_{wsyn} of Π_{wsyn}.
   2. Run $\mathcal{V}$ on input π^2_{sLS} thus obtaining the 3rd round π^3_{sLS} of sLS.
   3. Run $\mathsf{Sign}(\mathsf{sk}, \mathtt{msg})$ to obtain a signature σ of the message $\mathtt{msg}$.
   4. Send $(\pi^3_{\mathsf{wsyn}}, \pi^3_{\mathsf{sLS}}, \sigma)$ to $\mathsf{NM4Sen}$.
4. $\mathsf{NM4Sen} \rightarrow \mathsf{NM4Rec}$
   1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}, \sigma) \neq 1$ then abort, continue as follows otherwise.
   2. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input π^3_{wsyn} thus obtaining the 4th round π^4_{wsyn} of Π_{wsyn} and the decommitment information $\mathtt{dec}_{\mathsf{wsyn}}$.
   3. Set $x = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id}, \mathsf{vk})$ and $w = (m, \mathtt{dec}_{\mathsf{wsyn}}, \perp, \perp, \perp, \perp)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and π^3_{sLS} thus obtaining the 4th round π^4_{sLS} of sLS.
   4. Send $(\pi^4_{\mathsf{wsyn}}, \pi^4_{\mathsf{sLS}})$ to $\mathsf{NM4Rec}$.
5. $\mathsf{NM4Rec}$: Set $x = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id}, \mathsf{vk})$ and abort iff $(\pi^1_{\mathsf{sLS}}, \pi^2_{\mathsf{sLS}}, \pi^3_{\mathsf{sLS}}, \pi^4_{\mathsf{sLS}})$ is not accepting for $\mathcal{V}$ with respect to x.

Decommitment phase:

1. $\mathsf{NMSen} \rightarrow \mathsf{NMRec}$: Send $(\mathtt{dec}_{\mathsf{wsyn}}, m)$ to NMRec.
2. NMRec: accept m as the committed message if and only if $\mathsf{Rec}_{\mathsf{wsyn}}$, on input $(m, \mathtt{dec}_{\mathsf{wsyn}})$, accepts m as the committed message of $(\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id})$.

Figure 2: Our 4-round NM commitment scheme Π_{NM4Com}.

- The 2nd hybrid $\mathcal{H}^1(\lambda)$ differs from $\mathcal{H}^0(\lambda)$ only in the committed message. More specifically, $\mathsf{NM4Sen}$ runs $\mathsf{Sen}_{\mathsf{wsyn}}$ to commits to m_1 instead of m_0. The indistinguishability between $\mathcal{H}^0(\lambda)$ and $\mathcal{H}^1(\lambda)$ comes from the computationally-hiding property of Π_{wsyn}. Note that the reduction to the hiding of Π_{wsyn} is possible because the extraction of the signatures does not rewind the challenger of Π_{wsyn}.

The proof ends with the observation that by the adaptive-input statistical WI of sLS the experiments $\mathcal{H}^1(\lambda)$ and $\mathsf{ExpHiding}^1_{\mathcal{A},\Pi_{\mathsf{NM4Com}}}(\lambda)$ are statistically close. $\square$

We give now an overview of the non-malleability proof, while the formal proof can be found in App. B.1. We want to show that the committed value and the view of $\mathcal{A}_{\mathsf{NM4Com}}$ when interacting with $\mathsf{NM4Sen}$ that commits to a message m is indistinguishable from the committed value and the view of a simulator. The proof is divided in two cases. In the 1st case we consider an adversarial MiM $\mathcal{A}_{\mathsf{NM4Com}}$ that acts in a synchronized way, while in the 2nd case we deal with the case of a non-synchronized $\mathcal{A}_{\mathsf{NM4Com}}$. The proof of the 1st case goes through a sequence of hybrid experiments summarized below.

- The 1st hybrid is $\mathcal{H}_1^m(z)$ and in the left session $\mathsf{NM4Sen}$ commits to m, while in the right session $\mathsf{NM4Rec}$ interacts with $\mathcal{A}_{\mathsf{NM4Com}}$. We prove that in the right session $\mathcal{A}_{\mathsf{NM4Com}}$ does not commit to a message $\tilde{m} = \perp$. By contradiction if $\mathcal{A}_{\mathsf{NM4Com}}$ commits to $\tilde{m} = \perp$ then the witness used to complete an accepting transcript for sLS consists of two valid signatures of two different messages. Then, using the AoK property of sLS we can contradict the security property of $\mathsf{\Sigma}$. Note that this hybrid corresponds to the real experiment where $\mathcal{A}_{\mathsf{NM4Com}}$ interacts with $\mathsf{NM4Sen}$ in the left session.

- The 2nd hybrid is $\mathcal{H}_2^m(z)$ and differs from $\mathcal{H}_1^m(z)$ only in the witness used to compute the messages of sLS in the left session. In more details, we rewind the adversary $\mathcal{A}_{\mathsf{NM4Com}}$ from the 3rd to the 2nd round of the left session to extract two signatures σ_1, σ_2 of two different messages $(\mathtt{msg}_1, \mathtt{msg}_2)$ and we use them as witness to execute sLS in the left session. From the adaptive-input statistical WI property of sLS follows that the committed value and the view of $\mathcal{A}_{\mathsf{NM4Com}}$ do not change when moving from $\mathcal{H}_1^m(z)$ to $\mathcal{H}_2^m(z)$.

- The 3rd hybrid that we consider is $\mathcal{H}_1^0(z)$ and differs from the first hybrid experiment that we have considered $\mathcal{H}_1^m(z)$ in the committed message. Indeed in this case, the message committed in the left session is 0^λ. We observe that $\mathcal{H}_1^0(z)$ actually is the simulated game. Note that also in this hybrid we can argue that in the right session $\mathcal{A}_{\mathsf{NM4Com}}$ does not commit to a message $\tilde{m} = \perp$, for the same reason explained in hybrid $\mathcal{H}_1^m(z)$.

- The 4th hybrid experiment that we consider is $\mathcal{H}_2^0(m)$ and it differs from $\mathcal{H}_1^0(z)$ only in the witness used to compute the sLS transcript. In more details, we rewind the adversary $\mathcal{A}_{\mathsf{NM4Com}}$ from the 3rd to the 2nd round of the left session to extract two signatures σ_1, σ_2 of two different messages $(\mathtt{msg}_1, \mathtt{msg}_2)$ and we use them as witness to execute sLS in the left session. From the adaptive-input statistical WI of sLS we have that the committed value and the view of $\mathcal{A}_{\mathsf{NM4Com}}$ do not change when moving from $\mathcal{H}_1^0(z)$ to $\mathcal{H}_2^0(z)$.

To conclude the proof we need to show that the view and the committed message of $\mathcal{A}_{\mathsf{NM4Com}}$ that acts in $\mathcal{H}_1^m(z)$ are indistinguishable from the view and the committed message of $\mathcal{A}_{\mathsf{NM4Com}}$ that acts in $\mathcal{H}_1^0(z)$. Given the indistinguishability of the hybrids discussed above, it only remains to show that the view and the committed message of $\mathcal{H}_2^m(z)$ are indistinguishable from the view and the committed message of $\mathcal{H}_2^0(z)$. This is ensured by the synchronous weak one-one non-malleability of Π_{wsyn}. Indeed, observe that here we need only to use a *weak* synchronous one-one NM commitment because we are guaranteed (from the previous arguments) that whenever $\mathcal{A}_{\mathsf{NM4Com}}$ completes a commitment in the right session, the corresponding message committed through Π_{wsyn} is different from $\perp$ with overwhelming probability, and this holds both in $\mathcal{H}_2^m(z)$ and in $\mathcal{H}_2^0(z)$. One additional caveat that we have to address in this reduction is due to the rewinds executed in the experiment in the left session to extract signatures. These rewinds can affect the straight-line execution of the MiM adversary for Π_{wsyn} that we want to construct. This is the point where the

$\mathcal{P}_{\mathsf{ZK}}(\mathtt{id})$ $\qquad\qquad\qquad$ $\mathcal{V}_{\mathsf{ZK}}(\mathtt{id})$

$\xleftarrow{\pi^1_{\mathsf{ext}}(s_0),\, \pi^1_{\mathsf{sLS}},\, \mathsf{vk}}$

$\xrightarrow{\pi^2_{\mathsf{ext}}(s_0),\, \pi^2_{\mathsf{sLS}},\, \mathtt{msg}}$

$\xleftarrow{\pi^3_{\mathsf{ext}}(s_0),\, \pi^3_{\mathsf{sLS}},\, \sigma}$

Upon receiving x, w s.t. $(x, w) \in \mathsf{Rel_L}$ set $s_1 = w \oplus s_0$ $\xrightarrow{s_1,\, \pi^4_{\mathsf{ext}}(s_0),\, \pi^4_{\mathsf{sLS}}}$

- vk is a verification key of a signature scheme.
- $\tau = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}})$ is the transcript of $\langle \mathsf{Sen_{ext}}(m), \mathsf{Rec_{ext}} \rangle(\mathtt{id})$.
- $(\pi^1_{\mathsf{sLS}}, \pi^2_{\mathsf{sLS}}, \pi^3_{\mathsf{sLS}}, \pi^4_{\mathsf{sLS}})$ is the transcript of sLS proving knowledge of either the decommitment of τ to a message s_0 s.t. $(x, w = s_0 \oplus s_1) \in \mathsf{Rel_L}$ or of two valid signatures of two different messages w.r.t vk.

Figure 3: Informal description of our delayed-input 4-round NMZK AoK Π_{ZK}.

public-coin property of Π_{wsyn} is used. Indeed, in the reduction we will not need to rewind the external receiver of Π_{wsyn} because we can easily simulate his answers.

The proof for the asynchronous case is much simpler and relies on the hiding of Π_{NM4Com}. More precisely we observe that in the asynchronous scheduling it is possible to rewind the adversary $\mathcal{A}_{\mathsf{NM4Com}}$ (by changing the 3rd round), without rewinding the sender in the left session. This make us able to extract the witness used in sLS, that with overwhelming probability is the committed message. This allow us to reach a contradiction by breaking the hiding of Π_{NM4Com}.

Theorem 2. *Suppose Π_{wom} is a 4-round public-coin weak one-many NM commitment scheme and CRHFs exist then Π_{NM4Com} is a concurrent NM commitment scheme.*

The proof of this theorem is similar to the proof of Theorem 1. The main difference is in the NM security proof, where there is no need to distinguish two cases since Π_{wom} is only weak non-malleable (and not restricted to being synchronous). More specifically we can consider the sequence of hybrids listed above for the synchronized case of Theorem 1, and consider a MiM adversary that has no restriction on the scheduling of the messages. The proof still works since the indistinguishability between $\mathcal{H}_2^m$ and $\mathcal{H}_2^0$ can rely directly on the one-many non-malleability of Π_{wom}. We go from one-many to concurrent non-malleability by using Proposition 1.

4 4-Round NMZK from CRHFs

Our construction is based on a compiler that takes as input any 4-round public-coin extractable one-one NM commitment scheme $\Pi_{\mathsf{ext}} = (\mathsf{Sen_{ext}}, \mathsf{Rec_{ext}})$, a delayed-input adaptive-input statistical WI adaptive-input AoK $\mathsf{sLS} = (\mathcal{P}, \mathcal{V})$, a signature scheme, and outputs a delayed-input 4-round NMZK AoK $\Pi_{\mathsf{ZK}} = (\mathcal{P}_{\mathsf{ZK}}, \mathcal{V}_{\mathsf{ZK}})$ for the $\mathcal{NP}$-language L and corresponding relation $\mathsf{Rel_L}$.

The high-level idea of our compiler is depicted in Fig. 3. In the 1st round $\mathcal{V}_{\mathsf{ZK}}$ computes and sends the 1st round π^1_{sLS} of sLS and the 1st round π^1_{ext} of Π_{ext} to $\mathcal{P}_{\mathsf{ZK}}$. Also $\mathcal{V}_{\mathsf{ZK}}$ computes a pair of signature and verification keys $(\mathsf{sk}, \mathsf{vk})$ and sends vk to $\mathcal{P}_{\mathsf{ZK}}$. $\mathcal{P}_{\mathsf{ZK}}$ input the session-id $\mathtt{id}$, picks a random string s_0, then computes and sends to $\mathcal{V}_{\mathsf{ZK}}$ the 2nd round π^2_{ext} of Π_{ext} to commit to the message s_0 using $\mathtt{id}$ as session-id. Moreover $\mathcal{P}_{\mathsf{ZK}}$ computes the 2nd round π^2_{sLS} of sLS and sends it along with a random message $\mathtt{msg}$ to $\mathcal{V}_{\mathsf{ZK}}$. In the 3rd round $\mathcal{V}_{\mathsf{ZK}}$ sends the 3rd round π^3_{ext} of Π_{ext}, the 3rd round of sLS and a signature σ (computed using sk) of $\mathtt{msg}$ to $\mathcal{P}_{\mathsf{ZK}}$. In the last round upon receiving x, w s.t. $(x, w) \in \mathsf{Rel_L}$, $\mathcal{P}_{\mathsf{ZK}}$ verifies whether or not σ is a valid signature for $\mathtt{msg}$. If σ is a valid signature, then $\mathcal{P}_{\mathsf{ZK}}$ computes the last round π^4_{ext} of Π_{ext} and the 4th round π^4_{sLS} of sLS. Finally, $\mathcal{P}_{\mathsf{ZK}}$ sets $s_1 = s_0 \oplus w$ and sends $(\pi^4_{\mathsf{ext}}, \pi^4_{\mathsf{sLS}}, s_1)$ to $\mathcal{V}_{\mathsf{ZK}}$. The delayed-input statistical WIAoK protocol sLS is used by $\mathcal{P}_{\mathsf{ZK}}$ to prove either 1) knowledge of a message s_0 and randomness that are consistent with the transcript computed using Π_{ext} and s.t. $(x, s_1 \oplus s_0) \in \mathsf{Rel_L}$ or 2) knowledge of signatures of two different messages w.r.t vk.

For constructing our $\Pi_{\mathsf{ZK}} = (\mathcal{P}_{\mathsf{ZK}}, \mathcal{V}_{\mathsf{ZK}})$ for the $\mathcal{NP}$-language L and corresponding relation $\mathsf{Rel_L}$ we need the following tools:

1. a 4-round public-coin extractable one-one NM commitment scheme $\Pi_{\mathsf{ext}} = (\mathsf{Sen_{ext}}, \mathsf{Rec_{ext}})$;
2. a signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$;
3. a delayed-input adaptive-input statistical WIAoK protocol $\mathsf{sLS} = (\mathcal{P}, \mathcal{V})$ for the language

$$\Lambda = \Big\{\big(\tau = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}}), \mathtt{id}, \mathsf{vk}, x, s_1\big) : \exists\ (s_0, \mathtt{dec}, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2) \text{ s.t.}$$
$$\Big(\big(\mathsf{Rec_{ext}} \text{ on input } (\tau, s_0, \mathtt{dec}, \mathtt{id}) \text{ accepts } s_0 \text{ as a decommitment of } \tau \ \mathtt{AND}\ (x, s_0 \oplus s_1) \in \mathsf{Rel_L}\big)\ \mathtt{OR}$$
$$\big(\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma_1) = 1\ \mathtt{AND}\ \mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_2, \sigma_2) = 1\ \mathtt{AND}\ \mathtt{msg}_1 \neq \mathtt{msg}_2\big)\Big)\Big\}$$

that is adaptive-input statistical WI and adaptive-input AoK for the corresponding relation Rel_Λ.

Lemma 2. Π_{ZK} *enjoys delayed-input completeness.*

Proof. First we observe that completeness follows directly from the completeness of sLS, the correctness of Π_{ext} and the validity of the signature scheme Σ. Delayed-input completeness follows from the delayed-input completeness of sLS and from the observation that $\mathcal{P}_{\mathsf{ZK}}$ does not need to know the witness to run Π_{ext}. We stress that Π_{ext} is not required to enjoy a delayed-input property. □

Theorem 3. *If* Π_{ext} *is a 4-round public-coin extractable one-one NM commitment scheme and CRHFs exist then* Π_{ZK} *is a delayed-input NMZK AoK for* $\mathcal{NP}$*.*

We will refer to the simulated experiment as the experiment where $\mathsf{Sim_{ZK}}$ interacts with the adversary emulating both a prover and a verifier. The simulator works in a pretty straight-forward way. It commits to a random message, it extracts a second signature from the left session and completes the execution generating the first output according to Def. 4. Then it extracts the witness from the extractable commitment Π_{ext} played by the adversary in the right session (see Fig. 8 for a detailed description of $\mathsf{Sim_{ZK}}$). Obviously we will have to show that the probability that the message extracted is not a witness for the statement proved by the adversary in the right session is negligible.

We will give a lemma for each of the two properties of Def. 4.

Lemma 3. $\{\mathsf{Sim}^1_{\mathsf{ZK}}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*} \approx \{\mathsf{View}^{\mathcal{A}}(1^\lambda, z)\}_{\lambda \in \mathbb{N}, z \in \{0,1\}^*}$, *where* $\mathsf{Sim}^1_{\mathsf{ZK}}(1^\lambda, z)$ *denotes the 1st output of* $\mathsf{Sim_{ZK}}$*.*

In order to prove the above lemma we consider an hybrid experiment $\mathcal{H}_1(1^\lambda, z)$. $\mathcal{H}_1(1^\lambda, z)$ differs from the real execution of Π_{ZK} in the witness used to compute messages of sLS. In more

Common input: security parameter λ, the instance length ℓ of sLS and $\mathcal{P}_{\mathsf{ZK}}$'s identity $\mathtt{id} \in \{0,1\}^\lambda$, and the instance x available only at the last round.
Private input of $\mathcal{P}_{\mathsf{ZK}}$: w s.t. $(x, w) \in \mathsf{Rel_L}$, available only at the last round.

1. $\mathcal{V}_{\mathsf{ZK}} \rightarrow \mathcal{P}_{\mathsf{ZK}}$
    1. Run $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$.
    2. Run $\mathcal{V}$ on input 1^λ and ℓ thus obtaining the 1st round π^1_{sLS} of sLS.
    3. Run $\mathsf{Rec_{ext}}$ on input $1^\lambda, \mathtt{id}$ thus obtaining the 1st round π^1_{ext} of Π_{ext}.
    4. Send $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{ext}})$ to $\mathcal{P}_{\mathsf{ZK}}$.
2. $\mathcal{P}_{\mathsf{ZK}} \rightarrow \mathcal{V}_{\mathsf{ZK}}$
    1. Pick at random s_0 s.t. $|s_0|$ is the witness length for an instance of L.
    2. Run $\mathsf{Sen_{ext}}$ on input $1^\lambda, \mathtt{id}$, π^1_{ext} and s_0 thus obtaining the 2nd round π^2_{ext} of Π_{ext}.
    3. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the 2nd round π^2_{sLS} of sLS.
    4. Pick a message $\mathtt{msg} \leftarrow \{0,1\}^\lambda$.
    5. Send $(\pi^2_{\mathsf{ext}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg})$ to $\mathcal{V}_{\mathsf{ZK}}$.
3. $\mathcal{V}_{\mathsf{ZK}} \rightarrow \mathcal{P}_{\mathsf{ZK}}$
    1. Run $\mathsf{Rec_{ext}}$ on input π^2_{ext} thus obtaining the 3rd round π^3_{ext} of Π_{ext}.
    2. Run $\mathcal{V}$ on input π^2_{sLS} thus obtaining the 3rd round π^3_{sLS} of sLS.
    3. Run $\mathsf{Sign}(\mathsf{sk}, \mathtt{msg})$ to obtain a signature σ of the message $\mathtt{msg}$.
    4. Send $(\pi^3_{\mathsf{ext}}, \pi^3_{\mathsf{sLS}}, \sigma)$ to $\mathcal{P}_{\mathsf{ZK}}$.
4. $\mathcal{P}_{\mathsf{ZK}} \rightarrow \mathcal{V}_{\mathsf{ZK}}$
    1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}, \sigma) \neq 1$ then abort, continue as follows otherwise.
    2. Set $s_1 = s_0 \oplus w$.
    3. Run $\mathsf{Sen_{ext}}$ on input π^3_{ext} thus obtaining the 4th round π^4_{ext} of Π_{ext} and the decommitment information $\mathtt{dec}_{\mathsf{ext}}$.
    4. Set $x_{\mathsf{sLS}} = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and $w_{\mathsf{sLS}} = (s_0, \mathtt{dec}_{\mathsf{ext}}, \bot, \bot, \bot, \bot)$ with $|x_{\mathsf{sLS}}| = \ell$. Run $\mathcal{P}$ on input x_{sLS}, w_{sLS} and π^3_{sLS} thus obtaining the forth round π^4_{sLS} of sLS.
    5. Send $(\pi^4_{\mathsf{ext}}, \pi^4_{\mathsf{sLS}}, s_1)$ to $\mathcal{V}_{\mathsf{ZK}}$.
5. $\mathcal{V}_{\mathsf{ZK}}$: Set $x_{\mathsf{sLS}} = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and accept iff $(\pi^1_{\mathsf{sLS}}, \pi^2_{\mathsf{sLS}}, \pi^3_{\mathsf{sLS}}, \pi^4_{\mathsf{sLS}})$ is accepting for $\mathcal{V}$ with respect to x_{sLS}.

Figure 4: Our 4-round delayed-input NMZK argument of knowledge Π_{ZK}.

details in $\mathcal{H}_1(1^\lambda, z)$ $\mathcal{P}_{\mathsf{ZK}}$ extracts (through rewinds), two signatures of different messages[13] that are used as witness for sLS. Let $\{\mathsf{View}^{\mathcal{A}_{\mathsf{ZK}}}_{\mathcal{H}_1}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$ denote a random variable that describes the view of $\mathcal{A}_{\mathsf{ZK}}$ in $\mathcal{H}_1(1^\lambda, z)$. The adaptive-input statistical WI of sLS and the negligible probability of failing in extracting a second signature guarantee that $\{\mathsf{View}^{\mathcal{A}_{\mathsf{ZK}}}_{\mathcal{H}_1}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$ and $\{\mathsf{View}^{\mathcal{A}_{\mathsf{ZK}}}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$ are statistically close.

Observe now that the only difference between $\mathcal{H}_1(1^\lambda, z)$ and the simulated execution is the message committed using Π_{ext}. In more details, let x be the adaptively chosen statement proved by $\mathcal{P}_{\mathsf{ZK}}$. In $\mathcal{H}_1(1^\lambda, z)$ $\mathcal{P}_{\mathsf{ZK}}$ commits using Π_{ext} to a value s_0 s.t. $s_1 = w \oplus s_0$ (where $(x, w) \in \mathsf{Rel_L}$). Instead in the simulated experiment $\mathsf{Sim_{ZK}}$ commits to a random string. Now we can claim that $\{\mathsf{Sim}^1_{\mathsf{ZK}}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$ and $\{\mathsf{View}^{\mathcal{A}_{\mathsf{ZK}}}_{\mathcal{H}_1}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$ are computationally indistinguishable by using the computationally-hiding property of Π_{ext}. Informally, suppose by contradiction that there exist an adversary $\mathcal{A}_{\mathsf{ZK}}$ and a distinguisher $\mathcal{D}_{\mathsf{ZK}}$ such that $\mathcal{D}_{\mathsf{ZK}}$ distinguishes

[13] In the proof of Lemma 4 we show that the extraction fails with negligible probability. The same analysis applies here.

$\{\mathsf{Sim}^1_{\mathsf{ZK}}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$ from $\{\mathsf{View}^{\mathcal{A}_{\mathsf{ZK}}}_{\mathcal{H}_1}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$. Then we can construct an adversary $\mathcal{A}_{\mathsf{Hiding}}$ that breaks the computationally hiding of Π_{ext} in the following way. $\mathcal{A}_{\mathsf{Hiding}}$ sends to the challenger of the hiding game $\mathcal{C}_{\mathsf{Hiding}}$ two random messages (m_0, m_1). Then $\mathcal{A}_{\mathsf{Hiding}}$ acts as $\mathcal{P}_{\mathsf{ZK}}$ except for messages of Π_{ext} for which he acts as proxy between $\mathcal{C}_{\mathsf{Hiding}}$ and $\mathcal{A}_{\mathsf{ZK}}$. When $\mathcal{A}_{\mathsf{Hiding}}$ computes the last round of the left session $\mathcal{A}_{\mathsf{Hiding}}$ sets $s_1 = m_0 \oplus w$. At the end of the execution $\mathcal{A}_{\mathsf{Hiding}}$ runs $\mathcal{D}_{\mathsf{ZK}}$ and outputs what $\mathcal{D}_{\mathsf{ZK}}$ outputs. It easy to see that if $\mathcal{C}_{\mathsf{Hiding}}$ commits to m_0 then, $\mathcal{A}_{\mathsf{ZK}}$ acts as in $\mathcal{H}_1(1^\lambda, z)$, otherwise he acts as in the simulated experiment. Note that the reduction to the hiding property of Π_{ext} is possible because the rewinds to extract a second signature do not affect the execution with the challenger of Π_{ext} that remains straight-line. Thus we have proved that $\{\mathsf{View}^{\mathcal{A}_{\mathsf{ZK}}}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*} \equiv_s \{\mathsf{View}^{\mathcal{A}_{\mathsf{ZK}}}_{\mathcal{H}_1}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*} \approx \{\mathsf{Sim}^1_{\mathsf{ZK}}(1^\lambda, z)\}_{\lambda\in\mathbb{N}, z\in\{0,1\}^*}$.

Lemma 4. *Let $\tilde{x}$ be the right-session statement appearing in $\mathsf{View} = \mathsf{Sim}^1_{\mathsf{ZK}}(1^\lambda, z)$ and let $\mathtt{id}$ and $\tilde{\mathtt{id}}$ be the identities of the left and right sessions appearing in View. If the right session is accepting and $\mathtt{id} \neq \tilde{\mathtt{id}}$, then except with negligible probability, the second output of $\mathsf{Sim}_{\mathsf{ZK}}(1^\lambda, z)$ is $\tilde{w}$ such that $\mathsf{Rel}_{\mathsf{L}}(\tilde{x}, \tilde{w}) = 1$.*

The formal proof can be found in App. B.2. Here we give an overview. The proof relies on hybrid experiments to prove that $\mathcal{A}_{\mathsf{ZK}}$ commits to $\tilde{s}_0$ s.t. $(\tilde{x}, \tilde{s}_0 \oplus \tilde{s}_1) \in \mathsf{Rel}_{\mathsf{L}}$[14] through Π_{ext} in the simulated experiment.

- The 1st hybrid is $\mathcal{H}_1(z)$ in which in the left session $\mathcal{P}_{\mathsf{ZK}}$ interacts with $\mathcal{A}_{\mathsf{ZK}}$ and in the right session $\mathcal{A}_{\mathsf{ZK}}$ interacts with $\mathcal{V}_{\mathsf{ZK}}$. We refer to this hybrid experiment as $\mathcal{H}_1(z)$. Now we prove that in the right session of $\mathcal{H}_1(z)$ the MiM adversary $\mathcal{A}_{\mathsf{ZK}}$ commits to the witness. Let $\tilde{x}$ be the adaptively chosen theorem proved by $\mathcal{A}_{\mathsf{ZK}}$. By contradiction if $\mathcal{A}_{\mathsf{ZK}}$ commits to a message s_0' s.t. $(\tilde{x}, \tilde{s}_0' \oplus \tilde{s}_1) \notin \mathsf{Rel}_{\mathsf{L}}$, then the witness used to complete an accepting transcript for sLS consists of two valid signatures of two different messages. Then, by using the adaptive-input AoK property of sLS we can reach a contradiction by breaking the security of Σ. Note that this hybrid corresponds to the real experiment where $\mathcal{A}_{\mathsf{ZK}}$ interacts with $\mathcal{P}_{\mathsf{ZK}}$ in the left session.

- The 2nd hybrid is $\mathcal{H}_2(z)$ and differs from $\mathcal{H}_1(z)$ only in the witness used to compute messages of sLS in the left session. In more details, we rewind the adversary $\mathcal{A}_{\mathsf{ZK}}$ from the 3rd to the 2nd round of the left session to extract two signatures σ_1, σ_2 of two different messages $(\mathtt{msg}_1, \mathtt{msg}_2)$ and we use them as witness to execute sLS in the left session. From the adaptive-statistical WI of sLS it follows that the distribution of the message committed by $\mathcal{A}_{\mathsf{ZK}}$ does not change when moving from $\mathcal{H}_1(z)$ to $\mathcal{H}_2(z)$.

- The 3rd hybrid is $\mathcal{H}_3(z)$. The only difference between this hybrid and the previous one is that both s_0 and s_1 are random strings. From the non-malleability property of Π_{ext} it follows that the distribution of the message committed by $\mathcal{A}_{\mathsf{ZK}}$ does not change when switching from $\mathcal{H}_2(z)$ to $\mathcal{H}_3(z)$. This is again a delicate reduction because it requires to show a successful MiM for Π_{ext} that is supposed to work in straight-line. However the above experiment requires to rewind the adversary in order to extract a second signature. As already discussed in the previous section, this is the place where the public-coin property is used. Indeed this allows us to simulate the additional answers of the honest receiver of Π_{ext} that are needed because of the rewinds performed to extract a second signature.

[14]For simplicity in the rest of the proof we say that a player commits to a witness when he commits to s_0 and sends s_1 in the last round s.t. $(x, s_0 \oplus s_1) \in \mathsf{Rel}_{\mathsf{L}}$.

Note that $\mathcal{H}_3(z)$ corresponds to the the simulated experiment, this implies that also in the simulated game $\mathcal{A}_{\mathsf{ZK}}$ commits to the witness. Therefore, our simulator can use the extractor of Π_{ext} to get the witness $\tilde{w}$ s.t. $(\tilde{x}, \tilde{w}) \in \mathsf{Rel_L}$, where $\tilde{x}$ is the adaptively chosen theorem proved by $\mathcal{A}_{\mathsf{ZK}}$.

5 3-Round NM Commitments from Strong OWPs

Our construction is based on a compiler that takes as input a 3-round synchronous weak one-one NM commitment scheme $\Pi_{\mathsf{wsyn}} = (\mathsf{Sen_{wsyn}}, \mathsf{Rec_{wsyn}})$, a OWP f, an LS WIPoK for $\mathcal{NP}$ LS, and outputs a 3-round one-one NM commitment scheme $\Pi_{\mathsf{NMCom}} = (\mathsf{NMSen}, \mathsf{NMRec})$.

Let m be the message that NMSen wants to commit. The high-level idea of our compiler is depicted in Fig. 5. The sender NMSen, on input the session-id $\mathtt{id}$ and the message m, computes the 1st round of the protocol by sending the 1st round $\mathsf{a_{LS}}$ of LS and the 1st round $\mathsf{a_{wsyn}}$ of Π_{wsyn} (to commit to the message m using $\mathtt{id}$ as session-id). In the 2nd round the receiver NMRec sends challenges $\mathsf{c_{wsyn}}$ and $\mathsf{c_{LS}}$ of Π_{wsyn} and LS, also picks and sends an element Y in the range of f. In the 3rd round NMSen computes the 3rd round of Π_{wsyn} and completes the transcript for LS by sending $\mathsf{z_{wsyn}}$ and $\mathsf{z_{LS}}$. Let $\tau = (\mathsf{a_{wsyn}}, \mathsf{c_{wsyn}}, \mathsf{z_{wsyn}})$ be the transcript of the execution of Π_{wsyn}. LS is used by NMSen to prove knowledge of either a decommitment of τ to a message $\neq\perp$ or of a preimage of Y.

$$
\begin{array}{ccc}
\mathsf{NMSen}(m, \mathtt{id}) & & \mathsf{NMRec}(\mathtt{id}) \\
 & \xrightarrow{\quad \mathsf{a_{wsyn}}, \mathsf{a_{LS}} \quad} & \\
 & \xleftarrow{\quad \mathsf{c_{wsyn}}, \mathsf{c_{LS}}, Y \quad} & \\
 & \xrightarrow{\quad \mathsf{z_{wsyn}}, \mathsf{z_{LS}} \quad} &
\end{array}
$$

- Y is an element taken from the range of the OWP f.
- $\tau = (\mathsf{a_{wsyn}}, \mathsf{c_{wsyn}}, \mathsf{z_{wsyn}})$ is the transcript of $\langle \mathsf{Sen_{wsyn}}(m), \mathsf{Rec_{wsyn}} \rangle(\mathtt{id})$.
- $(\mathsf{a_{LS}}, \mathsf{c_{LS}}, \mathsf{z_{LS}})$ is the transcript of LS for proving knowledge of either the decommitment of τ to a message $\neq\perp$ or of the preimage of Y.

Figure 5: Informal description of our 3-round NM commitment scheme Π_{NMCom}.

Our compiler needs the following tools:

1. a OWP f that is secure against PPT adversaries and that is $\tilde{T}_f$-breakable;
2. a 3-round one-one synchronous weak NM commitment scheme $\Pi_{\mathsf{wsyn}} = (\mathsf{Sen_{wsyn}}, \mathsf{Rec_{wsyn}})$ that is T_{wsyn}-hiding/NM, and $\tilde{T}_{\mathsf{wsyn}}$-breakable;
3. the LS PoK $\mathsf{LS} = (\mathcal{P}, \mathcal{V})$ for the language

$$
\begin{aligned}
L = \big\{(a, c, z, Y, \mathtt{id}) : \exists\ (m, \mathtt{dec}, y) \text{ s.t. } \big(&\mathsf{Rec_{wsyn}} \text{ on input } (a, c, z, m, \mathtt{dec}, \mathtt{id}) \\
&\text{accepts } m \neq\perp \text{ as a decommitment of } (a, c, z, \mathtt{id}) \ \mathtt{OR}\ Y = f(y)\big)\big\}
\end{aligned}
$$

that is T_{LS}-WI for the corresponding relation $\mathsf{Rel_L}$.

Let λ be the security parameter of our scheme. We use w.l.o.g. λ also as security parameter for the one-wayness of f with respect to polynomial-time adversaries. We consider the following hierarchy of security levels: $\tilde{T}_f << T_{\mathsf{wsyn}} << \tilde{T}_{\mathsf{wsyn}} = \sqrt{T_{\mathsf{LS}}} << T_{\mathsf{LS}}$ where by "$T << T'$" we mean that "$T \cdot \mathsf{poly}(\lambda) < T'$".

Now, similarly to [PW10, COSV16], we define different security parameters, one for each tool involved in the security proof to be consistent with the hierarchy of security levels defined above. Given the security parameter λ of our scheme, we will make use of the following security parameters: 1) λ for the OWP f; 2) λ_{wsyn} for the synchronous weak one-one NM commitment scheme; 3) λ_{LS} for LS.

All of them are polynomially related to λ and they are such that the above hierarchy of security levels holds. In the construction we assume for simplicity to have a function Params that on input λ outputs $(\lambda_{\mathsf{wsyn}}, \lambda_{\mathsf{LS}}, \ell)$ where ℓ is the length of the theorem to be proved using LS.[15] The detailed scheme is described in Fig. 6 and a compact version is depicted in Fig. 5.

Common input: security parameters: λ, $(\lambda_{\mathsf{wsyn}}, \lambda_{\mathsf{LS}}, \ell) = \mathsf{Params}(\lambda)$, $\mathtt{id} \in \{0,1\}^\lambda$.
Input to NMSen: $m \in \{0,1\}^{\mathsf{poly}\{\lambda\}}$.
Commitment phase:

1. $\mathsf{NMSen} \rightarrow \mathsf{NMRec}$
   1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^{\lambda_{\mathsf{wsyn}}}$, $\mathtt{id}$ and m thus obtaining the 1st round $\mathsf{a}_{\mathsf{wsyn}}$ of Π_{wsyn}.
   2. Run $\mathcal{P}$ on input $1^{\lambda_{\mathsf{LS}}}$ and ℓ thus obtaining the 1st round a_{LS} of LS.
   3. Send $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{a}_{\mathsf{LS}})$ to NMRec.
2. $\mathsf{NMRec} \rightarrow \mathsf{NMSen}$
   1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $\mathtt{id}$ and $\mathsf{a}_{\mathsf{wsyn}}$ thus obtaining the 2nd round $\mathsf{c}_{\mathsf{wsyn}}$ of Π_{wsyn}.
   2. Run $\mathcal{V}$ on input a_{LS} thus obtaining the 2nd round c_{LS} of LS.
   3. Pick a random $Y \in \{0,1\}^\lambda$.
   4. Send $(\mathsf{c}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{LS}}, Y)$ to NMSen.
3. $\mathsf{NMSen} \rightarrow \mathsf{NMRec}$
   1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $\mathsf{c}_{\mathsf{wsyn}}$ thus obtaining the 3rd round $\mathsf{z}_{\mathsf{wsyn}}$ of Π_{wsyn} and the decommitment information $\mathtt{dec}_{\mathsf{wsyn}}$.
   2. Set $x = (\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id})$ and $w = (m, \mathtt{dec}_{\mathsf{wsyn}}, \perp)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w, and c_{LS} thus obtaining the 3rd round z_{LS} of LS.
   3. Send $(\mathsf{z}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{LS}})$ to NMRec.
4. NMRec: Set $x = (\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id})$ and abort iff $(\mathsf{a}_{\mathsf{LS}}, \mathsf{c}_{\mathsf{LS}}, \mathsf{z}_{\mathsf{LS}})$ is not accepted by $\mathcal{V}$ for $x \in L$.

Decommitment phase:

1. $\mathsf{NMSen} \rightarrow \mathsf{NMRec}$: Send $(\mathtt{dec}_{\mathsf{wsyn}}, m)$ to NMRec.
2. NMRec: accept m as the committed message if and only if $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $(m, \mathtt{dec}_{\mathsf{wsyn}})$ accepts m as a committed message of $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, \mathtt{id})$.

Figure 6: Our 3-round NM commitment scheme Π_{NMCom}.

Theorem 4. *Suppose there exist a synchronous weak one-one NM commitment scheme and OWPs, both secure against subexponential-time adversaries, then Π_{NMCom} is a NM commitment scheme.*

The proof is divided in two parts. First we prove that Π_{NMCom} is a commitment scheme. Then we prove that Π_{NMCom} is a NM commitment scheme.

Lemma 5. *Π_{NMCom} is a statistically-binding computationally-hiding commitment scheme.*

[15] To compute 1st and 2nd round of LS only the length ℓ of the instance is required.

Proof. **Correctness.** The correctness of Π_{NMCom} follows immediately from the completeness of LS, and the correctness of Π_{wsyn}.

Statistical Binding. Observe that the message given in output in the decommitment phase of Π_{NMCom} is the message committed using Π_{wsyn}. Moreover the decommitment phase of Π_{NMCom} coincides with the decommitment phase of Π_{wsyn}. Since Π_{wsyn} is binding we have that the same holds for Π_{NMCom}.

Hiding. Following Def. 11 to prove the hiding of Π_{NMCom} we have to show that the experiment $\mathsf{ExpHiding}^0_{\mathcal{A},\Pi_{\mathsf{NMCom}}}(\lambda)$ in which NMSen commits to a message m_0 is computationally indistinguishable from the experiment $\mathsf{ExpHiding}^1_{\mathcal{A},\Pi_{\mathsf{NMCom}}}(\lambda)$ in which NMSen commits to a message m_1. In order to prove this indistinguishability we consider the following hybrid experiments.

- The 1st hybrid experiment $\mathcal{H}^0(\lambda)$ is equal to the real game experiment $\mathsf{ExpHiding}^0_{\mathcal{A},\Pi_{\mathsf{NMCom}}}(\lambda)$, with the difference that a value y s.t. $Y = f(y)$ is computed and used as a witness for LS. Observe that in order to compute y the commitment phase takes time $\tilde{T}_f$. The indistinguishability between $\mathcal{H}^0(\lambda)$ and $\mathsf{ExpHiding}^0_{\mathcal{A},\Pi_{\mathsf{NMCom}}}(\lambda)$ comes from the adaptive-input WI of LS, that holds against adversaries with running time bounded by $T_{\mathsf{LS}} >> \tilde{T}_f$.

- The 2nd hybrid $\mathcal{H}^1(\lambda)$ differs from $\mathcal{H}^0(\lambda)$ in the message committed by the adversary using Π_{wsyn}. More precisely, Π_{wsyn} is used by NMSen to commit to the message m_1 instead of m_0. The indistinguishability between $\mathcal{H}^0(\lambda)$ and $\mathcal{H}^1(\lambda)$ comes from the hiding of Π_{wsyn} and noticing that the hiding of Π_{wsyn} still holds against adversaries with running time bounded by $T_{\mathsf{wsyn}} >> \tilde{T}_f$.

The proof ends with the observation that $\mathcal{H}^1(\lambda) \approx \mathsf{ExpHiding}^1_{\mathcal{A},\Pi_{\mathsf{NMCom}}}(\lambda)$. The indistinguishability between $\mathcal{H}^1(\lambda)$ and $\mathsf{ExpHiding}^1_{\mathcal{A},\Pi_{\mathsf{NMCom}}}(\lambda)$ comes from the adaptive-WI property of LS and from the observation that, as before, the adaptive-input WI of LS still holds against adversaries with running time bounded by $T_{\mathsf{LS}} >> \tilde{T}_f$.

□

The full proof of non-malleability can be found in App. B.3. Here we give an overview of the proof. The proof is divided in two cases, in the first case we consider an adversarial MiM $\mathcal{A}_{\mathsf{NMCom}}$ that acts in a synchronized way, while in the second case $\mathcal{A}_{\mathsf{NMCom}}$ is non-synchronized. In both cases we want to show that the committed value (and the view) of $\mathcal{A}_{\mathsf{NMCom}}$ when interacting with a prover NMSen that commits to a message m is indistinguishable from the committed value (and the view) of a simulator. The proof for the synchronous case goes through a series of hybrid experiments listed below.

- We consider the real game experiment $\mathcal{H}^m_1(z)$ in which in the left session NMSen commits to m, while in the right session NMRec interacts with $\mathcal{A}_{\mathsf{NMCom}}$. Now we prove that in the right session the MiM adversary $\mathcal{A}_{\mathsf{NMCom}}$ does not commit to a message $\tilde{m} = \perp$. By contradiction if $\mathcal{A}_{\mathsf{NMCom}}$ commits to $\tilde{m} = \perp$ then the witness used to complete an accepting transcript for LS is a value $\tilde{y}$ s.t. $f(\tilde{Y}) = \tilde{y}$. Then, by using the adaptive PoK property of LS we can reach a contradiction by inverting f in polynomial time.

- The 2nd hybrid is $\mathcal{H}^m_2(z)$ and it differs from $\mathcal{H}^m_1(z)$ only in the witness used to compute the sLS transcript. The adversary $\mathcal{A}_{\mathsf{NMCom}}$, running in sub-exponential time, computes a value y s.t. $f(y) = Y$, and uses it as witness for the execution of LS. From the adaptive-input WI (that is stronger than inverting the OWP and of breaking Π_{wsyn}) of sLS, the view and the committed message of $\mathcal{A}_{\mathsf{NMCom}}$ do not change between $\mathcal{H}_2^m(z)$ and $\mathcal{H}_1^m(z)$.

- We now consider the hybrid experiment is $\mathcal{H}_1^0(z)$ that differs from the first hybrid experiment that we have considered $\mathcal{H}_1^m(z)$ in the committed message. Indeed in this case, the message committed in the left session is 0^λ. We observe that $\mathcal{H}_1^0(z)$ actually is the simulated game. As for the hybrid experiment $\mathcal{H}_1^m(z)$ we need to prove that in the right session the MiM adversary $\mathcal{A}_{\mathsf{NMCom}}$ does not commit to a message $\tilde{m} = \perp$. By contradiction if $\mathcal{A}_{\mathsf{NMCom}}$ commits to $\tilde{m} = \perp$ then the witness used to complete an accepting transcript for LS is a value $\tilde{y}$ s.t. $f(\tilde{Y}) = \tilde{y}$. Then, by using the PoK property of LS we can reach a contradiction by inverting f in polynomial time.

- The last hybrid experiment that we consider is $\mathcal{H}_2^0(z)$ and it differs from $\mathcal{H}_1^0(z)$ only in the witness used to compute the sLS transcript. In more details the adversary $\mathcal{A}_{\mathsf{NMCom}}$, running in sub-exponential time, computes a value y s.t. $f(y) = Y$, and uses it as witness for the execution of LS. From the adaptive-input WI (that is stronger than inverting the OWP) of sLS, the view and the committed message of $\mathcal{A}_{\mathsf{NMCom}}$ do not change between $\mathcal{H}_2^0(z)$ and $\mathcal{H}_1^0(z)$.

To conclude this proof we show that the view and the committed message of $\mathcal{A}_{\mathsf{NMCom}}$ acting in $\mathcal{H}_1^m(z)$ are indistinguishable from the view and the committed message of $\mathcal{A}_{\mathsf{NMCom}}$ acting in $\mathcal{H}_1^0(z)$. Given the already argued indistinguishabilities, it remains to show that the view and the committed message of $\mathcal{H}_2^m(z)$ are indistinguishable from the view and the committed message of $\mathcal{H}_2^0(z)$. This is ensured by the synchronous weak non-malleability of Π_{wsyn}. Here we need only to use a *weak* synchronous one-one NM commitment since we are guaranteed from the above arguments that whenever $\mathcal{A}_{\mathsf{NMCom}}$ completes a commitment in a right session, the underlying commitment computed through Π_{wsyn} corresponds to $\perp$ with negligible probability only both in $\mathcal{H}_2^m(z)$ and in $\mathcal{H}_2^0(z)$.

The proof for the asynchronous case is much simpler and relies on the hiding of Π_{NMCom}. More precisely we observe that in case of asynchronous scheduling it is possible to rewind the adversary $\mathcal{A}_{\mathsf{NMCom}}$ without rewinding the sender in the left session. This allows us to extract (in polynomial time) the witness used by the adversary in the execution of LS, that with overwhelming probability corresponds to the committed message. Therefore we contradict the hiding of Π_{NMCom}.

6 Acknowledgments

We thank Vipul Goyal, and Silas Richelson for remarkable discussions on [GPR16].

Research supported in part by "GNCS - INdAM", EU COST Action IC1306, NSF grants 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported in part by DARPA Safeware program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

The work of 1st, 3rd and 4th authors has been done in part while visiting UCLA.

References

[ADL14] Divesh Aggarwal, Yevgeniy Dodis, and Shachar Lovett. Non-malleable codes from additive combinatorics. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 774–783, 2014. (Cited on page 3.)

[Bar02] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 345–355, 2002. (Cited on page 3.)

[BGR+15] Hai Brenner, Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. Fast non-malleable commitments. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 1048–1057, 2015. (Cited on pages 3 and 4.)

[BJY97] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-optimal zero-knowledge arguments based on any one-way function. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 280–305. Springer, 1997. (Cited on page 4.)

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 345–354, 2006. (Cited on page 4.)

[CKPR01] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires omega˜(log n) rounds. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 570–579, 2001. (Cited on page 4.)

[COP+14] Kai-Min Chung, Rafail Ostrovsky, Rafael Pass, Muthuramakrishnan Venkitasubramaniam, and Ivan Visconti. 4-round resettably-sound zero knowledge. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 192–216. Springer, 2014. (Cited on pages 4 and 7.)

[COSV16] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Concurrent non-malleable commitments (and more) in 3 rounds. Advances in Cryptology - CRYPTO 2016 - 36nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016. Proceedings, 2016. (Cited on pages 3, 4, 5, 7, and 19.)

[CPS13] Kai-Min Chung, Rafael Pass, and Karn Seth. Non-black-box simulation from one-way functions and applications to resettable security. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 231–240. ACM, 2013. (Cited on pages 7 and 27.)

[CPS+16a] Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Improved or-composition of sigma-protocols. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 112–141, 2016. (Cited on pages 4 and 7.)

[CPS+16b] Michele Ciampi, Giuseppe Persiano, Alessandra Scafuro, Luisa Siniscalchi, and Ivan Visconti. Online/offline OR composition of sigma protocols. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 63–92, 2016. (Cited on pages 4 and 28.)

[CVZ10] Zhenfu Cao, Ivan Visconti, and Zongyang Zhang. Constant-round concurrent non-malleable statistically binding commitments and decommitments. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 193–208. Springer, 2010. (Cited on page 3.)

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 542–552, 1991. (Cited on page 3.)

[DG03] Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 426–437, 2003. (Cited on page 3.)

[DPV04a] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 237–253. Springer, 2004. (Cited on page 4.)

[DPV04b] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Improved setup assumptions for 3-round resettable zero knowledge. In *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 530–544. Springer, 2004. (Cited on page 7.)

[Fei90] Uriel Feige. Alternative models for zero knowledge interactive proofs. Master's thesis, Weizmann Institute of Science, Rehovot, Israel, 1990. Ph.D. thesis. (Cited on page 29.)

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 416–426, New York, NY, USA, 1990. ACM. (Cited on page 6.)

[GJO+13] Vipul Goyal, Abhishek Jain, Rafail Ostrovsky, Silas Richelson, and Ivan Visconti. Constant-round concurrent zero knowledge in the bounded player model. In *Advances in Cryptology - ASIACRYPT 2013, 19th International Conference on the Theory and*

Application of Cryptology and Information Security, 2004, Proceedings, volume 8279 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2013. (Cited on page 7.)

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996. (Cited on page 43.)

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32, 1989. (Cited on page 3.)

[GLOV12] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 51–60, 2012. (Cited on page 3.)

[GMPP16] Sanjam Garg, Pratyay Mukherjee, Omkant Pandey, and Antigoni Polychroniadou. The exact round complexity of secure computation. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 448–476, 2016. (Cited on page 4.)

[Goy11] Vipul Goyal. Constant round non-malleable protocols using one way functions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 695–704, 2011. (Cited on page 3.)

[GPR16] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1128–1141, 2016. (Cited on pages 3, 5, 6, and 21.)

[GRRV14] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 41–50, 2014. (Cited on pages 3, 4, 5, 6, and 7.)

[GRRV16a] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. Personal communication, May 2016. (Cited on page 4.)

[GRRV16b] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. Personal communication, June 2016. (Cited on page 4.)

[GRRV16c] Vipul Goyal, Silas Richelson, Alon Rosen, and Margarita Vald. An algebraic approach to non-malleability. Cryptology ePrint Archive, Report 2014/586, June 2016. http://eprint.iacr.org/. (Cited on page 4.)

[HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. On the power of secure two-party computation. Cryptology ePrint Archive, Report 2016/074, 2016. http://eprint.iacr.org/. (Cited on page 4.)

[KO04] Jonathan Katz and Rafail Ostrovsky. Round-optimal secure two-party computation. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology-Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 335–354, 2004. (Cited on page 4.)

[Lin10] Yehuda Lindell. Foundations of cryptography 89-856. http://u.cs.biu.ac.il/~lindell/89-856/complete-89-856.pdf, 2010. (Cited on page 29.)

[LP11a] Huijia Lin and Rafael Pass. Concurrent non-malleable zero knowledge with adaptive inputs. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, pages 274–292, 2011. (Cited on page 9.)

[LP11b] Huijia Lin and Rafael Pass. Constant-round non-malleable commitments from any one-way function. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 705–714, 2011. (Cited on pages 3, 9, 28, and 30.)

[LP15] Huijia Lin and Rafael Pass. Constant-round nonmalleable commitments from any one-way function. *J. ACM*, 62(1):5:1–5:30, 2015. (Cited on page 3.)

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In *Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC 2008, New York, USA, March 19-21, 2008.*, pages 571–588, 2008. (Cited on pages 3, 8, and 9.)

[LS90] Dror Lapidot and Adi Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In *Advances in Cryptology - CRYPTO*, 1990. (Cited on pages 4, 6, 7, and 29.)

[MP12] Mohammad Mahmoody and Rafael Pass. The curious case of non-interactive commitments - on the power of black-box vs. non-black-box use of primitives. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 701–718, 2012. (Cited on page 3.)

[MV16] Arno Mittelbach and Daniele Venturi. Fiat-shamir for highly sound protocols is instantiable. Cryptology ePrint Archive, Report 2016/313, 2016. http://eprint.iacr.org/. (Cited on page 4.)

[OPV09] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Simulation-based concurrent non-malleable commitments and decommitments. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 91–108. Springer, 2009. (Cited on page 3.)

[Pas13] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In *TCC*, pages 334–354, 2013. (Cited on page 3.)

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In *Advances in Cryptology - CRYPTO 2008, 28th Annual*

International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings, pages 57–74, 2008. (Cited on page 3.)

[PR03] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 404–413, 2003. (Cited on page 3.)

[PR05a] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 563–572, 2005. (Cited on page 3.)

[PR05b] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 533–542, 2005. (Cited on page 3.)

[PR08a] Rafael Pass and Alon Rosen. Concurrent nonmalleable commitments. *SIAM J. Comput.*, 37(6):1891–1925, 2008. (Cited on page 3.)

[PR08b] Rafael Pass and Alon Rosen. New and improved constructions of nonmalleable cryptographic protocols. *SIAM J. Comput.*, 38(2):702–752, 2008. (Cited on page 3.)

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, pages 403–418, 2009. (Cited on page 30.)

[PW10] Rafael Pass and Hoeteck Wee. Constant-round non-malleable commitments from subexponential one-way functions. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 638–655, 2010. (Cited on pages 3 and 19.)

[SCO+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 566–598. Springer, 2001. (Cited on page 9.)

[Wee10] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 531–540. IEEE Computer Society, 2010. (Cited on pages 3 and 4.)

[YZ07] Moti Yung and Yunlei Zhao. Generic and practical resettable zero-knowledge in the bare public-key model. In *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*, pages 129–147, 2007. (Cited on page 4.)

A Standard Definitions and Tools

Definition 5 (One-way function (OWF))**.** *A function* $f : \{0,1\}^\star \rightarrow \{0,1\}^\star$ *is called one way if the following two conditions hold:*

- *there exists a deterministic polynomial-time algorithm that on input* y *in the domain of* f *outputs* $f(y)$*;*
- *for every* PPT *algorithm* $\mathcal{A}$ *there exists a negligible function* ν*, such that for every auxiliary input* $z \in \{0,1\}^{\mathsf{poly}(\lambda)}$*:*

$$\text{Prob}\left[\, y \leftarrow \{0,1\}^\star : \mathcal{A}(f(y), z) \in f^{-1}(f(y)) \,\right] < \nu(\lambda).$$

We say that a OWF f *is a* one-way permutation (OWP) *if* f *is a permutation.*

We will require that an algorithm that runs in time $\tilde{T} = 2^{\lambda^\alpha}$ *for some positive constant* $\alpha < 1$*, can invert a OWP* f*. In this case we say that* f *is* $\tilde{T}$*-breakable.*

Definition 6 (Strong Signatures [CPS13])**.** *A triple of* PPT *algorithms* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is called a* signature scheme *if it satisfies the following properties.*

Validity: *For every pair* $(s, v) \leftarrow \mathsf{Gen}(1^\lambda)$*, and every* $m \in \{0,1\}^\lambda$*, we have that*

$$\mathsf{Ver}(v, m, \mathsf{Sign}(s, m)) = 1.$$

Security: *For every* PPT $\mathcal{A}$*, there exists a negligible function* ν*, such that for all auxiliary input* $z \in \{0,1\}^\star$ *it holds that:*

$$\Pr[(s,v) \leftarrow \mathsf{Gen}(1^\lambda); (m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(s,\cdot)}(z, v) \wedge \mathsf{Ver}(v, m, \sigma) = 1 \wedge (m, \sigma) \notin Q] < \nu(\lambda)$$

where Q *denotes the list of query-answer pairs for all queries asked by* $\mathcal{A}$ *to the oracle* $\mathsf{Sign}(s, \cdot)$*.*

Definition 7 (Computational indistinguishability)**.** *Let* $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ *and* $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ *be ensembles, where* X_λ*'s and* Y_λ*'s are probability distribution over* $\{0,1\}^l$*, for same* $l = \mathsf{poly}(\lambda)$*. We say that* $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ *and* $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ *are* computationally indistinguishable*, denoted* $X \approx Y$*, if for every* PPT *distinguisher* $\mathcal{D}$ *there exists a negligible function* ν *such that for sufficiently large* $\lambda \in \mathbb{N}$*,*

$$\left| \text{Prob}\left[\, t \leftarrow X_\lambda : \mathcal{D}(1^\lambda, t) = 1 \,\right] - \text{Prob}\left[\, t \leftarrow Y_\lambda : \mathcal{D}(1^\lambda, t) = 1 \,\right] \right| < \nu(\lambda).$$

We note that in the usual case where $|X_\lambda| = \Omega(\lambda)$ and λ can be derived from a sample of X_λ, it is possible to omit the auxiliary input 1^λ. In this paper we also use the definition of *Statistical Indistinguishability*. This definition is the same as Definition 7 with the only difference that the distinguisher $\mathcal{D}$ is unbounded. In this case use $X \equiv_s Y$ to denote that two ensembles are statistically indistinguishable.

Definition 8 (Delayed-input proof/argument system)**.** *A pair of* PPT *interactive algorithms* $\Pi = (\mathcal{P}, \mathcal{V})$ *constitutes a* proof system *(resp., an* argument system*) for an* $\mathcal{NP}$*-language* L*, if the following conditions hold:*

Completeness: *For every* $x \in L$ *and* w *such that* $(x, w) \in \mathsf{Rel_L}$*, it holds that:*

$$\text{Prob}\left[\, \langle \mathcal{P}(w), \mathcal{V} \rangle(x) = 1 \,\right] = 1.$$

Soundness: *For every interactive (resp.,* PPT *interactive) algorithm $\mathcal{P}^\star$, there exists a negligible function ν such that for every $x \notin L$ and every z:*

$$\text{Prob}\left[\ \langle \mathcal{P}^\star(z), \mathcal{V}\rangle(x) = 1\ \right] < \nu(|x|).$$

A proof/argument system $\Pi = (\mathcal{P}, \mathcal{V})$ for an $\mathcal{NP}$-language L, enjoys *delayed-input* completeness if $\mathcal{P}$ needs x and w only to compute the last round and $\mathcal{V}$ needs x only to compute the output. Before that, $\mathcal{P}$ and $\mathcal{V}$ run having as input only the size of x. The notion of delayed-input completeness was defined in [CPS+16b].

An interactive protocol $\Pi = (\mathcal{P}, \mathcal{V})$ is *public coin* if, at every round, $\mathcal{V}$ simply tosses a predetermined number of coins (random challenge) and sends the outcome to the prover.

We say that the transcript τ of an execution $b = \langle \mathcal{P}(z), \mathcal{V}\rangle(x)$ is *accepting* if $b = 1$.

Witness indistinguishability. Let $\mathsf{View}^{\mathcal{P}(w)}_{\mathcal{V}^\star(z)}(x)$ be the random variable that denotes $\mathcal{V}^\star$'s view in an interaction with $\mathcal{P}$ when $\mathcal{V}^\star$ is given auxiliary input z, $\mathcal{P}$ is given witness w, and both parties are given common input x.

Definition 9 (Witness Indistinguishability (WI))**.** *An argument/proof system $\Pi = (\mathcal{P}, \mathcal{V})$ for $\mathcal{NP}$-language L, is* Witness Indistinguishable (WI) *for the corresponding relation $\mathsf{Rel_L}$ if, for every malicious* PPT *verifier $\mathcal{V}^\star$, for all auxiliary input $z \in \{0,1\}^\star$ and for all x, w, w' such that $(x, w) \in \mathsf{Rel_L}$ and $(x, w') \in \mathsf{Rel_L}$, the following ensembles are computationally indistinguishable:*

$$\{\mathsf{View}^{\mathcal{P}(w)}_{\mathcal{V}^\star(z)}(x)\} \approx \{\mathsf{View}^{\mathcal{P}(w')}_{\mathcal{V}^\star(z)}(x)\}.$$

The notion of a *statistically* WI proof/argument system is obtained by requiring that the two ensembles $\{\mathsf{View}^{\mathcal{P}(w)}_{\mathcal{V}^\star(z)}(x)\}$ and $\{\mathsf{View}^{\mathcal{P}(w')}_{\mathcal{V}^\star(z)}(x)\}$ are statistically indistinguishable.

Obviously one can generalize the above definitions of WI to their natural adaptive-input variant, where the adversarial verifier can select the statement and the witnesses adaptively, before the prover plays the last round.

In this paper we also consider a definition where the adaptive-WI property of the argument/proof system still holds against a distinguisher with running time bounded by $T = 2^{\lambda^\alpha}$ for some constant positive constant $\alpha < 1$. In this case we say that the instantiation of WI proof system is T-Witness Indistinguishable (T-WI).

Definition 10 (Proof of Knowledge [LP11b])**.** *A proof system $\Pi = (\mathcal{P}, \mathcal{V})$ is a* proof of knowledge *(PoK) for the relation $\mathsf{Rel_L}$ if there exist a probabilistic expected polynomial-time machine E, called the extractor, such that for every algorithm $\mathcal{P}^\star$, there exists a negligible function ν, every statement $x \in \{0,1\}^\lambda$, every randomness $r \in \{0,1\}^\star$ and every auxiliary input $z \in \{0,1\}^\star$,*

$$\text{Prob}\left[\ \langle \mathcal{P}^\star_r(z), \mathcal{V}\rangle(x) = 1\ \right] \leq \text{Prob}\left[\ w \leftarrow \mathsf{E}^{\mathcal{P}^\star_r(z)}(x) : (x, w) \in \mathsf{Rel_L}\ \right] + \nu(\lambda).$$

We also say that an argument system Π is a argument of knowledge *(AoK) if the above condition holds w.r.t. any* PPT *$\mathcal{P}^\star$.*

In this paper we also consider the *adaptive-input* PoK/AoK property. Adaptive-input PoK/AoK ensures that the PoK/AoK property still holds when a malicious prover can choose the statement adaptively at the last round. In this case, to be consistent with Definition 10 of PoK/AoK where the extractor algorithm E takes as input the statement proved by $\mathcal{P}^\star$, we have to consider a different extractor algorithm. This extractor algorithm takes as input the randomness r' of $\mathcal{V}$, the randomness r of $\mathcal{P}^\star$ and outputs the witness for $x \in L$, where x is selected by $\mathcal{P}^\star_r$ when interacting with $\mathcal{V}_{r'}$.

In this paper we use the 3-round public-coin WI PoK (WIPoK) proposed by Lapidot and Shamir [LS90], that we denote by LS. LS enjoys delayed-input completeness since the inputs for both $\mathcal{P}$ and $\mathcal{V}$ are needed only to play the last round, and only the length of the instance is needed earlier. LS also enjoys adaptive-input PoK and adaptive-input WI. We also use a 4-round delayed-input, adaptive-input AoK, and adaptive-input statistical WI argument of knowledge (WIAoK), that is a variant of LS [Fei90]. More in details, the WI of LS relies on the hiding property of the underlying commitment scheme, therefore if the prover of LS uses a 2-round statistically hiding commitment scheme, then we obtain adaptive-input statistical WIAoK. Note that compared to LS this variation of LS requires an additional round from verifier to prover in order to send the first round of the statistically hiding commitment scheme.

A.1 Commitment Schemes

Definition 11 (Commitment Scheme)**.** *Given a security parameter 1^λ, a commitment scheme* $\mathtt{CS} = (\mathsf{Sen}, \mathsf{Rec})$ *is a two-phase protocol between two* PPT *interactive algorithms, a sender* Sen *and a receiver* Rec*. In the commitment phase* Sen *on input a message m interacts with* Rec *to produce a commitment* $\mathtt{com}$*. In the decommitment phase,* Sen *sends to* Rec *a decommitment information* $\mathtt{d}$ *such that* Rec *accepts m as the decommitment of* $\mathtt{com}$*.*

Formally, we say that $\mathtt{CS} = (\mathsf{Sen}, \mathsf{Rec})$ *is a perfectly binding commitment scheme if the following properties hold:*

Correctness:

- *Commitment phase. Let* $\mathtt{com}$ *be the commitment of the message m given as output of an execution of* $\mathtt{CS} = (\mathsf{Sen}, \mathsf{Rec})$ *where* Sen *runs on input a message m. Let* $\mathtt{d}$ *be the private output of* Sen *in this phase.*
- *Decommitment phase*[16]*.* Rec *on input m and* $\mathtt{d}$ *accepts m as decommitment of* $\mathtt{com}$*.*

Statistical (resp. Computational) Hiding([Lin10]): *for any adversary (resp.* PPT *adversary) $\mathcal{A}$ and a randomly chosen bit $b \in \{0,1\}$, consider the following hiding experiment* $\mathsf{ExpHiding}^b_{\mathcal{A},\mathtt{CS}}(\lambda)$*:*

- *Upon input 1^λ, the adversary $\mathcal{A}$ outputs a pair of messages m_0, m_1 that are of the same length.*
- Sen *on input the message m_b interacts with $\mathcal{A}$ to produce a commitment of m_b.*
- *$\mathcal{A}$ outputs a bit b' and this is the output of the experiment.*

For any adversary (resp. PPT *adversary) $\mathcal{A}$, there exist a negligible function ν, s.t.:*

$$\left| \mathrm{Prob}\left[\ \mathsf{ExpHiding}^0_{\mathcal{A},\mathtt{CS}}(\lambda) = 1\ \right] - \mathrm{Prob}\left[\ \mathsf{ExpHiding}^1_{\mathcal{A},\mathtt{CS}}(\lambda) = 1\ \right] \right| < \nu(\lambda).$$

Statistical (resp. Computational) Binding: *for every commitment* $\mathtt{com}$ *generated during the commitment phase by a possibly malicious unbounded (resp. malicious* PPT*) sender* $\mathsf{Sen}^\star$ *there exists a negligible function ν such that* $\mathsf{Sen}^\star$*, with probability at most $\nu(\lambda)$, outputs two*

[16] In this paper we consider a non-interactive decommitment phase only.

decommitments (m_0, d_0) *and* (m_1, d_1)*, with* $m_0 \neq m_1$*, such that* Rec *accepts both decommitments.*

We also say that a commitment scheme is perfectly binding *iff* $\nu(\lambda) = 0$.

We also consider the definition of a commitment scheme where computational hiding still holds against an adversary $\mathcal{A}$ running in time bounded by $T = 2^{\lambda^\alpha}$ for some positive constant $\alpha < 1$. In this case we will say that a commitment scheme is T-hiding. We will also say that a commitment scheme is $\tilde{T}$-breakable to specify that an algorithm running in time $\tilde{T} = 2^{\lambda^\beta}$, for some positive constant $\beta < 1$, recovers the (if any) only message that can be successfully decommitment.

Extractable commitment schemes. Informally, a commitment scheme is extractable if there exists an efficient extractor that having black-box access to any efficient malicious PPT sender ExSen* that successfully performs the commitment phase, outputs the only committed string that can be successfully decommitted.

Definition 12 (Extractable Commitment Scheme [PW09]). *A perfectly (resp. statistically) binding commitment scheme* $\mathtt{ExCS} = (\mathrm{ExSen}, \mathrm{ExRec})$ *is an* extractable commitment scheme *if there exists an expected* PPT *extractor* $\mathtt{ExtCom}$ *that given oracle access to any malicious* PPT *sender* ExSen**, outputs a pair* $(\tau, \sigma^\star)$ *such that the following two properties hold:*

- **Simulatability:** τ *is identically distributed to the view of* ExSen* *(when interacting with an honest* ExRec*) in the commitment phase.*
- **Extractability:** *the probability that there exists a decommitment of* τ *to* σ*, where* $\sigma \neq \sigma^\star$ *is 0 (resp. negligible).*

B Formal Proofs

B.1 Proof of Non-Malleability of the 4-Round NM Commitment Scheme

Lemma 6. Π_{NM4Com} *is a one-one synchronous NM commitment scheme.*

Proof. We need to prove that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$

$$\{\mathsf{mim}_{\Pi_{\mathsf{NM4Com}}}^{\mathcal{A}_{\mathsf{NM4Com}}, m}(z)\}_{z\in\{0,1\}^\star} \approx \{\mathsf{sim}_{\Pi_{\mathsf{NM4Com}}}^{\mathsf{Sim}_{\mathsf{NM4Com}}}(1^\lambda, z)\}_{z\in\{0,1\}^\star}$$

where $\mathsf{Sim}_{\mathsf{NM4Com}}$ is the simulator depicted in Fig. 7.

We remark that in the security proofs we denote by $\tilde{\delta}$ a value associated with the right session (where the adversary plays with a receiver) where δ is the corresponding value in the left session. For example, the sender commits to v in the left session while the adversary commits to $\tilde{v}$ in the right session.

In order to prove the indistinguishability of the above two distributions we proceed by showing two experiments $\mathcal{H}_1^m(z), \mathcal{H}_2^m(z)$ where m is the message committed in the left session. Following [LP11b] we denote by $\{\mathsf{mim}_{\mathcal{H}_i^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z)\}_{z\in\{0,1\}^\star}$ the random variable describing the view of the MiM $\mathcal{A}_{\mathsf{NMCom}}$ combined with the value it commits to in the right interaction in hybrid $\mathcal{H}_i^m(z)$ (as usual, the committed value is replaced by $\perp$ if the right interaction does not correspond to a commitment that can be decommitted successfully or if $\mathcal{A}_{\mathsf{NMCom}}$ has copied the identity of the left interaction). The same notation is used also in the other proofs of non-malleability.

Let be p the probability that in the real experiment $\mathcal{A}_{\mathsf{NM4Com}}$ concludes the left session.

In the first experiment in the left session $\mathsf{NM4Sen}$ commits to m, while in the right session we let $\mathsf{NM4Rec}$ interacts with $\mathcal{A}_{\mathsf{NM4Com}}$. We refer to this hybrid experiment as $\mathcal{H}_1^m(z)$. Details follow below.

$\boldsymbol{\mathcal{H}_1^m(z)}$.

Left session:

1. Second round, upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{wsyn}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows:
   1.1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input 1^λ, $\mathtt{id}$, π^1_{wsyn} and m thus obtaining the second round π^2_{wsyn} of Π_{wsyn}.
   1.2. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the second round π^2_{sLS} of sLS.
   1.3. Pick a message $\mathtt{msg} \leftarrow \{0,1\}^\lambda$.
   1.4. Send $(\pi^2_{\mathsf{wsyn}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
2. Fourth round, upon receiving $(\pi^3_{\mathsf{wsyn}}, \pi^3_{\mathsf{sLS}}, \sigma)$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows:
   2.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}, \sigma) \neq 1$ then abort, continue as follows otherwise.
   2.2. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input π^3_{wsyn} thus obtaining the fourth round π^4_{wsyn} of Π_{wsyn} and the decommitment information $\mathtt{dec}_{\mathsf{wsyn}}$.
   2.3. Set $x = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id}, \mathsf{vk})$ and $w = (m, \mathtt{dec}_{\mathsf{wsyn}}, \perp, \perp, \perp, \perp)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w, and π^3_{sLS} thus obtaining the fourth round π^4_{sLS} of sLS.
   2.4. Send $(\pi^4_{\mathsf{wsyn}}, \pi^4_{\mathsf{sLS}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{NM4Com}}$ and $\mathsf{NM4Rec}$.

The distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_1^m}(z)$[17] clearly corresponds to the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\Pi_{\mathsf{NM4Com}}}(z)$.

We now prove that in the right session the MiM adversary $\mathcal{A}_{\mathsf{NM4Com}}$ does not commit to a message $\tilde{m} = \perp$. More formally prove the following claim.

Claim 1. *Let $\bar{p}$ be the probability that in the right session of $\mathcal{H}_1^m(z)$ $\mathcal{A}_{\mathsf{NM4Com}}$ successfully commits to a message $\tilde{m} = \perp$, then $\bar{p} < \nu(\lambda)$ for some negligible function ν.*

Proof. Suppose by contradiction that the claim does not hold, then we can construct an adversary $\mathcal{A}_{\Sigma}$ that breaks the security of the signature scheme Σ. Let vk be the challenge verification key. The intuition of the security proof is to create an adversary $\mathcal{A}_{\Sigma}$ that interacts against the MiM adversary $\mathcal{A}_{\mathsf{NM4Com}}$. $\mathcal{A}_{\Sigma}$ sends vk to $\mathcal{A}_{\mathsf{NM4Com}}$ in the first round of the right session and extracts the witness used by $\mathcal{A}_{\mathsf{NM4Com}}$ to execute sLS in the right session. Since by contradiction we are assuming that $\mathcal{A}_{\mathsf{NM4Com}}$ commits to a message $\tilde{m} = \perp$ then, with non-negligible probability, the witness extracted by sLS will be a pair of signatures (σ_1, σ_2) for a pair of distinct messages $(\mathtt{msg}_1, \mathtt{msg}_2)$ s.t. $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma_1) = 1$ and $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_2, \sigma_2) = 1$. In order to extract the witness used in sLS we use an extractor E (that exists from the property of adaptive-input AoK enjoyed by sLS). As discussed in App. A, in order to run E, the theorem x proved by a malicious prover $\mathcal{P}^\star$ is needed. In the case of adaptive-input AoK, E takes as input the randomnesses r and r' of the interaction between $\mathcal{V}_{r'}$ and $\mathcal{P}^\star_r$ where the theorem x is proved. Consider now a successful execution of $\mathcal{H}_1^m(z)$ where the theorem proved by sLS to a verifier $\mathcal{V}_{r'}$ is $\tilde{x} = (\tilde{\pi}^1_{\mathsf{wsyn}}, \tilde{\pi}^2_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^4_{\mathsf{wsyn}}, \tilde{\mathtt{id}}, \mathsf{vk})$.

We observe that to complete this execution it is necessary to compute a valid signature $\tilde{\sigma}$ (w.r.t. vk) of a message $\tilde{\mathtt{msg}}$ sent by $\mathcal{A}_{\mathsf{NM4Com}}$. This is not a problem because $\tilde{\sigma}$ can be computed by querying the signing oracle Sign.

[17] To simplify the notation here, and in the rest of the paper, we will omit to specify that the distribution is indexed for all $z \in \{0,1\}^\star$.

Now we are ready to construct the malicious prover $\mathcal{M}_{\mathcal{P}^\star}$ that will interacts with E. More specifically $\mathcal{M}_{\mathcal{P}^\star}$ internally runs $\mathcal{A}_{\mathsf{NM4Com}}$ and interacts with him as the sender $\mathsf{NM4Sen}$ does in the left session and as the receiver $\mathsf{NM4Rec}$ does in the right session. The only difference is that all the messages of sLS of the right session are sent to a verifier $\mathcal{V}$ and vice versa (i.e., messages received from a $\mathcal{V}$ are plugged in messages of the receiver in the right session). Formally $\mathcal{M}_{\mathcal{P}^\star}$ works as follows.

$\boldsymbol{\mathcal{M}_{\mathcal{P}^\star}(m, \mathsf{vk}, \tilde{\sigma}, r, z)}$.

Use r as randomness for all next steps.

Run $\mathcal{A}_{\mathsf{NM4Com}}$ and act as follows:

Left session: run as in the left session of $\mathcal{H}_1^m(z)$.

Right session:

1. Upon receiving the 1st round $\tilde{\pi}^1_{\mathsf{sLS}}$ of sLS from the external verifier $\mathcal{V}$, run as follows:
   1.1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $1^\lambda, \tilde{\mathtt{id}}$ thus obtaining the first round $\tilde{\pi}^1_{\mathsf{wsyn}}$ of Π_{wsyn}.
   1.2. Set $\tilde{\mathsf{vk}} = \mathsf{vk}$.
   1.3. Send $(\tilde{\mathsf{vk}}, \tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^1_{\mathsf{wsyn}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
2. Upon receiving $(\tilde{\pi}^2_{\mathsf{wsyn}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\mathtt{msg}})$ send $\tilde{\pi}^2_{\mathsf{sLS}}$ to the external verifier $\mathcal{V}$.
3. Upon receiving the 3rd round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS from the external verifier $\mathcal{V}$, run as follows:
   3.1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $\tilde{\pi}^2_{\mathsf{wsyn}}$ thus obtaining the third round $\tilde{\pi}^3_{\mathsf{wsyn}}$ of Π_{wsyn}.
   3.2. Send $(\tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
4. Upon receiving $(\tilde{\pi}^4_{\mathsf{wsyn}}, \tilde{\pi}^4_{\mathsf{sLS}})$ set $\tilde{x} = (\tilde{\pi}^1_{\mathsf{wsyn}}, \tilde{\pi}^2_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^4_{\mathsf{wsyn}}, \tilde{\mathtt{id}}, \tilde{\mathsf{vk}})$ and send $(\tilde{\pi}^4_{\mathsf{sLS}}, \tilde{x})$ to the external verifier $\mathcal{V}$.

Now we can conclude the proof of this claim by describing how $\mathcal{A}_\Sigma$ works. $\mathcal{A}_\Sigma$ runs the extractor E of sLS on input r' and r and setting $\mathcal{M}_{\mathcal{P}^\star}$ as $\mathcal{P}^\star$ (recall that an extractor of sLS plays having oracle access to a prover of sLS and receiving the randomnesses of the honest verifier and of the prover). The extractor, with non-negligible probability, outputs the witness for the statement proved with sLS. Since we are assuming that with non-negligible probability the commitment computed by $\mathcal{A}_{\mathsf{NM4Com}}$ using Π_{NM4Com} is not well formed, then the output of the extractor (with non-negligible probability) will be a pair of valid signature (σ_1, σ_2) for the messages $(\mathtt{msg}_1, \mathtt{msg}_2)$, with $\mathtt{msg}_1 \neq \mathtt{msg}_2$. The proof ends with the observation that in the reduction just one oracle query has been made to the signature oracle. □

The next experiment is $\mathcal{H}_1^0(z)$ and it corresponds to $\mathcal{H}_1^m(z)$ with the only difference that the message committed, using Π_{wsyn}, is 0^λ instead of m. We prove the following claim.

Claim 2. *Let $\bar{p}$ be the probability that in the right session of $\mathcal{H}_1^0(z)$ $\mathcal{A}_{\mathsf{NM4Com}}$ successfully commits to a message $\tilde{m} = \perp$. Then $\bar{p} < \nu(\lambda)$ for some negligible function ν.*

Proof. The security proof follows strictly the one of Claim 1. □

We now consider the hybrid experiment $\mathcal{H}_2^m(z)$ where in the left session, by rewinding the adversary $\mathcal{A}_{\mathsf{NM4Com}}$ from the third to the second round, two signatures σ_1, σ_2 for two distinct messages $(\mathtt{msg}_1, \mathtt{msg}_2)$ are extracted and used as witness to execute sLS. Note that after $1/p$ rewinds the probability of not obtaining a valid new signature is less than $1/2$. Therefore the probability that $\mathcal{A}_{\mathsf{NM4Com}}$ does not give a second valid signature for a randomly chosen message after λ/p rewinds is negligible in λ. For the above reason we can claim that the probability that in $\mathcal{H}_2^m(z)$ the experiment aborts is statistically close to the probability that in $\mathcal{H}_1^m(z)$ the output of the experiment is abort.

Because of the adaptive-input statistical WI of sLS we can also claim that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ $\mathsf{mim}_{\mathcal{H}_1^m}^{\mathcal{A}_{\mathsf{NM4Com}},m}(z) \equiv_s \mathsf{mim}_{\mathcal{H}_2^m}^{\mathcal{A}_{\mathsf{NM4Com}},m}(z)$. The formal description of $\mathcal{H}_2^m(z)$ is the follows below.

$\boldsymbol{\mathcal{H}_2^m(z)}$.

Left session:

1. Second round, upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{wsyn}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows.
   1.1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input 1^λ, $\mathtt{id}$, π^1_{wsyn} and m thus obtaining the second round π^2_{wsyn} of Π_{wsyn}.
   1.2. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the second round π^2_{sLS} of sLS.
   1.3. Pick a message $\mathtt{msg}_1 \leftarrow \{0,1\}^\lambda$.
   1.4. Send $(\pi^2_{\mathsf{wsyn}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg}_1)$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
2. Fourth round, upon receiving $(\pi^3_{\mathsf{wsyn}}, \pi^3_{\mathsf{sLS}}, \sigma_1)$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows.
   2.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma) \neq 1$ then abort, otherwise continue as follows.
   2.2. <u>Repeat Step 1.3, 1.4 and follow-up right-session messages up to λ/p times in order to obtain a signature σ_2 of a random message $\mathtt{msg}_2 \neq \mathtt{msg}_1$. Abort in case of failure in obtaining σ_2.</u>
   2.3. <u>Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input π^3_{wsyn} thus obtaining the fourth round π^4_{wsyn} of Π_{wsyn}.</u>
   2.4. Set $x = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id})$ and <u>$w = (\perp, \perp, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2)$</u> with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and π^3_{sLS} thus obtaining the forth round π^4_{sLS} of sLS.
   2.5. Send $(\pi^4_{\mathsf{wsyn}}, \pi^4_{\mathsf{sLS}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{NM4Com}}$ and $\mathsf{NM4Rec}$.

The next hybrid is $\mathcal{H}_2^0(m)$. The only differences between this hybrid and the previous one is that $\mathsf{Sen}_{\mathsf{wsyn}}$ commits, using Π_{wsyn}, to a message 0^λ instead of m. Formally $\mathcal{H}_2^0(z)$ is the following.

$\boldsymbol{\mathcal{H}_2^0(z)}$.

Left session:

1. Second round, upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{wsyn}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows.
   1.1. <u>Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^\lambda, \mathtt{id}$ and 0^λ thus obtaining the second round π^2_{wsyn} of Π_{wsyn}.</u>
   1.2. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the second round π^2_{sLS} of sLS.
   1.3. Pick a message $\mathtt{msg}_1 \leftarrow \{0,1\}^\lambda$.
   1.4. Send $(\pi^2_{\mathsf{wsyn}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg}_1)$ to $\mathsf{Sim}_{\mathsf{NM4Com}}$.
2. Fourth round, upon receiving $(\pi^3_{\mathsf{wsyn}}, \pi^3_{\mathsf{sLS}}, \sigma_1)$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows.
   2.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma) \neq 1$ then abort, continue as follows otherwise.
   2.2. Repeat Step 1.3, 1.4 and follow-up right-session messages up to λ/p times in order to obtain a signature σ_2 of a random message $\mathtt{msg}_2 \neq \mathtt{msg}_1$. Abort in case of failure in obtaining σ_2.
   2.3. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input π^3_{wsyn} thus obtaining the fourth round π^4_{wsyn} of Π_{wsyn}.
   2.4. Set $x = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id})$ and $w = (\perp, \perp, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and π^3_{sLS} thus obtaining the forth round π^4_{sLS} of sLS.
   2.5. Send $(\pi^4_{\mathsf{wsyn}}, \pi^4_{\mathsf{sLS}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{NM4Com}}$ and $\mathsf{NM4Rec}$.

Observe that from the adaptive-input statistical WI of sLS it follows that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_1^0}(z) \equiv_s \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_2^0}(z)$.

So far we have proved that $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_1^m}(z) \equiv_s \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_2^m}(z)$ and $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_2^0}(z) \equiv_s \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_1^0}(z)$ and that both in $\mathcal{H}_1^m(z)$ and $\mathcal{H}_1^0(z)$ the adversary $\mathcal{A}_{\mathsf{NM4Com}}$ commits to $\tilde{m} = \perp$ only with negligible probability. This implies that also in $\mathcal{H}_2^m(z)$ and $\mathcal{H}_2^0(z)$ $\mathcal{A}_{\mathsf{NM4Com}}$ commits to $\tilde{m} = \perp$ only with negligible probability. For this reason now we can prove the indistinguishability between $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_2^m}(z)$ and $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_2^0}(z)$ by relying only on the synchronous weak non-malleability of Π_{wsyn}. Formally we prove the following claim.

Claim 3. *For all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ it holds that $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_2^m}(z) \approx \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_2^0}(z)$.*

Proof. Suppose by contradiction that there exist an adversary $\mathcal{A}_{\mathsf{NM4Com}}$ and a distinguisher $\mathcal{D}_{\mathsf{NM4Com}}$ that can tell apart such two distributions. We can construct a distinguisher $\mathcal{D}_{\mathsf{wsyn}}$ and an adversary $\mathcal{A}_{\mathsf{wsyn}}$ that break the synchronous weak non-malleability of Π_{wsyn}. We observe that we can reduce the security of our scheme to the security of a *weak* non-malleable commitment because the previous claims ensure that the message that $\mathcal{A}_{\mathsf{NM4Com}}$ commits in the right session (using Π_{wsyn}) is valid with overwhelming probability. Let $\mathcal{C}_{\mathsf{wsyn}}$ be the challenger of the synchronous weak NM commitment scheme and let $(0^\lambda, m)$ be the two challenge messages.

Loosely speaking $\mathcal{A}_{\mathsf{wsyn}}$ acts as $\mathsf{NM4Sen}$ with $\mathcal{A}_{\mathsf{NM4Com}}$ with the following differences: 1) $\mathcal{A}_{\mathsf{wsyn}}$ plays as proxy between $\mathcal{C}_{\mathsf{wsyn}}$ and $\mathcal{A}_{\mathsf{NM4Com}}$ w.r.t. messages of Π_{wsyn} in the main thread; 2) a second signature is extracted from the left session through rewinds; 3) random strings are played to simulate the receiver of Π_{wsyn} during rewinds. Then $\mathcal{A}_{\mathsf{wsyn}}$ runs $\mathcal{D}_{\mathsf{wsyn}}$ on input the message $\tilde{m}$ committed by $\mathcal{A}_{\mathsf{wsyn}}$ and his randomness. Therefore $\mathcal{D}_{\mathsf{wsyn}}$ reconstructs the view of $\mathcal{A}_{\mathsf{NM4Com}}$ (by using the randomness received as input) and uses it along with the message $\tilde{m}$ as inputs of $\mathcal{D}_{\mathsf{NM4Com}}$ giving in output what $\mathcal{D}_{\mathsf{NM4Com}}$ outputs. Since by contradiction $\mathcal{D}_{\mathsf{NM4Com}}$ distinguishes between $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_2^m}(z)$ and $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}_2^0}(z)$ also $\mathcal{D}_{\mathsf{wsyn}}$ can tell apart which message has ben committed by the MiM adversary $\mathcal{A}_{\mathsf{wsyn}}$. We stress that to complete the reduction we need to extract two signatures for two distinct messages in the left session. This is done by rewinding the MiM adversary $\mathcal{A}_{\mathsf{NM4Com}}$ from the third to the second round of the left session. When the rewind occurs $\mathcal{A}_{\mathsf{NM4Com}}$ also rewinds the receiver of the right session, rewinding also the receiver of Π_{wsyn} involved in the security reduction. To avoid this issue in the reduction we answer as a receiver of Π_{wsyn} would have done (we remark that this can be done because Π_{wsyn} is public coin) for all rewinds that occur in the right session, allowing the reduction no to rewind the receiver of Π_{wsyn}. Formally the adversary $\mathcal{A}_{\mathsf{wsyn}}$ acts as follows (we recall that this reduction is possible because the message scheduling that we are considering is synchronous).

$\boldsymbol{\mathcal{A}}_{\mathsf{wsyn}}(0^\lambda, m, z)$

Left session:

1. Upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{wsyn}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$ forward π^1_{wsyn} to $\mathcal{C}_{\mathsf{wsyn}}$.
2. Upon receiving π^2_{wsyn} from $\mathcal{C}_{\mathsf{wsyn}}$, run as follows.
   2.1. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the second round π^2_{sLS} of sLS.
   2.2. Pick a message $\mathtt{msg}_1 \leftarrow \{0,1\}^\lambda$.
   2.3. Send $(\pi^2_{\mathsf{wsyn}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg}_1)$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
3. Upon receiving $(\pi^3_{\mathsf{wsyn}}, \pi^3_{\mathsf{sLS}}, \sigma_1)$ from $\mathcal{A}_{\mathsf{NM4Com}}$.
   3.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma) \neq 1$ then abort, otherwise continue as follows.

3.2. Repeat Step 1.3, 1.4 and follow-up right-session messages up to λ/p times in order to obtain a signature σ_2 of a random message $\mathtt{msg}_2 \neq \mathtt{msg}_1$. Abort if case of failure in obtaining σ_2.

3.3. Set $x = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id})$ and $w = (\perp, \perp, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and π^3_{sLS} thus obtaining the forth round π^4_{sLS} of sLS.

3.4. Send $(\pi^4_{\mathsf{wsyn}}, \pi^4_{\mathsf{sLS}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.

Right session:

1. Upon receiving $\tilde{\pi}^1_{\mathsf{wsyn}}$ from from $\mathsf{Rec}_{\mathsf{wsyn}}$, run as follows.

1.1. Run $(\tilde{\mathsf{sk}}, \tilde{\mathsf{vk}}) \leftarrow \mathsf{Gen}(1^\lambda)$.

1.2. Run $\mathcal{V}$ on input 1^λ thus obtaining the first round $\tilde{\pi}^1_{\mathsf{sLS}}$ of sLS.

1.3. Send $(\tilde{\mathsf{vk}}, \tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^1_{\mathsf{wsyn}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.

2. Upon receiving $(\tilde{\pi}^2_{\mathsf{wsyn}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\mathtt{msg}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows.

2.1. If there is not a rewind phase on the left-session send $\tilde{\pi}^2_{\mathsf{wsyn}}$ to $\mathsf{Rec}_{\mathsf{wsyn}}$ and execute the following steps.

i. Upon receiving $\tilde{\pi}^3_{\mathsf{wsyn}}$ from $\mathsf{Rec}_{\mathsf{wsyn}}$.

ii. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the third round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.

iii. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}$ of the message $\tilde{\mathtt{msg}}_1$.

iv. Send $(\tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma})$

2.2. Else if there is a rewind phase on the left then execute the following steps.

i. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the third round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.

ii. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}$ of the message $\tilde{\mathtt{msg}}_1$.

iii. Compute a random $\tilde{\pi}^3_{\mathsf{wsyn}}$.

iv. Send $(\tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.

3. Upon receiving $(\tilde{\pi}^4_{\mathsf{wsyn}}, \tilde{\pi}^4_{\mathsf{sLS}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$, run as follows.

3.1. Set $\tilde{x} = (\tilde{\pi}^1_{\mathsf{wsyn}}, \tilde{\pi}^2_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^4_{\mathsf{wsyn}}, \tilde{\mathtt{id}}, \tilde{\mathsf{vk}})$ and abort iff $(\tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\pi}^4_{\mathsf{sLS}})$ is not accepting for $\mathcal{V}$ with respect to $\tilde{x}$.

3.2. Send $\tilde{\pi}^4_{\mathsf{wsyn}}$ to $\mathsf{Rec}_{\mathsf{wsyn}}$.

Let $\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z)$ be the view and the committed message in the right session by $\mathcal{A}_{\mathsf{wsyn}}$. The distinguisher $\mathcal{D}_{\mathsf{wsyn}}$ takes as input $\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z)$ and runs as follows.

$\boldsymbol{\mathcal{D}_{\mathsf{wsyn}}(\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z))}$ **:** Let $\tilde{m}$ be the committed message sent in the right session by $\mathcal{A}_{\mathsf{wsyn}}$ to $\mathsf{NM4Rec}$. Reconstruct the view of $\mathcal{A}_{\mathsf{NM4Com}}$ (using randomness in $\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z)$) and give it and $\tilde{m}$ as input of the distinguisher $\mathcal{D}_{\mathsf{NM4Com}}$. Output what $\mathcal{D}_{\mathsf{NM4Com}}$ outputs.

The proof is concluded by observing that if $\mathcal{C}_{\mathsf{wsyn}}$ commits to m then the above execution of $\mathcal{A}_{\mathsf{wsyn}}$ corresponds to $\mathcal{H}^m_2(z)$, otherwise it corresponds $\mathcal{H}^0_2(z)$. $\square$

Observe that the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^m_1}(z)$ clearly corresponds to the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}},m}_{\Pi_{\mathsf{NM4Com}}}(z)$ while the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^0_1}(z)$ corresponds to the distribution of $\mathsf{sim}^{\mathsf{Sim}_{\mathsf{NM4Com}}}_{\Pi_{\mathsf{NM4Com}}}(1^\lambda, z)$. With those observations we have proved that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ the following holds:

$$\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}},m}_{\Pi_{\mathsf{NM4Com}}}(z) = \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^m_1}(z) \equiv_s \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^m_2}(z) \approx$$
$$\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^0_2}(z) \equiv_s \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^0_1}(z) = \mathsf{sim}^{\mathsf{Sim}_{\mathsf{NM4Com}}}_{\Pi_{\mathsf{NM4Com}}}(1^\lambda, z).$$

□

Lemma 7. Π_{NM4Com} *is a one-one NM commitment scheme.*

Proof. In Lemma 6 we have shown that Π_{NM4Com} is a synchronous non-malleable commitment scheme. Now we prove that Π_{NM4Com} is non-malleable also when the MiM adversary $\mathcal{A}_{\mathsf{NM4Com}}$ interacts with $\mathsf{NM4Sen}$ and $\mathsf{NM4Rec}$ in a non-synchronized way. More formally we want to argue that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$

$$\{\mathsf{mim}_{\Pi_{\mathsf{NM4Com}}}^{\mathcal{A}_{\mathsf{NM4Com}},m}(z)\}_{z\in\{0,1\}^\star} \approx \{\mathsf{sim}_{\Pi_{\mathsf{NM4Com}}}^{\mathsf{Sim}}(1^\lambda, z)\}_{z\in\{0,1\}^\star}.$$

We prove the indistinguishability through a sequence of hybrid experiments. The first hybrid experiment that we consider is $\mathcal{H}_1^m(z)$, that corresponds to $\mathcal{H}_1^m(z)$ showed in the proof of Lemma 6 with only difference that $\mathcal{A}_{\mathsf{NM4Com}}$ acts in a non-synchronized way. Is easy to see that Claim 1 is valid also in this case.

The second hybrid that we consider is $\mathcal{H}_1^0(z)$. The only difference between this hybrid and the previous one is that $\mathsf{NM4Rec}$ commits to a message 0^λ instead of m. It easy to see that Claim 2 is valid also in this case.

Claim 4. *For all* $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ $\mathsf{mim}_{\mathcal{H}_1^m}^{\mathcal{A}_{\mathsf{NM4Com}}}(z)_{z\in\{0,1\}^\star} \approx \mathsf{mim}_{\mathcal{H}_1^0}^{\mathcal{A}_{\mathsf{NM4Com}}}(z)_{z\in\{0,1\}^\star}$.

Proof. Suppose by contradiction that there exist adversary $\mathcal{A}_{\mathsf{NM4Com}}$ and a distinguisher $\mathcal{D}_{\mathsf{NM4Com}}$ that can tell apart such two distributions. We can construct an adversary $\mathcal{A}_{\mathsf{Hiding}}$ that breaks the hiding of Π_{NM4Com} (recall the hiding of Π_{NM4Com} comes from Lemma 5). Let $\mathcal{C}_{\mathsf{Hiding}}$ be the challenger of the hiding game, we consider the two challenge messages $(m, 0^\lambda)$. The high-level idea of this proof is that $\mathcal{A}_{\mathsf{Hiding}}$ can break the hiding of Π_{NM4Com} using the witness extracted from the sLS transcript computed by $\mathcal{A}_{\mathsf{NM4Com}}$ in the right session. In more details, if the witness extracted from the sLS transcript corresponds to the message committed by $\mathcal{A}_{\mathsf{NM4Com}}$ then $\mathcal{A}_{\mathsf{Hiding}}$ can win the hiding game by running $\mathcal{D}_{\mathsf{NM4Com}}$. Before continuing we observe that Claim 1 ensures that with overwhelming probability the witness extracted from sLS in $\mathcal{H}_1^m(z)$ is the committed message. Furthermore, Claim 2 ensures that with non-negligible probability the witness extracted from sLS in $\mathcal{H}_1^0(z)$ is the committed message.

Similarly to the security proof of Claim 1, in order to extract the witness from sLS we need to construct the augmented machine $\mathcal{M}_{\mathsf{Hiding}}$ that will be used by $\mathcal{A}_{\mathsf{Hiding}}$. $\mathcal{M}_{\mathsf{Hiding}}$ internally executes $\mathcal{A}_{\mathsf{NM4Com}}$, and interacts with an external verifier of the protocol sLS acting as the prover. Notice that there exists an extractor E from the adaptive-input AoK property of sLS. As discussed in App. A, in the case of adaptive-input AoK E takes as input the randomnesses r and r' used by the prover and verifier in an execution where in sLS x has been proved by $\mathcal{P}^\star$.

To construct $\mathcal{M}_{\mathsf{Hiding}}$, $\mathcal{A}_{\mathsf{Hiding}}$ runs $\mathcal{A}_{\mathsf{NM4Com}}$ and acts in the left session as a proxy between $\mathcal{C}_{\mathsf{Hiding}}$ and $\mathcal{A}_{\mathsf{NM4Com}}$ in order to obtain the transcript $\tau_{\mathsf{NM4Com}} = (\pi^1_{\mathsf{NM4Com}}, \pi^2_{\mathsf{NM4Com}}, \pi^3_{\mathsf{NM4Com}}, \pi^4_{\mathsf{NM4Com}})$ of Π_{NM4Com}. After that $\mathcal{A}_{\mathsf{Hiding}}$ uses $\mathcal{M}_{\mathsf{Hiding}}$ to extract the witness of the sLS transcript. The augmented machine $\mathcal{M}_{\mathsf{Hiding}}$ runs $\mathcal{A}_{\mathsf{NM4Com}}$ with same randomness as before acting in the left session with $\mathcal{A}_{\mathsf{NM4Com}}$ as the sender $\mathsf{NM4Sen}$ using the messages $\pi^2_{\mathsf{NM4Com}}, \pi^4_{\mathsf{NM4Com}}$ of τ_{NM4Com}. In the right session $\mathcal{A}_{\mathsf{Hiding}}$ interacts with $\mathcal{A}_{\mathsf{NM4Com}}$ as the receiver $\mathsf{NM4Rec}$ with the only difference that all the messages of sLS received by $\mathcal{A}_{\mathsf{NM4Com}}$ are forwarded to the extractor E and vice versa. Now we describe the augmented machine $\mathcal{M}_{\mathsf{Hiding}}$.

$\boldsymbol{\mathcal{M}_{\mathsf{Hiding}}(\tau_{\mathsf{NM4Com}}, r, z)}$.

Use r as randomness for all next steps.
Run $\mathcal{A}_{\mathsf{NM4Com}}$ and act as follows:

- Upon receiving π^1_{NM4Com} from $\mathcal{A}_{\mathsf{NM4Com}}$ send π^2_{NM4Com} to $\mathcal{A}_{\mathsf{NM4Com}}$.
- Upon receiving π^3_{NM4Com} from $\mathcal{A}_{\mathsf{NM4Com}}$ send π^4_{NM4Com} to $\mathcal{A}_{\mathsf{NM4Com}}$.
- Upon receiving $\tilde{\pi}^1_{\mathsf{sLS}}$ of sLS from the external verifier $\mathcal{V}$.
  1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $1^\lambda, \tilde{\mathtt{id}}$ thus obtaining the first round $\tilde{\pi}^1_{\mathsf{wsyn}}$ of Π_{wsyn}.
  2. Run $(\tilde{\mathsf{sk}}, \tilde{\mathsf{vk}}) \leftarrow \mathsf{Gen}(1^\lambda)$.
  3. Send $(\tilde{\mathsf{vk}}, \tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^1_{\mathsf{wsyn}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
- Upon receiving $(\tilde{\pi}^2_{\mathsf{wsyn}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\mathtt{msg}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$ send $\tilde{\pi}^2_{\mathsf{sLS}}$ to the external verifier $\mathcal{V}$.
- Upon receiving $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS from the external verifier $\mathcal{V}$ execute the following steps.
  1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $\tilde{\pi}^2_{\mathsf{wsyn}}$ thus obtaining the third round $\tilde{\pi}^3_{\mathsf{wsyn}}$ of Π_{wsyn}.
  2. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}})$ to obtain a signature $\tilde{\sigma}$ of the message $\tilde{\mathtt{msg}}$.
  3. Send $(\tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
- Upon receiving $(\tilde{\pi}^4_{\mathsf{wsyn}}, \tilde{\pi}^4_{\mathsf{sLS}})$ set $\tilde{x} = (\tilde{\pi}^1_{\mathsf{wsyn}}, \tilde{\pi}^2_{\mathsf{wsyn}}, \tilde{\pi}^3_{\mathsf{wsyn}}, \tilde{\pi}^4_{\mathsf{wsyn}}, \tilde{\mathtt{id}}, \tilde{\mathsf{vk}})$ and send $(\tilde{\pi}^4_{\mathsf{sLS}}, \tilde{x})$ to the external verifier $\mathcal{V}$.

Now we can conclude the proof of this claim by describing how $\mathcal{A}_{\mathsf{Hiding}}$ works. $\mathcal{A}_{\mathsf{Hiding}}$ runs the extractor E of sLS on input r' (to be used as randomness of the hoenst verifier) and r (to be used as randomness of the malicious prover) using $\mathcal{M}_{\mathsf{Hiding}}$ as $\mathcal{P}^\star$ (recall that an extractor of sLS plays having access to a prover of sLS). We know from Claim 1 and Claim 2 that with overwhelming probability the witness extracted from sLS is the committed message $\tilde{m}$. Therefore $\mathcal{A}_{\mathsf{Hiding}}$ runs $\mathcal{D}_{\mathsf{NM4Com}}$ on input the view of $\mathcal{A}_{\mathsf{NM4Com}}$ of the above execution and the message $\tilde{m}$, and outputs what $\mathcal{D}_{\mathsf{NM4Com}}$ outputs.

Note that $\mathcal{C}_{\mathsf{Hiding}}$ is never rewound for all possible non-trivial non-synchronizing schedulings of $\mathcal{A}_{\mathsf{NM4Com}}$[18]. Therefore, it is always possible to execute the extractor when $\mathcal{A}_{\mathsf{NM4Com}}$ acts in a non-synchronized way.

The proof ends with the observation that if $\mathcal{C}_{\mathsf{Hiding}}$ commits to m then $\mathcal{A}_{\mathsf{NM4Com}}$ acts as in $\mathcal{H}^m_1(z)$, otherwise he acts as in $\mathcal{H}^0_1(z)$. □

Observe that the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}},m}_{\mathcal{H}^m_1}(z)$ clearly corresponds to the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}},m}_{\Pi_{\mathsf{NM4Com}}}(z)$ and the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}},m}_{\mathcal{H}^0_1}(z)$ corresponds to the distribution of $\mathsf{sim}^{\mathsf{Sim}_{\mathsf{NM4Com}}}_{\Pi_{\mathsf{NM4Com}}}(1^\lambda, z)$. With this observation the entire security proof now is almost over because we have proved that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ the following relation holds:

$$\mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}},m}_{\Pi_{\mathsf{NM4Com}}}(z) = \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^m_1(z)}(z) \approx \mathsf{mim}^{\mathcal{A}_{\mathsf{NM4Com}}}_{\mathcal{H}^0_1(z)}(z) = \mathsf{sim}^{\mathsf{Sim}_{\mathsf{NM4Com}}}_{\Pi_{\mathsf{NM4Com}}}(1^\lambda, z).$$

□

The proof of Theorem 1 follows from Lemma 1, Lemma 6 and Lemma 7. We show in Fig. 7 the description of $\mathsf{Sim}_{\mathsf{NM4Com}}$. □

[18] There can be some non-synchronizing scheduling such that the rewind of the extractor would rewind also the challenger. However such schedulings are trivial since such man-in-the-middle adversaries can always be simulated by synchronizing adversaries. Therefore the proof for synchronizing adversaries applies.

Common input: Security parameters: λ. $\mathsf{NM4Sen}$'s identity: $\mathtt{id} \in \{0,1\}^\lambda$.
Internal simulation of the left session:

1. Upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{wsyn}})$ from $\mathcal{A}_{\mathsf{NM4Com}}$.
   1.1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^\lambda, \mathtt{id}$, π^1_{wsyn} and 0^λ thus obtaining the second round π^2_{wsyn} of Π_{wsyn}.
   1.2. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the second round π^2_{sLS} of sLS.
   1.3. Pick a message $\mathtt{msg} \leftarrow \{0,1\}^\lambda$.
   1.4. Send $(\pi^2_{\mathsf{wsyn}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.
2. Upon receiving $(\pi^3_{\mathsf{wsyn}}, \pi^3_{\mathsf{sLS}}, \sigma)$ from $\mathcal{A}_{\mathsf{NM4Com}}$.
   2.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}, \sigma) \neq 1$ then abort, continue as follows otherwise.
   2.2. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input π^3_{wsyn} thus obtaining the fourth round π^4_{wsyn} of Π_{wsyn} and the decommitment information $\mathtt{dec}_{\mathsf{wsyn}}$.
   2.3. Set $x = (\pi^1_{\mathsf{wsyn}}, \pi^2_{\mathsf{wsyn}}, \pi^3_{\mathsf{wsyn}}, \pi^4_{\mathsf{wsyn}}, \mathtt{id}, \mathsf{vk})$ and $w = (m, \mathtt{dec}_{\mathsf{wsyn}}, \perp, \perp, \perp, \perp)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and π^3_{sLS} thus obtaining the forth round π^4_{sLS} of sLS.
   2.4. Send $(\pi^4_{\mathsf{wsyn}}, \pi^4_{\mathsf{sLS}})$ to $\mathcal{A}_{\mathsf{NM4Com}}$.

Stand-alone commitment:

1. $\mathsf{Sim}_{\mathsf{NM4Com}}$ acts as a proxy between $\mathcal{A}_{\mathsf{NM4Com}}$ and $\mathsf{NM4Rec}$.

Figure 7: The simulator $\mathsf{Sim}_{\mathsf{NM4Com}}$.

B.2 Last Part of the Proof of 4-Round NMZK

The security proof goes through a sequence of hybrid experiments that prove that $\mathcal{A}_{\mathsf{ZK}}$ commits to $\tilde{s}_0$ s.t. $(\tilde{x}, \tilde{s}_0 \oplus \tilde{s}_1) \in \mathsf{Rel_L}$ during the simulated experiment. Once we have ensured that in all the hybrids the distribution of the message committed by $\mathcal{A}_{\mathsf{ZK}}$ does not change, we show that if the right session is accepting and $\mathtt{id} \neq \tilde{\mathtt{id}}$ we can recover the witness used by $\mathcal{A}_{\mathsf{ZK}}$ (that is internally executed by $\mathsf{Sim}_{\mathsf{ZK}}$).

Let p be the probability that in the real game $\mathcal{A}_{\mathsf{ZK}}$ concludes the left session. We start considering the hybrid $\mathcal{H}_1$ in which in the left session $\mathcal{P}_{\mathsf{ZK}}$ interacts with $\mathcal{A}_{\mathsf{ZK}}$ and in the right session $\mathcal{V}_{\mathsf{ZK}}$ interacts with $\mathcal{A}_{\mathsf{ZK}}$. We refer to this hybrid experiment as $\mathcal{H}_1(z)$. Details follow below.

$\boldsymbol{\mathcal{H}_1(z)}$.
Left session:

1. Second round, upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{ext}})$ from $\mathcal{A}_{\mathsf{ZK}}$.
   1.1. Pick at random s_0.
   1.2. Run $\mathsf{Sen}_{\mathsf{ext}}$ on input $1^\lambda, \mathtt{id}$, π^1_{ext} and s_0 thus obtaining the 2nd round π^2_{ext} of Π_{ext}.
   1.3. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the 2nd round π^2_{sLS} of sLS.
   1.4. Pick a message $\mathtt{msg} \leftarrow \{0,1\}^\lambda$.
   1.5. Send $(\pi^2_{\mathsf{ext}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg})$ to $\mathcal{A}_{\mathsf{ZK}}$.
2. Fourth round, upon receiving $(\pi^3_{\mathsf{ext}}, \pi^3_{\mathsf{sLS}}, \sigma, x, w)$ from $\mathcal{A}_{\mathsf{ZK}}$.
   2.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}, \sigma) \neq 1$ then abort, continue as follows otherwise.
   2.2. Set $s_1 = s_0 \oplus w$.

2.3. Run $\mathsf{Sen}_{\mathsf{ext}}$ on input $(\pi^1_{\mathsf{ext}}, \pi^3_{\mathsf{ext}})$ thus obtaining the 4th round π^4_{ext} of Π_{ext} and the decommitment information $\mathtt{dec}_{\mathsf{ext}}$.

2.4. Set $x_{\mathsf{sLS}} = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and $w_{\mathsf{sLS}} = (s_0, \mathtt{dec}_{\mathsf{ext}}, \bot, \bot, \bot, \bot)$ with $|x_{\mathsf{sLS}}| = \ell$. Run $\mathcal{P}$ on input x_{sLS}, $w_{\mathsf{sLS}}, \pi^1_{\mathsf{sLS}}$ and π^3_{sLS} thus obtaining the 4th round π^4_{sLS} of sLS.

2.5. Send $(\pi^4_{\mathsf{ext}}, \pi^4_{\mathsf{sLS}}, s_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{ZK}}$ and $\mathcal{V}_{\mathsf{ZK}}$.

We now prove that in the right session of $\mathcal{H}_1(z)$ the MiM adversary $\mathcal{A}_{\mathsf{ZK}}$ does not complete successfully the right session committing to a message s'_0 s.t. $(\tilde{x}, \tilde{s}'_0 \oplus \tilde{s}_1) \notin \mathsf{Rel}_{\mathsf{L}}$. More formally we want to prove the following claim.

Claim 5. *Let $\bar{p}$ be the probability that in the right session of $\mathcal{H}_1(z)$ $\mathcal{A}_{\mathsf{ZK}}$ successfully commits to a message s'_0 s.t. $(\tilde{x}, \tilde{s}'_0 \oplus \tilde{s}_1) \notin \mathsf{Rel}_{\mathsf{L}}$, and the verifier outputs 1. Then $\bar{p} < \nu(\lambda)$ for some negligible function ν.*

The highl-level idea of the proof of this claim follows below. Suppose by contradiction that the claim does not hold, then we can construct an adversary $\mathcal{A}_{\Sigma}$ that breaks the security of the signature scheme Σ. Let vk be the challenge verification key. The idea of the security proof is to create an adversary $\mathcal{A}_{\Sigma}$ that interacts against the MiM adversary $\mathcal{A}_{\mathsf{ZK}}$ sending vk in the 1st round of the right session and extracting the witness used by $\mathcal{A}_{\mathsf{ZK}}$ to execute sLS. Because by contradiction we are assuming that $\mathcal{A}_{\mathsf{ZK}}$ does not commit to a witness then, with non-negligible probability, the witness extracted by sLS will be a pair of signatures (σ_1, σ_2) for a pair of different messages $(\mathtt{msg}_1, \mathtt{msg}_2)$ s.t. $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma_1) = 1$ and $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_2, \sigma_2) = 1$. From the above informal description one can notice that the formal proof strictly follows the one of Claim 1, and for this reason we omit further details.

The 2nd hybrid that we consider is $\mathcal{H}_2(z)$ and it differs from $\mathcal{H}_1(z)$ only in the way the transcript of sLS is computed. In more details, by rewinding the adversary $\mathcal{A}_{\mathsf{ZK}}$ from the 3rd to the 2nd round it is possible to extract two signatures σ_1, σ_2 of two different messages $(\mathtt{msg}_1, \mathtt{msg}_2)$ and use them as a witness to execute the WIAoK sLS. As discussed earlier, after λ/p rewinds a second signature is obtained with overwhelming probability. For the above reason we can claim that the probability that in $\mathcal{H}_2(z)$ the output of the experiment is abort is statistically close to the probability that in $\mathcal{H}_1(z)$ the output of the experiment is abort. The formal description of $\mathcal{H}_2(z)$ is the following experiment.

$\boldsymbol{\mathcal{H}_2(z)}$.

Left session:

1. Second round, upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{ext}})$ from $\mathcal{A}_{\mathsf{ZK}}$.

   1.1. Pick at random s_0.

   1.2. Run $\mathsf{Sen}_{\mathsf{ext}}$ on input 1^λ, $\mathtt{id}$, π^1_{ext} and s_0 thus obtaining the 2nd round π^2_{ext} of Π_{ext}.

   1.3. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the 2nd round π^2_{sLS} of sLS.

   1.4. Pick a message $\mathtt{msg}_1 \leftarrow \{0,1\}^\lambda$.

   1.5. Send $(\pi^2_{\mathsf{ext}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg}_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

2. Fourth round, upon receiving $(\pi^3_{\mathsf{ext}}, \pi^3_{\mathsf{sLS}}, \sigma_1, x, w)$ from $\mathcal{A}_{\mathsf{ZK}}$.

   2.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma) \neq 1$ then abort, continue as follows otherwise.

   2.2. <u>Repeat Step 1.4, 1.5 and follow-up right-session messages up to λ/p times in order to obtain a signature σ_2 of a random message $\mathtt{msg}_2 \neq \mathtt{msg}_1$. Abort if case of failure in obtaining σ_2.</u>

2.3. Run $\mathsf{Sen}_{\mathsf{ext}}$ on input $(\pi^1_{\mathsf{ext}}, \pi^3_{\mathsf{ext}})$ thus obtaining the 4th round π^4_{ext} of Π_{ext}.

2.4. Set $x_{\mathsf{sLS}} = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and $w_{\mathsf{sLS}} = (\bot, \bot, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2)$ with $|x_{\mathsf{sLS}}| = \ell$. Run $\mathcal{P}$ on input x_{sLS}, $w_{\mathsf{sLS}}, \pi^1_{\mathsf{sLS}}$ and π^3_{sLS} thus obtaining the 4th round π^4_{sLS} of sLS.

2.5. Set $s_1 = w \oplus s_0$.

2.6. Send $(\pi^4_{\mathsf{ext}}, \pi^4_{\mathsf{sLS}}, s_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{ZK}}$ and $\mathcal{V}_{\mathsf{ZK}}$.

By the adaptive-input statistical WI of sLS the distribution of the message committed by $\mathcal{A}_{\mathsf{ZK}}$ does not change when moving from $\mathcal{H}_1(z)$ to $\mathcal{H}_2(z)$.

The next hybrid is $\mathcal{H}_3(z)$. The only differences between this hybrid and the previous one is that now $s_0 \oplus s_1$ is a random string. Formally $\mathcal{H}_3(z)$ is the following experiment.

$\boldsymbol{\mathcal{H}_3(z)}$.

Left session:

1. Second round, upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{ext}})$ from $\mathcal{A}_{\mathsf{ZK}}$.

1.1. Pick at random s_0.

1.2. Run $\mathsf{Sen}_{\mathsf{ext}}$ on input $1^\lambda, \mathtt{id}$ and s_0 thus obtaining the 2nd round π^2_{ext} of Π_{ext}.

1.3. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the 2nd round π^2_{sLS} of sLS.

1.4. Pick a message $\mathtt{msg}_1 \leftarrow \{0,1\}^\lambda$.

1.5. Send $(\pi^2_{\mathsf{ext}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg}_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

2. Fourth round, upon receiving $(\pi^3_{\mathsf{ext}}, \pi^3_{\mathsf{sLS}}, \sigma_1, x, w)$ from $\mathcal{A}_{\mathsf{ZK}}$.

2.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma_1) \neq 1$ then abort, continue as follows otherwise.

2.2. Repeat Step 1.4, 1.5 and follow-up right-session messages up to λ/p times in order to obtain a signature σ_2 of a random message $\mathtt{msg}_2 \neq \mathtt{msg}_1$. Abort if case of failure in obtaining σ_2.

2.3. Run $\mathsf{Sen}_{\mathsf{ext}}$ on input $(\pi^1_{\mathsf{ext}}, \pi^3_{\mathsf{ext}})$ thus obtaining the 4th round π^4_{ext} of Π_{ext}.

2.4. Set $x_{\mathsf{sLS}} = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and $w_{\mathsf{sLS}} = (\bot, \bot, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2)$ with $|x_{\mathsf{sLS}}| = \ell$. Run $\mathcal{P}$ on input x_{sLS}, $w_{\mathsf{sLS}}, \pi^1_{\mathsf{sLS}}$ and π^3_{sLS} thus obtaining the 4th round π^4_{sLS} of sLS.

2.5. Pick at random s_1.

2.6. Send $(\pi^4_{\mathsf{ext}}, \pi^4_{\mathsf{sLS}}, s_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{ZK}}$ and $\mathcal{V}_{\mathsf{ZK}}$.

Claim 6. *The distribution of the message committed by $\mathcal{A}_{\mathsf{ZK}}$ does not change between $\mathcal{H}_2(z)$ and $\mathcal{H}_3(z)$.*

Proof. Suppose by contradiction that the claim does not hold. Then $\mathcal{A}_{\mathsf{ZK}}$ in right session commits to a witness with non-negligible probability only when $\mathcal{P}_{\mathsf{ZK}}$ commits to a witness in the left session too. Based on this observation we can construct a distinguisher $\mathcal{D}_{\mathsf{ext}}$ and an adversary $\mathcal{A}_{\mathsf{ext}}$ that break the non-malleability of Π_{ext}. Let $\mathcal{C}_{\mathsf{ext}}$ be the challenger of the NM commitment scheme and let (m_0, m_1) be the two random challenge messages.

Loosely speaking $\mathcal{A}_{\mathsf{ext}}$ acts as $\mathcal{P}_{\mathsf{ZK}}$ with $\mathcal{A}_{\mathsf{ZK}}$ in the left session and as $\mathcal{V}_{\mathsf{ZK}}$ in the right session with the following differences: 1) $\mathcal{A}_{\mathsf{ext}}$ plays as proxy between $\mathcal{C}_{\mathsf{ext}}$ and $\mathcal{A}_{\mathsf{ZK}}$ w.r.t. messages of Π_{ext} in the main thread; 2) a second signature is extracted from the left session through rewinds; 3) random strings are played to simulate the receiver of Π_{ext} during rewinds. 4) $\mathcal{A}_{\mathsf{ext}}$ in the last round of the left session sends s_1 s.t. $s_1 = m_0 \oplus w$.

Then $\mathcal{D}_{\mathsf{ext}}$, on input the message $\tilde{m}$ committed by $\mathcal{A}_{\mathsf{ext}}$ and his randomness, reconstructs the view of $\mathcal{A}_{\mathsf{ZK}}$ and recovers the adaptively chosen statement $\tilde{x}$ proved by $\mathcal{A}_{\mathsf{ZK}}$ and the messages $\tilde{s}_1$ sent by $\mathcal{A}_{\mathsf{ZK}}$ in the last round. If $\tilde{s}_1 \oplus \tilde{m}$ is s.t. $(\tilde{x}, \tilde{s}_1 \oplus \tilde{m}) \in \mathsf{Rel_L}$ then $\mathcal{D}_{\mathsf{ext}}$ outputs 0, and a random bit otherwise. Since by contradiction $\mathcal{A}_{\mathsf{ZK}}$ commits to the witness for $\tilde{x}$ with overwhelming probability only when $\mathcal{P}_{\mathsf{ZK}}$ commits to a witness for x, then $\mathcal{D}_{\mathsf{ext}}$ can tell apart which message has ben committed by the MiM adversary $\mathcal{A}_{\mathsf{ext}}$. We notice that the reduction queries to query only once the receiver of Π_{ext} involved in the reduction. Formally the adversary $\mathcal{A}_{\mathsf{ext}}$ acts as follows.

$\boldsymbol{\mathcal{A}}_{\mathsf{ext}}(m_0, m_1, z)$.

Set $\mathtt{round}_2 = \perp, \mathtt{round}_3 = \perp$.

Left session:

1. Upon receiving $(\mathsf{vk}, \pi^1_{\mathsf{sLS}}, \pi^1_{\mathsf{ext}})$ from $\mathcal{A}_{\mathsf{ZK}}$ forward π^1_{ext} to $\mathcal{C}_{\mathsf{ext}}$.
2. Upon receiving π^2_{ext} from $\mathcal{C}_{\mathsf{ext}}$.
    - 2.1. Run $\mathcal{P}$ on input 1^λ, ℓ and π^1_{sLS} thus obtaining the 2nd round π^2_{sLS} of sLS.
    - 2.2. Pick a message $\mathtt{msg}_1 \leftarrow \{0,1\}^\lambda$.
    - 2.3. Send $(\pi^2_{\mathsf{ext}}, \pi^2_{\mathsf{sLS}}, \mathtt{msg}_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.
3. Upon receiving $(\pi^3_{\mathsf{ext}}, \pi^3_{\mathsf{sLS}}, \sigma_1, x, w)$ from $\mathcal{A}_{\mathsf{ZK}}$.
    - 3.1. If $\mathsf{Ver}(\mathsf{vk}, \mathtt{msg}_1, \sigma) \neq 1$ then abort, continue with the following steps otherwise.
    - 3.2. Repeat Step 2.2, 2.3 and follow-up right-session messages up to λ/p times in order to obtain a signature σ_2 of a random message $\mathtt{msg}_2 \neq \mathtt{msg}_1$. Abort in case of failure in obtaining σ_2.
    - 3.3. Set $x_{\mathsf{sLS}} = (\pi^1_{\mathsf{ext}}, \pi^2_{\mathsf{ext}}, \pi^3_{\mathsf{ext}}, \pi^4_{\mathsf{ext}}, \mathtt{id}, \mathsf{vk}, x, s_1)$ and $w_{\mathsf{sLS}} = (\perp, \perp, \mathtt{msg}_1, \mathtt{msg}_2, \sigma_1, \sigma_2)$ with $|x_{\mathsf{sLS}}| = \ell$. Run $\mathcal{P}$ on input x_{sLS}, $w_{\mathsf{sLS}}, \pi^3_{\mathsf{sLS}}$ and π^3_{sLS} thus obtaining the 4th round π^4_{sLS} of sLS.
    - 3.4. Set $s_1 = m_0 \oplus w$.
    - 3.5. Send $(\pi^4_{\mathsf{ext}}, \pi^4_{\mathsf{sLS}}, s_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

Right session:

1. Upon receiving $\tilde{\pi}^1_{\mathsf{ext}}$ from from $\mathsf{Rec}_{\mathsf{ext}}$.
    - 1.1. Run $(\tilde{\mathsf{sk}}, \tilde{\mathsf{vk}}) \leftarrow \mathsf{Gen}(1^\lambda)$.
    - 1.2. Run $\mathcal{V}$ on input 1^λ thus obtaining the 1st round $\tilde{\pi}^1_{\mathsf{sLS}}$ of sLS.
    - 1.3. Send $(\tilde{\mathsf{vk}}, \tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^1_{\mathsf{ext}})$ to $\mathcal{A}_{\mathsf{ZK}}$.
2. Upon receiving $(\tilde{\pi}^2_{\mathsf{ext}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\mathtt{msg}})$ from $\mathcal{A}_{\mathsf{ZK}}$.
    - 2.1. If there is no rewind phase on the left-session then send $\tilde{\pi}^2_{\mathsf{ext}}$ to $\mathsf{Rec}_{\mathsf{ext}}$ and execute the following steps. execute the following steps.
        - i. Upon receiving $\tilde{\pi}^3_{\mathsf{ext}}$ from $\mathsf{Rec}_{\mathsf{wsyn}}$.
        - ii. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the 3rd round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.
        - iii. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}_1$ of the message $\tilde{\mathtt{msg}}_1$.
        - iv. Send $(\tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma}_1)$.
    - 2.2. Else if there is a rewind phase in the left-session then execute the following steps.
        - i. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the 3rd round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.

ii. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}_1$ of the message $\tilde{\mathtt{msg}}_1$.

iii. Compute a random $\tilde{\pi}^3_{\mathsf{ext}}$.

iv. Send $(\tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma}_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

3. Upon receiving $(\tilde{\pi}^4_{\mathsf{ext}}, \tilde{\pi}^4_{\mathsf{sLS}}, \tilde{s}_1, \tilde{x})$ from $\mathcal{A}_{\mathsf{ZK}}$.

3.1. Set $\tilde{x}_{\mathsf{sLS}} = (\tilde{\pi}^1_{\mathsf{ext}}, \tilde{\pi}^2_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^4_{\mathsf{ext}}, \tilde{\mathtt{id}}, \tilde{\mathsf{vk}}, \tilde{x}, \tilde{s}_1)$ and abort iff $(\tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\pi}^4_{\mathsf{sLS}})$ is not accepting for $\mathcal{V}$ with respect to $\tilde{x}$.

3.2. Send $\tilde{\pi}^4_{\mathsf{ext}}$ to $\mathsf{Rec}_{\mathsf{ext}}$.

Let $\mathsf{mim}^{\mathcal{A}_{\mathsf{ext}}}(z)$ be the view and the committed message in the right session by $\mathcal{A}_{\mathsf{ext}}$. The distinguisher $\mathcal{D}_{\mathsf{ext}}$ takes as input $\mathsf{mim}^{\mathcal{A}_{\mathsf{ext}}}(z)$ and acts as follows.

$\mathcal{D}_{\mathsf{ext}}(\mathsf{mim}^{\mathcal{A}_{\mathsf{ext}}}(z))$: Let $\tilde{m}$ be the committed message sent in the right session by $\mathcal{A}_{\mathsf{ext}}$ to $\mathcal{V}_{\mathsf{ZK}}$. Reconstruct the view of $\mathcal{A}_{\mathsf{ZK}}$ (using randomness in $\mathsf{mim}^{\mathcal{A}_{\mathsf{ext}}}(z)$) and recover the adaptively chosen statement $\tilde{x}$ proved by $\mathcal{A}_{\mathsf{ZK}}$ and the messages $\tilde{s}_1$ sent by $\mathcal{A}_{\mathsf{ZK}}$ in the last round. Since by contradiction $\mathcal{A}_{\mathsf{ZK}}$ contradicts the claim, we have that $\mathcal{A}_{\mathsf{ext}}$ breaks the non-malleability of Π_{ext} because $(\tilde{x}, \tilde{s}_1 \oplus \tilde{m}) \in \mathsf{Rel}_{\mathsf{L}}$ with non-negligible probability in $\mathcal{H}_2(z)$ where $m_0 = \tilde{m}$ is committed in $\mathtt{com}$, while the same happens with negligible probability only in $\mathcal{H}_3(z)$ where m_1 is a random string. Therefore if $(\tilde{x}, \tilde{s}_1 \oplus \tilde{m}) \in \mathsf{Rel}_{\mathsf{L}}$ then $\mathcal{A}_{\mathsf{ext}}$ outputs 0 otherwise $\mathcal{A}_{\mathsf{ext}}$ outputs a random bit.

The proof is concluded by observing that if $\mathcal{C}_{\mathsf{ext}}$ commits to m_0 then the above execution of $\mathcal{A}_{\mathsf{ext}}$ corresponds to $\mathcal{H}_2(z)$, otherwise it corresponds to $\mathcal{H}_3(z)$. □

We now describe how $\mathsf{Sim}_{\mathsf{ZK}}$ of Figure 8 works. Let $\mathtt{ExtCom}$ be the extractor of Π_{ext}. $\mathsf{Sim}_{\mathsf{ZK}}$ runs $\mathtt{ExtCom}$ in order to get the witness $\tilde{w}$ s.t. $(\tilde{x}, \tilde{w}) \in \mathsf{Rel}_{\mathsf{L}}$, where $\tilde{x}$ is the adaptively chosen theorem proved by $\mathcal{A}_{\mathsf{ZK}}$. Before formally describing $\mathsf{Sim}_{\mathsf{ZK}}$ we need to construct an augmented machine $\mathcal{M}_{\mathsf{ext}}$ that is a malicious sender that will be black-box accessed by $\mathtt{ExtCom}$.

$\mathcal{M}_{\mathsf{ext}}(1^\lambda, z)$.

Run $\mathcal{A}_{\mathsf{ZK}}$ with randomness φ.

Left session: Interact with $\mathcal{A}_{\mathsf{ZK}}$ as in $\mathcal{H}_3(z)$.

Right session:

1. Upon receiving $\tilde{\pi}^1_{\mathsf{ext}}$ from from $\mathsf{Rec}_{\mathsf{ext}}$.

1.1. Run $(\tilde{\mathsf{sk}}, \tilde{\mathsf{vk}}) \leftarrow \mathsf{Gen}(1^\lambda)$.

1.2. Run $\mathcal{V}$ on input 1^λ thus obtaining the 1st round $\tilde{\pi}^1_{\mathsf{sLS}}$ of sLS.

1.3. Send $(\tilde{\mathsf{vk}}, \tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^1_{\mathsf{ext}})$ to $\mathcal{A}_{\mathsf{ZK}}$.

2. Upon receiving $(\tilde{\pi}^2_{\mathsf{ext}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\mathtt{msg}})$ from $\mathcal{A}_{\mathsf{ZK}}$.

2.1. If there is no rewind phase on the left-session then send $\tilde{\pi}^2_{\mathsf{ext}}$ to $\mathsf{Rec}_{\mathsf{ext}}$ and execute the following steps. execute the following steps.

i. Upon receiving $\tilde{\pi}^3_{\mathsf{ext}}$ from $\mathsf{Rec}_{\mathsf{wsyn}}$.

ii. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the 3rd round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.

iii. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}_1$ of the message $\tilde{\mathtt{msg}}_1$.

iv. Send $(\tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma}_1)$.

2.2. Else if there is a rewind phase in the left-session then execute the following steps.

i. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the 3rd round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.

Input: Security parameters: λ, auxiliary input: z.

1. Run ExtCom using $\mathcal{M}_{\mathsf{ext}}(1^\lambda, z)$ as a sender, and let $(\tilde{w}, \mathsf{View}_{\mathsf{ext}})$ be the output of ExtCom where $\tilde{w}$ denote the extracted value and $\mathsf{View}_{\mathsf{ext}}$ is the view of $\mathcal{M}_{\mathsf{ext}}(1^\lambda, z)$ that contains the transcript $\tau = (\tilde{\pi}^1_{\mathsf{ext}}, \tilde{\pi}^2_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^4_{\mathsf{ext}})$ (see App. A.1).
2. Use the same randomness φ used by $\mathcal{M}_{\mathsf{ext}}(1^\lambda, z)$ and $\mathcal{A}_{\mathsf{ZK}}$, and reconstruct the view View of $\mathcal{A}_{\mathsf{ZK}}$ by executing the following steps.
   2.1. Run $\mathcal{A}_{\mathsf{ZK}}$.
   2.2. Interact in the left session with $\mathcal{A}_{\mathsf{ZK}}$ as in $\mathcal{H}_3(z)$.
   2.3. Run $(\tilde{\mathsf{sk}}, \tilde{\mathsf{vk}}) \leftarrow \mathsf{Gen}(1^\lambda)$.
   2.4. Run $\mathcal{V}$ on input 1^λ thus obtaining the 1st round $\tilde{\pi}^1_{\mathsf{sLS}}$ of sLS.
   2.5. Send $(\tilde{\mathsf{vk}}, \tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^1_{\mathsf{ext}})$ to $\mathcal{A}_{\mathsf{ZK}}$.
   2.6. Upon receiving $(\tilde{\pi}^2_{\mathsf{ext}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\mathtt{msg}})$ from $\mathcal{A}_{\mathsf{ZK}}$.
      i. If there is no rewind phase in the left session then execute the following steps.
         A. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the 3rd round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.
         B. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}_1$ of the message $\tilde{\mathtt{msg}}_1$.
         C. Send $(\tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma}_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.
      ii. Else if there is a rewind phase in the left-session then execute the following steps.
         A. Run $\mathcal{V}$ on input $\tilde{\pi}^2_{\mathsf{sLS}}$ thus obtaining the 3rd round $\tilde{\pi}^3_{\mathsf{sLS}}$ of sLS.
         B. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}_1$ of the message $\tilde{\mathtt{msg}}_1$.
         C. Compute a random third round $\tilde{\pi}^\star_{\mathsf{ext}}$ of Π_{ext}.
         D. Send $(\tilde{\pi}^\star_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma}_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.
   2.7. Upon receiving $(\tilde{\pi}^4_{\mathsf{ext}}, \tilde{\pi}^4_{\mathsf{sLS}}, \tilde{s}_1, \tilde{x})$ from $\mathcal{A}_{\mathsf{ZK}}$, set $\tilde{x}_{\mathsf{sLS}} = (\tilde{\pi}^1_{\mathsf{ext}}, \tilde{\pi}^2_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^4_{\mathsf{ext}}, \tilde{\mathtt{id}}, \tilde{\mathsf{vk}}, \tilde{x}, \tilde{s}_1)$ and abort iff $(\tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\pi}^4_{\mathsf{sLS}})$ is not accepting for $\mathcal{V}$ with respect to $\tilde{x}$.
3. Let T be the transcript of the main thread in the above execution. Output $(\mathsf{View} = (\varphi, T), \tilde{w})$.

Figure 8: The simulator $\mathsf{Sim}_{\mathsf{ZK}}$.

ii. Run $\mathsf{Sign}(\tilde{\mathsf{sk}}, \tilde{\mathtt{msg}}_1)$ to obtain a signature $\tilde{\sigma}_1$ of the message $\tilde{\mathtt{msg}}_1$.
iii. Compute a random $\tilde{\pi}^3_{\mathsf{ext}}$.
iv. Send $(\tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\sigma}_1)$ to $\mathcal{A}_{\mathsf{ZK}}$.

3. Upon receiving $(\tilde{\pi}^4_{\mathsf{ext}}, \tilde{\pi}^4_{\mathsf{sLS}}, \tilde{s}_1, \tilde{x})$ from $\mathcal{A}_{\mathsf{ZK}}$.
   3.1. Set $\tilde{x}_{\mathsf{sLS}} = (\tilde{\pi}^1_{\mathsf{ext}}, \tilde{\pi}^2_{\mathsf{ext}}, \tilde{\pi}^3_{\mathsf{ext}}, \tilde{\pi}^4_{\mathsf{ext}}, \tilde{\mathtt{id}}, \tilde{\mathsf{vk}}, \tilde{x}, \tilde{s}_1)$ and abort iff $(\tilde{\pi}^1_{\mathsf{sLS}}, \tilde{\pi}^2_{\mathsf{sLS}}, \tilde{\pi}^3_{\mathsf{sLS}}, \tilde{\pi}^4_{\mathsf{sLS}})$ is not accepting for $\mathcal{V}$ with respect to $\tilde{x}$.
   3.2. Send $\tilde{\pi}^4_{\mathsf{ext}}$ to $\mathsf{Rec}_{\mathsf{ext}}$.

Similarly to the black-box simulator of [GK96] we assume w.l.o.g. that if a transcript τ appears in the final output of a black-box extractor ExtCom, then ExtCom has queried the sender of the extractable commitment Π_{ext} on every prefix of τ. $\mathsf{Sim}_{\mathsf{ZK}}$, in order to reconstruct the full transcript T of the entire execution, interacts in the right session with $\mathcal{A}_{\mathsf{ZK}}$ by playing messages of τ. See Figure 8 for more details.

Common input: security parameters: λ, $(\lambda_{\mathsf{wsyn}}, \lambda_{\mathsf{LS}}, \ell) = \mathsf{Params}(\lambda)$.
Identity: $\mathtt{id} \in \{0,1\}^\lambda$.
Internal simulation of the left session:

1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^{\lambda_{\mathsf{wsyn}}}$, $\mathtt{id}$ and 0^λ thus obtaining the first round $\mathsf{a}_{\mathsf{wsyn}}$ of Π_{wsyn}.
2. Run $\mathcal{P}$ on input $1^{\lambda_{\mathsf{LS}}}$ and ℓ thus obtaining the first round a_{LS} of LS.
3. Send $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{a}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
4. Upon receiving $(\mathsf{c}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{LS}}, Y)$ from $\mathcal{A}_{\mathsf{NMCom}}$.
   4.1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $\mathsf{c}_{\mathsf{wsyn}}$ thus obtaining the third round $\mathsf{z}_{\mathsf{wsyn}}$ of Π_{wsyn}.
   4.2. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ thus obtaining the decommitment information $\mathtt{dec}_{\mathsf{wsyn}}$ of Π_{wsyn}.
   4.3. Set $x = \big(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id}\big)$ and $w = (m, \mathtt{dec}_{\mathsf{wsyn}}, \perp)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w, and c_{LS} thus obtaining the third round z_{LS} of LS.
   4.4. Send $(\mathsf{z}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.

Stand-alone commitment:

1. $\mathsf{Sim}_{\mathsf{NMCom}}$ acts as a proxy between $\mathcal{A}_{\mathsf{NMCom}}$ and NMRec.

Figure 9: The simulator $\mathsf{Sim}_{\mathsf{NMCom}}$.

B.3 Proof of NM of the 3-Round NM Commitment Scheme

We now formally prove that the commitment scheme Π_{NMCom} is non-malleable. This security proof consists of two parts. In the first part we consider a MiM adversary $\mathcal{A}_{\mathsf{NMCom}}$ that interacts only in a synchronized way with NMSen and NMRec showing that our scheme is synchronous one-one non-malleable. In the second part we argue that the commitment scheme is non-malleable also when $\mathcal{A}$ acts in a non-synchronized way. Putting together these two arguments we are able to conclude the proof on non-malleability.

Lemma 8. Π_{NMCom} *is a synchronous one-one NM commitment scheme.*

Proof. We show that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ it holds that:

$$\{\mathsf{mim}_{\Pi_{\mathsf{NMCom}}}^{\mathcal{A}_{\mathsf{NMCom}},m}(z)\}_{z\in\{0,1\}^\star} \approx \{\mathsf{sim}_{\Pi_{\mathsf{NMCom}}}^{\mathsf{Sim}_{\mathsf{NMCom}}}(1^\lambda, z)\}_{z\in\{0,1\}^\star}$$

where $\mathsf{Sim}_{\mathsf{NMCom}}$ is the simulator depicted in Fig. 9.

In the first experiment, in the left session NMSen commits to m playing with $\mathcal{A}_{\mathsf{NMCom}}$, while in the right session $\mathcal{A}_{\mathsf{NMCom}}$ commits on the right by playing with NMRec. We refer to this experiment as $\mathcal{H}_1^m(z)$. Details follow below.

$\boldsymbol{\mathcal{H}_1^m(z)}$.
Left session:

1. First round.
   1.1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^{\lambda_{\mathsf{wsyn}}}$, $\mathtt{id}$ and m thus obtaining the first round $\mathsf{a}_{\mathsf{wsyn}}$ of Π_{wsyn}.
   1.2. Run $\mathcal{P}$ on input $1^{\lambda_{\mathsf{LS}}}$ and ℓ thus obtaining the first round a_{LS} of LS.
   1.3. Send $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{a}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
2. Third round, upon receiving $(\mathsf{c}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{LS}}, Y)$ from $\mathcal{A}_{\mathsf{NMCom}}$, run as follows.

2.1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $\mathsf{c}_{\mathsf{wsyn}}$ thus obtaining the third round $\mathsf{z}_{\mathsf{wsyn}}$ of Π_{wsyn} and the decommitment information $\mathtt{dec}_{\mathsf{wsyn}}$.
2.2. Set $x = (\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id})$ and $w = (m, \mathtt{dec}_{\mathsf{wsyn}}, \perp)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and c_{LS} thus obtaining the third round z_{LS} of LS.
2.3. Send $(\mathsf{z}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{NMCom}}$ and NMRec.

The distribution of $\mathsf{mim}_{\mathcal{H}_1^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$ clearly corresponds to the distribution of $\mathsf{mim}_{\Pi_{\mathsf{NMCom}}}^{\mathcal{A}_{\mathsf{NMCom}},m}(z)$. We now prove that in the right session $\mathcal{A}_{\mathsf{NMCom}}$ does not commit to a message $\tilde{m} =\perp$. We can do so by proving that the LS proof of the right session is computed by $\mathcal{A}_{\mathsf{NMCom}}$ without using as witness a value $\tilde{y}$ s.t. $f(\tilde{y}) = \tilde{Y}$, where $\tilde{Y}$ is the value sent to $\mathcal{A}_{\mathsf{NMCom}}$ in the second round of the right session. Formally we want have the following claim.

Claim 7. *Let $\bar{p}$ be the probability that in the right session $\mathcal{A}_{\mathsf{NMCom}}$ successfully commits to $\tilde{m} =\perp$. Then $\bar{p} < \nu(\lambda)$ for some negligible function ν.*

Proof. Suppose by contradiction that the claim does not hold, then we can construct an adversary $\mathcal{A}_f$ that inverts the OWP f in polynomial time. Formally we consider a challenger $\mathcal{C}_f$ of f that chooses a random $Y \in \{0,1\}^\lambda$ and sends it to $\mathcal{A}_f$. $\mathcal{A}_f$ wins if it gives as output y s.t. $Y = f(y)$. Before describing the adversary we need to consider the augmented machine $\mathcal{M}_f$ that will be used by $\mathcal{A}_f$ to extract the witness from LS by using the extractor E (that exists from the property of adaptive-input PoK enjoyed by LS). Recall that in the case of an adaptive-input PoK, the extractor takes as input the randomnesses r of the prover and r' of the verifier of an execution of LS when theorem x has been proved by $\mathcal{P}^\star$. Now we are ready to describe how $\mathcal{M}_f$ works. $\mathcal{M}_f$ internally runs $\mathcal{A}_{\mathsf{NMCom}}$ with randomness r and interacts with him as the sender NMSen does in the left session and as the receiver NMRec does in the right session. The only difference is that all messages of LS of the right session are forwarded to the verifier $\mathcal{V}$ and vice versa. Formally $\mathcal{M}_f$ acts as follows.

$\boldsymbol{\mathcal{M}}_f(z, Y, r)$.

Execute the following steps with randomness r

- Run NMSen on input m with $\mathcal{A}_{\mathsf{NMCom}}$ as in $\mathcal{H}_1^m(z)$.
- Upon receiving $(\tilde{\mathsf{a}}_{\mathsf{wsyn}}, \tilde{\mathsf{a}}_{\mathsf{LS}})$ from $\mathcal{A}_{\mathsf{NMCom}}$, send $\tilde{\mathsf{a}}_{\mathsf{LS}}$ to $\mathcal{V}$.
- Upon receiving $\tilde{\mathsf{c}}_{\mathsf{LS}}$ from $\mathcal{V}$, run as follows.
  1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $\tilde{\mathtt{id}}$ and $\tilde{\mathsf{a}}_{\mathsf{wsyn}}$ thus obtaining the second round $\tilde{\mathsf{c}}_{\mathsf{wsyn}}$ of Π_{wsyn}.
  2. Set $\tilde{Y} = Y$.
  3. Send $(\tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{LS}}, \tilde{Y})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
- Upon receiving the 3rd round of the right session $(\tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{LS}})$ set $\tilde{x} = (\tilde{\mathsf{a}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{Y}, \tilde{\mathtt{id}})$ and send $(\tilde{\mathsf{z}}_{\mathsf{LS}}, \tilde{x})$ to $\mathcal{V}$.

Now we can conclude the proof of this claim by describing how $\mathcal{A}_f$ works. $\mathcal{A}_f$ runs E on input the randomness r' (used by the verifier in an execution where x has been proved) and uses $\mathcal{M}_f$ as prover with randomness r (recall that an extractor of LS plays only having access to a prover of LS). Notice that the above execution of $\mathcal{M}_f$ is distributed identically to $\mathcal{H}_1^m(z)$. Since by contradiction

$\mathcal{A}_{\mathsf{NMCom}}$ is successful with non-negligible probability, we have that with non-negligible probability $\mathcal{A}_f$ in polynomial time[19] outputs the value y such that $f(y) = Y$. $\square$

The next hybrid experiment that we consider is $\mathcal{H}_1^0(z)$ that is equal to $\mathcal{H}_1^m(z)$ with the only difference that the message committed using Π_{wsyn} is 0^λ instead of m. Similarly to $\mathcal{H}_1^m(z)$, we have for $\mathcal{H}_1^0(z)$ the following claim.

Claim 8. *The probability that in the right session $\mathcal{A}_{\mathsf{NMCom}}$ successfully commits to a message $\tilde{m} = \perp$ is $p < \nu(\lambda)$ for some negligible function ν.*

Proof. The security proof strictly follows the one of Claim 7. $\square$

The next hybrid that we consider is $\mathcal{H}_2^m(z)$. $\mathcal{H}_2^m(z)$ differs from $\mathcal{H}_1^m(z)$ only in the witness used to compute the LS transcript. Formally $\mathcal{H}_2^m(z)$ is the following experiment.

$\boldsymbol{\mathcal{H}_2^m(z)}$.

Left session:

1. First round
   1.1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^{\lambda_{\mathsf{wsyn}}}$, $\mathtt{id}$ and m thus obtaining the first round $\mathsf{a}_{\mathsf{wsyn}}$ of Π_{wsyn}.
   1.2. Run $\mathcal{P}$ on input $1^{\lambda_{\mathsf{LS}}}$ and ℓ thus obtaining the first round a_{LS} of LS.
   1.3. Send $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{a}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
2. Third round, upon receiving $(\mathsf{c}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{LS}}, Y)$ from $\mathcal{A}_{\mathsf{NMCom}}$, run as follows.
   2.1. <u>Run in time $\tilde{T}_f$ to compute y s.t. $Y = f(y)$.</u>
   2.2. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $\mathsf{c}_{\mathsf{wsyn}}$ thus obtaining the third round $\mathsf{z}_{\mathsf{wsyn}}$ of Π_{wsyn}.
   2.3. Set $x = (\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id})$ and $\underline{w = (\perp, \perp, y)}$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and c_{LS} thus obtaining the third round z_{LS} of LS.
   2.4. Send $(\mathsf{z}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{NMCom}}$ and NMRec.

Claim 9. *For all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ it holds that $\mathsf{mim}_{\mathcal{H}_1^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z) \approx \mathsf{mim}_{\mathcal{H}_2^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$.*

Proof. Suppose by contradiction that there exist adversary $\mathcal{A}_{\mathsf{NMCom}}$ and a distinguisher $\mathcal{D}_{\mathsf{NMCom}}$ that can tell apart such two distributions. We can use this adversary and the associated distinguisher to construct ad adversary $\mathcal{A}_{\mathsf{LS}}$ for the T_{LS}-witness-indistinguishable of LS. Let $\mathcal{C}_{\mathsf{LS}}$ be the adaptive-input WI challenger. In the left session $\mathcal{A}_{\mathsf{LS}}$ acts as NMSen with $\mathcal{A}_{\mathsf{NMCom}}$ except for the messages of LS for which he acts as a proxy between $\mathcal{C}_{\mathsf{LS}}$ and $\mathcal{A}_{\mathsf{NMCom}}$. In the right session he acts as NMRec with $\mathcal{A}_{\mathsf{NMCom}}$. After the execution of the right session, $\mathcal{A}_{\mathsf{LS}}$ runs in time $\tilde{T}_{\mathsf{wsyn}}$ to obtain the message $\tilde{m}$ committed by $\mathcal{A}_{\mathsf{NMCom}}$ in the right session using Π_{wsyn}. Finally $\mathcal{A}_{\mathsf{LS}}$ gives $\tilde{m}$ and the output view of $\mathcal{A}_{\mathsf{NMCom}}$ as input to the distinguisher $\mathcal{D}_{\mathsf{NMCom}}$ and outputs what $\mathcal{D}_{\mathsf{NMCom}}$ outputs. Since by contradiction $\mathcal{D}_{\mathsf{NMCom}}$ distinguishes $\mathsf{mim}_{\mathcal{H}_1^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$ from $\mathsf{mim}_{\mathcal{H}_2^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$ we have

[19] The extractor is an expected polynomial-time algorithm while $\mathcal{A}_f$ must be a strict polynomial-time algorithm. Therefore $\mathcal{A}_f$ will run the extractor up to a given upperbounded number of steps that is higher than the expected running time of the extractor. Obviously with non-negligible probability the *truncated* extraction procedure will be completed successfully and this is sufficient for $\mathcal{A}_f$ to invert f. The same standard argument about truncating the execution of an expected polynomial-time algorithm is used in another proofs but for simplicity we will not repeat this discussion.

that $\mathcal{A}_{\mathsf{LS}}$ can tell apart with non-negligible advantage which witness has been used to compute the transcript of LS. Formally the adversary $\mathcal{A}_{\mathsf{LS}}$ works as follows.

$\boldsymbol{\mathcal{A}_{\mathsf{LS}}(z)}$.

- Act as a honest receiver NMRec with $\mathcal{A}_{\mathsf{NMCom}}$, when $\mathcal{A}_{\mathsf{NMCom}}$ plays as as a sender.
- Upon receiving a_{LS} from $\mathcal{C}_{\mathsf{LS}}$, run as follows.
  1. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^{\lambda_{\mathsf{wsyn}}}$ $\mathtt{id}$ and m thus obtaining the first round $\mathsf{a}_{\mathsf{wsyn}}$ of Π_{wsyn}.
  2. Send $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{a}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
- Upon receiving $(\mathsf{c}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{LS}}, Y)$ from $\mathcal{A}_{\mathsf{NMCom}}$, run as follows.
  1. Run in time $\tilde{T}_f$ to compute y s.t. $Y = f(y)$.
  2. Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $\mathsf{c}_{\mathsf{wsyn}}$ thus obtaining the third round $\mathsf{z}_{\mathsf{wsyn}}$ of Π_{wsyn} and the decommitment information $\mathtt{dec}_{\mathsf{wsyn}}$.
  3. Set $x = (\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id})$, $w_0 = (m, \mathtt{dec}_{\mathsf{wsyn}}, \perp)$, $w_1 = (\perp, \perp, y)$ and send $(x, \mathsf{c}_{\mathsf{LS}}, w_0, w_1)$ to $\mathcal{C}_{\mathsf{LS}}$.
  4. Upon receiving z_{LS} from $\mathcal{C}_{\mathsf{LS}}$, send $(\mathsf{z}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.

  After the execution with $\mathcal{A}_{\mathsf{NMCom}}$, $\mathcal{A}_{\mathsf{LS}}$ computes the following steps:

1. Let $(\tilde{\mathsf{a}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{\mathtt{id}})$ be the commitment received by NMRec when playing as in Π_{wsyn}. Run in time $\tilde{T}_{\mathsf{wsyn}}$ to compute $\tilde{m} : \exists\ \tilde{\mathtt{dec}}_{\mathsf{wsyn}}$ s.t. $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $(\tilde{m}, \tilde{\mathtt{dec}}_{\mathsf{wsyn}})$ accepts $\tilde{m}$ as a decomitment of $(\tilde{\mathsf{a}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{\mathtt{id}})$.
2. Give $\tilde{m}$ and the view of $\mathcal{A}_{\mathsf{NMCom}}$ to the distinguisher $\mathcal{D}_{\mathsf{NMCom}}$.
3. Output what $\mathcal{D}_{\mathsf{NMCom}}$ outputs.

The proof ends with the observation that if $\mathcal{C}_{\mathsf{LS}}$ has used w_0 as a witness then $\mathcal{A}_{\mathsf{NMCom}}$ acts as in $\mathcal{H}_1^m$, otherwise he acts as in $\mathcal{H}_2^m$. □

The next hybrid is $\mathcal{H}_2^0(z)$. The only differences between this hybrid and $\mathcal{H}_2^m(z)$ is that $\mathsf{Sen}_{\mathsf{wsyn}}$ commits, using Π_{wsyn}, to a message 0^λ instead of m. Formally $\mathcal{H}_2^0(z)$ is the following.

$\boldsymbol{\mathcal{H}_2^0(z)}$.

Left session:

1. First round.
   - 1.1. <u>Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $1^{\lambda_{\mathsf{wsyn}}}$ $\mathtt{id}$, and 0^λ thus obtaining the first round $\mathsf{a}_{\mathsf{wsyn}}$ of Π_{wsyn}.</u>
   - 1.2. Run $\mathcal{P}$ on input $1^{\lambda_{\mathsf{LS}}}$ and ℓ thus obtaining the first round a_{LS} of LS.
   - 1.3. Send $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{a}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
2. Third round, upon receiving $(\mathsf{c}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{LS}}, Y)$ from $\mathcal{A}_{\mathsf{NMCom}}$, run as follows.
   - 2.1. Run in time $\tilde{T}_f$ to compute y s.t. $Y = f(y)$.
   - 2.2. <u>Run $\mathsf{Sen}_{\mathsf{wsyn}}$ on input $\mathsf{c}_{\mathsf{wsyn}}$ thus obtaining the third round $\mathsf{z}_{\mathsf{wsyn}}$ of Π_{wsyn}.</u>
   - 2.3. Set $x = (\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id})$ and $w = (\perp, \perp, y)$ with $|x| = \ell$. Run $\mathcal{P}$ on input x, w and c_{LS} thus obtaining the third round z_{LS} of LS.
   - 2.4. Send $(\mathsf{z}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.

Right session: act as a proxy between $\mathcal{A}_{\mathsf{NMCom}}$ and NMRec.

Claim 10. *For all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ it holds that* $\mathsf{mim}_{\mathcal{H}_2^0}^{\mathcal{A}_{\mathsf{NMCom}}}(z) \approx \mathsf{mim}_{\mathcal{H}_1^0}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$.

Proof. The security proof follows the same idea of the proof of Claim 9. □

Until now we have proved that $\mathsf{mim}_{\mathcal{H}_1^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z) \approx \mathsf{mim}_{\mathcal{H}_2^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$ and $\mathsf{mim}_{\mathcal{H}_2^0}^{\mathcal{A}_{\mathsf{NMCom}}}(z) \approx \mathsf{mim}_{\mathcal{H}_1^0}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$ and that both in $\mathcal{H}_1^m(z)$ and $\mathcal{H}_1^0(z)$ the adversary $\mathcal{A}_{\mathsf{NMCom}}$ commits to $\tilde{m} = \perp$ only with negligible probability. This implies that also in $\mathcal{H}_2^m(z)$ and $\mathcal{H}_2^0(z)$ $\mathcal{A}_{\mathsf{NMCom}}$ commits to $\tilde{m} = \perp$ only with negligible probability. For this reason now we can prove the indistinguishability between $\mathsf{mim}_{\mathcal{H}_2^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$ and $\mathsf{mim}_{\mathcal{H}_2^0}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$ by relying only on the synchronous weak one-one non-malleability of Π_{wsyn}. Formally we prove the following claim.

Claim 11. *For all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ it holds that* $\mathsf{mim}_{\mathcal{H}_2^m}^{\mathcal{A}_{\mathsf{NMCom}}}(z) \approx \mathsf{mim}_{\mathcal{H}_2^0}^{\mathcal{A}_{\mathsf{NMCom}}}(z)$.

Proof. Suppose by contradiction that there exists an adversary $\mathcal{A}_{\mathsf{NMCom}}$ and a distinguisher $\mathcal{D}_{\mathsf{NMCom}}$ that can tell apart such two distributions. We can construct a distinguisher $\mathcal{D}_{\mathsf{wsyn}}$ and an adversary $\mathcal{A}_{\mathsf{wsyn}}$ that break the synchronous weak one-one non-malleability of Π_{wsyn}. It is important to observe that we can reduce the security of our scheme to the security of a synchronous weak one-one NM commitment because the previous claims ensure that the message that $\mathcal{A}_{\mathsf{NMCom}}$ commits in the right session (using Π_{wsyn}) is valid with overwhelming probability. Let $\mathcal{C}_{\mathsf{wsyn}}$ be the challenger of the *synchronous weak* one-one NM commitment and let $(0^\lambda, m)$ be the two challenge messages given by $\mathcal{C}_{\mathsf{wsyn}}$.

Loosely speaking in the left session $\mathcal{A}_{\mathsf{wsyn}}$ acts as NMSen with $\mathcal{A}_{\mathsf{NMCom}}$ with the difference that w.r.t. to the messages of Π_{wsyn} he acts as a proxy between $\mathcal{C}_{\mathsf{wsyn}}$ and $\mathcal{A}_{\mathsf{NMCom}}$. In the right session he acts as NMRec with $\mathcal{A}_{\mathsf{NMCom}}$ and, as in the left session, acts as a proxy w.r.t. the messages of Π_{wsyn} exchanged between $\mathsf{Rec}_{\mathsf{wsyn}}$ and $\mathcal{A}_{\mathsf{wsyn}}$. Then $\mathcal{A}_{\mathsf{wsyn}}$ runs $\mathcal{D}_{\mathsf{wsyn}}$ on input the message $\tilde{m}$ committed by $\mathcal{A}_{\mathsf{wsyn}}$ and his randomness. $\mathcal{D}_{\mathsf{wsyn}}$ reconstructs the view of $\mathcal{A}_{\mathsf{NMCom}}$ (by using the same randomness) and uses it and the message $\tilde{m}$ as inputs of $\mathcal{D}_{\mathsf{NMCom}}$ giving in output what $\mathcal{D}_{\mathsf{NMCom}}$ outputs. Since by contradiction $\mathcal{D}_{\mathsf{NMCom}}$ distinguishes between $\mathsf{mim}_{\mathcal{H}_2^m}^{\mathcal{A}_{\mathsf{NM4Com}}}(z)$ and $\mathsf{mim}_{\mathcal{H}_2^0}^{\mathcal{A}_{\mathsf{NM4Com}}}(z)$, we have that $\mathcal{D}_{\mathsf{wsyn}}$ tells apart which message has ben committed by the MiM adversary $\mathcal{A}_{\mathsf{wsyn}}$.

The adversary $\mathcal{A}_{\mathsf{wsyn}}$ acts as follows (we recall that this reduction is possible only because the message scheduling that we are considering is synchronous).

$\boldsymbol{\mathcal{A}_{\mathsf{wsyn}}(0^\lambda, m, z)}$.

Left session:

1. Upon receiving $\mathsf{a}_{\mathsf{wsyn}}$ from $\mathcal{C}_{\mathsf{wsyn}}$, run as follows.
   - 1.1. Run $\mathcal{P}$ on input $1^{\lambda_{\mathsf{LS}}}$ and ℓ thus obtaining the first round a_{LS} of LS.
   - 1.2. Send $(\mathsf{a}_{\mathsf{wsyn}}, \mathsf{a}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
2. Upon receiving $(\mathsf{c}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{LS}}, Y)$ from $\mathcal{A}_{\mathsf{NMCom}}$, run as follows.
   - 2.1. Run in time $\tilde{T}_f$ to compute y s.t. $Y = f(y)$.
   - 2.2. Set $x = (\mathsf{a}_{\mathsf{wsyn}}, \mathsf{c}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{wsyn}}, Y, \mathtt{id})$, $w = (\perp, \perp, y)$. Run $\mathcal{P}$ on input x, w and c_{LS} thus obtaining the third round z_{LS} of LS.
3. Upon receiving $\mathsf{z}_{\mathsf{wsyn}}$ from $\mathcal{C}_{\mathsf{wsyn}}$, send $(\mathsf{z}_{\mathsf{wsyn}}, \mathsf{z}_{\mathsf{LS}})$ to $\mathcal{A}_{\mathsf{NMCom}}$.

Right session:

1. Forward $\tilde{\mathsf{a}}_{\mathsf{wsyn}}$ to $\mathsf{Rec}_{\mathsf{wsyn}}$.

2. Upon receiving $\tilde{\mathsf{c}}_{\mathsf{wsyn}}$ from $\mathsf{Rec}_{\mathsf{wsyn}}$, run as follows.
   2.1. Pick a random $\tilde{Y}$.
   2.2. Run $\mathcal{V}$ on input $\tilde{\mathsf{a}}_{\mathsf{LS}}$ thus obtaining the second round $\tilde{\mathsf{c}}_{\mathsf{LS}}$ of Π_{LS}.
   2.3. Send $(\tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{LS}}, Y)$ to $\mathcal{A}_{\mathsf{NMCom}}$.
3. Upon receiving $\tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{LS}}$ from $\mathcal{A}_{\mathsf{NMCom}}$, run as follows:
   3.1. Set $\tilde{x} = (\tilde{\mathsf{a}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{Y}, \tilde{\mathtt{id}})$ and abort iff $(\tilde{\mathsf{a}}_{\mathsf{LS}}, \tilde{\mathsf{c}}_{\mathsf{LS}}, \tilde{\mathsf{z}}_{\mathsf{LS}})$ is not accepted by $\mathcal{V}$ for $\tilde{x} \in L$.
   3.2. Send $\tilde{\mathsf{z}}_{\mathsf{wsyn}}$ to $\mathsf{Rec}_{\mathsf{wsyn}}$.

Let $\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z)$ be the view and the committed message in the right session by $\mathcal{A}_{\mathsf{wsyn}}$. The distinguisher $\mathcal{D}_{\mathsf{wsyn}}$ takes as input $\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z)$ and acts as follows.

$\mathcal{D}_{\mathsf{wsyn}}(\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z))$: Let $\tilde{m}$ be the committed message sent in the right session by $\mathcal{A}_{\mathsf{wsyn}}$ to $\mathsf{Rec}_{\mathsf{wsyn}}$. Reconstruct the view of $\mathcal{A}_{\mathsf{NMCom}}$ (using the randomness given in $\mathsf{mim}^{\mathcal{A}_{\mathsf{wsyn}}}(z)$) and give it and $\tilde{m}$ to the distinguisher $\mathcal{D}_{\mathsf{NMCom}}$. Output what $\mathcal{D}_{\mathsf{NMCom}}$ outputs. We observe that the reduction could fail if $\mathcal{A}_{\mathsf{NMCom}}$ commit to $\perp$ when $\mathcal{C}_{\mathsf{wsyn}}$ commits to 0^λ (by definition Π_{wsyn} is not secure against MiM adversary that can commit to $\perp$ when the commitment on the left is honestly generated). Actually the probability that $\mathcal{A}_{\mathsf{NMCom}}$ commits to $\perp$ is negligible. This is because $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_2^0}(z) \approx \mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_1^0}(z)$ and because of Claim 8. The proof ends with the observation that if $\mathcal{C}_{\mathsf{wsyn}}$ commits to m, $\mathcal{A}_{\mathsf{NMCom}}$ acts as in $\mathcal{H}_2^m$, otherwise he acts as in $\mathcal{H}_2^0$. □

Now we can conclude the security proof of Lemma 8 by observing that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ the following holds:

$$\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}},m}_{\Pi_{\mathsf{NMCom}}}(z) = \mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_1^m}(z) \approx \mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_2^m}(z) \approx$$
$$\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_2^0}(z) \approx \mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_1^0}(z) = \mathsf{sim}^{\mathsf{Sim}_{\mathsf{NMCom}}}_{\Pi_{\mathsf{NMCom}}}(1^\lambda, z).$$

□

We conclude the proof of Theorem 4 by proving the following lemma.

Lemma 9. Π_{NMCom} *is a one-one NM commitment scheme.*

Proof. The proofs starts with the observation that the only non-trivial adversary using a non-synchronizing scheduling[20] is the sequential scheduling where $\mathcal{A}_{\mathsf{NMCom}}$ lets the left interaction complete before beginning the right. Considering this scheduling we now prove again that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ it holds that

$$\{\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}},m}_{\Pi_{\mathsf{NMCom}}}(z)\}_{z\in\{0,1\}^\star} \approx \{\mathsf{sim}^{\mathsf{Sim}_{\mathsf{NMCom}}}_{\Pi_{\mathsf{NMCom}}}(1^\lambda, z)\}_{z\in\{0,1\}^\star}.$$

We prove the indistinguishability through a sequence of two hybrid experiments. The first hybrid experiment that we consider is $\mathcal{H}_1^m(z)$, that corresponds to the $\mathcal{H}_1^m(z)$ showed in the proof of Lemma 8 with the only difference that $\mathcal{A}_{\mathsf{NMCom}}$ acts in a non-synchronized way. Therefore Claim 7 holds also in this case.

The second hybrid that we consider is $\mathcal{H}_1^0(z)$. The only differences between this hybrid and the previous one is that NMSen commits to the message 0^λ instead of m. We observe that Claim 8 holds also in this case.

[20] As discussed earlier, an adverary using a trivial non-syncrhonizing scheduling can be simulated by an adversary using a syncrhonizing scheduling. Therefore the security proof for the synchronizing case applies.

Claim 12. *For all* $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ *it holds that* $\{\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_1^m}(z)\}_{z\in\{0,1\}^\star} \approx \{\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}_1^0}(z)\}_{z\in\{0,1\}^\star}$.

Proof. Suppose by contradiction that there exist an adversary $\mathcal{A}_{\mathsf{NMCom}}$ and a distinguisher $\mathcal{D}_{\mathsf{NMCom}}$ that can tell apart such two distributions. We can construct an adversary $\mathcal{A}_{\mathsf{Hiding}}$ that breaks the hiding of Π_{NMCom} (recall the hiding property of Π_{NMCom} comes from Lemma 5). Let $\mathcal{C}_{\mathsf{Hiding}}$ be the challenger of the hiding game and let $(m, 0^\lambda)$ be two challenge messages of the hiding game sent by $\mathcal{A}_{\mathsf{Hiding}}$. The high-level idea of this proof is that $\mathcal{A}_{\mathsf{Hiding}}$ can break the hiding of Π_{NMCom} using the witness extracted from the LS transcript computed by $\mathcal{A}_{\mathsf{NMCom}}$ in the right session. In more details if the witness extract from the LS transcript corresponds to the message committed by $\mathcal{A}_{\mathsf{NMCom}}$ then $\mathcal{A}_{\mathsf{Hiding}}$ can win the hiding game by running $\mathcal{D}_{\mathsf{NMCom}}$. We observe that Claim 7 and Claim 8 ensure that with non-negligible probability the witness extracted from LS in $\mathcal{H}_1^m$ and also in $\mathcal{H}_1^0$ is the committed message $\tilde{m}$.

Before describing the adversary we need to consider the augmented machine $\mathcal{M}_{\mathsf{Hiding}}$ that will be used by $\mathcal{A}_{\mathsf{Hiding}}$ to extract the witness from LS by using the extractor (that exists from the property of adaptive-input PoK enjoyed by LS). Recall that the extractor takes as input a randomness r for the prover and a randomness r' of $\mathcal{V}_{r'}$ in an execution of LS where x has been proved by $\mathcal{P}_r^\star$. Therefore $\mathcal{A}_{\mathsf{Hiding}}$ runs $\mathcal{A}_{\mathsf{NMCom}}$ and interacts in the left session acting as a proxy between $\mathcal{C}_{\mathsf{Hiding}}$ and $\mathcal{A}_{\mathsf{NMCom}}$ in order to obtain the transcript $\tau_{\mathsf{NMCom}} = (\mathsf{a}_{\mathsf{NMCom}}, \mathsf{c}_{\mathsf{NMCom}}, \mathsf{z}_{\mathsf{NMCom}})$ of Π_{NMCom}. In the right session $\mathcal{A}_{\mathsf{Hiding}}$ acts as NMRec with $\mathcal{A}_{\mathsf{NMCom}}$.

Then $\mathcal{A}_{\mathsf{Hiding}}$ uses $\mathcal{M}_{\mathsf{Hiding}}$ to extract the witness of the LS transcript. The augmented machine $\mathcal{M}_{\mathsf{Hiding}}$ runs $\mathcal{A}_{\mathsf{NMCom}}$ acting in the left session with $\mathcal{A}_{\mathsf{NMCom}}$ as the sender NMSen using the messages $\mathsf{a}_{\mathsf{NMCom}}, \mathsf{z}_{\mathsf{NMCom}}$ of τ_{NMCom}. In the right session $\mathcal{M}_{\mathsf{Hiding}}$ interacts with $\mathcal{A}_{\mathsf{NMCom}}$ as the receiver NMRec with the only difference that all the messages of LS received by $\mathcal{A}_{\mathsf{NMCom}}$ are forwarded to the verifier $\mathcal{V}$ and vice versa. Now we describe the augmented machine $\mathcal{M}_{\mathsf{Hiding}}$.

$\boldsymbol{\mathcal{M}_{\mathsf{Hiding}}(\tau_{\mathsf{NMCom}}, r, z)}$.

Let r be the randomness used for all next steps.

- Send $\mathsf{a}_{\mathsf{NMCom}}$ to $\mathcal{A}_{\mathsf{NMCom}}$.
- Upon receiving $\mathsf{c}_{\mathsf{NMCom}}$ from $\mathcal{A}_{\mathsf{NMCom}}$, send $\mathsf{z}_{\mathsf{NMCom}}$ to $\mathcal{A}_{\mathsf{NMCom}}$.
- Upon receiving $(\tilde{\mathsf{a}}_{\mathsf{wsyn}}, \tilde{\mathsf{a}}_{\mathsf{LS}})$ from $\mathcal{A}_{\mathsf{NMCom}}$, send $\tilde{\mathsf{a}}_{\mathsf{LS}}$ to $\mathcal{V}$.
- Upon receiving $\tilde{\mathsf{c}}_{\mathsf{LS}}$ from $\mathcal{V}$, run as follows.
  1. Run $\mathsf{Rec}_{\mathsf{wsyn}}$ on input $\tilde{\mathtt{id}}$ and $\tilde{\mathsf{a}}_{\mathsf{wsyn}}$ thus obtaining the second round $\tilde{\mathsf{c}}_{\mathsf{wsyn}}$ of Π_{wsyn}.
  2. Pick a random $\tilde{Y}$.
  3. Send $(\tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{LS}}, \tilde{Y})$ to $\mathcal{A}_{\mathsf{NMCom}}$.
- Upon receiving $(\tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{LS}})$, set $\tilde{x} = (\tilde{\mathsf{a}}_{\mathsf{wsyn}}, \tilde{\mathsf{c}}_{\mathsf{wsyn}}, \tilde{\mathsf{z}}_{\mathsf{wsyn}}, \tilde{Y}, \tilde{\mathtt{id}})$ and send $(\tilde{\mathsf{z}}_{\mathsf{LS}}, \tilde{x})$ to $\mathcal{V}$.

Now we can conclude the proof of this claim by describing how $\mathcal{A}_{\mathsf{Hiding}}$ works. $\mathcal{A}_{\mathsf{Hiding}}$ runs the extractor of LS (on input the randomnesses r and r') with oracle access to $\mathcal{M}_{\mathsf{Hiding}}$ (recall that an extractor of LS plays having oracle access to an adversarial prover of LS). We know from Claim 7 and from Claim 8 that with overwhelming probability the witness extracted from LS in $\mathcal{H}_1^m$ and in $\mathcal{H}_1^0$ is the committed message $\tilde{m}$. Therefore, $\mathcal{A}_{\mathsf{Hiding}}$ runs the distinguisher $\mathcal{D}_{\mathsf{NMCom}}$ on input $\tilde{m}$ and the view of $\mathcal{A}_{\mathsf{NMCom}}$, and outputs what $\mathcal{D}_{\mathsf{NMCom}}$ outputs. The proof ends with the observation that if $\mathcal{C}_{\mathsf{Hiding}}$ commits to m $\mathcal{A}_{\mathsf{NMCom}}$ acts as in $\mathcal{H}_1^m(z)$, otherwise he acts as in $\mathcal{H}_1^0(z)$. $\square$

Now, observe that the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}^m_1}(z)$ corresponds to the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}},m}_{\Pi_{\mathsf{NMCom}}}(z)$ and that the distribution of $\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}^0_1}(z)$ corresponds to the distribution of $\mathsf{sim}^{\mathsf{Sim}_{\mathsf{NMCom}}}_{\Pi_{\mathsf{NMCom}}}(1^\lambda, z)$. With this observation we have proved that for all $m \in \{0,1\}^{\mathsf{poly}(\lambda)}$ the following relation holds:

$$\mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}},m}_{\Pi_{\mathsf{NMCom}}}(z) = \mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}^m_1}(z) \approx \mathsf{mim}^{\mathcal{A}_{\mathsf{NMCom}}}_{\mathcal{H}^0_1}(z) = \mathsf{sim}^{\mathsf{Sim}_{\mathsf{NMCom}}}_{\Pi_{\mathsf{NMCom}}}(1^\lambda, z).$$

□

The proof of Theorem 4 follows from Lemma 5, Lemma 8 and Lemma 9.