Constant-Round Concurrent Non-Malleable Commitments and Decommitments

Rafail Ostrovsky* Giuseppe Persiano† Ivan Visconti‡

Abstract

In this paper we consider commitment schemes that are secure against *concurrent* poly-time man-in-the-middle (cMiM) attacks. Under such attacks, two possible notions of security for commitment schemes have been proposed in the literature: concurrent non-malleability with respect to *commitment* and concurrent non-malleability with respect to *decommitment* (i.e., opening).

After the original notion of non-malleability introduced by [Dolev, Dwork and Naor STOC 91] that is based on the independence of the committed and decommitted message, a new and stronger notion of non-malleability has been given in [Pass and Rosen STOC 05] by requiring that for any man-in-the-middle adversary there is a stand-alone adversary that succeeds with the same probability.

Under this stronger security notion, a constant-round commitment scheme that is concurrent non-malleable *only* with respect to commitment has been given in [Pass and Rosen FOCS 05] for the plain model, thus leaving as an open problem the construction of a constant-round concurrent non-malleable commitments with respect to decommitment. In other words, in [Pass and Rosen FOCS 05] security against adversaries that mount concurrent man-in-the-middle attacks is guaranteed only during the commitment phase (under their stronger notion of non-malleability).

The main result of this paper is a commitment scheme that is concurrent non-malleable with respect to both commitment and *decommitment*, under the stronger notion of [Pass and Rosen STOC 05]. This property protects against cMiM attacks mounted during both commitments and decommitments which is a crucial security requirement in several applications, as in some digital auctions, in which players have to perform both commitments and decommitments. Our scheme uses a constant number of rounds of interaction in the plain model and is the first scheme that enjoys all these properties under the definitions of [Pass and Rosen FOCS 05].

We stress that, exactly as in [Pass and Rosen FOCS 05], we assume that commitments and decommitments are performed in two distinct phases that do not overlap in time.

Keywords: commitments, definitions, non-malleability, concurrency.

The work of the first author has been supported in part by Intel equipment grant, NSF Cybertrust grant No. 0430254, Xerox Innovation group Award and IBM Faculty Award. The work of the last two authors has been supported in part by the European Commission through the FP6 program under contracts FP6-1596 AEOLUS and IST-2002-507932 ECRYPT. The work of the last author was done in part while he was visiting IPAM.

*UCLA, Los Angeles, CA 90095, USA. rafail@cs.ucla.edu.

†Dip. di Informatica ed App., Università di Salerno, 84084 Fisciano (SA), Italy. giuper@dia.unisa.it.

‡Dip. di Informatica ed App., Università di Salerno, 84084 Fisciano (SA), Italy. visconti@dia.unisa.it.

1 Introduction

Commitment is a fundamental two-party protocol and constitutes the digital equivalent of a safe. One party, called the *committer*, commits to a value without disclosing it to another party called the *receiver*. This property is called *hiding*. The value can be later revealed by the committer and the receiver is guaranteed that the revealed value is the one that the committer originally committed to. This property is called *binding*. Commitments have been used in the design of more complex cryptographic protocols since the early 80's (e.g., for coin flipping [Blu82] and for zero-knowledge for NP [GMW86]).

This basic setting can be extended to several different scenarios that need stronger notions of commitment schemes. In some application scenarios, one wants to be able to guarantee that an adversary $\mathcal{A}$, playing as a receiver in an execution in which a honest committer commits to message m, is not able to commit to a related value $\tilde{m}$ to a honest receiver in another execution in which $\mathcal{A}$ plays as a committer. It is easy to observe that the hiding property does not guarantee this extra property. This type of adversary is called a *man-in-the-middle* adversary (as the adversary plays in between two honest players). Commitment schemes secure with respect to these attacks are called *non-malleable* commitments.

Two notions of non-malleable commitments have been considered in the literature. A commitment scheme that is *non-malleable with respect to commitment* (in short NMc) guarantees that no polynomial-time man-in-the-middle adversary $\mathcal{A}$ can *commit* to a message $\tilde{m}$ that is related to the message m committed by the honest committer. Instead, a commitment scheme that is *non-malleable with respect to decommitment* (also known as non-malleable with respect to opening), (in short NMd) guarantees that after the commitment phase, no polynomial-time man-in-the-middle adversary $\mathcal{A}$, observing the decommitment to m of the honest committer, can *decommit* its commitment to a message $\tilde{m}$ that is related to m.

The need to design non-malleable cryptographic primitives has been first pointed out in the seminal paper by Dolev, Dwork and Naor [DDN91] who also gave constructions for non-malleable encryption, non-malleable zero-knowledge proofs and non-malleable commitments. The constructions for non-malleable commitments of [DDN91] required $O(\log k)$ rounds, where k is the security parameter. Subsequently, assuming the existence of some trusted parameters, non-malleable commitments were constructed in [CIO98, DKOS01, FF00, DG03].

The first constant-round non-malleable commitment scheme in the standard model (i.e., without setup assumptions) has been given by Barak [Bar02] under the assumption of the existence of trapdoor permutations and hash functions that are collision resistant against sub-exponential-time adversaries.

Pass and Rosen studied in [PR05b] several issues about non-malleability, we now consider two of those contributions. First of all they introduced a stronger notion of non-malleability where it is required that the success probability of a man-in-the-middle adversary is maintained by a stand-alone adversary. Therefore, in this definition the sole requirement that the messages decommitted by the man-in-the-middle have to be independent of the ones of the honest committer is not enough. We will show that this is a critical difference that makes non-malleability with respect to decommitment much more challenging to achieve. More concretely, a man-in-the-middle could correctly decommit or could fail the decommitment of a message depending on the message decommitted by the honest sender. This attack can be applied according to the [DDN91] definition while it fails when the [PR05b] definition is used instead.

The second main contribution of [PR05b] is that Pass and Rosen reduced the assumption for constructing non-malleable commitment schemes to the existence of hash functions that are collision resistant against polynomial-time adversaries. They gave two different schemes: one that is NMc and one that is NMd.

More recently, Pass and Rosen [PR05a] have considered *concurrent* man-in-the-middle attacks (cMiM attacks). In such an attack, the adversary can be active in any polynomial number of executions as a receiver and as a committer. A commitment scheme that is secure against cMiM attacks is called *concurrent non-malleable*. As before, we can have two notions of concurrent non-malleable commitment schemes: concurrent NMc and NMd commitment schemes. Pass and Rosen in [PR05a] showed that the constant-round scheme that is NMc of [PR05b] is actually *concurrent* NMc. This implies that security is guaranteed if the commitments are concurrently executed but decommitments are not.

Their paper leaves as an open problem the construction of constant-round commitment schemes that are concurrent NMc, under their stronger definition (instead, their scheme enjoys the weaker notion of non-malleability with respect to decommitment that only focuses on the indipendence of the opened messages). We stress that the concurrent non-malleability of the scheme of [PR05a] relies on the assumption that commitments and decommitments do not overlap in time. We retain this assumption in our schemes. This assumption is motivated by the fact that several important applications have such separation. (e.g., electronic auctions where first all parties send their hidden bids, and only in a second phase they decommit their bids).

We also remark that the recent work of Barak et al. [BPS06] obtains concurrent non-malleable zero-knowledge with a poly-logarithmic round complexity, and thus can not be used for achieving constant-round non-malleable commitments.

Our results. Our main result consists in the construction of a constant-round commitment scheme that is concurrent NMc and NMd under the stronger definition of [PR05b, PR05a]. This implies that security is preserved when polynomially many commitment phases are concurrently executed and when polynomial many decommitment phases are concurrently executed. This solves a problem left open by the results of [PR05a] and allows one to securely run some commitment-based applications (e.g., some digital auctions) by only requiring a constant number of rounds. We follow [PR05a] in that concurrent non-malleability is guaranteed only if commitments and decommitments do not overlap in time.

Our scheme builds and extends multiple techniques. In particular, our scheme uses the perfect NMZK argument of knowledge of [PR05b, PR05a, PR06] but in a critically different manner. Indeed, whereas in [PR05b, PR05a] the statistical NMZK argument of knowledge is simply combined with a (potentially malleable) commitment scheme and a signature scheme, to achieve security in a concurrent setting we also employ a technique by Feige [Fei90] and a more sophisticated rewind technique. Furthermore, the simulator used by [PR05b, PR05a] works in a straight-line fashion including non-black-box techniques. Our result, instead, combines the straight-line simulation with a new rewinding simulation that still avoids the well known problems of using rewinds in concurrent settings [DNS98]. Our approach also includes and extends some of the techniques developed for building concurrent NMZK in the bare public-key model [OPV06].

Finally we stress that in [PR05b] non-malleability with respect to commitment is considered only with respect to statistically binding commitments. Here we show that it is possible

to have non-malleable commitments with respect to commitments that are not statistically binding. This is crucially used in our main result since the constant-round NMc and NMd commitment scheme that we show is not statistically binding.

2 Non-Malleable Commitments

We review the needed background in Appendix A. Here we start by considering concurrent non-malleable commitment schemes; that is, commitments schemes that are secure under *Concurrent Man-in-the-Middle* attacks (cMiM attacks). Informally speaking, a non-malleable commitment scheme guarantees that the value committed to (or the value that is decommitted) by a polynomial-time adversary $\mathcal{A}$ is independent of the value simultaneously committed (or the value that is decommitted) to $\mathcal{A}$ by a honest committer. We assume that $\mathcal{A}$ has full power over the scheduling of the messages in the two sessions (the one in which $\mathcal{A}$ is a committer and the one in which $\mathcal{A}$ is a receiver). Following [PR05b, PR05a], we formalize this notion by comparing two executions: the *man-in-the-middle* execution (the MiM execution) and the *simulated* execution. We denote the security parameter by k and consider the concurrent case where the adversary $\mathcal{A}$ receives and send a polynomial number of commitments.

Discussion. To define NMc we will use the concept of "message committed to by an adversary $\mathcal{A}$ during the commitment phase." By this we mean the following. We will consider commitment schemes in which, for all adversaries $\mathcal{A}$, and for each possible transcript `trans` of the interaction between adversary $\mathcal{A}$ and a honest receiver R such that R accepts the commitment, there exists (statistically) only one message m that is consistent with `trans`; that is, for which there exist random coin tosses that give `trans`. We stress that statistically hiding commitment schemes do not have the above property and thus our definition is not suitable for these commitment schemes. On the other hand, this does not mean that the notion of non-malleability with respect to commitment that we are to define makes sense only for statistically binding commitments (this restriction was instead considered in [PR05b]). Indeed, as we shall see, it is possible to construct NMc commitment schemes that are not statistically binding. This is done by observing that in some commitment schemes while an unbounded adversarial sender could violate the binding property (since the scheme is not statistically binding), when the adversarial (polynomial time) man-in-the-middle successfully plays the commitment phase there always is a well defined and unique message that he can decommit. Therefore such a message can be used to define non-malleability with respect to commitment without having statistical binding.

Informally speaking, non-malleability with respect to commitment guarantees that the commitment computed by the MiM adversary corresponds to a message that is independent of the one committed by the honest committer. Then, decommitting the commitment should not harm and thus non-malleability with respect to decommitment should hold as well. The definition of non-malleability with respect to commitment from [DDN91] essentially implies non-malleability with respect to decommitment [FF00]. In [DDN91], dependency of the values m and $\tilde{m}$ has been formalized through the existence of a poly-time computable relation $\mathcal{R}$ for which $\mathcal{R}(m, \tilde{m}) = 1$. Non-malleability with respect to commitment requires that for any man-in-the-middle adversary $\mathcal{A}$ and any polynomial time computable relation $\mathcal{R}$, there exists a poly-time stand-alone adversary S whose success probability in committing to a value $\tilde{m}$

so that $\mathcal{R}(m, \tilde{m}) = 1$ is at least as good as $\mathcal{A}$'s success probability. Non-malleability with respect to decommitment instead considers the ability of $\mathcal{A}$ to decommit to a value $\tilde{m}$ that is related to m. Notice that under the [DDN91] definition, if $\mathcal{A}$ is no more likely to commit to a related value than S and the commitment is statistically binding, then A is also no more likely to decommit to a related value. This is true regardless of whether $\mathcal{A}$ is given the decommitment information or not. So under this definition, any commitment that is NMc is also NMd.

A stronger notion of non-malleable commitments. The more recent and stronger definition of Pass and Rosen given in [PR05b, PR05a], which we adopt for our schemes requires that the value $\tilde{m}$ committed to by S in the stand-alone execution is computationally indistinguishable from the value committed to by $\mathcal{A}$ in the man-in-the-middle execution. To make non-malleability with respect to decommitment possible, since $\mathcal{A}$ is assumed to obtain the decommitment (say m) of the sender before decommitting its commitment, S is assumed to obtain m before decommitting its commitment. In particular, it is not clear that when using this definition non-malleability with respect to commitment implies non-malleability with respect to decommitment. The problem here is that (in contrast to the [DDN91] definition), one would like the success probability of S (i.e., the probability that the stand-alone simulator playing with a honest receiver correctly completes the decommitment phase) to be only negligibly far from $\mathcal{A}$'s success probability. Indeed, in the schemes of [PR05b, PR05a], the simulator S generates a bogus commitment that is being fed to $\mathcal{A}$. However, after having committed to some value, S is stuck with the bogus value and it is not clear how to enable S to decommit it to $\mathcal{A}$ as m.

From the above discussion we have that the constant-round commitment scheme $\mathcal{NMC}$ (see Figure 2) of [PR05b] that is proved to be NMc, does not seem to be NMd (according to the stronger [PR05b] definition), at least no evidence of that is given in the proof of [PR05b]. Specifically, the simulator that computes $c = \mathsf{SBCom}(0^k, s)$ in the commitment phase cannot open c as m since the decommitment phase simply consists in the decommitment phase of SBCom which is *statistically* binding. Therefore the proof that $\mathcal{NMC}$ is an NMc commitment scheme does not seem to extend to prove that $\mathcal{NMC}$ is also NMd. We stress that in [PR05b, PR05a], only the commitment phase is considered for proving NMc, and since the decommitment phase as discussed above is quite problematic, their security proof implicitly requires that the commitment and decommitment phases do not overlap in time.

Next we remark on the possibility of obtaining NMc commitments that are not statistically binding. When statistically binding commitments are considered, the commitment phase encodes the unique message to which the commitment can be later decommitted. Indeed, even in case the adversarial committer is unbounded there is no way for him to violate the binding property. Since NMc considers the message committed in the commitment phase, the statistical binding property guarantees that this non-malleability notion is well defined, and indeed in [PR05b] the authors consider the notion of NMc only for statistically binding commitment schemes. Intuitively, this claim is correct since in case the scheme is not statistically binding, then the commitment phase does not specify yet and uniquely the message that is going to be decommitted. Therefore, it would be unclear the meaning of NMc as an unbounded adversarial committer could decommit a commitment to different messages. However, we observe that NMc commitments are secure against polynomial-time MiM adversaries for which the binding property still holds. It is therefore possible to have a commitment

scheme that is not statistically binding (i.e., binding does necessarily hold in case the adversarial committer is unbounded) but however is NMc as at the end of the commitment phase it is always possible to determine the message committed by the bounded (i.e., polynomial time) MiM and by the honest sender. Indeed we show commitment schemes that are not statistically binding but that are NMc commitment schemes and, at the same time, NMd.

In Appendix B, we review two recent schemes (due to [PR05b]) of non-malleable commitment: $\mathcal{NMC}c$ that is NMc and $\mathcal{NMC}d$ that is NMd. We show a commitment scheme that combines $\mathcal{NMC}c$ and $\mathcal{NMC}d$ in Appendix C. In the main body of this paper we will concentrate on our main result: a constant-round concurrent NMc and NMd commitment scheme.

3 Concurrent Non-Malleable Commitments

Following [PR05b, PR05a], we now formalize the concept of a *concurrent* non-malleable commitment scheme by comparing two executions: the *concurrent man-in-the-middle* execution (the cMiM execution) and the *simulated* execution. We denote the security parameter by k.

The cMiM execution. In the cMiM execution, the cMiM adversary $\mathcal{A}$ is simultaneously participating in $\mathsf{poly}(k)$ left and $\mathsf{poly}(k)$ right interactions.

Consider a cMiM execution in which the cMiM adversary $\mathcal{A}$ with auxiliary information z interacts in the i-th left interaction with a honest committer running on input a message m_i of length $\mathsf{poly}(k)$ and in the right interactions $\mathcal{A}$ interacts with honest receivers. We denote by $\mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(M, z)$, where $M = (m_1, \ldots, m_{\mathsf{poly}(k)})$, the random variable that associates to the cMiM execution a vector $\tilde{M}$ whose i-th component $\tilde{m}_i$ is defined as follows. If the commitment phase of the i-th right interaction terminates successfully and its transcript is different from the commitment phase of all the left interactions, then $\tilde{m}_i$ is the message that $\mathcal{A}$ has *committed to* in the i-th right interaction. Otherwise, $\tilde{m}_i = \perp$.

Similarly, we denote by $\mathsf{cmim}^{\mathcal{A}}_{\mathsf{Dec}}(M, z)$ the vector $\tilde{M}$ whose i-th component $\tilde{m}_i$ is the message that $\mathcal{A}$ has *decommitted* in the right interaction. If the i-th right interaction is not successful or its transcript (including commitment and decommitment phase) is identical to the transcript of one of the left interactions then $\tilde{m}_i = \perp$.

The simulated execution. In the simulated execution we have one party S (called the *simulator*) that interacts with $\mathsf{poly}(k)$ honest receivers. S works in two phases: in the commitment phase S receives security parameter 1^k and auxiliary information z and interacts with the honest receivers. We denote by $\mathsf{csis}^{S}_{\mathsf{Com}}(1^k, z)$ the vector $\tilde{M}$ whose i-th component $\tilde{m}_i$ is the value committed to by S if the i-th commitment phase has been successfully completed. Otherwise $\tilde{m}_i$ is set equal to $\perp$.

Once the commitment phases have been completed, S receives input vector M and interacts with the honest receiver to complete the decommitment phase. We denote by $\mathsf{csis}^{S}_{\mathsf{Dec}}(M, z)$ the vector $\tilde{M}$ whose i-th component $\tilde{m}_i$ is the value decommitted by S in the i-th decommitment phase if it has been successfully completed. Otherwise $\tilde{m}_i$ is set equal to $\perp$.

We have the following definitions (see also [PR05b, PR05a]).

Definition 3.1 *A commitment scheme is* concurrent non-malleable with respect to commitment *(*a concurrent NMc commitment scheme*) if, for every probabilistic polynomial-time*

cMiM adversary $\mathcal{A}$, there exists a probabilistic polynomial time simulator S such that following ensembles are computationally indistinguishable:

$$\{\mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(M,z)\}_{M\in(\{0,1\}^{\mathsf{poly}(k)})^{\mathsf{poly}(k)},z\in\{0,1\}^\star} \text{ and } \{\mathsf{csis}^{S}_{\mathsf{Com}}(1^k,z)\}_{z\in\{0,1\}^\star}.$$

Definition 3.2 *A commitment scheme is* concurrent non-malleable with respect to decommitment *(*a concurrent NMd commitment scheme*) if, for every probabilistic polynomial-time cMiM adversary $\mathcal{A}$, there exists a probabilistic polynomial time simulator S such that the following ensembles are computationally indistinguishable:*

$$\{\mathsf{cmim}^{\mathcal{A}}_{\mathsf{Dec}}(M,z)\}_{M\in(\{0,1\}^{\mathsf{poly}(k)})^{\mathsf{poly}(k)},z\in\{0,1\}^\star} \text{ and } \{\mathsf{csis}^{S}_{\mathsf{Dec}}(M,z)\}_{M\in(\{0,1\}^{\mathsf{poly}(k)})^{\mathsf{poly}(k)},z\in\{0,1\}^\star}.$$

3.1 Commitment Scheme $c\mathcal{NM}cd$

In this section we present a constant-round commitment scheme $c\mathcal{NM}cd$ that enjoys both concurrent NMc and concurrent NMd. We stress that this is the first protocol that is concurrent NMd in the plain model and requires only a constant number of rounds. Therefore this protocol gives the best security guarantees for important applications as electronic auctions.

We will use a constant-round tag-based perfect NMZK argument of knowledge $\mathsf{nmZK} = \{\mathcal{P}_t, \mathcal{V}_t\}_t$ for all $\mathtt{NP}$ [PR05b], a constant-round witness indistinguishable $(\mathsf{wi}\mathcal{P}, \mathsf{wi}\mathcal{V})$ proof of knowledge (WIPoK) for all $\mathtt{NP}$ [Blu86, FS90], a non-interactive statistically binding commitment scheme Com and a secure signature scheme $SS = (\mathtt{SG}, \mathtt{Sig}, \mathtt{SVer})$. See Appendix A.3 and A.4 for details about these tools. A description of commitment scheme $c\mathcal{NM}cd$ is found in Figure 1.

How we achieve concurrent NMd. First of all, we notice that a straight-forward combination of the two commitment schemes of [PR05b] achieves non-malleability with respect to both commitment and decommitment, when concurrency is not considered. This scheme is shown in Appendix C and is called $\mathcal{NM}cd$. Let us briefly discuss why commitment scheme $\mathcal{NM}cd$ is not a *concurrent* NMd commitment scheme. In proving the NMd property we crucially relied on the existence of a simulator extractor for the NMZK argument nmZK. If we try to argue that $\mathcal{NM}cd$ is a concurrent NMd commitment scheme along the same lines, we would need a simulator that simulates concurrent executions; in other words, we would need a concurrent NMZK argument of knowledge. Unfortunately, the existence of a constant-round concurrent NMZK argument system in the plain model is still an open problem.

We use instead a more sophisticated protocol and prove its properties by blending the straight-line simulator of the concurrent NMc commitment scheme of [PR05a] with a sophisticated rewind technique. In using rewinding we have to be careful as the nested sessions can potentially make the running time super polynomial[1]. Instead, we perform rewinds "in advance," to extract information from the adversary. The simulator is then able to simulate in a straight-line fashion the decommitment phase by using the information extracted by means of rewinds. Our security proof also employs the two-witness technique by [Fei90] and the well known FLS-technique [FLS99].

More in details, we extend the commitment phase of the concurrent non-malleable commitment scheme of [PR05a] by requiring that the receiver gives a proof of knowledge of a secret. The decommitment phase consists in sending a message and in proving with a NMZK

[1]The study of this problem started with the notion of concurrent zero knowledge [DNS98].

Security Parameter: 1^k.
Input to Committer: $m \in \{0,1\}^k$.
Commitment Phase:

$C \to R$: pick $s \in \{0,1\}^k$, set $c = \mathsf{Com}(m,s)$ and send c to R.

$C \to R$: set $(\mathtt{PK}, \mathtt{SK}) \leftarrow \mathtt{SG}(1^k)$ and send $\mathtt{PK}$ to R.

$C \leftrightarrow R$: C executes the code of $\mathcal{P}_{\mathtt{PK}}$ on input c to prove knowledge of $m, s \in \{0,1\}^k$ such that $c = \mathsf{Com}(m,s)$. R executes the code of $\mathcal{V}_{\mathtt{PK}}$ on input c. If $\mathcal{V}_{\mathtt{PK}}$ rejects then R aborts.

$R \to C$: pick $m_0, s_0, m_1, s_1 \in \{0,1\}^k$, set $c_0 = \mathsf{Com}(m_0, s_0), c_1 = \mathsf{Com}(m_1, s_1)$ and send c_0 and c_1 to C.

$R \leftrightarrow C$: R select a random bit b and executes the code of $\mathsf{wi}\mathcal{P}$ on input (c_0, c_1) to prove knowledge of $\hat{m}, \hat{s} \in \{0,1\}^k$ such that $c_0 = \mathsf{Com}(\hat{m}, \hat{s})$ or $c_1 = \mathsf{Com}(\hat{m}, \hat{s})$ using (m_b, s_b) as witness. C executes the code of $\mathsf{wi}\mathcal{V}$ on input (c_0, c_1). If $\mathsf{wi}\mathcal{V}$ rejects then C aborts.

$C \to R$: let $\mathtt{trans}_0$ be the transcript so far. Set $\sigma_0 \leftarrow \mathtt{Sig}(\mathtt{trans}_0, \mathtt{SK})$ and send σ_0 to R.

R: if $\mathtt{SVer}(\mathtt{trans}_0, \sigma_0, \mathtt{PK}) \neq 1$ abort.

Decommitment Phase:

$C \to R$: send m.

$C \leftrightarrow R$: C executes the code of $\mathcal{P}_{\mathtt{PK}}$ on input (c, c_0, c_1) to prove knowledge of $\hat{m}, \hat{s} \in \{0,1\}^k$ such that $c = \mathsf{Com}(m, \hat{s})$ or $c_0 = \mathsf{Com}(\hat{m}, \hat{s})$ or $c_1 = \mathsf{Com}(\hat{m}, \hat{s})$, using (m, s) as witness. R executes the code of $\mathcal{V}_{\mathtt{PK}}$ on input (c, c_0, c_1). If $\mathcal{V}_{\mathtt{PK}}$ rejects then R aborts.

$C \to R$: let $\mathtt{trans}_1$ be the transcript so far. Set $\sigma_1 \leftarrow \mathtt{Sig}(\mathtt{trans}_1, \mathtt{SK})$ and send σ_1 to R.

R: if $\mathtt{SVer}(\mathtt{trans}_1, \sigma_1, \mathtt{PK}) \neq 1$ abort.

Figure 1: Our concurrent NMc and concurrent NMd commitment scheme $c\mathcal{NMM}cd$.

proof that either the message corresponds to the committed one or the sender knows the secret (this is the FLS-technique [FLS99]). Our simulator will extract the secrets of all receivers in the commitment phase and will use them as fake witnesses in the decommitment phase. Instead an adversary will not be able to use such secret, since we show that any successful adversary can be reduced to break a standard complexity-theoretic assumption by using the two-witness technique of [Fei90] and the non-malleability of nmZK.

In the next section we prove the properties of commitment scheme $c\mathcal{NMM}cd$. We will often use the simulation-extractability property of nmZK. Notice that this property is guaranteed only in case the tag used by the adversary is different from the one used by the other parties. Since in our scheme we use as tag the public key of a signature scheme, and since each phase is only correctly completed if there is a signature under that public key of the transcript of the phase, we assume that the simulation-extractability property always holds, since otherwise the security of the signature scheme is broken. We will detail this argument only when we prove the NMc property for the one-left many-right case (see the discussion below the description of Expt_2), in the other cases the argument is quite similar and is omitted.

Binding. In the proof of concurrent NMd we show that any man-in-the-middle adversary that completes the commitment phase, can later open that commitment only in one way. This property is even stronger than binding (since the classical adversary for the binding property can not play as receiver) thus that proof properly contains the proof of the binding property.

Hiding. Assume by contradiction that there exists an adversarial receiver $\mathcal{A}$ that, after the commitment phase distinguishes a commitment to m_0 from a commitment to m_1 with non-negligible advantage. We show how to reduce $\mathcal{A}$ to an adversary $\mathcal{A}'$ that breaks the hiding property of Com. Indeed, $\mathcal{A}'$ on input a challenge com (i.e., a commitment of either m_0 or m_1), plays the honest committer algorithm with the following two exceptions: com is sent in the commitment phase and the simulator for $\mathsf{nmZK}_{\mathsf{PK}}$ is used instead of the honest prover algorithm. Since the simulation for $\mathsf{nmZK}_{\mathsf{PK}}$ is perfect, the only chance $\mathcal{A}$ has to guess concerns the value of com. Therefore, $\mathcal{A}'$ by simply giving in output the same bit given in output by $\mathcal{A}$ succeeds in guessing with non-negligible advantage the message committed in com.

3.1.1 Concurrent NMc

We start by considering the simpler case in which the adversary $\mathcal{A}$ is active in one left commitment and in polynomially many right commitments (a one-left many-right adversary).

The one-left many-right case. For every one-left many-right MiM adversary $\mathcal{A}$, we consider simulator $S(z)$ that internally runs $\mathcal{A}(z)$ and provides $\mathcal{A}$ with a left commitment by executing the code of the honest committer to commit to 0^k (k is the security parameter). For the right commitments instead S relays messages between the polynomially many honest receivers and $\mathcal{A}$. We stress that for NMc we only have to consider the commitment phase.

We now prove that for all messages $m \in \{0,1\}^k$ and all z $\Big|\text{Prob}[\ D(m, \mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(m,z)) = 1\ ] - \text{Prob}[\ D(m, \mathsf{csis}^{S}_{\mathsf{Com}}(1^k,z)) = 1\ ]\Big|$ is negligible in k for all distinguishers D. We consider hybrid experiments starting with $\mathsf{Expt}_0(v,z)$.

$\mathsf{Expt}_0(v,z)$ is the experiment in which $\mathcal{A}(z)$ interacts in the left commitment with a honest committer committing to v and with honest receivers in the right commitments. We denote by $\tilde{M}$ the vector whose i-th component $\tilde{m}_i$ is defined as follows. If the i-th right commitment is successfully completed by $\mathcal{A}$ and its transcript differs from the one of the left commitment then $\tilde{m}_i$ is the message $\mathcal{A}$ has committed to[2] in the i-th right commitment. Otherwise $\tilde{m}_i = \perp$. $\mathsf{Expt}_0(v,z)$ returns $D(v,\tilde{M})$. We set $p_0(v,z) = \text{Prob}[\ \mathsf{Expt}_0(v,z) = 1\ ]$. Obviously, we have that for all z, k and $m \in \{0,1\}^k$, $p_0(m,z) = \text{Prob}[\ D(m, \mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(m,z)) = 1\ ]$ and that $p_0(0^k,z) = \text{Prob}[\ D(m, \mathsf{csis}^{S}_{\mathsf{Com}}(1^k,z)) = 1\ ]$.

To define the next experiment, we observe that $\mathcal{A}$ naturally defines a one-left many-right MiM adversary $\mathcal{A}'$ for nmZK. Specifically, consider the following adversary $\mathcal{A}'$. $\mathcal{A}'(z)$ internally runs $\mathcal{A}(z)$. $\mathcal{A}'$ forwards externally all $\mathcal{A}$'s messages of all the executions of nmZK. For the execution of $(\mathsf{wi}\mathcal{P}, \mathsf{wi}\mathcal{V})$ of each right commitment (here $\mathcal{A}$ acts as a verifier), $\mathcal{A}'$ computes the commitment of two random messages and executes the code of $\mathsf{wi}\mathcal{P}$. For the

[2] This is the message that is consistent with the transcript. Since we use a statistically binding commitment scheme there is a unique such message.

executions of $(\mathsf{wi}\mathcal{P}, \mathsf{wi}\mathcal{V})$ of the left commitments, $\mathcal{A}'$ executes the code of $\mathsf{wi}\mathcal{V}$. Now let $\mathcal{S}'$ be the simulator-extractor of nmZK for adversary $\mathcal{A}'$.

Experiment $\mathsf{Expt}_1(v, z)$ differs from $\mathsf{Expt}_0(v, z)$ in that we have the simulator $\mathcal{S}'$ for adversary $\mathcal{A}'$ instead of $\mathcal{A}$ that is playing with the honest prover and honest verifiers for nmZK. More precisely, in the left commitment of $\mathsf{Expt}_1(v, z)$, we first compute $\mathsf{com} = \mathsf{Com}(v, s)$ and $(\mathtt{PK}, \mathtt{SK}) = \mathtt{SG}(1^k)$ and then run $\mathcal{S}'$ on input com, tag $\mathtt{PK}$ and z. All other steps (executions of $(\mathsf{wi}\mathcal{P}, \mathsf{wi}\mathcal{V})$ and signatures) are performed just like in $\mathsf{Expt}_0(v, z)$. Let View be the view output by $\mathcal{S}'$ and define vector $\tilde{M}$ as follows. If the i-th right commitment in View is successfully completed and its transcript differs from the one of the left commitment, then set $\tilde{m}_i$ equal to the message committed to (again, this message is unique since Com is statistically binding) by $\mathcal{A}$. Otherwise, set $\tilde{m}_i = \perp$. Finally, $\mathsf{Expt}_1(v, z)$ outputs $D(v, \tilde{M})$. We set $p_1(v, z) = \text{Prob}[\ \mathsf{Expt}_1(v, z) = 1\ ]$. By the perfect NMZK property of nmZK, we have that $p_0(v, z) = p_1(v, z)$ for all v and z.

Experiment $\mathsf{Expt}_2(v, z)$ differs from $\mathsf{Expt}_1(v, z)$ in the way in which vector $\tilde{M}$ (and consequently the output) is computed. Specifically, in $\mathsf{Expt}_2(v, z)$ we set $\tilde{m}_i$ as the message that has been extracted by $\mathcal{S}'$ as part of the witness for the i-th right execution of nmZK. If no message is extracted then $\tilde{m}_i = \perp$. We set $p_2(v, z) = \text{Prob}[\ \mathsf{Expt}_2(v, z) = 1\ ]$.

Denote by $\tilde{\mathtt{PK}}_i$ the signature public key used as a tag for the i-th right execution of nmZK in View and by $\mathtt{PK}$ the signature public key used as a tag for the left execution of nmZK in View. First of all observe that, for all i, if the transcript of the i-th right commitment of View differs from the one of the left commitment then, by the security of the signature scheme, the probability that $\tilde{\mathtt{PK}}_i = \mathtt{PK}$ is negligible. Therefore, for each i, only two cases have non-negligible probability. In the first case the transcript of the i-th right commitment is equal to the one of the left commitment (and thus $\tilde{\mathtt{PK}}_i = \mathtt{PK}$). Then we observe that in this case $\tilde{m}_i = \perp$ both in $\mathsf{Expt}_1(v, z)$ and in $\mathsf{Expt}_2(v, z)$. If instead the transcript of the i-th right commitment differs from the one of the left commitment and $\tilde{\mathtt{PK}}_i \neq \mathtt{PK}$ then, by the extraction properties of $\mathcal{S}'$, the value $\tilde{m}_i$ extracted by $\mathcal{S}'$ is not the value committed to by $\mathcal{A}$ in View with negligible probability. Therefore we conclude that $|p_2(v, z) - p_1(v, z)|$ is negligible for all v and z.

We now conclude the proof by showing that for all k and for all $v \in \{0,1\}^k$, $|p_2(v, z) - p_2(0^k, z)|$ is negligible. Suppose that it is not and thus for infinitely many k there exists $v_k \in \{0,1\}^k$ and z such that $|p_2(v_k, z) - p_2(0^k, z)| \geq 1/\mathsf{poly}(k)$. Then, we can construct the following adversary B that breaks the hiding of Com. B receives $\hat{c}$ that is a commitment to either 0^k or v_k and executes $\mathsf{Expt}_2(v_k, z)$ by setting in the left commitment phase $c = \hat{c}$. We notice that Expt_2 can be executed in polynomial time even though the message committed to by c in the left interaction is not known. From the output of the experiment B has a non-negligible advantage in guessing the committed bit.

We have shown that both $|p_0(v_k, z) - p_2(v^k, z)|$ and $|p_2(v_k, z) - p_2(0^k, z)|$ are negligible, Using again the same arguments, it follows that $|p_2(0^k, z) - p_0(0^k, z)|$ is negligible. Therefore, we have that $\left|\text{Prob}[\ D(m, \mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(m, z)) = 1\ ] - \text{Prob}[\ D(m, \mathsf{csis}^{S}_{\mathsf{Com}}(1^k, z)) = 1\ ]\right| = |p_0(v^k, z) - p_0(0^k, z)|$ is negligible.

The many-left many-right case for concurrent NMc. We now consider the many-left many-right case. For concurrent MiM adversary $\mathcal{A}$, we consider simulator $S(z)$ that runs $\mathcal{A}(z)$ internally and executes the code of the honest committer on input 0^k for all left commitments. For the right interactions, S relays messages between the external receivers

and $\mathcal{A}$. Notice that if we have only one left commitment S coincides with the simulator we used for proving non-malleability with respect to one-left many-right MiM.

Assume by contradiction that there exists a distinguisher D that distinguishes $\mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(M,z)$ and $\mathsf{csis}^{S}_{\mathsf{Com}}(1^k,z)$. Let $l=\mathsf{poly}(k)$ be the number of left commitments and, for $i=0,\ldots,l$, consider hybrid experiment $\mathsf{Expt}^{\mathcal{A}}_i$ defined as follows. Let $M=(m_1,\ldots,m_l)$ be a vector of messages. In $\mathsf{Expt}^{\mathcal{A}}_i(M,z)$, adversary $\mathcal{A}$ is run on input z and the j-th honest left committer commits to m_j if $j\leq i$ and to 0^k otherwise. $\mathsf{Expt}^{\mathcal{A}}_i(M,z)$ outputs a vector whose i-th component consists of the messages committed to by $\mathcal{A}$ in the i-th right commitment if it has been successfully completed by $\mathcal{A}$ and if its transcript differs from the transcripts of all the left commitments. If this is not the case then the i-th component of the output of $\mathsf{Expt}^{\mathcal{A}}_i(M,z)$ is set equal to $\perp$. Obviously, for all M and z, $\mathsf{Expt}^{\mathcal{A}}_0(M,z)$ coincides with $\mathsf{csis}^{S}_{\mathsf{Com}}(1^k,z)$ and $\mathsf{Expt}^{\mathcal{A}}_l(M,z)$ with $\mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(M,z)$. If there exists a probabilistic polynomial time distinguisher D that distinguishes between $\mathsf{csis}^{S}_{\mathsf{Com}}(1^k,z)$ and $\mathsf{cmim}^{\mathcal{A}}_{\mathsf{Com}}(M,z)$ then there must be $i\in\{0,\ldots l-1\}$ such that D distinguishes the output of $\mathsf{Expt}^{\mathcal{A}}_i(M,z)$ and the output of $\mathsf{Expt}^{\mathcal{A}}_{i+1}(M,z)$. We stress that the only difference between experiment $\mathsf{Expt}^{\mathcal{A}}_i$ and experiment $\mathsf{Expt}^{\mathcal{A}}_{i+1}$ is that in the $(i+1)$-st left commitment of $\mathsf{Expt}^{\mathcal{A}}_i$ we are committing to 0^k (just like the simulator) whereas in $\mathsf{Expt}^{\mathcal{A}}_{i+1}$ we are committing to m_{i+1}. We can therefore construct a successful MiM adversary $\mathcal{A}'$ for the one-left many-right case. Adversary $\mathcal{A}'$ internally runs all left sessions with the only exception of the $(i+1)$-st session that is played either with a honest committer committing to m_{i+1} or with the simulator of the one-left many-right case. Therefore $\mathcal{A}'$ breaks the one-left many-right non-malleability which is a contradiction.

3.1.2 Concurrent NMd

For every cMiM adversary $\mathcal{A}$, we describe a simulator S that interacts with polynomially many honest receivers and performs with each of them a commitment and a decommitment phase. To satisfy Definition 3.2, we will show that, for every vector M of messages, S decommits its commitments to a vector $\tilde{M}$ of messages that is indistinguishable from the messages decommitted by $\mathcal{A}$ when interacting on the left with honest committers committing to M.

The simulator. Since now we also have to care about decommitments, we extend the simulator in the following way. S first runs the left and the right commitment phases with $\mathcal{A}$ executing the code of the honest receiver in the right commitment phases and the code of the honest committer on input message 0^k in the left commitment phase. Notice that $\mathcal{A}$ is interacting solely with S and no honest receiver is involved. Then S runs the extractors for all the proofs (both in left and right commitment phases) provided by $\mathcal{A}$ in order to get the corresponding witnesses. More precisely, for each right commitment phase, S runs the extractor of nmZK and we denote by (m_i,s_i) the witness extracted in the i-th right commitment phase; for each left commitment phase, S runs the extractor of the WIPoK and we denote by $(m_{b_i,i},s_{b_i,i})$, with $b_i\in\{0,1\}$, the witness extracted in the i-th left commitment phase. Extractions are executed sequentially and thus the running time of S is polynomial.

Next, S plays the commitment phases with the honest receivers. S does so by executing the code of the honest committer and using, for the i-th commitment phase, message m_i as input.

After the commitment phases have been completed, S receives vector $M^\star=(m_1^\star,\ldots,m_l^\star)$ and has to perform the decommitment phases with $\mathcal{A}$. S does so by resuming the interactions with $\mathcal{A}$ in the following way. In the left decommitment phase corresponding to the i-th left

commitment phase, S uses knowledge of $m_{b_i,i}$ to open the commitment (that was originally computed by S as a commitment to 0^k) to $m_i^\star$. In the right decommitment phases, S acts as a honest receiver. Then, for each i, if $\mathcal{A}$ has successfully completed the i-th right decommitment phase, then S completes the i-th decommitment phase with the honest receiver decommitting the commitment to m_i (notice that in the i-th commitment phase with honest receivers, S had committed to m_i). This ends the description of the simulator S.

The above simulator combines the techniques we propose in this paper to overcome the limitations of the [PR05a] result. Our simulator not only guarantees concurrent NMc as we proved previously, but it will also guarantee concurrent NMd. Notice that the [PR05a] simulator only works for concurrent NMc, while for NMd it immediately fails when a single decommitment phase is executed. We now turn to proving that the described simulator S satisfies Definition 3.2.

We now prove that the distribution of the messages decommitted by $\mathcal{A}$ when interacting with honest committers and honest receivers is indistinguishable from the distribution of the messages decommitted by $\mathcal{A}$ when interacting with S.

Indistinguishability of the simulation. We start with the one-left many-right case and then we will consider the many-left many-right case. We consider a sequence of experiments $\mathsf{Expt}_i^{\mathcal{A}}(m,z)$ and show that any distinguisher D between the experiments can be used to produce a contradiction. Therefore, the output of each experiment is the output of a distinguisher D (which existence is assumed by contradiction) on input a message m and a vector $\tilde{M}$ whose i-th component $\tilde{m}_i$ is defined as follows. If the decommitment phase of the i-th right interaction terminates successfully and its transcript is different from all the left interactions, then $\tilde{m}_i$ is the message that $\mathcal{A}$ has decommitted in the i-th right interaction. Otherwise, $\tilde{m}_i = \perp$. We also set $p_i^{\mathcal{A}}(m,z) = \text{Prob}[\,\mathsf{Expt}_i^{\mathcal{A}}(m,z) = 1\,]$.

$\mathsf{Expt}_0^{\mathcal{A}}(m,z)$ is the experiment in which $\mathcal{A}$ plays with S that behaves as a honest receivers in the right interactions and as a honest committer on input m in the left interaction. We notice that, since S is acting as honest receiver and honest committer, $p_0^{\mathcal{A}}(m,z)$ is the probability that D outputs 1 on input distributed according to $\mathsf{cmim}_{\mathsf{Dec}}^{\mathcal{A}}(m,z)$.

Experiment $\mathsf{Expt}_1^{\mathcal{A}}(m,z)$ differs from Expt_0 only because in the left commitment phase, S runs the extractor of the WIPoK used by $\mathcal{A}$. Since there is no other deviation, we have that $p_1^{\mathcal{A}}(m,z) = p_0^{\mathcal{A}}(m,z)$.

Experiment $\mathsf{Expt}_2^{\mathcal{A}}(m,z)$ differs from Expt_1 in that in the left decommitment phase, S executes the code of the honest prover but uses a fake witness (that is the witness extracted in the left commitment phase from $\mathcal{A}$'s WIPoK). Next we prove that $|p_2^{\mathcal{A}}(m,z) - p_1^{\mathcal{A}}(m,z)|$ is negligible. Assume by contradiction that this difference is non-negligible; as the only difference between the two games consists in the witness used in the nmZK played in the decommitment phase, we show how to break the witness indistinguishability of nmZK. Specifically, we play the following game with an external prover P. We perform the commitment phase like in game $\mathsf{Expt}_1^{\mathcal{A}}(m,z)$. In particular, in the left commitment phase S has computed and sent to $\mathcal{A}$ commitment $c = \mathsf{Com}(m,s)$ and $\mathcal{A}$ has produced commitments c_0 and c_1 and proved knowledge of the message committed to by one of the two. We denote by (m_b, s_b) the witness extracted by S from $\mathcal{A}$'s WIPoK. The decommitment phase proceeds as in game Expt_1 with the exception of the execution of nmZK in the left decommitment phase which is performed by the external prover P. P is fed with the real witness (m,s) and the fake witness (m_b, s_b) and performs the code of the honest prover using one of the two. Notice that the decommit-

ment phase is straight-line. We observe that if P uses the fake witness then we are actually playing game $\mathsf{Expt}_2^{\mathcal{A}}(m,z)$ whereas if P uses the real witness we are playing $\mathsf{Expt}_1^{\mathcal{A}}(m,z)$. Therefore if D distinguishes these two games, we break the witness indistinguishability of nmZK. We stress that in this reduction we have not used the extractor of the nmZK of the decommitment phase, therefore we can relay messages with P without rewinding it.

Next we consider $\mathsf{Expt}_3^{\mathcal{A}}(m,z)$ in which S uses the simulator of nmZK in the left commitment phase. Since the simulation is perfect we have that $p_3^{\mathcal{A}}(m,z) = p_2^{\mathcal{A}}(m,z)$.

Next we consider $\mathsf{Expt}_4^{\mathcal{A}}(m,z)$ in which S commits to 0^k in the left commitment phase. Any distinguisher between $\mathsf{Expt}_4^{\mathcal{A}}(m,z)$ and $\mathsf{Expt}_3^{\mathcal{A}}(m,z)$ can be easily reduced to a distinguisher between a commitment of 0^k and a commitment of m using Com, by simply playing this commitment as c, completing the experiment and then giving in output the same output of the distinguisher. Therefore by the computational hiding of Com we have that $|p_4^{\mathcal{A}}(m,z) - p_3^{\mathcal{A}}(m,z)|$ is negligible.

This sequence of experiments shows that the distribution of the messages decommitted by $\mathcal{A}$ during the man-in-the-middle game when the honest sender commits and decommits to m and $\mathcal{A}$ commits and decommits with the honest receiver R (i.e., $\mathsf{Expt}_0^{\mathcal{A}}(m,z)$), is indistinguishable from the distribution of the messages that $\mathcal{A}$ decommits in the simulated game where S plays both as sender committing to 0^k and as receiver (i.e., $\mathsf{Expt}_4^{\mathcal{A}}(m,z)$).

Epilogue. We now show that S is actually a stand-alone adversary, i.e., it can commit and open to a honest receiver R the same messages that $\mathcal{A}$ can open and decommit during a man-in-the-middle game.

Following the description of S, we know that S commits to R the messages that it extracts from $\mathcal{A}$ at the end of the commitment phase of the simulated game. The proof of non-malleability with respect to commitment given previously, says that the messages committed by S to R have the same distribution of the ones committed by $\mathcal{A}$ in the real game. Then the description of S says that S decommits to R the commitments that correspond to the ones that $\mathcal{A}$ decided to decommit to S in the decommitment phase of the simulated game. Since the indistinguishability of the simulation proved so far says that $\mathcal{A}$ decommits to S the same messages that $\mathcal{A}$ decommits in the real game, we have that S decommits to R the same messages decommitted by $\mathcal{A}$ in the real game, unless $\mathcal{A}$ in the real game decommits messages different with respect to the committed ones (indeed, S never decommits to R a message that is different from the committed one).

Therefore we now show that in the real game $\mathcal{A}$ can not open to different messages, this will imply that S decommits to R messages with the same distribution of the ones decommitted by $\mathcal{A}$.

In the real game $\mathcal{A}$ cannot open in a different way. Assume by contradiction that, with some non-negligible probability, in the real game (i.e., when $\mathcal{A}$ plays with a honest prover committing to m and with honest receivers) there exists i such that the decommitted message m_i' is different from the committed message m_i[3]. We denote by c_0 and c_1 the two commitments computed by the R in the i-th commitment phase of $\mathcal{A}$ and by $b \in \{0,1\}$ the bit such that the receiver R used knowledge of the message committed to by c_b to perform the WIPoK of the i-th commitment phase. Given that $\mathcal{A}$ successfully completes the i-th decommitment phase then, we can consider the following experiment. Adversary $\mathcal{A}$ plays with a real sender

[3]The committed message is the one uniquely specified by the statistically binding commitment scheme used as subprotocol.

and a receiver-extractor. The real sender commits to m, while the receiver-extractor runs the honest receiver algorithm for all right commitments and runs the extractor of nmZK of the i-th decommitment phase. The receiver-extractor with overwhelming probability outputs a pair $(\hat{m}, \hat{s})$ such that either $c_b = \mathsf{Com}(\hat{m}, \hat{s})$ or $c_{1-b} = \mathsf{Com}(\hat{m}, \hat{s})$ (i.e., since $\mathcal{A}$ decommitted to a different message, the witness must be a fake one).

Suppose that with some non-negligible probability it happens that $c_{1-b} = \mathsf{Com}(\hat{m}, \hat{s})$. Then we break the hiding property of Com. Consider the following adversary $\mathcal{B}$ that receives a commitment $\hat{c}$ and would like to compute the message committed to by $\hat{c}$ with some non-negligible probability. $\mathcal{B}$ interacts with $\mathcal{A}$ and plays all commitment phases as the honest senders and receivers, with the only exception of the i-th commitment phase played as receiver. Here $\mathcal{B}$ picks a random $b \in \{0,1\}$, a random $m_b \in \{0,1\}^k$ and random $s_b \in \{0,1\}^k$ and computes commitment $c_b = \mathsf{Com}(m_b, s_b)$ and sets $c_{1-b} = \hat{c}$. Then $\mathcal{B}$ continues the commitment phase by running the code of the honest prover wiP of the WIPoK using (m_b, s_b) as witness. By our hypothesis, with some non-negligible probability, the extractor gives the message committed to by $\hat{c}$, this gives to $\mathcal{B}$ a non-negligible advantage for breaking the hiding property of Com.

Suppose instead that, except with negligible probability, it happens that $c_b = \mathsf{Com}(\hat{m}, \hat{s})$. We show that the witness indistinguishability of the WIPoK is violated. More specifically, we consider a WI adversary $\mathcal{B}$ that executes internally all the previous interactions with the only exception that the WIPoK of the i-th right commitment phase is played by relaying messages with an external prover (that uses a witness for $c_{b^\star}$ for some $b^\star \in \{0,1\}$). $\mathcal{B}$ then plays internally the decommitment phases with the exception of the i-th decommitment phase for which the extractor is used. By looking at the extracted witness, $\mathcal{B}$ will guess the witness used by the external prover.

Summing up. We have therefore shown that $\mathcal{A}$ decommits successfully only the committed messages. Moreover, we have shown that in the simulated game $\mathcal{A}$'s choices for which commitment have to be decommitted are indistinguishable from its choices in the simulated game. These two properties guarantee that S decommits to R messages indistinguishable from the ones decommitted by $\mathcal{A}$ in the real game.

This terminates the proof for the one-left many-right case.

The many-left many-right case for concurrent NMd. Let $l = \mathsf{poly}(k)$ be the size of the vector of messages M, we consider the hybrid games $\{\mathsf{Expt}_i^{\mathcal{A}}\}_{0 \le i \le l}$, where $\mathsf{Expt}_i^{\mathcal{A}}$ for $i = 0, \ldots l$ is defined as follows. In the game $\mathsf{Expt}_i^{\mathcal{A}}$ the committer commits to m_j as the j-th commitments if $j \le i$, and to 0^k if $j > i$. Moreover in $\mathsf{Expt}_i^{\mathcal{A}}$ the i-th commitment is decommitted using a legal witness if $j \le i$ and a fake witness if $j > i$. Obviously $\mathsf{Expt}_0^{\mathcal{A}}$ corresponds to the game played by the simulator (including both the commitment and decommitment phases) while $\mathsf{Expt}_l^{\mathcal{A}}$ corresponds to game played by the honest committer (again, including both the commitment and decommitment phases). For all M and z we denote by $\{\mathsf{csis}_{\mathsf{Dec}}^{\mathsf{Expt}_i^{\mathcal{A}}}(M, z)\}$ the random variable that associates to each successfully completed decommitment phase of $\mathsf{Expt}_i^{\mathcal{A}}$ the messages decommitted by $\mathcal{A}$. Instead $\{\mathsf{csis}_{\mathsf{Dec}}^{\mathsf{Expt}_i^{\mathcal{A}}}(M, z)\}$ associates the value $\perp$ to interactions that have not been completed by $\mathcal{A}$.

Assume by contradiction that the scheme is not concurrent non-malleable with respect to decommitment. It follows that there must be an index $i \in \{0, \ldots l-1\}$ such that D distinguishes with non-negligible probability between $\{\mathsf{csis}_{\mathsf{Dec}}^{\mathsf{Expt}_i^{\mathcal{A}}}(M, z)\}$ and $\{\mathsf{csis}^{\mathsf{Expt}_{i+1}^{\mathcal{A}}\mathsf{Dec}}(M, z)\}$.

The only difference between game $\mathsf{Expt}_i^{\mathcal{A}}$ and game $\mathsf{Expt}_{i+1}^{\mathcal{A}}$ for $i \in \{0, \ldots l-1\}$ is that the $i+1$ commitment is computed for message 0^k in $\mathsf{Expt}_i^{\mathcal{A}}$ while it is computed for message m_i in $\mathsf{Expt}_{i+1}^{\mathcal{A}}$. Moreover the corresponding decommitment uses a fake witness in $\mathsf{Expt}_i^{\mathcal{A}}$ and a legal witness in $\mathsf{Expt}_{i+1}^{\mathcal{A}}$.

We can therefore construct a successful MiM adversary $\mathcal{A}'$ for the one-left many-right case. Adversary $\mathcal{A}'$ internally runs all left sessions with the only exception of the $(i+1)$-st commitment and the corresponding decommitment that is played either with a honest committer committing to m_{i+1} or with the simulator of the one-left many-right case. Therefore $\mathcal{A}'$ breaks the one-left many-right non-malleability which is a contradiction.

From the previous discussion and by observing that existence of a family of claw-free permutations is sufficient for the tools we use, we have the following theorem and corollary.

Theorem 3.3 *Under the assumption of existence of a tag-based one-left many-right perfect cNMZK arguments of knowledge for all* NP*, of a secure signature scheme and of a secure non-interactive commitment scheme, commitment scheme $\mathcal{NM}cd$ is is both concurrent NMc and concurrent NMd.*

Corollary 3.4 *Under the existence of a family of claw-free permutations there exists a constant-round commitment scheme that is both concurrent NMc and concurrent NMd.*

References

[Bar02] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd Annual Symposium on Foundations of Computer Science*, pages 345–355, Vancouver, British Columbia, Canada, November 16–19, 2002. IEEE Computer Society Press.

[Blu82] Manuel Blum. Coin flipping by telephone. In *Proc. IEEE Spring COMPCOM*, pages 133–137, 1982.

[Blu86] Manuel Blum. How to Prove a Theorem So No One Else Can Claim It. In *Proceedings of the International Congress of Mathematicians*, pages 1444–1451, 1986.

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *47th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 2006.

[CIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *30th Annual ACM Symposium on Theory of Computing*, pages 141–150, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *23rd Annual ACM Symposium on Theory of Computing*, pages 542–552, New Orleans, Louisiana, USA, May 6–8, 1991. ACM Press.

[DDO+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 566–598, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.

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

[DKOS01] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Efficient and non-interactive non-malleable commitment. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 40–59, Innsbruck, Austria, May 6–10, 2001. Springer-Verlag, Berlin, Germany.

[DNS98] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *30th Annual ACM Symposium on Theory of Computing*, pages 409–418, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[Fei90] Uriel Feige. *Alternative Models for Zero Knowledge Interactive Proofs*. Weizmann Institute of Science, 1990.

[FF00] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 413–431, Santa Barbara, CA, USA, August 20–24, 2000. Springer-Verlag, Berlin, Germany.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple NonInteractive Zero Knowledge Proofs under General Assumptions. *SIAM Journal on Computing*, 29:1–28, 1999.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd Annual ACM Symposium on Theory of Computing*, pages 416–426, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.

[GK96] Oded Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. *Journal of Cryptology*, 9(2):167–189, 1996.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *27th Annual Symposium on Foundations of Computer Science*, pages 174–187, Toronto, Ontario, Canada, October 27–29, 1986. IEEE Computer Society Press.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

[Nao91] Moni Naor. Bit Commitment using Pseudorandomness. *Journal of Cryptology*, 4:151–158, 1991.

[OPV06] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Concurrent non-malleable witness indistinguishability and its applications. Technical Report ECCC Report TR06-095, ECCC, 2006.

[PR05a] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th Annual Symposium on Foundations of Computer Science*, pages 563–572. IEEE Computer Society Press, 2005.

[PR05b] Rafael Pass and Alon Rosen. New and Improved Constructions of Non-Malleable Cryptographic Protocols. In *37th Annual ACM Symposium on Theory of Computing*, pages 533–542. ACM Press, 2005.

[PR06] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments (full version). http://www.eecs.harvard.edu/∼alon/PAPERS/conc-nmc/conc-nmc.ps, 2006.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pages 387–394, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.

A Background

A polynomial-time relation R is a relation for which it is possible to verify in time polynomial in $|x|$ whether $R(x,w) = 1$. We will consider NP-languages L and denote by R_L the corresponding polynomial-time relation such that $x \in L$ if and only if there exists w such that $R_L(x,w) = 1$. We will call such a w a *valid witness for* $x \in L$ and denote by $W_L(x)$ the set of valid witnesses for $x \in L$. We will slightly abuse notation and, whenever L is clear from the context, we will simply write $W(x)$ instead of $W_L(x)$. Also for sequences $X = (x_1, \ldots, x_m)$ and $W = (w_1, \ldots, w_m)$, by the writing "$W \in W(X)$" we mean that $w_i \in W(x_i)$ for $i = 1, \ldots, m$.

For a language L we will denote by L_n^m the set of sequences of m elements of L each of length at most n. A *negligible* function $\nu(k)$ is a function such that for any constant $c < 0$ and for all sufficiently large k, $\nu(k) < k^c$.

Indistinguishability. Let $\mathcal{S}$ be a set of strings. An *ensemble* of random variables $X = \{X_s\}_{s\in\mathcal{S}}$ is a sequence of random variables indexed by elements of $\mathcal{S}$.

Definition A.1 *Two ensembles of random variables $X = \{X_s\}_{s\in\mathcal{S}}$ and $Y = \{Y_s\}_{s\in\mathcal{S}}$ are* computationally indistinguishable *if for every probabilistic polynomial-time algorithm D there exists a negligible function ν such that for any $s \in \mathcal{S}$*

$$|\text{Prob}[\ \alpha \leftarrow X_s : D(s,\alpha) = 1\ ] - \text{Prob}[\ \alpha \leftarrow Y_s : D(s,\alpha) = 1\ ]| < \nu(|s|).$$

Definition A.2 *Two ensembles of random variable $X = \{X_s\}_{s\in\mathcal{S}}$ and $Y = \{Y_s\}_{s\in\mathcal{S}}$ are* statistically indistinguishable *if there exists a negligible function ν such that for all $s \in \mathcal{S}$*

$$\sum_{\alpha} |\text{Prob}[\ X_s = \alpha\ ] - \text{Prob}[\ Y_s = \alpha\ ]| < \nu(|s|).$$

Definition A.3 *Two ensembles of random variables $X = \{X_s\}_{s\in\mathcal{S}}$ and $Y = \{Y_s\}_{s\in\mathcal{S}}$ are* perfectly indistinguishable *if for all $s \in \mathcal{S}$*

$$\sum_{\alpha} |\text{Prob}[\ X_s = \alpha\ ] - \text{Prob}[\ Y_s = \alpha\ ]| = 0.$$

One-way functions. One of the tools we will use is that of a one-way function.

Definition A.4 *A polynomial-time computable function $f : \{0,1\}^\star \to \{0,1\}^\star$ is called* one-way *if for every probabilistic polynomial time algorithm A there exists a negligible function ν such that*

$$\text{Prob}[\ x \leftarrow \{0,1\}^k; y \leftarrow A(f(x), 1^k) : f(y) = f(x)\ ] < \nu(n).$$

Interactive argument/proof systems. An *interactive proof (resp., argument) system* [GMR89] for a language L is a pair of interactive Turing machines $\langle P, V\rangle$, satisfying the requirements of *completeness* and *soundness*. Informally, completeness requires that for any $x \in L$, at the end of the interaction between P and V, where P has on input a valid witness for $x \in L$, V rejects with negligible probability. Soundness requires that for any $x \notin L$, for any computationally unbounded (resp., probabilistic polynomial-time for arguments) $P^\star$, at the end

of the interaction between $P^\star$ and V, V accepts with negligible probability. We denote by $\langle P, V\rangle(x)$ the output of the verifier V when interacting on common input x with prover P. Also, sometimes we will use the notation $\langle P(w), V\rangle(x)$ to stress that prover P receives as additional input witness w for $x \in L$.

Formally, we have the following definition.

Definition A.5 *A pair of interactive Turing machines $\langle P, V\rangle$ is an* interactive proof system *for the language L, if V is probabilistic polynomial-time and*

1. *Completeness: There exists a negligible function $\nu(\cdot)$ such that for every $x \in L$ and for every $w \in W(x)$*

$$\text{Prob}[\ \langle P(w), V\rangle(x) = 1\ ] \geq 1 - \nu(|x|).$$

2. *Soundness: For every $x \notin L$ and for every interactive Turing machines $P^\star$ there exists a negligible function $\nu(\cdot)$ such that*

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

If the soundness condition holds only with respect to probabilistic polynomial-time interactive Turing machines $P^\star$ then $\langle P, V\rangle$ is called an argument*.*

Since all protocols we give are actually argument (rather than proof) systems, we will now focus on argument systems only. Also from now on we assume that all interactive Turing machines are probabilistic polynomial-time.

Zero knowledge. The classical notion of zero knowledge has been introduced in [GMR89]. In a zero-knowledge argument system a prover can prove the validity of a statement to a verifier without releasing any additional information. This concept is formalized by requiring the existence of an expected polynomial-time algorithm, called the *simulator*, whose output is indistinguishable from the view of the verifier.

We start by defining the concept of a view of an interactive Turing machine. Let A and B be two interactive Turing machines that run on common input x and assume that A and B have additional information z_A and z_B. We denote by $\mathsf{View}_B^A(x, z_A, z_B)$ the random variable describing the *view of* B; that is, B's random coin tosses, internal state sequence, and messages received by B during its interaction with A.

We are now ready to present the notion of a zero-knowledge argument.

Definition A.6 *An interactive argument system $\langle P, V\rangle$ for a language L is* zero-knowledge *if for all polynomial-time verifiers $V^\star$, there exists an expected polynomial-time algorithm S running in expected polynomial time such that the following ensembles are computationally indistinguishable:*

$$\{\mathsf{View}_{V^\star}^P(x, w, z)\}_{x\in L, w\in W(x), z\in\{0,1\}^\star} \text{ and } \{S(x, z)\}_{x\in L, z\in\{0,1\}^\star}.$$

If the two ensembles are statistically/perfectly *indistinguishable then $\langle P, V\rangle$ is* statistical/perfect zero-knowledge*.*

A.1 Witness Indistinguishability

The notion of a witness indistinguishable argument was introduced in [FS90] and requires the *view* of the (adversarial) verifier when interacting with a honest prover to be independent of the witness used by the prover. This notion therefore concerns NP statements for which there exists more than one witness. Even though witness indistinguishability yields weaker security guarantees than zero knowledge, in several cases witness indistinguishability is sufficient for the specific task at hand and it gives very efficient protocols. Furthermore, the celebrated FLS technique [FLS99] can be used for obtaining zero knowledge from witness indistinguishability.

Let us now proceed more formally. Let $\Pi = \langle P, V \rangle$ be an argument system for language L. A *witness indistinguishability adversary* V' for Π receives as input $x \in L$, $w^0, w^1 \in W(x)$ and auxiliary information z. V' interacts with machine $P^\star$ that has a bit $b \in \{0,1\}$ wired-in. $P^\star$ receives as input (x, w^0, w^1) and executes the code of the honest prover P on input (x, w^b). For $b \in \{0,1\}$, we denote by $\mathsf{WIExpt}^b_{P,V'}(x, w^0, w^1, z)$ the random variable describing the output of V' when interacting on input (x, w^0, w^1, z) with prover $P^\star$ running on input (x, w^0, w^1) and b is the wired-in bit of $P^\star$.

Definition A.7 *Argument system $\Pi = \langle P, V \rangle$ for the language L is* witness indistinguishable *if for all probabilistic polynomial-time witness indistinguishability adversaries V' there exists a negligible function ν such that for all $x \in L$, all witnesses $w^0, w^1 \in W(x)$ and all $z \in \{0,1\}^\star$*

$$|\text{Prob}[\ \mathsf{WIExpt}^0_{P,V'}(x, w^0, w^1, z) = 1\ ] - \text{Prob}[\ \mathsf{WIExpt}^1_{P,V'}(x, w^0, w^1, z) = 1\ ]| < \nu(|x|).$$

We stress that witness indistinguishability holds with respect to adversaries that know both witnesses.

A stronger notion can be obtained if we consider adversaries that can concurrently execute several proofs. More precisely, a *concurrent witness indistinguishability adversary* V' for argument system $\Pi = \langle P, V \rangle$ for language L receives as input security parameter 1^k, sequence $X = (x_1, \ldots, x_m)$ of $m = \mathsf{poly}(k)$ elements of L each of length $n = \mathsf{poly}(k)$, two sequences $W^0 = (w^0_1, \ldots, w^0_m)$ and $W^1 = (w^1_1, \ldots, w^1_m)$ such that $w^0_i, w^1_i \in W(x_i)$ and auxiliary information z. V' interacts with m copies of machine $P^\star$ (one copy for each x_i). All copies of machine $P^\star$ have the same random bit $b \in \{0,1\}$ wired-in and the i-th copy of $P^\star$ receives as input (x_i, w^0_i, w^1_i) and executes the code of the honest prover P on input (x_i, w^b_i). V' has control of the network and decides in which order messages from different executions are delivered. For $b \in \{0,1\}$, we define the random variable $\mathsf{WIExpt}^b_{P,V'}(X, W^0, W^1, z)$ as the output of V' when interacting with m copies of machine $P^\star$ with bit b wired-in. We have the following definition.

Definition A.8 *An argument system $\Pi = \langle P, V \rangle$ for the language L is* concurrent witness indistinguishable *(*cWI argument system*) if for all efficient non-uniform adversaries V', for all k, for all $m = \mathsf{poly}(k)$ and $n = \mathsf{poly}(k)$, there exists a negligible function ν such that for all sequences X of m elements of L of length n, for all sequences $W^0, W^1 \in W(X)$ and for all $z \in \{0,1\}^*$ it holds that*

$$|\text{Prob}[\ \mathsf{WIExpt}^0_{P,V'}(X, W^0, W^1, z) = 1\ ] - \text{Prob}[\ \mathsf{WIExpt}^1_{P,V'}(X, W^0, W^1, z) = 1\ ]| < \nu(k).$$

It is known (see [FS90]) that witness indistinguishability is closed under concurrent composition. Finally we stress that the FLS paradigm [FLS99] that allows one to obtain zero knowledge from witness indistinguishability is the most used technique for designing zero knowledge protocols.

A.2 Non-Malleable Argument Systems

The notion of non-malleability has been first considered in [DDN91]. Non-malleability is concerned with an adversary $\mathcal{A}$ that mounts a so-called *man-in-the-middle attack* on two concurrent executions of a protocol Π.

Let $\Pi = \langle P, V \rangle$ be an argument system for the language L. A *man-in-the-middle adversary* $\mathcal{A}$ for Π acts as a verifier in one proof (called the *left* proof) and verifies the validity of a statement "$x \in L$" being proved by a honest party running P; and acts as a prover in another proof (called the *right* proof) in which $\mathcal{A}$ tries to convince a honest party running V of the validity of a statement $\tilde{x} \in L$ of its choice. It is assumed that $\mathcal{A}$ has complete control of the communication channel and therefore decides the scheduling of the messages. Very informally, Π is non-malleable if, whenever $x \neq \tilde{x}$, the left proof does not help $\mathcal{A}$ in the right proof.

Let us proceed more formally. For a man-in-the-middle adversary $\mathcal{A}$, we consider two executions: the *man-in-the-middle* execution and the *stand-alone* execution.

In the man-in-the-middle execution we have three parties: a honest prover P, a honest verifier V and man-in-the-middle adversary $\mathcal{A}$. In the left proof P and $\mathcal{A}$ (acting as a verifier) interact on common input $x \in L$; P receives $w \in W(x)$ as private input and $\mathcal{A}$ receives auxiliary information $z \in \{0,1\}^\star$. In the right proof $\mathcal{A}$ (acting as a prover) and V interact on common input $\tilde{x}$ chosen by $\mathcal{A}$. We denote by $\mathsf{mim}_V^{\mathcal{A}}(x, w, z)$ the random variable describing the output of V in this scenario which is V's decision and the right input $\tilde{x}$ chosen by $\mathcal{A}$. If $x = \tilde{x}$ then $\mathsf{mim}_V^{\mathcal{A}}(x, w, z)$ is the random variable that assigns positive probability only to $\perp$.

In the stand-alone execution we have only two parties: a machine S (the simulator) and a verifier V. S with access to $\mathcal{A}$ and auxiliary information $z \in \{0,1\}^\star$, interacts with V on common input x. We denote by $\mathsf{sta}_V^S(x, z)$ the random variable describing the output of V in this interaction.

Definition A.9 (non-malleable argument system) *An argument system $\Pi = \langle P, V \rangle$ for a language L is* non-malleable *if for every probabilistic polynomial-time man-in-the-middle adversary $\mathcal{A}$, there exists a probabilistic algorithm S running in expected polynomial time and a negligible function ν such that, for every $x \in L$, $w \in W(x)$, and for every $z \in \{0,1\}^\star$*

$$|\text{Prob}[\ \mathsf{mim}_V^{\mathcal{A}}(x, w, z) = 1\ ] - \text{Prob}[\ \mathsf{sta}_V^S(x, z) = 1\ ]| < \nu(|x|).$$

Tag-based non-malleability. The above definition does not say anything about the case in which $\mathcal{A}$ proves in the right proof the same theorem P proved in the left proof (that is, $\tilde{x} = x$). Actually, there is no way of preventing $\mathcal{A}$ from relaying messages from the left proof to the right proof and vice versa. The next definition requires that if, $x = \tilde{x}$, then $\mathcal{A}$'s proof must be somehow different from P's.

Consider a family $\{\langle P_{\mathtt{tag}}, V_{\mathtt{tag}} \rangle\}_{\mathtt{tag}}$ of argument systems indexed by a string $\mathtt{tag}$. As before, we will consider the man-in-the-middle execution and the stand-alone execution. More specifically, in the man-in-the-middle execution we consider $\mathcal{A}$ that, on input x and auxiliary information z, interacts in the left proof with the prover $P_{\mathtt{tag}}$ on input (x, w) and in the right proof with verifier $V_{\tilde{\mathtt{tag}}}$ on input $\tilde{x}$. The tag $\tilde{\mathtt{tag}}$ of the right proof as well the input $\tilde{x}$ of the right proof are chosen adaptively by $\mathcal{A}$. We denote by $\mathsf{mim}_V^{\mathcal{A}}(\mathtt{tag}, x, w, z)$ the random

variable describing the output of V in this scenario (it is V's decision, the tag $\tilde{\mathtt{tag}}$ and the statement $\tilde{x} \in L$). Similarly $\mathsf{sta}_V^S(\mathtt{tag}, x, z)$ is defined as the output of the verifier while interacting with S. Similarly to the previous case, if the right proof contains the same tag used in the left proof, then $\mathsf{mim}_V^{\mathcal{A}}(\mathtt{tag}, x, w, z)$ gives positive probability only to the string $\perp$.

Definition A.10 (tag-based non-malleable argument) *A family of argument systems $\Pi = \{\langle P_{\mathtt{tag}}, V_{\mathtt{tag}}\rangle\}_{\mathtt{tag}}$ for a language L is a* tag-based non-malleable argument with tags of length ℓ *if for every probabilistic polynomial-time man-in-the-middle adversary $\mathcal{A}$, there exist a probabilistic algorithm S running in expected polynomial time and a negligible function ν such that for every $x \in L$, $w \in W(x)$, for every $\mathtt{tag} \in \{0,1\}^\ell$, and for every $z \in \{0,1\}^\star$*

$$|\mathrm{Prob}[\ \mathsf{mim}_V^{\mathcal{A}}(\mathtt{tag}, x, w, z) = 1\ ] - \mathrm{Prob}[\ \mathsf{sta}_V^S(\mathtt{tag}, x, z) = 1\ ]| < \nu(|x|).$$

Non-malleable zero knowledge. Consider an argument system $\Pi = \langle P, V\rangle$ for an NP-language L. Let $\mathcal{A}$ be a man-in-the-middle adversary attacking Π. Then, with a slight abuse of notation, by $\mathsf{View}_{\mathcal{A}}^P(x, w, z)$ we denote the random variable describing the view obtained by $\mathcal{A}$ in the left and the right proof (including the sequence of its internal states and messages sent and received by $\mathcal{A}$) when given auxiliary information z. In the left proof $\mathcal{A}$ is interacting with a honest prover P on common input "$x \in L$" and P receives a valid witness w for x as private input. In the right proof $\mathcal{A}$ interacts with the honest verifier on input $\tilde{x}$ chosen by $\mathcal{A}$.

Definition A.11 (NMZK argument system) *A non-malleable argument system $\Pi = \langle P, V\rangle$ for a language L is* non-malleable zero-knowledge *(in short* NMZK*) if for any probabilistic polynomial-time man-in-the-middle adversary $\mathcal{A}$, there exists a probabilistic algorithm S running in expected polynomial time such that, the ensembles*

$$\{\mathsf{View}_{\mathcal{A}}^P(x, w, z)\}_{x\in L, w\in W(x), z\in\{0,1\}^\star} \text{ and } \{S(x, z)\}_{x\in L, z\in\{0,1\}^\star}$$

are computationally indistinguishable.

If the two ensembles are statistically/perfectly indistinguishable then Π is said to be non-malleable statistical/perfect zero-knowledge.

NMZK arguments of knowledge. The notion of non-malleable zero knowledge argument of *knowledge* is obtained by requiring that the simulator also outputs the witness encoded in the right proof (in which the man-in-the-middle adversary $\mathcal{A}$ plays as a prover). This notion was introduced by [DDO+01] for non-interactive NMZK and clearly implies non-malleability.

Definition A.12 (NMZK arguments of knowledge) *An argument system $\Pi = \langle P, V\rangle$ for a language L is a* non-malleable zero-knowledge argument of knowledge *if for every probabilistic polynomial-time man-in-the-middle adversary $\mathcal{A}$, there exists a probabilistic algorithm S (called the simulator-extractor) running in expected polynomial time such that by denoting as $S(x, z) = (S_0(x, z), S_1(x, z))$, the output of $S(x, z)$, we have that:*

1. *$\{S_0(x, z)\}_{x\in L, z\in\{0,1\}^\star}$ is computationally indistinguishable from $\{\mathsf{View}_{\mathcal{A}}^P(x, w, z)\}_{x\in L, w\in W(x), z\in\{0,1\}^\star}$;*
2. *$S_1(x, z) = \tilde{w}$ and if the right proof given in $\{S_0(x, z)\}_{x\in L, z\in\{0,1\}^\star}$ is accepting with common input $\tilde{x} \neq x$ we have that, except with negligible probability, $\tilde{w} \in W(\tilde{x})$.*

The notion of tag-based NMZK argument of knowledge is obtained by requiring that the extraction procedure is successful if the right proof has a tag different from the one of the left proof. We also stress that we allow the adversary $\mathcal{A}$ to pick the theorem and the tag to be used in the right proof.

A.3 Concurrent Non-Malleable Zero-Knowledge Arguments of Knowledge

In a more powerful man-in-the-middle attack, the adversary $\mathcal{A}$ is not restricted to one proof on the left and one proof on the right but instead $\mathcal{A}$ is allowed to concurrently play polynomially many left and right proofs. We call such an adversary a *concurrent man-in-the-middle adversary*. Specifically, let k be the security parameter. Consider a vector $X = (x_1, \ldots, x_m)$ of $m = \mathsf{poly}(k)$ inputs each of length $n = \mathsf{poly}(k)$ and a vector $W = (w_1, \ldots, w_m)$, such that $w_1 \in W(x_1), \ldots, w_m \in W(x_m)$. In the man-in-the-middle execution, the i-th left proof, for $1 \leq i \leq m$, is played by (an instance of) the honest prover P on input (x_i, w_i) and by the adversary $\mathcal{A}$ on input (x_i, z), for some auxiliary information z; the j-th right proof, for $1 \leq j \leq l$, is played by $\mathcal{A}$ on input $\tilde{x}_j$ chosen by $\mathcal{A}$ and auxiliary information z and by (an instance of) the verifier V on input $\tilde{x}_j$. We assume that $\mathcal{A}$ has complete control over the network and thus decides when each message of each proof is delivered.

cNMZK arguments of knowledge. The next definition extends the notion of an NMZK Argument of Knowledge (as defined in Definition A.12) to the concurrent scenario.

Definition A.13 (cNMZK arguments of knowledge) *An argument system $\Pi = \langle P, V \rangle$ for the language L is a* concurrent non-malleable zero-knowledge argument of knowledge *(a* cNMZK argument of knowledge*) if for every probabilistic polynomial-time concurrent man-in-the-middle adversary $\mathcal{A}$ there exists a probabilistic algorithm S (called the simulator-extractor) running in expected polynomial time such that for all $m = \mathsf{poly}(k)$ and $n = \mathsf{poly}(k)$, by denoting with $S(X, z) = (S_0(X, z), S_1(X, z))$ the output of S on input (X, z) , we have that:*

1. *$\{S_0(X, z)\}_{X \in L_n^m, z \in \{0,1\}^\star}$ and $\{\mathsf{View}_{\mathcal{A}}^P(X, W, z)\}_{X \in L_n^m, W \in W(X), z \in \{0,1\}^\star}$ are computationally indistinguishable;*
2. *$S_1(X, z) = (\tilde{w}_1, \ldots, \tilde{w}_m)$ where, except with negligible probability, $\tilde{w}_j \in W(\tilde{x}_j)$ for $1 \leq j \leq m$ and $\tilde{x}_j \notin X$ is the common input of the j-th accepting right proof given in $\{S_0(X, z)\}_{X \in L_n^m, z \in \{0,1\}^\star}$*

To define the notion of tag-based cNMZK argument of knowledge we define the view $\mathsf{View}_{\mathcal{A}}^P(T, X, W, z)$ of a tag-based man-in-the-middle adversary $\mathcal{A}$ when T is the sequence of tags, X is the sequence of inputs and W is the sequence of witnesses used in the left proofs as all messages receives by $\mathcal{A}$ in left and right proofs along with $\mathcal{A}$'s internal coin tosses.

Definition A.14 (tag-based cNMZK arguments of knowledge) *A family $\Pi = \{\langle P_{\mathtt{tag}}, V_{\mathtt{tag}} \rangle\}_{\mathtt{tag}}$ of argument systems for the language L is a* tag-based concurrent non-malleable zero-knowledge argument of knowledge with tags of length ℓ *(a* cNMZK argument of knowledge*) if for every probabilistic polynomial-time tag-based concurrent man-in-the-middle adversary $\mathcal{A}$ there exists a probabilistic algorithm S (called the simulator-extractor) running in expected polynomial time such that for all $m = \mathsf{poly}(k)$ and $n = \mathsf{poly}(k)$, and for all sequences T of m tags of length ℓ by denoting with $S(T, X, z) = (S_0(T, X, z), S_1(T, X, z))$ the output of S on input (T, X, z), we have that:*

1. *$\{S_0(T, X, z)\}_{T \in \{0,1\}^{ml}, X \in L_n^m, z \in \{0,1\}^\star}$ and $\{\mathsf{View}_{\mathcal{A}}^P(T, X, W, z)\}_{T \in \{0,1\}^{ml}, X \in L_n^m, W \in W(X), z \in \{0,1\}^\star}$ are computationally indistinguishable;*
2. *$S_1(T, X, z) = (\tilde{w}_1, \ldots, \tilde{w}_m)$ and for all accepting right proofs j with tag $\tilde{\mathtt{tag}}_j \notin T$ we have that, except with negligible probability, $\tilde{w}_j \in W(\tilde{x}_j)$.*

One-left many-right cNMZK arguments of knowledge. Weaker notions of cNMZK can be obtained by restricting the power of the concurrent man-in-the-middle adversary $\mathcal{A}$.

If we allow the adversary to be active in only one left proof, then we obtain the notion of a *one-left many-right cNMZK argument of knowledge.* The following theorem is from [PR05b, PR05a, PR06][4].

Theorem A.15 ([PR05b, PR05a, PR06]) Assume that there exists a family of claw-free permutations. Then for any NP language L there exists a constant-round tag-based one-left many-right perfect cNMZK arguments of knowledge $\mathsf{nmZK} = \{\langle \mathcal{P}_{\mathtt{tag}}, \mathcal{V}_{\mathtt{tag}} \rangle\}_{\mathtt{tag}}$ for all NP.

According to the above definition, this theorem says that for any efficient one-left many-right concurrent man-in-the-middle adversary $\mathcal{A}$ that is restricted to one left session there exists an efficient simulator S that guarantees:

1. the view (including the left proof and all the right proofs) given in output by S is perfectly indistinguishable from the interaction of $\mathcal{A}$ with honest prover and honest verifiers;
2. the extraction succeeds for all accepting right proofs in which the one-left many-right concurrent man-in-the-middle adversary has used a tag not appearing in the left proof;

The last property is based on a technique referred to as simulation-extraction that combines non-black-box simulation with black-box extraction. Indeed, while the simulator is simulating a proof to the adversary, the extractor rewinds the adversary (and thus the simulation itself) and still extracts a valid witness from the proof given by the adversary.

A.4 Other Tools

Secure signature schemes. A secure signature scheme $SS = (\mathtt{SG}, \mathtt{Sig}, \mathtt{SVer})$ is a triple of efficient algorithms. The key generation algorithm SG on input a security parameter 1^k returns a pair $(\mathsf{pk}, \mathsf{sk})$ that are respectively a public and a secret key. The secret key sk is used to sign a message m by running the signature algorithm Sig on input sk and the message m and obtains the signature s. The public key pk instead is used to verify signatures by means of the SVer algorithm that runs on input the public key pk, the message m and the signature s and outputs a bit. The security requirement guarantees that no polynomial-time adversary that is given access to a signature oracle is able to produce a signature of a message for which it has not queried the oracle, or to produce a new signature of a message for which it has queried the oracle (this last requirement defines a *strong* secure signature scheme). See [Rom90] for a construction of a secure signature scheme based on one-way functions.

Commitment Schemes. Informally, a commitment scheme is a two-phase two-party protocol played by the *committer* C (sometimes also called the committer) and the *receiver* R.

In the *commitment phase* C receives as input a message m. The commitment phase has one common output com, called the *commitment*, and one private input, to be used in the next phase, for each party.

In the *decommitment phase* both parties receive as input the common output com of the commitment phase and their respective private auxiliary information. During the decommitment phase C decommits com to m by sending m to R and then engaging in a protocol with

[4]The use of claw-free permutations and the perfect zero-knowledge property are in particular discussed in [PR06].

R. At the end of the decommitment phase R may accept the decommitment of m as com (in which case we say that decommitment was *successful*) or reject the decommitment.

The *hiding* property requires that the commitment phase does not reveal any information about m to a polynomial-time adversarial receiver. If the hiding property holds with respect to unbounded adversarial receivers then we say that the commitment scheme is *statistically* hiding.

The *binding* property requires that a polynomial-time adversarial committer can not produce a commitment com and later decommit it to two different messages m_0 and m_1. If the binding property holds with respect to unbounded adversarial committer then we say that the commitment scheme is *statistically* binding.

Non-interactive commitment schemes. In our schemes we will use *non-interactive* commitment schemes. These are schemes in which the commitment phase consists of a single message from committer to receiver. We will describe such schemes by means of a commitment algorithm Com. Specifically, let k be a security parameter. To commit to a message m of length $\mathsf{poly}(k)$, the committer picks a random k-bit string s and computes $\mathsf{com} = \mathsf{Com}(m, s)$ and sends it to the receiver. To decommit commitment com to message m, the committer sends message m and string s to the receiver that verifies that indeed com has been computed as $\mathsf{com} = \mathsf{Com}(m, s)$.

It is possible to construct non-interactive statistically binding commitment schemes based on any one-to-one one-way function (see for example [Gol01]). If we allow a setup message from the receiver to the committer, then we can construct non-interactive statistically binding commitment schemes based on any one-way function [Nao91] and non-interactive statistically hiding commitment schemes based on certified claw-free functions [GK96].

B Previous Schemes

In this section we review two recent schemes (due to [PR05b]) of non-malleable commitment: one that is NMc and one that is NMd. In Section C, we show how to combine the two schemes to get a non-malleable commitment scheme that is simultaneously NMc and NMd. Moreover in Section 3 we will show that using new techniques, we can keep both security guarantees in the concurrent setting.

These schemes are based on TheoremA.15 of [PR05b, PR05a, PR06] (that is also an ingredient for our schemes)

A statistically binding NMc scheme $\mathcal{NM}c$. Denote by SBCom a non-interactive statistically binding commitment scheme and by $\mathsf{nmZK} = \{\langle \mathcal{P}_t, \mathcal{V}_t \rangle\}$ a tag-based perfect NMZK argument of knowledge (see [PR05b]) for all $\mathtt{NP}$. $\mathcal{NM}c$ is conceptually very simple (see Figure 2). The committer commits using (the possibly malleable) commitment scheme SBCom and then proves knowledge of the committed value by using nmZK. For the proof, we construct a simulator that commits to 0^k. Since we are arguing NMc, there is no need to consider the decommitment phase. The proof then follows from the fact that nmZK is a perfect non-malleable argument of knowledge and that the commitment scheme Com is hiding (see [PR05b]).

A statistically hiding NMd scheme $\mathcal{NM}d$. Denote by SHCom a statistically hiding commitment scheme and by $\mathsf{nmZK} = \{\langle \mathcal{P}_t, \mathcal{V}_t \rangle\}$ a tag-based perfect NMZK argument of knowledge (see [PR05b]) for all $\mathtt{NP}$. The scheme is conceptually the dual of NMc (see Figure 3).

Security Parameter: 1^k.
Input to Committer: $m \in \{0,1\}^k$.
Commitment Phase:
$C \to R$: Pick $s \in \{0,1\}^k$ and send $c = \mathsf{SBCom}(m, s)$.
$C \leftrightarrow R$: C proves to R using nmZK_c that there exist $m, s \in \{0,1\}^k$ so that $c = \mathsf{SBCom}(m, s)$.
R: Verify that nmZK_c is accepting.
Decommitment Phase:
$C \to R$: Send m, s.
R: Verify that $c = \mathsf{SBCom}(m, s)$.

Figure 2: The NMc commitment scheme $\mathcal{NM}c$ of [PR05b].

In the commitment phase the committer computes (the possibly malleable) commitment com of m using commitment scheme SHCom. In the decommitment phase, the committer sends m to the receiver and then proves, using nmZK, that m is the correct decommitment.

Security Parameter: 1^k.
Input to Committer: $m \in \{0,1\}^k$.
Commitment Phase:
$R \to C$: Pick $r \in \{0,1\}^k$ and send it to C;
$C \to R$: Pick $s \in \{0,1\}^k$ and send $c = \mathsf{SHCom}(r, m, s)$.
Decommitment Phase:
$C \to R$: Send m.
$C \leftrightarrow R$: C proves to R using $\mathsf{nmZK}_{c,r,m}$ that there exist $s \in \{0,1\}^k$ so that $c = \mathsf{SHCom}(r, m, s)$.
R: Verify that $\mathsf{nmZK}_{c,r,m}$ is accepting.

Figure 3: The NMd commitment scheme $\mathcal{NM}d$ of [PR05b].

C The Constant Round Commitment Scheme $\mathcal{NM}cd$

In Figure 4 we describe commitment scheme $\mathcal{NM}cd$ that is both NMc and NMd. We will use a non-interactive statistically binding commitment scheme Com in which the committer can commit to a message m by picking a random s and by setting $c = \mathsf{Com}(m, s)$. Such commitment schemes can be constructed using 1-to-1 one-way functions (see [Gol01]). Also we denote by $\mathsf{nmZK} = \{\langle \mathcal{P}_t, \mathcal{V}_t \rangle\}$ a tag-based perfect NMZK argument of knowledge (see Theorem A.15) for all $\mathtt{NP}$. See Appendix A.3 and A.4 for details about these tools.

At a very high level, $\mathcal{NM}cd$ can be seen as a combination of the schemes $\mathcal{NM}c$ and $\mathcal{NM}d$. Specifically, the commitment phase consists of committing m using the (possibly

malleable) non-interactive commitment scheme Com and then proving knowledge of the committed message using nmZK. In the decommitment phase, the committer sends message m to the receiver and then proves, using nmZK, that m is the message originally committed to.

In the next sections we prove the properties of $\mathcal{NMC}d$. We stress that, being simultaneously NMc and NMd, guarantees security if the MiM adversary is active both in the commitment phase and in the decommitment phase.

Security Parameter: 1^k.
Input to Committer: $m \in \{0,1\}^k$.
Commitment Phase:
$C \to R$: Pick $s \in \{0,1\}^k$ and send $c = \mathsf{Com}(m,s)$.
$C \leftrightarrow R$: C proves to R that there exist $m, s \in \{0,1\}^k$ so that $c = \mathsf{Com}(m,s)$ by running $\mathcal{P}_c$ on input c and witness (m,s).
R: Verify that $\mathcal{V}_c$ on input c is accepting.
Set $\mathtt{tag} = (c,t)$ where t is the transcript of the interaction between $\mathcal{P}_c$ and $\mathcal{V}_c$.
Decommitment Phase:
$C \to R$: Send m.
$C \leftrightarrow R$: C proves to R that there exist $s \in \{0,1\}^k$ so that $c = \mathsf{Com}(m,s)$ by running $\mathcal{P}_{m,\mathtt{tag}}$ on input (m,c) and witness s.
R: Verify that $\mathcal{V}_{m,\mathtt{tag}}$ is accepting.

Figure 4: Commitment scheme $\mathcal{NMC}d$ enjoying both NMc and NMd.

We have the following theorem.

Theorem C.1 *Under the existence of a family of claw-free permutations there exists a constant-round commitment scheme that enjoys both NMc and NMd.*

Proof.
Hiding and binding. The proof of the hiding property is precisely the same of the statistically binding NMc commitment scheme of Fig. 2 as the commitment phase is identical. The proof of the binding property is instead very similar to (actually it is properly contained in) the proof of the binding property of the statistically hiding NMd commitment scheme of Fig. 3. We remark that $\mathcal{NMC}d$ is *not* statistically binding as an unbounded adversary might be able to cheat in the execution of nmZK (which is an argument and not a proof) and open a commitment in any way.

$\mathcal{NMC}d$ **is NMc.** In this section we prove that $\mathcal{NMC}d$ is NMc by giving, for every MiM adversary $\mathcal{A}$, a simulator S that satisfies Definition 3.1 when there is just one left and one right commitment.. S, on input z and security parameter 1^k, interacts with a honest receiver R and runs adversary $\mathcal{A}(z)$ internally. S simulates the left and the right commitment phases for $\mathcal{A}$ in the following way. In the left interaction, S behaves as a honest committer committing to $m = 0^k$; that is, S picks s at random, computes c as $c = \mathsf{Com}(0^k, s)$ and then executes the code of the honest prover $\mathcal{P}_c$ to prove knowledge of the message committed to by c. In the right interaction, S acts as a relayer between $\mathcal{A}(z)$ and R.

Suppose, for the sake of contradiction, that there exists a probabilistic polynomial-time distinguisher D and infinite set $\mathcal{K} \subseteq \mathbb{N}$ such that for all $k \in \mathcal{K}$ there exists $v_k \in \{0,1\}^k$ such that

$$|\text{Prob}[\ D(v_k, \mathsf{sis}^S_{\mathsf{Com}}(1^k, z)) = 1\ ] - \text{Prob}[\ D(v_k, \mathsf{mim}^{\mathcal{A}}_{\mathsf{Com}}(v_k, z)) = 1\ ]| \geq 1/q(k),$$

for some polynomial $q(\cdot)$ and consider the following sequence of experiments involving $\mathcal{A}$, starting with $\mathsf{Expt}^{\mathcal{A}}_0(v, z)$.

$\mathsf{Expt}^{\mathcal{A}}_0(v, z)$:

Left commitment phase. $\mathcal{A}(z)$ interacts with honest committer that commits to v. Let c be the commitment sent by the honest committer.

Right commitment phase. $\mathcal{A}(z)$ interacts with honest receiver. Let $\tilde{c}$ be the commitment sent by $\mathcal{A}$ in the right commitment phase.

Output. If the right commitment phase is successful and $\tilde{c} \neq c$, then let $\tilde{v}$ be the value committed to by $\tilde{c}$; otherwise let $\tilde{v} = \perp$. Output $D(v, \tilde{v})$.

We set $p^{\mathcal{A}}_0(v, z) = \text{Prob}[\ \mathsf{Expt}^{\mathcal{A}}_0(v, z) = 1\ ]$. We observe that, for all $v \in \{0,1\}^k$, $p^{\mathcal{A}}_0(0^k, z) = \text{Prob}[\ D(v, \mathsf{sis}^S_{\mathsf{Com}}(z)) = 1\ ]$ and $p^{\mathcal{A}}_0(v, z) = \text{Prob}[\ D(v, \mathsf{mim}^{\mathcal{A}}_{\mathsf{Com}}(v, z)) = 1\ ]$. Therefore, by assumption,

$$|p^{\mathcal{A}}_0(v, z) - p^{\mathcal{A}}_0(0^k, z)| \geq 1/q(k)$$

for all $k \in \mathcal{K}$.

To define experiment $\mathsf{Expt}^{\mathcal{A}}_1$, we observe that the MiM adversary $\mathcal{A}$ naturally defines an adversary $\mathcal{A}'$ for nmZK. Therefore there exists a simulator-extractor S' for nmZK that simulates a left interaction with $\mathcal{A}'$ and extracts the witness for the right proof provided by $\mathcal{A}'$.

$\mathsf{Expt}^{\mathcal{A}}_1(v, z)$:

Left commitment phase. $\mathcal{A}(z)$ interacts with the following committer $\hat{C}$. $\hat{C}$ picks s at random and computes $c = \mathsf{Com}(v, s)$ and then executes the simulator-extractor S' for nmZK on input c and tag c.

Right commitment phase. $\mathcal{A}(z)$ interacts with the simulator extractor S' for nmZK running on input $\tilde{c}$ and tag $\tilde{c}$.

Output. Let View be the view of $\mathcal{A}'$ as output by S'. Let $\tilde{c}$ be the commitment sent by $\mathcal{A}$ in the right commitment phase of View.
If the right commitment phase of View is successful and $\tilde{c} \neq c$, then let $\tilde{v}$ be the value committed to by $\tilde{c}$; otherwise let $\tilde{v} = \perp$. Output $D(v, \tilde{v})$.

We set $p^{\mathcal{A}}_1(v, z) = \text{Prob}[\ \mathsf{Expt}^{\mathcal{A}}_1(v, z) = 1\ ]$. Since nmZK is perfect zero-knowledge, we have that, for all $v \in \{0,1\}^k$, $p^{\mathcal{A}}_1(v, z) = p^{\mathcal{A}}_0(v, z)$ and thus

$$|p^{\mathcal{A}}_1(v, z) - p^{\mathcal{A}}_1(0^k, z)| \geq 1/q(k),$$

for all $k \in \mathcal{K}$.

Experiment $\mathsf{Expt}^{\mathcal{A}}_2(v, z)$ is similar to $\mathsf{Expt}^{\mathcal{A}}_1(v, z)$ with the only difference that the output is computed in a different way.

$\mathsf{Expt}_2^{\mathcal{A}}(v,z)$:

Left commitment phase. $\mathcal{A}(z)$ interacts with the following committer $\hat{C}$. $\hat{C}$ picks s at random and computes $c = \mathsf{Com}(v,s)$ and then executes the simulator-extractor S' for nmZK on input c and tag c.

Right commitment phase. $\mathcal{A}(z)$ interacts with the simulator extractor S' for nmZK running on input $\tilde{c}$ and tag $\tilde{c}$.

Output. Let View be the view of $\mathcal{A}'$ output by S', $\tilde{c}$ be the commitment sent by $\mathcal{A}$ in the right commitment phase of View and let $\tilde{w} = (v', s')$ be the witness for $\tilde{c}$ output by S'.
If the right commitment phase of View is successful and $\tilde{c} \neq c$, then let $\tilde{v} = v'$; otherwise let $\tilde{v} = \perp$. Output $D(v, \tilde{v})$.

We set $p_2^{\mathcal{A}}(v,z) = \text{Prob}[\ \mathsf{Expt}_2^{\mathcal{A}}(v,z) = 1\ ]$. If $\tilde{c} \neq c$ then the non-malleable argument of knowledge of the right commitment phase uses a tag different from the one used in the non-malleable argument of knowledge of the left commitment phase and thus, with overwhelming probability, S' outputs the unique valid witness. Therefore, conditioned on $\tilde{c} \neq c$, we have that

$$|p_2^{\mathcal{A}}(v,z) - p_1^{\mathcal{A}}(v,z)| \leq \nu(k)$$

for some negligible function ν. On the other hand, conditioned on $\tilde{c} = c$, we have $p_2^{\mathcal{A}}(v,z) = p_1^{\mathcal{A}}(v,z)$. Therefore, there exists a poly $q_1(\cdot)$ such that

$$|p_2^{\mathcal{A}}(v_k,z) - p_2^{\mathcal{A}}(0^k,z)| \geq 1/q_1(k),$$

for all $k \in \mathcal{K}$. Now observe that $\mathsf{Expt}_2^{\mathcal{A}}$ (unlike $\mathsf{Expt}_1^{\mathcal{A}}$ and $\mathsf{Expt}_0^{\mathcal{A}}$) can be performed in polynomial time and thus the above inequality implies that the commitment scheme is not hiding thus reaching a contradiction. More precisely, we can construct an adversary $\mathcal{B}$ for breaking commitment scheme Com that receives as input a commitment $\hat{c}$ of either 0^k or of v_k. $\mathcal{B}$ executes $\mathsf{Expt}_2^{\mathcal{A}}(v_k,z)$ and, instead of setting $c = \mathsf{Com}(v_k,s)$ sets $c = \hat{c}$. We notice that $\mathcal{B}$ can perform $\mathsf{Expt}_2^{\mathcal{A}}(v,z)$ even without knowing the value committed to by $\hat{c}$. This concludes the proof that $\mathcal{NMC}d$ is NMc.

$\mathcal{NMC}d$ is NMd. For every MiM adversary $\mathcal{A}$, we describe a simulator S such that $\mathsf{mim}_{\mathsf{Dec}}^{\mathcal{A}}$ and $\mathsf{sis}_{\mathsf{Dec}}^{S}$ are indistinguishable. For the sake of simplifying the notation, we will denote by $\mathcal{A}_{\mathsf{Dec}}$ the state of $\mathcal{A}$ (i.e., $\mathcal{A}$'s a description along with its configuration at that time slot) after the commitment phases (i.e., before the decommitment phases starts).

Let us now describe simulator S for $\mathcal{A}$. S, on input z and security parameter 1^k, interacts with a honest receiver R and runs adversary $\mathcal{A}(z)$ internally. S simulates the left and the right commitment phases for $\mathcal{A}$ in the following way. For the left commitment phase, S behaves like a honest committer that commits to message 0^k. For the right commitment phase, S relays message between $\mathcal{A}$ and R. Once the commitment phase has terminated, S receives message m.

The basic idea is that now S will play the decommitment phases internally with $\mathcal{A}_{\mathsf{Dec}}$ (we stress that $\mathcal{A}_{\mathsf{Dec}}$ is precisely $\mathcal{A}$ after the commitment phases) playing as both sender in the left decommitment phase and as receiver in the right decommitment phase. This allows S to obtain the decommitment of the right commitment by means of the simulation-extractability

property of nmZK. Then S will be able to open to R the message that it committed in the commitment phase.

More formally, let c and $\tilde{c}$ be the commitments of the left and right commitment phases and let t and $\tilde{t}$ be the two transcripts. Since the decommitment phase is simply a message that specifies a statement and an execution of nmZK for that statement, the existence of $\mathcal{A}_{\mathsf{Dec}}$ implies the existence of a simulator-extractor S_{Dec} for nmZK. S then runs S_{Dec} on input (c, m), using tag $\mathtt{tag} = (c, t)$ and obtains the view View and witness $\tilde{w}$ for the right theorem proved by $\mathcal{A}_{\mathsf{Dec}}$ (i.e., by $\mathcal{A}$). If the right decommitment phase in View terminates successfully then it must be the case that $\mathcal{A}_{\mathsf{Dec}}$ has successfully proved theorem $(\tilde{c}, \tilde{m})$ for some string $\tilde{m}$. If $\tilde{w}$ is a witness for $(\tilde{c}, \tilde{m})$ then S performs the decommitment phase with R by executing $\mathcal{P}_{\mathtt{tag}}$ on input $(\tilde{c}, \tilde{m})$ and witness $\tilde{w}$. Otherwise S aborts. This ends the description of S.

We now prove that S is a good simulator; that is, for all probabilistic polynomial-time distinguishers D, for all z, for all $k \in \mathbb{N}$ and for all $m \in \{0,1\}^k$,

$$|\text{Prob}[\ D(m, \mathsf{sis}^S_{\mathsf{Dec}}(m, z)) = 1\ ] - \text{Prob}[\ D(m, \mathsf{mim}^{\mathcal{A}}_{\mathsf{Dec}}(m, z)) = 1\ ]|$$

is negligible in k. To this aim, consider the following sequence of experiments, starting with $\mathsf{Expt}_0(m, z)$. We denote by $p_i(m, z)$ the probability that $\mathsf{Expt}_i(m, z)$ returns 1. Experiment $\mathsf{Expt}_0(m, z)$ consists in sampling $\mathsf{mim}^{\mathcal{A}}(m, z)$ obtaining $\tilde{m}$ and returning $D(m, \tilde{m})$. Therefore, $p_0(m, z) = \text{Prob}[\ D(m, \mathsf{mim}^{\mathcal{A}}(m, z)) = 1\ ]$.

Let us now consider experiment $\mathsf{Expt}_1(m, z)$ which differs from $\mathsf{Expt}_0(m, z)$ only in the decommitment phase. As observed in the description of S, $\mathcal{A}_{\mathsf{Dec}}$ implies the existence of S_{Dec}. Then in $\mathsf{Expt}_1(m, z)$, instead of performing the decommitment phase, we run S_{Dec} on input (m, c) (where c is the commitment that appears in the left commitment phase), and tag $\mathtt{tag} = (c, t)$. S_{Dec} outputs $\mathsf{View}_{\mathsf{Dec}}$ and witness $\tilde{w}$. Notice that in these decommitment phases of Expt_1 the simulator is playing both as sender in the left decommitment and as receiver in the right decommitment. If in $\mathsf{View}_{\mathsf{Dec}}$ the right decommitment phase is successful and different from the left one, the output of Expt_1 is $D(m, \tilde{m})$ where $\tilde{m}$ is the decommitment of $\tilde{c}$ in $\mathsf{View}_{\mathsf{Dec}}$. Otherwise Expt_1 returns $D(m, \perp)$. It follows from the perfect simulation of S that $p_1(m, z) = p_0(m, z)$.

In $\mathsf{Expt}_2(m, z)$ we use the simulator of nmZK for the commitment phase too and leave the decommitment phase as in $\mathsf{Expt}_1(m, z)$. Specifically, since the commitment phase is again the composition of a message that defines a statement and of an execution of nmZK for that statement, we observe that $\mathcal{A}$ is actually a man-in-the-middle for nmZK and we denote by S_{Com} the associated simulator. The commitment phase is then performed by computing a commitment c of m and then running S_{Com} on input c and z as auxiliary input. We denote by $\mathsf{View}_{\mathsf{Com}}$ the obtained view. Notice that S_{Com} also plays the role of honest receiver in the right commitment phase, therefore R is not part of this experiment. By the fact that S_{Com} is a perfect simulator (including both the right and left commitment phases), we obtain $p_2(m, z) = p_1(m, z)$.

$\mathsf{Expt}_3(m, z)$ is similar to $\mathsf{Expt}_2(m, z)$ with the only difference that c is computed as $\mathsf{Com}(0^k)$. The hiding property of the commitment guarantees that $|p_3(m, z) - p_2(m, z)|$ is negligible. Indeed, any distinguisher between $\mathsf{Expt}_3(m, z)$ and $\mathsf{Expt}_2(m, z)$ can be used to distinguish a commitment of either 0^k or m by simply playing this commitment as c, completing the experiment and then giving in output the same output of the distinguisher.

$\mathsf{Expt}_4(m,z)$ is similar to $\mathsf{Expt}_3(m,z)$ with the only difference that in $\mathsf{Expt}_4(m,z)$ we use the algorithm of the prover of nmZK in the left commitment phase instead of the simulator of nmZK. Since the simulation is perfect $p_4(m,z) = p_3(m,z)$.

$\mathsf{Expt}_5(m,z)$ is similar to $\mathsf{Expt}_4(m,z)$ with the following difference. The right commitment phases is played by relying with R all messages. Since in $\mathsf{Expt}_4(m,z)$ the receiver of the right commitment phase is played by S_{Com} running the honest receiver algorithm, we have that $p_5(m,z) = p_4(m,z)$.

$\mathsf{Expt}_6(m,z)$ is similar to $\mathsf{Expt}_5(m,z)$ with the following difference. The right decommitment phase is performed by running with R the code of the honest prover of nmZK using the witness $\tilde{w}$ obtained by S_{Dec}. As before, the output of the experiment is $D(m,\tilde{m})$ if the right decommitment phase is successful and differs from the left one. Otherwise the experiment outputs $D(m,\perp)$. We observe that, since S_{Dec} obtains the unique valid witness (we stress that the commitment scheme used as subprotocol is statistically binding) except with negligible probability, $|p_6(m,z) - p_5(m,z)|$ is negligible.

Finally note that $\mathsf{Expt}_6(m,z)$ corresponds to $\mathsf{sis}(m,z)$, therefore the proof is completed.

The theorem then follows from the observation that claw-free permutations are sufficient for the existence of nmZK (see Theorem A.15) and non-interactive statistically binding commitment schemes. □