Concurrent Non-Malleable Zero Knowledge with Adaptive Inputs

Huijia Lin[*] Rafael Pass[**]

Cornell University
{huijia,rafael}@cs.cornell.edu

Abstract. Concurrent non-malleable zero-knowledge ($\mathcal{CNMZK}$) considers the concurrent execution of zero-knowledge protocols in a setting where the attacker can simultaneously corrupt multiple provers and verifiers. We provide the first construction of a $\mathcal{CNMZK}$ protocol that, without any trusted set-up, remains secure even if the attacker may adaptively select the statements to receive proofs of. Previous works [BPS06,LPTV10] only handle scenarios where the statements are fixed at the beginning of the execution, or chosen adaptively from a restricted set of statements.

1 Introduction

Zero-knowledge ($\mathcal{ZK}$) interactive proofs [GMR89] are fundamental constructs that allow the Prover to convince the Verifier of the validity of a mathematical statement $x \in L$, while providing *zero additional knowledge* to the Verifier. *Concurrent* $\mathcal{ZK}$, first introduced and achieved by Dwork, Naor and Sahai [DNS04], considers the execution of zero-knowledge protocols in an asynchronous and concurrent setting. In this model, an adversary acts as verifiers in many concurrent executions of the zero-knowledge protocol, and launches a coordinated attack on multiple independent provers to gain knowledge. *Non-malleable* $\mathcal{ZK}$, first introduced and achieved by Dolev, Dwork and Naor [DDN00], also considers the concurrent execution of zero-knowledge protocols, but in a different manner. In this model, an adversary concurrently participates in only two executions, but plays different roles in the two executions; in the first execution (called the left execution), it acts as a verifier, whereas in the second execution (called the right execution) it acts as a prover. The notion of *Concurrent Non-malleable* $\mathcal{ZK}$ ($\mathcal{CNMZK}$) considers both of the above attacks; the adversary may participate in an unbounded number of concurrent executions, playing the role of a prover in some, and the role of a verifier in others. Despite the generality of such an attacks scenario, this notion of security seems most appropriate for modeling the execution of cryptographic protocols in open networks, such as the Internet. Barak, Prabhakaran and Sahai (BPS) [BPS06] provided the the first $\mathcal{CNMZK}$

[*] Lin is supported by a Microsoft Research Fellowship.

[**] Pass is supported in part by a Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR Award FA9550-08-1-0197, BSF Grant 2006317 and I3P grant 2006CS-001-0000001-02.

argument for $\mathcal{NP}$ in the plain model (i.e., without any set-up assumptions); see also the more efficient instantiation of Ostrovsky, Pandey and Visconti [OPV10]. More recently, Lin, Pass, Tseng and Venkitasubramaniam (LPTV) [LPTV10] provided a somewhat different approach to constructing $\mathcal{CNMZK}$ protocols, improving the round-complexity of the BPS construction, as well as providing a construction of a $\mathcal{CNMZK}$ proof.

Adaptive inputs selection. All the above-mentioned feasibility results for $\mathcal{CNMZK}$, however, consider a quite restricted form of input selection: More precisely, whereas the attacker is allowed to adaptively select the statements it gives proofs of (on the right), the statements to receive proofs of (on the left) are assumed to be fixed before the execution begins.

Indeed, there is a sense in which this is necessary: as argued by Lindell [Lin03], if we consider a scenario where the left statement are chosen adaptively by an "environement" (think of this as some other arbitrary protocol running in the network), then the notion of $\mathcal{CNMZK}$ collapses down to the notion of *Universally Composable* $\mathcal{ZK}$ [Can01], which is known to be unachievable without set-up [CKL03].

We here focus on the simpler case of just "self-composition": that is, we only consider the security of the $\mathcal{ZK}$ protocols (and thus we do not allow them to interact with other protocols in the network; this is similar to the original setting studied in the context of Concurrent $\mathcal{ZK}$). Yet, we want to capture a notion of security where also the statements in the left executions are adaptively chosen. The natural way to do this is to (just as in the definition of security of signature schemes [GMR89]) allow the attacker to adaptively select the statements it wants to hear proofs of on the left (as well as the statements it gives proofs of on the right); additionally we must restrict the attacker to only ask to hear proofs of statements that are true (or else we can never expect the conversation with the provers to be $\mathcal{ZK}$—if the statement is false, then the prover on the left will not be able to provide a proof, which thus reveals information). However, one problem arises when statements in the left interactions are chosen adaptively: to provide a $\mathcal{ZK}$ proof of a statement, the prover (on the left) needs to receive a valid witness; however, it might be hard to compute valid witnesses for general languages in $\mathcal{NP}$. We can consider the following three possibilities:

– There is a non-uniform $\mathcal{PPT}$ machine, called the *witness-selecting machine*, such that, whenever the adversary chooses a statement to hear a proof of, the witness-selecting machine on input the statement, generates a witness for the prover. We call this weak adaptive input selection. Since the witness-selecting machine is computationally bounded, the adversary is implicitly restricted to only choose statements for which a valid witness can be computed efficiently. In essence, this case is similar to the static input selection case; instead of having all the witnesses fixed at the beginning of the execution, there is a fixed "master witness" (i.e., the non-uniform advise), from which all the witnesses can be generated efficiently. Not surprisingly, previous works on $\mathcal{CNMZK}$ with static input selection [BPS06,LPTV10] can already be extended to handle weak adaptive input selection.

- We can slightly strengthen the power of the witness-selecting machine by allowing it to also receive as input the view of the honest left prover (as well as the views of the adversary and right receiver). This further enables, for instance, scenarios where the adversary can request for a proof about the execution of other proof-instances in the execution (e.g., prove that the prover has behaved honestly in previous proofs).
- Finally, we may even model the witness-selecting machines to be computationally unbounded (in addition to receiving the views of all parties as input), which allows the adversary to adaptively choose any true statements to hear proofs of.

In this work, we consider adaptive input selection with the most powerful witness-selecting machine. We call a $\mathcal{ZK}$ protocol that is secure in this setting a $\mathcal{CNMZK}$ *with Adaptive Input Selection*, or for short *Adaptive* $\mathcal{CNMZK}$ ($\mathcal{ACNMZK}$). More precisely, a $\mathcal{ZK}$ protocol is $\mathcal{ACNMZK}$ if for every adversary A, there exists a computationally efficient simulator-extractor that can simulate both the left and the right interactions for A, while outputting a witness for every statement proved by the adversary in the right interactions. Our main result is the construction of an $\mathcal{ACNMZK}$ proof:

Theorem 1 *Assume the existence of collision-resistant hash functions. Then there exists a* $\omega(\log^2 n)$*-round concurrent non-malleable zero-knowledge proof with adaptive input selection (and with a black-box simulator) for all of* $\mathcal{NP}$*.*

Related Work. We mention that there are several works constructing $\mathcal{CNMZK}$ protocols in various trusted set-up models. For instance, [SCO$^+$01,CF01,DN02]) provide constructions of Universall Composable $\mathcal{ZK}$ in the Common Reference String (CRS) model; these protocol are thus also $\mathcal{ACNMZK}$.

An interesting recent work by Yao, Yao and Zhao [YYZ09] provide a construction of a $\mathcal{CNMZK}$ protocol in the Bare Public Key Model; their protocol is not UC secure but satisfies a notion CNMZK with adaptive inputs selection (for both the left and the right interaction); our definition of $\mathcal{ACNMZK}$ in the plain model is heavily inspired by their work.

Techniques. Our protocol is a close variant of the LPTV protocol; let us start by reviewing it. The protocol uses two main components. The first component is the notion of *concurrently extractable commitments* (CECom) introduced by Micciancio, Ong, Sahai, and Vadhan [MOSV06]. Informally, values committed to using a CECom can be extracted by a rewinding simulator even in the concurrent setting. In our protocol (as in most concurrent $\mathcal{ZK}$ protocols), the verifier commits to a random trapdoor using CECom, so that our $\mathcal{ZK}$ simulator may extract this trapdoor to perform simulation. The second component is the notion of *robust non-malleable commitments* (an extension due to Lin and Pass [LP09] of the notion of non-malleable commitments as defined by Dolev, Dwork, and Naor [DDN00]); roughly speaking these are non-malleable commitment schemes with an additional robustness property that makes them convienent to compose with other protocols.

The high-level idea behind the LPTV protocol (just as in the protocol of [BPS06]) is to start off with a *preamble phase* where the verifier uses a CECom to commit to a trapdoor; next in a *commit phase*, the prover commits to a witness of the proof statement using both a CECom and robust non-malleable commitments; and finally during a *proof phase*, the prover proves using a (stand-alone) $\mathcal{ZK}$ protocol that it has either committed to a valid witness, or a valid trapdoor in the commit phase. To prove security, LPTV provides a simulator that uses rewindings to extract out trapdoors (from the CECom in the preamble phase) to simulate the commit and proof phases of the left interactions, and uses rewindings again to extract the witnesses committed to by the adversary (from the CECom in the commit phase) on the right. The crux of the proof is then to show that even during the simulation, when the simulator commits to trapdoors (instead of real witnesses) in left interactions, the adversary still cannot commit to a trapdoor in right interactions, so that the values extracted out from the right interactions must be real witnesses. Very roughly speaking, this follows from the security guarantees of robust non-malleable commitments.

When considering adaptive input selection (for the left executions) a problem arises. First, proving indistinguishability of the simulation becomes problematic: in fact, getting a concurrent $\mathcal{ZK}$ protocol with adaptive input selection is already non-trivial (we call it Adaptive Concurrent Zero-Knowledge ($\mathcal{ACZK}$)); our core technical contribution is to provide a solution to this problem. The reason for this is that proving indistinguishability of the simulation requires performing a hybrid argument, where we switch the witness used in the left interactions from the trapdoors (used by the simulator) to the real witness (used by the prover). More precisely, we consider a hybrid H_i, where the first i left interactions are simulated using the trapdoors, and the later ones use the real witnesses. The problem is that the real witnesses might not be efficiently recoverable since the statements are chosen adaptively by the adversary (it is computed by a computationally unbounded witness-selecting machine); so the hybrid is not efficiently computable!

Our idea for circumventing this problem can be described as follows:

- First, we switch the order of the hybrids. We consider hybrids H_i where the first i left interactions are emulated using real witness and the later ones are simulated using trapdoors. The reason for doing this is that we can now non-uniformly fix the real witnesses of the first i left interactions by hardcoding the "prefix" of hybrid H_i before the i^{th} left interaction; and then the remaining execution can be efficiently emulated using the real witnesses.
- But now the obstacle is that arguing indistinguishability of H_i and H_{i+1} becomes problematic. To show indistinguishability we need to show that simulating the i^{th} left interaction using a real witness or trapdoor is indistinguishable (other interactions are simulated identically in the two hybrids). It seems that this should just follow from the hiding and $\mathcal{ZK}$ property of the commit and proof phases of the left interaction. However, the problem is that (when trying to extract the trapdoors of the latter left interactions), we might be rewindings the i^{th} left interaction. Our way around this problem is to add more CECom to the preamble phase; the idea is to show that there

exists some alternative simulator, that generates a statistically close distribution, but is able to avoid rewinding the messages in the commit and proof phases of the left interaction that we want to violate indistinguishability of.

To also complete the proof of non-malleability, a second (very related problem) arises: namely, we need to argue that the witness committed to by the adversary on the right are valid even in simulation; this is usually done through a hybrid argument as well and relies on the robust non-malleability of the commitment scheme used in the commit phase (instead of the hiding and $\mathcal{ZK}$ properties). When doing this, we again run into the same problem as when showing indistinguishability of the simulation. Here the issue is that we need to ensure that the robust non-malleability property holds even under rewindings. We use the same idea to overcome this problem: as long as there are sufficiently many CECom in the preamble phase, we can describe an alternative simulator that produces a statistically close distribution without rewinding these commitments that we want to violate robust non-malleability of.

Overview. Section 2 contains the basic notations and definitions of $\mathcal{ACNMZK}$ and other primitives. In Section 3, we present our main result, a $\omega(\log^2 n)$-round $\mathcal{ACNMZK}$ proof system for all of $\mathcal{NP}$, from collision resistant hash functions. In Section 4.2, we first focus on showing the $\mathcal{ACZK}$ property of the protocol, which contains the main technical content of this paper; then in Section 5 we sketch how to extend this proof to also show the $\mathcal{ACNMZK}$ property.

2 Preliminaries

Let N denote the set of all positive integers. For any integer $n \in N$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$, and let $\{0, 1\}^n$ denote the set of n-bit strings, and ε the empty string. We assume familiarity with interactive Turing machines, interactive protocols, statistical/computational indistinguishability, notions of interactive proofs, zero-knowledge, (strong) witness-indistinguishability, and notions of statistically binding/hiding commitments. (See [Gol01] for formal definitions.)

2.1 Adpative Concurrent Non-Malleable Zero-Knowledge

Our definition of adpative concurrent non-malleable zero-knowledge is very similar to that of concurrent non-malleable zero-knowledge from [BPS06] (which in turn closely follows the definition of simulation extractability of [PR05]), with the only difference that now the adversary is allowed to adaptively select the statements it receives proofs to, subject to that they are true statements.

Let $\langle P, V \rangle$ be an interactive proof for a language $L \in \mathcal{NP}$ with witness relation R_L, and let n be the security parameter. Consider a man-in-the-middle adversary A that participates in many left and right interactions in which $m = m(n)$ proofs take place. In the left interactions, the adversary A verifies the validity of statements $x_1, \ldots, x_m$ by interacting with an honest prover P, using identities $\mathsf{id}_1, \ldots, \mathsf{id}_m$. In the right interactions, A proves the validity of statements $\tilde{x}_1, \ldots, \tilde{x}_m$ to an honest verifier V, using identities $\tilde{\mathsf{id}}_1, \ldots, \tilde{\mathsf{id}}_m$. Prior to the interactions, all parties in the system receives as common input the security parameter in unary 1^n, and A receives as auxiliary input $z \in \{0,1\}^*$. Furthermore, at the beginning of each left (respectively right) interaction, the adversary adaptively selects the statement x_i (respectively $\tilde{x}_i$) and the identity id_i (respectively $\tilde{\mathsf{id}}_i$), with the only restriction that all the statements $x_1, \ldots, x_m$ chosen in the left interactions have to be true. Additionally in each left interaction, the prover P receives as local input a witness $w_i \in R_L(x_i)$, chosen adaptively by a *witness-selecting machine* M. More specifically, M is a (randomized) Turing machine that runs in *exponential* time, and whenever the adversary chooses a statement x_i for a left interaction, M on inputs the statement x_i and the current view of all parties (including the adversary, provers, and receivers), picks a witness $w_i \in R_L(x_i)$ as the private input of the prover P. Let $\mathsf{view}_{A,M}(n,z)$ denote a random variable that describes the view of A in the above experiment. Loosely speaking, an interactive proof is adaptive concurrent non-malleable zero-knowledge ($\mathcal{ACNMZK}$) if for all man-in-the-middle adversary A, there exists a probabilistic polynomial time machine (called the simulator-extractor) that can simulate both the left and the right interactions for A, while outputting a witness for every statement proved by the adversary in the right interactions.

Definition 1 *An interactive proof* (P,V) *for a language* L *with witness relation* R_L *is said to be* adaptive concurrent non-malleable zero-knowledge *if for every polynomial* m*, and every probabilistic polynomial-time man-in-the-middle adversary* A *that participates in at most* $m = m(n)$ *concurrent executions, there exists a probabilistic polynomial time machine* S*, such that, for every input-selecting machine* M*:*

1. *The following ensembles are computationally indistinguishable over* $n \in N$
   - $\{\mathsf{view}_{A,M}(n,z)\}_{n\in N, z\in\{0,1\}^*}$
   - $\{S_1(1^n,z)\}_{n\in N, z\in\{0,1\}^*}$

   where $S_1(1^n,z)$ *denotes the first output of* $S(1^n,z)$*.*
2. *Let* $z \in \{0,1\}^*$ *and* $(\mathsf{view}, \boldsymbol{w})$ *denote the output of* $S(1^n,z)$*. Let* $\tilde{x}_1, \ldots, \tilde{x}_m$ *be the statements of the right-interactions in* view*, and let* $\mathsf{id}_1, \ldots, \mathsf{id}_m$ *and* $\tilde{\mathsf{id}}_1, \ldots, \tilde{\mathsf{id}}_m$ *be the identities of the left-interactions and right-interactions in* view*. Then for every* $i \in [m]$*, if the* i^{th} *right-interaction is accepting and* $\tilde{\mathsf{id}}_i \neq \mathsf{id}_j$*,* $\boldsymbol{w}$ *contains a witness* w_i *such that* $R_L(\tilde{x}_i, w_i) = 1$*.*

We also consider concurrent $\mathcal{ZK}$ with adaptive input selection. We say that an interactive proof (P,V) is adaptive concurrent $\mathcal{ZK}$ ($\mathcal{ACZK}$) if it satisfies the above definition with respect to adversaries that only receive proofs (and do not give proofs).

Remark 1. As mentioned before, the security proof in [BPS06,LPTV10] can be extended to show that their constructions of $\mathcal{CNMZK}$ protocols satisfy a notion of $\mathcal{CNMZK}$ with *weak* adaptive input selection, where the adversary can only choose to hear proofs of statements for which a witness can be computed efficiently. Formally, the witness-selecting machine is restricted to be computationally bounded (i.e., a non-uniform $\mathcal{PPT}$) and only receives a statement as input (but not the views of the adversary, the left prover and right receiver[1]).

Remark 2. Universal Composability [Can01] considers a more generalized form of adaptive input selection, where both the statements and witnesses are chosen adaptively by a separate entity called the "environment", which may communicate with the adversary in an arbitrarily way. In contrast, our definition of $\mathcal{ACNMZK}$ only allows the witnesses to be selected by a separate entity, whereas the statements are chosen directly by the adversary. Unfortunately, it has been shown that Universal Composable $\mathcal{ZK}$ is unachievable without set-up [CKL03]. Our construction can actually satisfy a slight strengthening of the above definition of $\mathcal{ACNMZK}$, where the statements are adaptively chosen by a *stateless non-uniform* $\mathcal{PPT}$ machine—called the statement-selecting machine—that both the adversary and the simulator have oracle accesses to. This will bring us closer to the fully adaptive case of Universal Composability; we leave details to the full version.

Non-Malleable Commitment Schemes We recall the definition of non-malleability from [LPV08] (which builds upon the definition of [DDN00,PR05]). Let $\langle C, R\rangle$ be a tag-based statistically binding commitment scheme, and let $n \in N$ be a security parameter. Consider a man-in-the-middle adversary A that, on auxiliary inputs n and z, participates in one left and one right interaction simultaneously. In the left interaction, the man-in-the-middle adversary A interacts with C, receiving a commitment to value v, using identity id of its choice. In the right interaction A interacts with R attempting to commit to a related value $\tilde{v}$, again using identity $\tilde{\mathsf{id}}$ of its choice. If the right commitment is invalid, or undefined, its value is set to $\perp$. Furthermore, if $\tilde{\mathsf{id}} = \mathsf{id}$, $\tilde{v}$ is also set to $\perp$—i.e., a commitment where the adversary copies the identity of the left interaction is considered invalid. Let $\mathsf{nmc}^A_{\langle C,R\rangle}(v, z)$ denote a random variable that describes the value $\tilde{v}$ and the view of A, in the above experiment.

Definition 2 *A statistically binding commitment scheme $\langle C, R\rangle$ is said to be* non-malleable (with respect to itself) *if for every polynomial $p(\cdot)$, and every probabilistic polynomial-time man-in-the-middle adversary A, the following ensembles are computationally indistinguishable.*

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

Non-Malleable Commitment Robust w.r.t. *k*-round Protocols The notion of non-malleability w.r.t. arbitrary k-round protocols is introduced in [LP09]. Unlike traditional definitions of non-malleability, which only consider man-in-the

[1] A slight generalization would be allowing the witness-selecting machine to also see the view of the adversary, but not the views of the left prover and right receiver.

middle adversaries that participate in two (or more) executions of the *same* protocol, non-malleability w.r.t. arbitrary protocols considers a class of adversaries that can participate in a left interaction of any arbitrary protocol. Below we recall the definition. Consider a one-many man-in-the-middle adversary A that participates in one left interaction—communicating with a machine B—and one right interaction—acting as a commiter using the commitment scheme $\langle C, R\rangle$. As in the standard definition of non-malleability, A can adaptively choose the identity in the right interaction. We denote by $\mathsf{nmc}^{B,A}_{\langle C,R\rangle}(y,z)$ the random variable consisting of the view of $A(z)$ in a man-in-the-middle execution when communicating with $B(y)$ on the left and an honest receiver on the right, combined with the value $A(z)$ commits to on the right. Intuitively, we say that $\langle C, R\rangle$ is non-malleable w.r.t. B if $\mathsf{nmc}^{B,A}_{\langle C,R\rangle}(y_1,z)$ and $\mathsf{nmc}^{B,A}_{\langle C,R\rangle}(y_2,z)$ are indistinguishable, whenever interactions with $B(y_1)$ and $B(y_2)$ cannot be distinguished.

Definition 3 *Let B be a probabilistic polynomial time machine. We say the statistically binding commitment scheme $\langle C, R\rangle$ is* non-malleable w.r.t. B*, if for every probabilistic polynomial-time man-in-the-middle adversary A, and every two sequences $\{y^1_n\}_{n\in N}$ and $\{y^2_n\}_{n\in N}$ such that, for all probabilistic polynomial-time machine $\tilde{A}$, it holds that*

$$\left\{\langle B(y^1_n), \tilde{A}(z)\rangle(1^n)\right\}_{n\in \mathbf{N}, z\in\{0,1\}^*} \approx \left\{\langle B(y^2_n), \tilde{A}(z)\rangle(1^n)\right\}_{n\in \mathbf{N}, z\in\{0,1\}^*}$$

where $\langle B(y), \tilde{A}(z)\rangle(1^n)$ denotes the view of $\tilde{A}$ in interaction with B on common input 1^n, and private inputs z and y respectively, then it holds that:

$$\left\{\mathsf{nmc}^{B,A}_{\langle C,R\rangle}(y^1_n,z)\right\}_{n\in \mathbf{N}, z\in\{0,1\}^*} \approx \left\{\mathsf{nmc}^{B,A}_{\langle C,R\rangle}(y^2_n,z)\right\}_{n\in \mathbf{N}, z\in\{0,1\}^*}$$

We say that $\langle C, R\rangle$ is non-malleable w.r.t. k-round protocols if $\langle C, R\rangle$ is non-malleable w.r.t. any $\mathcal{PPT}$ machine B that interacts with the man-in-the-middle adversary in k rounds. Below, we focus on commitment schemes that are non-malleable w.r.t. itself and arbitrary $\ell(n)$-round protocols, where ℓ is a super-logarithmic function. We say that such a commitment scheme is robust w.r.t. $\ell(n)$-round protocols. It has been shown in [LPV08] that,

Lemma 1 ([LPV08]) *Let $\ell(n)$ be a super-logarithmic function. Then there exists a $O(\ell(n))$-round statistically binding commitment scheme that is robust w.r.t. $\ell(n)$-round protocols, assuming that one-way functions exist.*

Concurrently Extractable Commitment Schemes Micciancio, Ong, Sahai and Vadhan introduce and construct *concurrently extractable commitment schemes*, CECom, in [MOSV06]. The commitment scheme is an abstraction of the preamble stage of the concurrent zero-knowledge protocol of [PRS02]. Informally, values committed by CECom can be extracted by a rewinding extractor (e.g., the zero-knowledge simulator of [KP01,PRS02,PTV08]), even in the concurrent setting. In this work, we use the same construction as in [PRS02,MOSV06], but are unable to employ their analysis.

3 An $\mathcal{ACNMZK}$ Proof

In this section we construct an adaptive concurrent non-malleable zero-knowledge proof based on collision-resistant hash-functions. The construction is almost identical to the $\mathcal{CNMZK}$ proof system in [LPTV10], except that, the verifier is asked to provide more CECom commitments to its trapdoor at the beginning of the protocol, which, in the proof, facilitates the simulator-extractor to extract the trapdoor while strategically avoiding rewinding certain messages.

Let $\ell(n)$ be any super logarithmic function. Our adaptive concurrent non-malleable zero-knowledge protocol, $\mathsf{ACNMZKProof}$, employs several commitment protocols. Let Com_{sh} be a 2-round statistically *hiding* commitment (based on collision-resistant hash-functions), Com_{sb} be a 2-round statistically *binding* commitment (based on one-way functions), and NMCom be an $O(\ell(n))$-round statistically binding commitment scheme that is robust w.r.t. $\ell(n)$-round protocols (based on one-way functions).

Our protocol also employs $\ell(n)$-round statistically hiding (respectively statistically binding) concurrently-extractable commitment schemes, CECom_{sh} (respectively CECom_{sb}). These schemes are essentially instantiations of the PRS preamble [PRS02], and can be constructed given Com_{sh} and Com_{sb}. Below we repeat their definitions.

To commit a n-bit string v, the commiter chooses $n \times \ell(n)$ pairs of random n-bit strings $(\alpha^0_{i,j}, \alpha^1_{i,j}), i \in [n], j \in [\ell(n)]$, such that $\alpha^0_{i,j} \oplus \alpha^1_{i,j} = v$ for every i and j. The sender then commits to v and each of the $2n\ell(n)$ strings in parallel using Com_{sh}. This is followed by $\ell(n)$ rounds of interactions. In the j^{th} interaction, the receiver sends a random n-bit challenge $b_j = b_{1,j} \ldots b_{n,j}$, and the commiter decommits the commitments of $\alpha^{b_{1,j}}_{1,j}, \ldots, \alpha^{b_{n,j}}_{n,j}$ according to the challenge.

A valid decommitment of CECom_{sh} requires the commiter to decommit all initial commitments under scheme Com_{sh} (i.e., reveal the randomness of the commitments), and that the decommitted values satisfy $\alpha^0_{i,j} \oplus \alpha^1_{i,j} = v$ for every i and j.

A $\ell(n)$-round statistically binding concurrently-extractable commitment scheme, CECom_{sh}, is defined analogously as CECom_{sh} with the initial commitment Com_{sh} replaced by Com_{sb}. Additionally, we say a transcript of CECom_{sh} is *valid* if there exists a valid decommitment.

We now describe $\mathsf{ACNMZKProof}$, our adaptive concurrent non-malleable zero-knowledge protocol. Protocol $\mathsf{ACNMZKProof}$ for a language $L \in \mathcal{NP}$ proceeds in six stages given a security parameter n, a common input statement $x \in \{0,1\}^n$, an identity id, and a private input $w \in R_L(x)$ to the Prover.

Stage 1: The Verifier chooses a random string $r \in \{0,1\}^n$ and commits to r using $k(n)+1$ invocations of CECom_{sh}, where $k(n)$ is the number of rounds in Stage 2-6 of the protocol; r is called the "fake witness".

Stage 2: The Prover commits to the witness w using CECom_{sb}.

Stage 3: The Prover commits to the witness w using NMCom with identity id.

Stage 4: The Prover commits to the witness w using NMCom with identity id, again.

Stage 5: The Verifier decommits the Stage 1 commitment to value r'.

Stage 6: The Prover using a $\omega(1)$-round $\mathcal{ZK}$ proof (e.g., [Blu86]), proves that the commitments in Stages 2, 3 and 4 all commit to the same value $\tilde{w}$ (with identity id), and that either $\tilde{w} \in R_L(x)$ or $\tilde{w} = r'$.

A formal description of the protocol can be found in Figure 1.

On Round Complexity: Since the protocol NMCom has $O(\ell(n))$ rounds, we have that $k(n) = O(\ell(n))$. Therefore, the round complexity of the protocol $\mathsf{ACNMZKProof}$ is $O(\ell^2(n)) = \omega(\log^2 n)$.

The above protocol is an extension of the Goldreich-Kahan protocol [GK96]. Completeness and Soundness follows using stand techniques; since the protocol is essentially the same as the $\mathcal{CNMZK}$ protocol in [LPTV10] (except that Stage 1 now contains many CECom's), we refer the reader to [LPTV10] for more details.

Protocol ACNMZKProof

Common Input: an instance x of a language L with witness relation R_L, an identifier id, and a security parameter n.

Auxiliary Input for Prover: a witness w, such that $(x, w) \in R_L(x)$.

Stage 1:

V uniformly chooses $r \in \{0,1\}^n$ (the "fake witness").

V commits to r using $k+1$ invocations of the protocol CECom_{sh}, where k is the number of rounds in Stage 2-6 of the protocol. Let $\mathcal{T}_1$ be the commitment transcript.

Stage 2:

P commits to w using protocol CECom_{sb}. Let $\mathcal{T}_2$ be the commitment transcript.

Stage 3:

P commits to w using protocol NMCom and identity id. Let $\mathcal{T}_3$ be the commitment transcript.

Stage 4:

P commits to w using protocol NMCom and identity id. Let $\mathcal{T}_4$ be the commitment transcript.

Stage 5:

V decommits $\mathcal{T}_1$ to value r; P aborts if no valid decommitment is given.

Stage 6:

P $\leftrightarrow$ V: a $\omega(1)$-round $\mathcal{ZK}$ proof [Blu86] of the statement: There exists $\tilde{w}$ such that

– $\tilde{w}$ is a valid decommitment of $\mathcal{T}_2$,

– *and* $\tilde{w}$ is a valid decommitment of $\mathcal{T}_3$ and $\mathcal{T}_4$ under identity id,

– *and* $\tilde{w} \in R_L(x)$ or $\tilde{w} = r$.

Fig. 1. An Adaptive Concurrent Non-Malleable $\mathcal{ZK}$ Proof for $\mathcal{NP}$

4 Proof of Security

The definition of $\mathcal{ACNMZK}$ requires a simulator-extractor S that is able to simulate the view of a man-in-the-middle adversary A (including both left and right interactions), while simultaneously extracting the witnesses to statements proved in the right interactions. We describe the construction of our simulator in Section 4.1, show that it is a correct $\mathcal{ACZK}$ simulator in Section 4.2, and extend this proof to show the $\mathcal{ACNMZK}$ property in Section 5.

4.1 Our Simulator-Extractor

Our simulator-extractor, S, is almost identical to the simulator extractor of the $\mathcal{CNMZK}$ protocol in [LPTV10], except that now, given more CECom_{sh}'s in Stage 1 of the protocol, S tries to extract a "fake" witness from every CECom_{sh} from the adversary in the left interactions, and aborts if the extraction fails for any of the commitment or the extracted value does not equal to the value that the adversary decommitment to later. Roughly speaking, S follows this strategy:

Simulating the view of the right interactions. S simply follows the honest verifier strategy.

Simulating the view of the left interactions. In each protocol execution, S first extracts the "fake witness" r from the $k(n)+1$ CECom_{sh}'s committed by A in Stage 1, then commits to r in Stage 2, 3, and 4, and finally simulates the $\mathcal{ZK}$ proof using r as a witness in Stage 6.

Extracting the witnesses. In each right interaction that completes successfully, S extracts a witness w from CECom_{sb} committed by A in Stage 2 of the protocol.

Thus, the main task of S is to extract the values committed by A, using CECom, in Stage 1 and 2 of the protocol. This is done by rewinding A during each CECom. To that end, we employ the "lazy KP" simulator of [PTV08], an oblivious simulator that is nearly identical to the Killian-Petrank (KP) simulator [KP01]. We also follow the analysis of [PTV08], which is in turn based on the analysis of [PRS02].

On a very high-level, S attempts to simulate the view of A (with "fake witnesses") in one continuous, straight-line manner (so as to not skew the output distribution); this is aided by numerous auxiliary rewinds that allows S to extract the "fake witnesses" in time. As implied by our simulation strategy, the view of A generated by S depends on the extracted "fake witnesses", but is otherwise independent of the interaction in auxiliary rewinds. (The simulator S is essentially identical to the simualtor of the $\mathcal{CNMZK}$ protocol in [LPTV10]; we refer the reader to [LPTV10] for a more detailed description.)

It is useful to know that S may abort in two manners. At the end of a CECom, if S is unable to extract the committed value (the rewinds were unhelpful), S outputs $\perp_{ext}$. Or, in Stage 5 of a left interaction, if A decommits its Stage 1 CECom_{sh}'s to a value that is different from any of the $k(n)+1$ extracted values, S outputs $\perp_{bind}$. Conversely if S does not abort, then it must have extracted the committed value from every Stage 1 CECom_{sh} that it has encountered, and A must decommit to the extracted values (if A decommits at all). The following claim bounds the abort probability of S.

Claim 2 *S outputs $\perp_{ext}$ and $\perp_{bind}$ with negligible probability.*

The proof is identical to the proof of Claim 2 in [LPTV10], which in turn follow directly from the analysis of [PTV08] in the setting of concurrent $\mathcal{ZK}$; we refer the reader to [LPTV10] for a formal proof.

4.2 Proof of $\mathcal{ACZK}$

We first show that S is a valid $\mathcal{ACZK}$ simulator for the protocol $\mathsf{ACNMZKProof}$, that is, the view generated by S is indistinguishable from the real view of A.

Lemma 3 *For every witness-selecting machine M, the following ensembles are computationally indistinguishable over $n \in \mathsf{N}$:*

$$\{S_1(1^n, z)\}_{n \in \mathsf{N}, z \in \{0,1\}^*}$$
$$\{\mathsf{view}_{A,M}(1^n, z)\}_{n \in \mathsf{N}, z \in \{0,1\}^*}$$

To show Lemma 3, we introduce a series of hybrid simulators; the same hybrid simulators will also be helpful later in showing the $\mathcal{ACNMZK}$ property in Section 5. Hybrids hyb^i, $0 \le i \le m+1$ proceed in three steps.

Real Excution Phase: Run the honest man-in-the-middle execution with A until the i^{th} left interaction starts: in left iteration $j < i$, run the witness-selecting machine M (on input the statement x_j of this interaction, and the current view of the adversary, prover and verifier) to compute a valid witnesses w_j and execute the honest prover strategy. Note that the Real Exeuciton Phase may take exponential time. Let $\mathcal{V}_A$ be the view of A, and $\mathcal{V}_P$, $\mathcal{V}_V$ the view of the prover and verifier produced in this phase.

Simulation Phase: Feed A with $\mathcal{V}_A$. Run the following simulation strategy with A to complete the partial execution defined by $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$.

- For every right interaction, emulate the interaction by following the honest verifier strategy from $\mathcal{V}_V$.
- For left interaction $j < i$, emulate the interaction by following the honest prover strategy from $\mathcal{V}_P$.
- For left interaction $j \ge i$, simulate the interaction using a "fake" witness, as S does.

Formally, this simulation strategy can be implemented as follows: construct another machine A' that internally incorporates A and simulates the first $i-1$ left and all right interactions for A honestly from $\mathcal{V}_P$ and $\mathcal{V}_V$, and forwards the rest $m-i+1$ left interactions externally. Then simply run S on A' and outputs the embedded view of A in the view of A' produced by S.

Output Phase: Output $\perp_{ext}$ or $\perp_{bind}$ if S returns $\perp_{ext}$ or $\perp_{bind}$; otherwise, output the view $\mathcal{V}$ of A embeded in the view of A' produced by S.

We also define hybrids hyb^i_+ that proceed identically to hyb^i except that, in the Simulation Phase, the i^{th} left interaction is simulated using a real witness (rather than the "fake" witness). This can be done as the Real Execution Phase runs till the i^{th} left interaction starts, and can also compute the real witness of the i^{th} interaction. Note that these hybrids $\{\mathsf{hyb}^i\}$, $\{\mathsf{hyb}^i_+\}$ are only concerned with producing a view of A, and do not extract the witnesses of the right interactions.

By construction, hyb^i and hyb^i_+ abort only when S aborts. Hence by Claim 2, we have,

Claim 4 *For all i, hyb^i and hyb^i_+ output $\bot$ with negligible probability.*

By Claim 4, the output of hyb^1 is statistically close to the output of S running with A in its entirety. (They only differ when S aborts due to trying to extract witnesses of the right interactions from the CECom_{sb}'s committed by A.) The output of hyb^{m+1}, on the other hand, is identical to the real view of A. Therefore Lemma 3 directly follows from the next two claims:

Lemma 5 *The outputs of hyb^i_+ and hyb^{i+1} are statistically close.*

Proof. Ignoring the fact that hyb^i_+ and hyb^{i+1} may abort, their outputs are identical. This is because hyb^i_+ differs from hyb^{i+1} only in that when generating the output view, from the beginning of the i^{th} left interaction until the beginning of the $i + 1^{\text{st}}$ left interactions, hyb^i_+ employs *rewinds*. However, these rewinds do not extract any new "fake witnesses" for use in the output view, and do not skew the output distribution because the rewinding schedule (including which rewind determines the output view) is oblivious. Since both machines abort at most with negligible probability by Claim 4, their outputs are statistically close.

Lemma 6 *The outputs of hyb^i and hyb^i_+ are computationally indistinguishable.*

Proof. Assume for contradiction that there exists an adversary A and a polynomial p, such that, for infinitely many $n \in N$, hyb^i and hyb^i_+ are distinguishable with probability $1/p(n)$. Towards reaching a contradiction, note that hyb^i and hyb^i_+ differ only in how the i^{th} left interaction is simulated (fake or real witness) in the rewindings. We thus want to violate the computational hiding property of Stage 2-4 of the protocol, or the strongly witness-indistinguishable property (implied by the $\mathcal{ZK}$ property) of Stage 6. However, two problems arise: (1) the Real Execution Phase of the two hybrids takes exponential steps, and (2) Stage 2-6 of the i^{th} left interaction maybe *rewound* by the simualtor S. Fix a $n \in N$ for which our hypothesis holds. To overcome the first problem, by our hypothesis, there must exist an execution of the Real Execution Phase—defined by the views of the adversary $\mathcal{V}_A$, the left prover $\mathcal{V}_P$ and the right verifier $\mathcal{V}_V$ produced in this phase—such that, conditioned on $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$ occurring in the two hybrids, hyb^i and hyb^i_+ are still distinguishable with probability $1/p(n)$. Given $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$, the rest of the hybrids (i.e., the Simulation Phase and Output Phase) can be generated efficiently.

Now it only remains to handle the second problem, that is, Stage 2-6 of the i^{th} left interaction may be rewound in the Simulation Phase. We consider another two hybrids $\tilde{\mathsf{hyb}}^i$ and $\tilde{\mathsf{hyb}}^i_+$, which proceed identically to hyb^i and hyb^i_+

respectively, except that, in the Simulation Phase, they employ the following alternative simulation strategy that avoids rewinding Stage 2-6 of the i^{th} left interactions.

The alternative simulation strategy of $\tilde{\mathsf{hyb}}^i$: The goal this simulation strategy is to complete the partial execution $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$ produced by the Real Execution Phase, without rewinding Stage 2-6 of the i^{th} left interaction. Let $\{m_1, \ldots, m_t\}$ for $t = k(n)/2$, be the messages that A sends in Stage 2-6 of the i^{th} left interaction; and a_i the reply to m_i from the left prover. Then the execution of A continuing from $\mathcal{V}_A$ is "equivalent" to the sequential execution of the following $t+1$ machines $\tilde{A}_1, \ldots, \tilde{A}_{t+1}$.

Machine A_i on input a partial view $\mathcal{V}_{i-1}$ of A up until the message m_{i-1} is sent and the reply a_{i-1} ($\mathcal{V}_0 = \mathcal{V}_A$ and $a_0 = \varepsilon$), continue the execution of A from $\mathcal{V}_{i-1}$, by feeding $\mathcal{V}_{i-1}$ and a_{i-1} to A, and forwarding every message from A externally; finally, it aborts when A terminates or sends the message m_i, and output the newly generated view $\mathcal{V}_i$ of A.

The alternative simulation strategy, instead of producing a simulated view of A "in one shot", produces the view "progressively" by simulating the view of $A_1, \ldots, A_{t+1}$ in sequence. Furthermore, the simulation strategy remembers all the "fake witnesses" it has extracted so far, and to simulate the view of A_i, it can use the "fake" witnesses extracted when simulating the views of A_j's with $j < i$. More precisely, let $\mathcal{S}$ (initialized to empty set) denote the set of "fake witnesses" extracted so far; let $(\mathcal{V}_A^j, \mathcal{V}_P^j, \mathcal{V}_V^j)$ for $j \in [t+1]$, and $(\mathcal{V}_A^0, \mathcal{V}_P^0, \mathcal{V}_V^0) = (\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$ the partial execution with A that is produced after j steps. In step $j \in [t+1]$,

1. Simulate the view of A_j continuing from $(\mathcal{V}_A^{j-1}, \mathcal{V}_P^{j-1}, \mathcal{V}_V^{j-1})$ as in hyb^i—that is, emulate the first $i-1$ left and all the right interactions honestly from $\mathcal{V}_P^{j-1}$ and $\mathcal{V}_V^{j-1}$, and simulate the rest $m-i+1$ left interactions using "fake" witnesses—except that now the "fake" witnesses can be obtained through extracting from some CECom_{sh}'s in this step, or in previous steps, found in $\mathcal{S}$. (Output $\perp_{ext}$ if no such fake witness is available, and $\perp_{bind}$, if A_i decommits to a value different from any of the "fake" witnesses extracted.)
2. Set $\mathcal{V}_A^j$ to the view of A embedded in the simulated view of A_i (set $\mathcal{V}_P^j$ and $\mathcal{V}_V^j$ appropriately as well); add all the "fake" witnesses extracted in this step to $\mathcal{S}$.

Finally, $\tilde{\mathsf{hyb}}^i$ outputs $\mathcal{V} = \mathcal{V}_A^{t+1}$.

We remark that in step j, the only message that A_j receives belonging to Stage 2-6 of the i^{th} left is a_{j-1}. This is because A in A_j starts its execution from $\mathcal{V}_A^{j-1}$, after messages m_1 to m_{j-1} are sent, and is cutoff immediately after m_j is sent. Therefore, during the simulation with A_i, in every rewinding, A never sends m_1 to m_{j-1} again, and never receives a reply to m_j (as every time it does send m_j, it is cutoff immediately). Hence the only message it receive is a_{j-1}. Therefore, overall, the alternative simulation strategy never rewinds Stage 2-6 of the i^{th} left interaction.

The alternative simulation strategy of $\tilde{\mathsf{hyb}}_+^i$: Define $\tilde{\mathsf{hyb}}_+^i$ analogously for

hyb^i_+. $\widetilde{\mathsf{hyb}}^i_+$ proceeds identically to $\widetilde{\mathsf{hyb}}^i$, except that in the simulation with A_j's, messages in Stage 2-6 of the i^{th} left interaction are emulated using the real witness (as in hyb^i_+). As $\widetilde{\mathsf{hyb}}^i$, $\widetilde{\mathsf{hyb}}^i_+$ never rewinds Stage 2-6 of the i^{th} left interaction.

Claim 7 *For all i, $\widetilde{\mathsf{hyb}}^i$ and $\widetilde{\mathsf{hyb}}^i_+$ output $\perp$ with negligible probability.*

Proof. It essentially follows from Claim 4 that the probabilities that $\widetilde{\mathsf{hyb}}^i$ and $\widetilde{\mathsf{hyb}}$ outputs $\perp_{bind}$ are negligible.

On the other hand, $\widetilde{\mathsf{hyb}}^i$ (respectively $\widetilde{\mathsf{hyb}}^i_+$) outputs $\perp_{ext}$ only if it fails to extract a "fake" witness for some left interaction $j \geq i$ (respectively $j > i$). Fix one such j, since left interaction j starts completely after $\mathcal{V}_A$ (the view generated in the Real Execution Phase), the execution of this interaction occurs completely inside machines $A_1, \ldots, A_{t+1}$, where $t = k/2$. Then since the number of CECom_{sh}'s in Stage 1 of the left interaction is $k+1 > t+1$, there exists a machine $A_{j'}$, such that, during its execution, a complete CECom_{sh} from A is sent. Then in Step j' of $\widetilde{\mathsf{hyb}}^i$ (respectively $\widetilde{\mathsf{hyb}}^i_+$), the alternative simulation strategy must try to extract a "fake" witness from this CECom_{sh}, and by Claim 4, it succeeds except with negligible probability. Therefore, by union bound, the probability that $\widetilde{\mathsf{hyb}}^i$ (respectively $\widetilde{\mathsf{hyb}}^i_+$) outputs $\perp_{ext}$ is negligible.

Furthermore, ignoring the fact that $\widetilde{\mathsf{hyb}}^i$ and hyb^i (respectively $\widetilde{\mathsf{hyb}}^i_+$ and hyb^i_+) may abort, their outputs are identical, since the views of A in $\widetilde{\mathsf{hyb}}^i$ and hyb^i are simulated identically. Therefore,

Claim 8 *For all i, it holds that,*

- *the outputs of $\widetilde{\mathsf{hyb}}^i$ and hyb^i are statistically close, and*
- *the outputs of $\widetilde{\mathsf{hyb}}^i_+$ and hyb^i_+ are statistically close.*

Combining Claim 8 with our hypothesis, we have that conditioned on $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$ occurring in the two hybrids, $\widetilde{\mathsf{hyb}}^i$ and $\widetilde{\mathsf{hyb}}^i_+$ are distinguishable with probability at least $1/2p(n)$. Note that continuing from $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$ the rest of the two hybrids can be efficiently generated, and the only difference between the two hybrids lies in how Stage 2-6 of the i^{th} left interaction are simulated (using a fake or a real witness), which are *never rewound* in the two hybrids. Then it follows directly from the computational hiding property of Stage 2-4, and the strongly witness-indistinguishable property (implied by the $\mathcal{ZK}$ property) of Stage 6 that conditioned on $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$, $\widetilde{\mathsf{hyb}}^i$ and $\widetilde{\mathsf{hyb}}^i_+$ are indistinguishable. This gives a contradiction.

Remark 3. Note that Claim 4 is crucial to the analysis of the hybrids. The analysis of [PRS02,MOSV06] can only realize Claim 4 for *committed-verifier* protocols. Since $\mathsf{ACNMZKProof}$ is not committed-verifier, we instead turn to the analysis of [PTV08]. Alternatively, it seems we can also utilize the analysis of [KP01], at the cost of $O(\log^2 n)$ round complexity.

5 Proof of $\mathcal{ACNMZK}$

As shown in the last section, the simulator constructed in Section 4.1 is a correct $\mathcal{ACZK}$ simulator, that is, the first output of S (i.e., $\mathsf{view} = S_1(1^n, z)$) is computationally indistinguishable from the real view of the adversary. To further show that S is also a correct $\mathcal{ACNMZK}$ simulator-extractor, it remains to show that the second output of S contains the valid $\mathcal{NP}$ witnesses of the statements proved in the right interactions (in view).

By construction, the witnesses that S outputs are just values it extracts out from the CECom_{sb}'s in Stage 2 of the right interactions. Therefore, if A always commits to valid witnesses using CECom_{sb} in the right interactions, by Claim 2 the simulator S would extract the valid witnesses except with negligible probability. Therefore, the following lemma establishes the correctness of the output witnesses:

Lemma 9 *For every $\mathcal{PPT}$ adversary A, there exists a negligible function ν, such that for every $n \in N$ and $z \in \{0,1\}^*$, the probability that A fails to commit to a valid witness in Stage 2 of a right interaction that is accepting and uses a different identity from all left interactions in* $\mathsf{view} = S_1(1^n, z)$*, is less than $\nu(n)$.*

Proof. Assume for contradiction that there exists a man-in-the-middle adversary A that participates in $m = m(n)$ left and right interactions, and a polynomial function p, such that for infinitely many $n \in N$ and $z \in \{0,1\}^*$, A *cheats* in an outcome of $S_1(1^n, z)$ with probability $1/p(n)$; by cheating, we mean that A fails to commit to a valid witness in Stage 2 of any right interaction that is accepting and uses a different identity from all the left interactions. (Note that A is not considered cheating if the simulator fails to output a view of A).

Consider again the series of hybrids, hyb^i and hyb^i_+, defined in section 4.2. Since the output of hyb^1 is statistically close to the output of S, by our hypothesis, the probability that A cheats in hyb^1 is non-negligible. On the other hand, in hyb^{m+1}, it follows from the soundness of Stage 6 that, except with negligible probability, in every accepting right interaction, A commits to either a real or a "fake" witness; it further follows from the statistically hiding property of Stage 1 and the (stand-alone) extractability of Stage 2 that, except with negligible probability, A never commits to a "fake" witness in any accepting right interactions. Hence, by union bound, except with negligible probability, A never cheats in hyb^{m+1}. It follows from Claim 6 that the probabilities of A cheating in hyb^i_+ and hyb^{i+1} differ by at most a negligible amount. Therefore, for infinitely many n, there must exist an $i = i(n)$, such that, the probabilities of A cheating in hyb^i and hyb^i_+ differ by at least a polynomial amount. Since the total number of right interactions is bounded by a polynomial, this implies that the probabilities that A cheats in a *randomly chosen* right interaction in the two hybrids differ by a polynomial amount.

Notice that the hybrids hyb^i and hyb^i_+ proceed identically up until the i^{th} left interaction starts. After that, the only difference between the two experiments lies in how the i^{th} left interaction is simulated (using either the fake or real witnesses). Towards reaching a contradiction, we want to claim that, by the non-malleability and $\ell(n)$-robustness of NMCom, the value A commits to in a randomly chosen right interaction is "computationally independent" from how Stage 2-6 of the i^{th} left interaction are simulated. However, (as in proof of Lemma 5) two problems arise: one is that the Real Execution Phase of the two hybrids can not be generated efficiently, and the other is that both Stage 2-6 of the i^{th} left and the randomly chosen right interactions might be rewound by S. We solve the two problem in the same way as in proof of Lemma 5: to overcome the first problem, we fix one execution of the Real Execution Phase $(\mathcal{V}_A, \mathcal{V}_P, \mathcal{V}_V)$ such that conditioned on it occurring, the two hybrids are still distinguishable with high probability; to overcome the second problem, we again consider two alternative hybrids $\hat{\mathsf{hyb}}^i$ and $\hat{\mathsf{hyb}}^i_+$, which proceed identically to hyb^i and hyb^i_+ respectively, except that, in the Simulation Phase, they employ an alternative simulation strategy that avoids rewinding Stage 2-6 of the i^{th} left and the randomly picked right interactions. More precisely, $\hat{\mathsf{hyb}}^i$ and $\hat{\mathsf{hyb}}^i_+$ proceed almost identically to $\tilde{\mathsf{hyb}}^i$ and $\tilde{\mathsf{hyb}}^i_+$ in the proof of Lemma 5, except that now it "chops" up the execution of A into $k(n) + 1$ phases $A_1, \ldots, A_{k+1}$, according to messages in Stage 2-6 of the i^{th} left and the randomly picked right interactions, and simualtes the views of $A_1, \ldots, A_{k+1}$ sequentially. It follows using the same argument that the outputs of $\hat{\mathsf{hyb}}^i$ and hyb^i, as well as that of $\hat{\mathsf{hyb}}^i_+$ and hyb^i_+, are statistically close.

Therefore by our hypothesis, the probabilities that A cheats in a *randomly chosen* right interaction in $\hat{\mathsf{hyb}}^i$ and $\hat{\mathsf{hyb}}^i_+$ differ by a polynomial amount. However, the only difference between the two hybrids lies in how Stage 2-6 of the i^{th} left interaction are simulated (using a fake or a real witness), AND the Stage 2-6 of the i^{th} left and the randomly chosen right interactions *are never rewound* in the two hybrids. Then it follows using the same proof of Lemma 7 in [LPTV10] that, essentially by the non-malleability and $\ell(n)$-robustness of NMCom that the probability that A commits to a "fake" witness in Stage 2 of the randomly chosen right interaction differ by at most a negligible amount, which gives a contradiction.

References

[Blu86] M. Blum. How to prove a theorem so no one else can claim it. *Proc. of the International Congress of Mathematicians*, pages 1444–1451, 1986.

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *FOCS*, pages 345–354, 2006.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 136, Washington, DC, USA, 2001. IEEE Computer Society.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *CRYPTO '01*, pages 19–40, 2001.

[CKL03] Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In *EUROCRYPT*, pages 68–86, 2003.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DN02] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *CRYPTO*, pages 581–596, 2002.

[DNS04] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6):851–898, 2004.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 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.

[Gol01] Oded Goldreich. *Foundations of Cryptography — Basic Tools*. Cambridge University Press, 2001.

[KP01] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in poly-loalgorithm rounds. In *STOC '01*, pages 560–569, 2001.

[Lin03] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *STOC '03*, pages 683–692, 2003.

[LP09] Huijia Lin and Rafael Pass. Non-malleability amplification. In *STOC '09*, pages 189–198, 2009.

[LPTV10] Huijia Lin, Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable zero knowledge proofs. In *CRYPTO*, pages 429–446, 2010.

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In *TCC '08*, pages 571–588, 2008.

[MOSV06] Daniele Micciancio, Shien Jin J Ong, Amit Sahai, and Salil Vadhan. Concurrent zero knowledge without complexity assumptions. *TCC '06*, pages 1–20, 2006.

[OPV10] Rafail Ostrovsky, Omkant Pandey, and Ivan Visconti. Efficiency preserving transformations for concurrent non-malleable zero knowledge. In *TCC*, pages 535–552, 2010.

[PR05] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *STOC '05*, pages 533–542, 2005.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *FOCS '02*, pages 366–375, 2002.

[PTV08] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishan Venkitasubramaniam. Concurrent zero knowledge: Simplifications and generalizations. Manuscript, 2008. `http://hdl.handle.net/1813/10772`.

[SCO$^+$01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO*, pages 566–598, 2001.

[YYZ09] Andrew Chi-Chih Yao, Moti Yung, and Yunlei Zhao. Adaptive concurrent non-malleability with bare public-keys. *CoRR*, abs/0910.3282, 2009.