Concurrent Non-Malleable Zero Knowledge Proofs

Huijia Lin[⋆], Rafael Pass[⋆⋆], Wei-Lung Dustin Tseng[⋆⋆⋆], and
Muthuramakrishnan Venkitasubramaniam

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

Abstract. Concurrent non-malleable zero-knowledge (NMZK) considers the concurrent execution of zero-knowledge protocols in a setting where the attacker can simultaneously corrupt multiple provers and verifiers. Barak, Prabhakaran and Sahai (FOCS'06) recently provided the first construction of a concurrent NMZK protocol without any set-up assumptions. Their protocol, however, is only computationally sound (a.k.a., a concurrent NMZK *argument*). In this work we present the first construction of a concurrent NMZK *proof* without any set-up assumptions. Our protocol requires poly(n) rounds assuming one-way functions, or $\tilde{O}(\log n)$ rounds assuming collision-resistant hash functions.
As an additional contribution, we improve the round complexity of concurrent NMZK arguments based on one-way functions (from poly(n) to $\tilde{O}(\log n)$), and achieve a near linear (instead of cubic) security reductions. Taken together, our results close the gap between concurrent ZK protocols and concurrent NMZK protocols (in terms of feasibility, round complexity, hardness assumptions, and tightness of the security reduction).

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,

[⋆] Supported in part by a Microsoft Research PhD Fellowship.

[⋆⋆] Supported in part by a Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR Award FA9550-08-1-0197 and BSF Grant 2006317.

[⋆⋆⋆] Supported in part by a NSF Graduate Research Fellowship.

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] recently constructed the first $\mathcal{CNMZK}$ protocol for $\mathcal{NP}$ in the plain model (i.e., without any set-up assumptions).[1] They provide a $\text{poly}(n)$-round construction based on one-way functions, and a $\tilde{O}(\log n)$-round construction based on collision-resistant hash-functions. Their constructions, however, are only computationally sound; that is, they only show the existence of $\mathcal{CNMZK}$ interactive *arguments* (as defined by [BCC88]). In contrast, for both concurrent $\mathcal{ZK}$ and non-malleable $\mathcal{ZK}$, interactive *proofs* (as originally defined by [GMR89]) are known [RK99,KP01,PRS02,DDN00].

Main result. In this work, we provide the first construction of a $\mathcal{CNMZK}$ proof in the plain model.[2]

Theorem 1 *Assume the existence of one-way functions. Then there exists a* $\text{poly}(n)$*-round concurrent non-malleable zero-knowledge proof (with a black-box simulator) for all of* $\mathcal{NP}$*. Furthermore, assuming the existence of collision-resistant hash-functions, the round complexity is only* $\tilde{O}(\log n)$.

Due to the $\tilde{\Omega}(\log n)$-round lower bound for black-box concurrent $\mathcal{ZK}$ of [CKPR01], the round complexity of our construction based on collision-resistant hash-functions is essentially optimal (unless $\mathcal{NP} \subseteq \mathcal{BPP}$).

Efficiency improvements. As an additional contribution, we improve the round-complexity of $\mathcal{CNMZK}$ arguments based on one-way functions (recall that the BPS protocol requires $\text{poly}(n)$ rounds).

Theorem 2 *Assume the existence of one-way functions. Then there is a* $\tilde{O}(\log n)$*-round concurrent non-malleable zero-knowledge argument (with a black-box simulator) for all of* $\mathcal{NP}$.

Combined with the black-box lower bounds of [CKPR01], this settles the round-complexity of $\mathcal{CNMZK}$ arguments based on minimal assumptions.

Finally, whereas the "knowledge security" [GMW91] of the BPS reduction (i.e., the overhead of the simulator w.r.t. to the adversary) is cubic, our analysis

[1] See also the more efficient construction of [OPV10].

[2] We mention that there are several works constructing $\mathcal{CNMZK}$ proofs in the Common Reference String (CRS) model (see e.g., [SCO$^+$01,DN02]). A potential approach for getting $\mathcal{CNMZK}$ proofs in the plain model would thus be to try to implement the CRS in a way that prevents man-in-the-middle attacks. This task seems harder than constructing $\mathcal{CNMZK}$ proofs from scratch, so we have not pursued this approach.

(for both proofs and arguments) achieves a near linear security reduction; in fact, our protocols achieve the stronger notion of *precise zero-knowledge* [MP06] which bounds the overhead of the simulator in an execution-by-execution fashion (as opposed to only bounding the worst-case running time), and achieve the same level of security as the best concurrent $\mathcal{ZK}$ protocols [PPS$^+$08].

Techniques. Our protocol attempts to combine previous techniques in concurrent and non-malleable $\mathcal{ZK}$ in a modular way. As a result, our $\mathcal{CNMZK}$ protocol largely consists of sub-protocols, more precisely commitments, that are developed in previous works.

To leverage existing techniques for concurrent $\mathcal{ZK}$, we follow the abstraction of *concurrently extractable commitments* (CECom) introduced by Micciancio, Ong, Sahai, and Vadhan [MOSV06]. Informally, values committed by 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 complete the simulation. Correspondingly, to leverage existing techniques for non-malleable $\mathcal{ZK}$, we employ *non-malleable commitments* as defined by Dolev, Dwork, and Naor [DDN00]. In our protocol (as in the work of [BPS06]), the prover commits to a witness of the proof statement using a non-malleable commitment, and next proves (using a stand-alone) $\mathcal{ZK}$ protocol that it either committed to a valid witness, or a valid trapdoor.

The crux of the proof is then to show that even during 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. Intuitively this should follow from the security guarantees of the non-malleable commitments. The problem, however, is that even if the non-malleable commitments do not "leak" information about the simulator's trapdoors, other parts of the protocol, such as the zero-knowledge proof, might affect the values of the adversary commitments. On a high-level, BPS overcame this problem by relying on *statistical* zero-knowledge protocols for $\mathcal{NP}$; such protocols can only be computationally sound (unless the polynomial hierarchy collapses [AH91]), and known constructions based on one-way functions require $\mathrm{poly}(n)$ rounds.

Instead, we overcome this obstacle by relying on the notion of *robust non-malleable commitments* introduced by [LP09];[3] informally, a robust non-malleable commitment is non-malleable with respect to any protocol that has small round complexity. As shown in [LP09], most known constructions of non-malleable commitment schemes are already robust, or can be made robust easily. Roughly speaking, by relying on this notion we can ensure that the witness used in the $\mathcal{ZK}$ protocol does not affect the witness committed by the adversary (using robust non-malleable commitments) in other executions; in particular, this is used to argue that the adversary essentially never commits to a trapdoor. The actual application of this technique, however, is not direct and requires a subtle treatment—in particular, for technical reasons, we require the prover to use two

[3] Robustness was originally referred to as *naturality*.

robust non-malleable commitments (the same technique is used in [LPV09] for constructing another primitive called strong non-malleable $\mathcal{WI}$ proofs). Furthermore, to make our simulation go through, we are unable to apply the original analysis of CECom as presented in [PRS02,MOSV06], but instead rely on the recent analysis of [PTV08]. Roughly speaking, the reason for this is that concurrently extractable commitments are traditionally used and analyzed in so-called *committed-verifier* protocols [MOSV06], where the verifier commits and *fixes* all of its messages at the start of the protocol. Our protocol does not fall into this category.

Finally, to improve the efficiency of the simulation we have the prover commit to its witness also using a CECom; doing this ensures that the concurrent non-malleability simulator becomes as efficient as the extractor of CECom. Our final result regarding precision is then obtained by relying on the precise $\mathcal{ZK}$ approach from [PPS$^+$08] to implement CECom.

Discussion and Perspectives. Our work closes the "gap" between known constructions of concurrent $\mathcal{ZK}$ and $\mathcal{CNMZK}$ for the plain model (without set-up); that is, we have shown that all known results for concurrent $\mathcal{ZK}$ in the plain model extend to $\mathcal{CNMZK}$ (under the same assumptions, the same round complexity, and the same efficiency of security reductions). In essence, we reduce that task of constructing $\mathcal{CNMZK}$ protocols to constructing concurrently extractable commitments, and thus, concurrent non-malleability come for free. It seems promising that the same approach could be extended also to models with set-up. For instance, in the Bare Public Key model of [CGGM00], $O(1)$-round concurrent $\mathcal{ZK}$ with black-box simulation is known, whereas the only $O(1)$-round protocol for $\mathcal{CNMZK}$ of [OPV08] requires non-black-box simulation. Similar gaps exists for the Timing model [DNS04], and for the model of quasi-polynomial time security [Pas03]. We believe that, by providing appropriate implementations of concurrently extractable commitments (in line with the work on concurrent $\mathcal{ZK}$ in these models), our technique extends to close these gaps. We leave an exploration of these questions for future work.

Overview. Section 2 contains the basic notations and definitions of $\mathcal{CNMZK}$ and other primitives. In Section 3, we present our main result, a $\tilde{O}(\log n)$-round $\mathcal{CNMZK}$ proof system for all of $\mathcal{NP}$, from collision resistant hash functions, and provide the proof of security in Section 4. We also modify the protocol to obtain constructions of a poly(n)-round $\mathcal{CNMZK}$ proof, and a $\tilde{O}(\log n)$-round $\mathcal{CNMZK}$ argument system, from one-way functions at the end of Section 4. We defer our result on Precise $\mathcal{CNMZK}$ to the full version.

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. We assume familiarity with interactive Turing machines, interactive protocols, statistical/computational indistinguishability, zero-knowledge, (strong) witness-indistinguishability (see [Gol01] for formal definitions).

2.1 Concurrent Non-Malleable Zero-Knowledge

We recall the definition of concurrent non-malleable zero-knowledge from [BPS06], which in turn closely follows the definition of simulation extractability of [PR05]. 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, both P and A receives as common input the security parameter in unary 1^n and the statements $x_1, \ldots, x_m$. Additionally, P receives as local input the witnesses $w_1, \ldots, w_m$, $w_i \in R_L(x_i)$, while A receives as auxiliary input $z \in \{0,1\}^*$, which in particular might contain a-priori information about $x_1, \ldots, x_m$ and $w_1, \ldots, w_m$. On the other hand, the statements proved in the right interactions $\tilde{x}_1, \ldots, \tilde{x}_m$ and the identities in both the left and right interactions, $\mathsf{id}_1, \ldots, \mathsf{id}_m$ and $\tilde{\mathsf{id}}_1, \ldots, \tilde{\mathsf{id}}_m$, are chosen by A. Let $\mathsf{view}_A(n, x_1, \ldots, x_m, z)$ denote a random variable that describes the view of A in the above experiment. Loosely speaking, an interactive proof is concurrent non-malleable zero-knowledge ($\mathcal{CNMZK}$) 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* 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:*

1. *The following ensembles are computationally indistinguishable over $n \in N$*
   - $\{\mathsf{view}_A(n, x_1, \ldots, x_m, z)\}_{n \in N, x_1, \ldots, x_m \in L \cap \{0,1\}^n, z \in \{0,1\}^*}$
   - $\{S_1(1^n, x_1, \ldots, x_m, z)\}_{n \in N, x_1, \ldots, x_m \in L \cap \{0,1\}^n, z \in \{0,1\}^*}$

   where $S_1(n, x_1, \ldots, x_m, z)$ denotes the first output of $S(1^n, x_1, \ldots, x_m, z)$.
2. *Let $x_1, \ldots, x_m \in L \cap \{0,1\}^n$, $z \in \{0,1\}^*$, and let $(\mathsf{view}, \boldsymbol{w})$ denote the output of $S(1^n, x_1, \ldots, x_m, z)$. Let $\tilde{x}_1, \ldots, \tilde{x}_m$ be the statements of the right-interactions in view* 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-interaction and right-interactions, respectively, in view* view. *Then for every $i \in [m]$, if the i^{th} right-interaction is accepting and $\tilde{\mathsf{id}}_i \neq \mathsf{id}_j$ for all $j \in [m]$, $\boldsymbol{w}$ contains a witness w_i such that $R_L(\tilde{x}_i, w_i) = 1$.*

2.2 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 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 $\mathsf{nmc}^{A}_{\langle C,R\rangle} v_1, \ldots, v_m, z$ denote a random variable that describes the value $\tilde{v}$ and the view of A, in the above experiment.

Definition 2 *A 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.*

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

Remark 1. The main difference of this definition compared to previous ones [PR03,DDN00] is that it considers not only the values the adversary commits to, but also the view of the adversary. This is particularly important in our analysis later. (See Hybrid H_3 and H_4 in case $j = 2$ in the proof of Lemma 7.)

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 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 committer 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. More formally, let $\mathsf{view}_A[\langle B(y), A(z)\rangle]$ denote the view of $A(z)$ in an interaction with $B(y)$.

Definition 3 *Let $\langle C, R\rangle$ be a commitment scheme, and B a probabilistic polynomial-time machine. We say the 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 $\{y_n^1\}_{n\in N}$ and $\{y_n^2\}_{n\in N}$ such that, for all probabilistic polynomial-time machine $\tilde{A}$, it holds that*

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

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:

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

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 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

Lemma 1 *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.*

The protocol is essentially identical to the $O(\log n)$-round protocol in [LPV08]. A formal proof of this lemma will appear in the full version.

2.3 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 A Concurrent Non-Malleable Zero-Knowledge Proof

In this section we construct a concurrent non-malleable zero-knowledge proof based on collision-resistant hash-functions. Let $\ell(n)$ be any super logarithmic function. Our concurrent non-malleable zero-knowledge protocol, CNMZKProof, 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}. We repeat their definitions below.

To commit a n-bit string v under scheme CECom_{sh}, the committer choses $n \times \ell(n)$ 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 committer 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 committer to decommit all initial commitments under scheme Com_{sh} (i.e., reveal the randomness of the commitments), and that the decommited values satisfy $\alpha^0_{i,j} \oplus \alpha^1_{i,j} = v$ for every i and j.

CECom_{sb} is defined analogously as CECom_{sh} with the initial commitment Com_{sh} replaced by Com_{sb}. Additionally, we say a transcript of CECom_{sb} is *valid* if there exists a valid decommitment. Formal definitions of CECom_{sh} and CECom_{sb} are shown in Fig. 1.

```
Protocol $\mathsf{CECom}_{sh}$(resp. $\mathsf{CECom}_{sb}$)

Inputs: A security parameter $n$, and a value $v \in \{0,1\}^n$ given to the committer (to
    be committed).
Protocol:
    The Committer selects $n\ell(n)$ pairs of random $n$-bit strings $(\alpha^0_{i,j}, \alpha^1_{i,j})$, $i \in [n], j \in$
        $[\ell(n)]$ such that for all $i, j$ $\alpha^0_{i,j} \oplus \alpha^1_{i,j} = v$ and commits to $v$ and $\alpha^0_{i,j}$, $\alpha^1_{i,j}$ for
        every $i \in [n], j \in [\ell(n)]$ using scheme $\mathsf{Com}_{sh}$ (resp. $\mathsf{Com}_{sb}$) to the Receiver.
  For $i = 1$ to $\ell(n)$:
      The Receiver sends a $n$-bit string $b_j = b_{1,j} \ldots b_{n,j}$
      The Committer decommits to $\alpha^{b_{1,j}}_{1,j}, \ldots, \alpha^{b_{n,j}}_{n,j}$.
Decommitment: The Committer decommits to all $n\ell(n) + 1$ commitments made
    under scheme $\mathsf{Com}_{sh}$ (resp. $\mathsf{Com}_{sb}$), and show that $\alpha^0_{i,j} \oplus \alpha^1_{i,j} = v$ for all $i$ and $j$.
```

Fig. 1. Concurrently extractable commitments [MOSV06,PRS02].

We now describe $\mathsf{CNMZKProof}$, our concurrent non-malleable zero-knowledge protocol. Protocol $\mathsf{CNMZKProof}$ 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 of the Prover, and a private input $w \in R_L(x)$ to the Prover.

Stage 1: The Verifier choses a random string $r \in \{0,1\}^n$ and commits to r using CECom_{sh}; 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 under identity id.
Stage 4: The Prover commits to the witness w using NMCom under identity id, again.
Stage 5: The Verifier decommits the Stage 1 commitment to value v.
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 $\tilde{w}$ (under identity id), and that either $\tilde{w} \in R_L(x)$, or $\tilde{w} = r$.

Protocol $\mathsf{CNMZKProof}$, in essence, is a modification of the Goldreich-Kahan protocol [GK96]. The protocol is trivially complete, and below we intuitively argue that the protocol is sound. To cheat in the protocol, because the Stage 2 commitment is statistically binding (and the Stage 6 protocol is a proof), the Prover must know the value r committed by the Verifier in Stage 1, before the conclusion of Stage 2 (i.e., before the Verifier decommits to r). This violates that statistical hiding property of the commitment scheme CECom_{sh}. A formal description of protocol $\mathsf{CNMZKProof}$ is shown in Figure 2.

4 Proof of Security

The definition of $\mathcal{CNMZK}$ 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 the Sect. 4.1 and show its correctness in Sect. 4.2 and 4.3.

4.1 Our Simulator-Extractor

Our simulator-extractor, S, roughly 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 a "fake witness" r from the CECom_{sh} committed by A in Stage 1, then commits to r in Stage 2, 3, and 4, and finally simulates the proof of knowledge 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.

Protocol $\mathsf{CNMZKProof}$

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 protocol CECom_{sh}. 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. 2. Concurrent Non-Malleable $\mathcal{ZK}$ argument for $\mathcal{NP}$

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 oblivious Killian-Petrank simulator [KP01] We also rely on 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 continuously 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.

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 $\bot_{ext}$. Or, in Stage 5 of a left interaction, if A decommits its Stage 1 CECom_{sh} to a value that is different from the extracted value, S outputs $\bot_{bind}$. The following claim bounds the abort probability of S.

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

Proof. This follows essentially from the analysis of [PTV08] in the setting of concurrent $\mathcal{ZK}$. We present the complete proof in the full version of the paper.

4.2 The View Generated by the Simulator

We next show that the view generated by S is indistinguishable from the real view of A.

Lemma 3 *The following ensembles are computationally indistinguishable over $n \in N$:*

$$\{S(1^n, x_1, \dots, x_m, z)\}_{n\in N, x_1,\dots,x_m \in \{0,1\}^n \cap L, z\in\{0,1\}^*}$$
$$\{\mathsf{view}_A(1^n, x_1, \dots, x_m, z)\}_{n\in N, x_1,\dots,x_m \in \{0,1\}^n \cap L, z\in\{0,1\}^*}$$

To show Lemma 3, we introduce a series of hybrid simulators; the same hybrid simulators will also be helpful later in Sect. 4.3. Hybrids hyb^i, $0 \le i \le m+1$, receive the witnesses of the statements proved in any left interactions (i.e., "real witnesses"), and proceed in three steps. In the following description, we order the left interactions by the order in which Stage 1 is completed.

Step 1: Run the simulator S with the adversary A *in its entirety*. Output $\bot_{ext}$ or $\bot_{bind}$ if S outputs $\bot_{ext}$ or $\bot_{bind}$. Otherwise, let $\mathcal{V}$ be the view of A produced by S, and r_j be the "fake witness" extracted by S from the j^{th} left interaction in $\mathcal{V}$.

Step 2: Let $\mathcal{V}_i$ be the prefix of $\mathcal{V}$ up until the i^{th} left interaction has completed Stage 1 of the protocol. Simulate a new man-in-the-middle execution with A, continuing from $\mathcal{V}_i$, in a *straight-line manner*. In each of the following cases, we need to make sure that the view $\mathcal{V}_i$ can be completed in a consistent way. Note that we can continue any partial commitment or zero-knowledge proof contained in $\mathcal{V}_i$ as long as we don't change the committed value or proof witness.[4]

- Continue of the simulation of right interactions by following the honest verifier strategy (just like S).
- Continue the simulation of the first i left interactions in the same manner as S: use the "fake witnesses" r_j's for the commitments in Stage 2, 3 and 4, and the proof in Stage 6. This can be done in a straight line manner since the first i extracted "fake witnesses" ($r_j, j \le i$) are still useful; they correspond to the Stage 1 commitments of the first i left interactions that are present in $\mathcal{V}_i$. Similar to S, if A decommits the Stage 1 CECom_{sh} to a value different from the extracted "fake witness" r, hyb_i outputs $\bot_{bind}$.

[4] Recall that S follows the honest committer and prover strategy in each stage of the protocol; it only cheats by using "fake witnesses". Formally, we can continue any partial commitment or zero-knowledge proof, for example, by requiring S to output the state of every partial commitment and zero-knowledge proofs, for every prefix of the view $\mathcal{V}$.

- Continue the simulation of the $i+1^{\text{st}}$ and later left interactions by following the honest prover strategy using the given "real witnesses". This does not conflict with the partial view $\mathcal{V}_i$, since Stage 2 of these left interactions have not yet started.

Step 3: Output the newly completed view of A from step 2.

We also define hybrids hyb^i_+ that proceed identically as hyb^i except that in step 2, it simulates the i^{th} left interaction following the honest prover strategy, using the given "real witness" (all other interactions are handled identically as before). Note that these hybrids are only concerned with producing a view of A, and do not extract the witnesses of the right interactions.

We start with a claim bounding the abort probability of the hybrids.

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

Proof. hyb^i and hyb^i_+ abort when S aborts, or if they output $\perp_{bind}$ during the second pass of the simulation (while mimicking S). The first event is bounded by Claim 2. The second event occurs with negligible probability due to the binding property of CECom;

By Claim 4, the output of hyb^0 is statistically close to the real view of A (they only differ when hyb^0 aborts, which occurs with negligible probability). The output of hyb^{m+1}, on the other hand, is identical to the output of simulator S. Therefore Lemma 3 directly follows from the next two claims:

Claim 5 *The output of hyb^i and hyb^i_+ are computationally indistinguishable.*

Proof. hyb^i and hyb^i_+ differs only in how the i^{th} left interaction is simulated (real or fake witness), which is done in a straight line fashion by both hybrids. Therefore they are computationally indistinguishable by the computational hiding property of the Stage 2, 3, and 4 commitments, and the strongly witness-indistinguishable property (implied by the $\mathcal{ZK}$ property) of the Stage 6 proof.

Claim 6 *The output 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 end of the $i-1^{\text{st}}$ Stage 1 until the end of the i^{th} Stage 1 of the 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.

Remark 2. 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 $\mathsf{CNMZKProof}$ 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.

4.3 The Witnesses Output by the Simulator

We now show that the extracted witnesses are indeed the $\mathcal{NP}$ witnesses of the statements proved in the right interactions; this is the main technical contribution of our work.

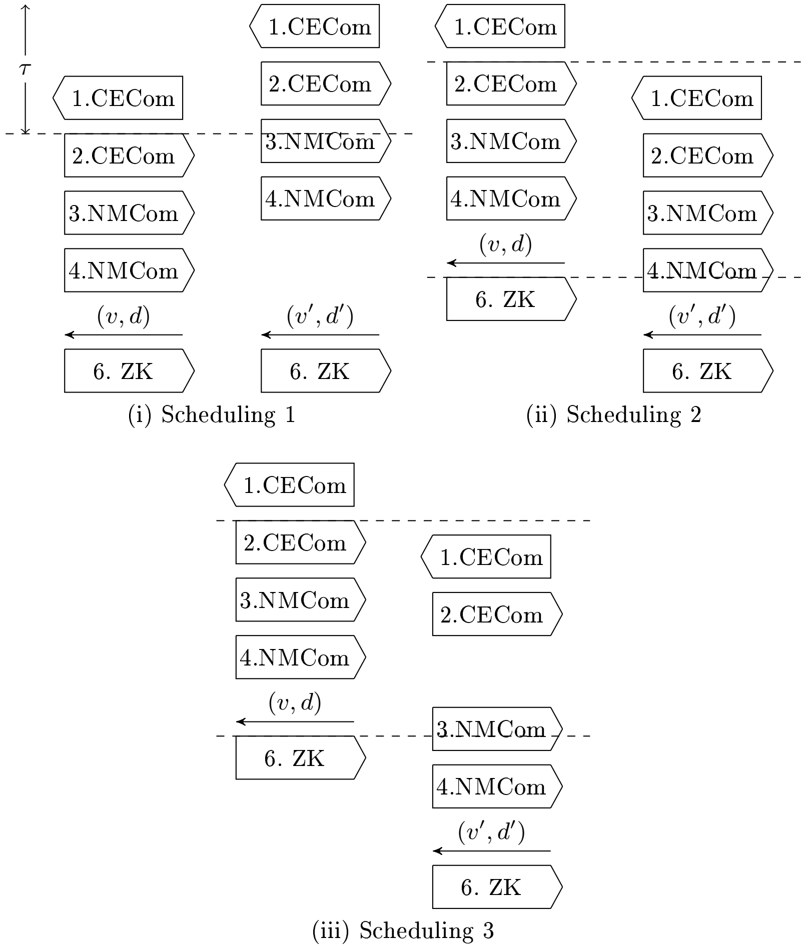
Observe that if A commits to a valid witness using CECom_{sb} in Stage 2 of a right interaction, then by Claim 2, the simulator S would extract this witness except with negligible probability. Therefore, the following lemma establishes the correctness of the output witnesses:

Lemma 7 *For every $\mathcal{PPT}$ adversary A, there exists a negligible function ν, such that for every $n \in N$, $x_1, \ldots, x_m \in \{0,1\}^n \cap L$ 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, 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 $n \in N$, there exists $x_1, \ldots, x_m \in \{0,1\}^n \cap L$ and $z \in \{0,1\}^*$, such that A *cheats* in an outcome of $S_1(n, x_1, \ldots, x_{m(n)}, z)$ 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 the series of hybrids, hyb^i and hyb^i_+, defined in section 4.2. Since hyb^{m+1} is identical to S, by our hypothesis, the probability that A cheats in hyb^{m+1} is non-negligible. On the other hand, in hyb^0, it follows from the soundness of Stage 6 that, except with negligible probability, in every accepting right interaction, A commits (successfully) 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^0. In addition, 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 probability of cheating differ by at least a polynomial amount in hyb^i_+ and hyb^i. 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 has completed Stage 1 of the protocol—we call it the *cutoff point*. After the cutoff point, the only difference between the two experiments lies in how the i^{th} left interaction are simulated (using either the real or fake witness.) Recall that the adversary A controls the message scheduling in the network; it can thus arrange messages in the i^{th} left-proof and the randomly chosen right-proof in one of the following three ways; see figure 3. Below we omit specifying the i^{th} left interaction and the randomly chosen right interaction, when it is clear in the context.



(i) Scheduling 1

(ii) Scheduling 2

(iii) Scheduling 3

Fig. 3. The three scheduling in a man-in-the-middle execution of A.

Scheduling 1: A completes the Stage 2 commitment on the right before the cutoff point.

Scheduling 2: A completes the Stage 2 commitment after the cutoff point, but completes the Stage 3 commitment before the Stage 6 proof starts on the left.

Scheduling 3: A completes the Stage 2 commitment after the cutoff point, and completes the Stage 3 commitment after the Stage 6 proof starts on the left.

Now consider a variant of hyb^i, $\mathsf{hyb}^{i,j}$ where $j \in \{1, 2, 3\}$, which proceeds identically to hyb^i, except that it outputs $\perp$ if scheduling j does not occur in the output view; define $\mathsf{hyb}_{+}^{i,j}$ correspondingly for hyb_{+}^{i}. Since every man-in-the-middle execution must follow one of the three scheduling above, it holds that, there exists a $j \in \{1, 2, 3\}$, such that for infinitely many $n \in N$, the probabilities that A cheats in a randomly chosen right interaction in $\mathsf{hyb}_{+}^{i,j}$ and $\mathsf{hyb}^{i,j}$ differ by a polynomial amount,

Towards reaching a contradiction, let $\mathsf{hyb}^{i,j}(n, x_1, \dots, x_m, z)$ denote the combined view of A and the value v it commits to in Stage 2 of a randomly chosen right interaction in $\mathsf{hyb}^{i,j}$; v is replaced with $\perp$ if any of the following three events happens: the hybrid experiment fails, or the right interaction j fails, or the right interaction copies the identity of one of the left interactions. Define $\mathsf{hyb}_{+}^{i,j}(n, x_1, \dots, x_m, z)$ correspondingly for $\mathsf{hyb}_{+}^{i,j}$. (For convenience, we refer to v as the committed value of the right interaction.) Below we show that, for every b, and every function $i : N \to N$, $\left\{\mathsf{hyb}^{i,j}(n, x_1, \dots, x_m, z)\right\}$ and $\left\{\mathsf{hyb}_{+}^{i,j}(n, x_1, \dots, x_m, z)\right\}$ are computationally indistinguishable, which implies that the probabilities that A cheats in a *randomly chosen* right interaction differ by at most a negligible amount in the two hybrid experiments, which is a contradiction. The lemma thus follows.

When $j = 1$, A completes the Stage 2 commitment on the right *before the cutoff point*, in hybrids $\mathsf{hyb}^{i,1}$ and $\mathsf{hyb}_{+}^{i,1}$. Since the two hybrid experiments proceed identically before the cutoff point, the values A commits to in Stage 2 on the right are identical in the two experiments. It then follows using essentially the same argument as in Lemma 3 (by relying on the hiding property of Stage 2 to 4 and the strongly $\mathcal{WI}$ property of Stage 6) that the view and the committed value on the right are indistinguishable, i.e.,

Claim 8 *For every function $i : N \to N$, the following ensembles are computationally indistinguishable:*

- $\left\{\mathsf{hyb}^{i(n),1}(n, x_1, \dots, x_m, z)\right\}_{n \in \mathbf{N}, x_1, \dots, x_m \in (L \cap \{0,1\}^n)^m, z \in \{0,1\}^*}$
- $\left\{\mathsf{hyb}_{+}^{i(n),1}(n, x_1, \dots, x_m, z)\right\}_{n \in \mathbf{N}, x_1, \dots, x_m \in (L \cap \{0,1\}^n)^m, z \in \{0,1\}^*}$

When $j = 2$, Stage 3 to 6 of the right interaction are simulated completely after the cutoff point in a straight line fashion, in $\mathsf{hyb}^{i,2}$ and $\mathsf{hyb}_{+}^{i,2}$. It then follows from the soundness of Stage 6 that, except from negligible probability, A always commits to the same value in Stage 2, 3 and 4 on the right, provided that the right interaction is *accepting*. Hence to show the indistinguishability of the view and the value A commits to on the right, it suffices to show the indistinguishability of the view $\mathcal{V}$ and the value v that A commits to in Stage 3 (This is because the committed value on the right can be efficiently reconstructed from $\mathcal{V}$ and v, by replacing v with $\perp$ appropriately according to $\mathcal{V}$). Then consider the following hybrids, $H_0 = \mathsf{hyb}_{+}^{i,2}$ to $H_5 = \mathsf{hyb}^{i,2}$.

Hybrid H_1 proceeds identically to H_0, except that, in H_1, Stage 6 of the left interaction is simulated using the simulator of the $\mathcal{ZK}$ protocol $\langle P, V \rangle$. Since in Scheduling 2, the Stage 3 commitment on the right completes before the Stage 6 proof starts, the value A commits to in Stage 3 is independent of the $\mathcal{ZK}$ proof. Therefore, the view and the value A commits to in Stage 3 are indistinguishable in H_0 and H_1.

Hybrid H_2 proceeds identically to H_1, except that the Stage 2 CECom_{sb} of the left interaction is now a commitment to the "fake witness" (whereas in H_1, it is a commitment to a valid witness). It then follows from the non-malleability w.r.t. $\ell(n)$-round protocols of NMCom, (and the fact that Stage 2 of the protocol consists of $\ell(n)$ rounds) that, the view and the value A commits to in Stage 3 are indistinguishable in H_1 and H_2.

Hybrid H_3 **(and** H_4 **resp.)** proceeds identically to H_2 (and H_3 resp.), except that, Stage 3 (and Stage 4 resp.) of the left interaction is now a commitment to the "fake witness". It follows using a similar argument as in H_2, but relying on the non-malleability w.r.t. itself of NMCom that the view and the value A commits to in Stage 3 are indistinguishable in H_2 and H_3 (and in H_3 and H_4 resp.).

Hybrid H_5 proceeds identically to H_4, except that Stage 6 of the left interaction is simulated by proving that Stage 2, 3 and 4 are valid commitments to the value revealed by A in Stage 5 on the left. Note that, by defintion, H_5 proceeds identically to the experiment $\mathsf{hyb}^{i,2}$. Furthermore, it follows using the same argument as in H_1 that the view and the values A commits to in Stage 3 are indistinguishable in H_4 and H_5.

Finally, it follows using a hybrid argument that the combined view and the value A commits to in Stage 3 are indistinguishable in $\mathsf{hyb}^{i,2}$ and $\mathsf{hyb}_+^{i,2}$. Therefore,

Claim 9 *For every function $i : N \to N$, the following ensembles are computationally indistinguishable:*

- $\left\{\mathsf{hyb}^{i(n),2}(n, x_1, \ldots, x_m, z)\right\}_{n\in N, x_1,\ldots,x_m\in(L\cap\{0,1\}^n)^m, z\in\{0,1\}^*}$
- $\left\{\mathsf{hyb}_+^{i(n),2}(n, x_1, \ldots, x_m, z)\right\}_{n\in N, x_1,\ldots,x_m\in(L\cap\{0,1\}^n)^m, z\in\{0,1\}^*}$

When $j = 3$, by the same argument as in the case when $j = 2$, A always commits to the same value in Stage 2, 3 and 4 of every accepting right interaction, and thus, it suffices to show that the view and the value A commits to in Stage 4 are indistinguishable. In $\mathsf{hyb}^{i,3}$ and $\mathsf{hyb}_+^{i,3}$, (as A completes the Stage 3 commitment on the right after the Stage 6 proof starts on the left), the Stage 4 commitment on the right starts completely after the Stage 6 proof on the left, which (by definition) consists of only $\omega(1)$ rounds. It thus follows from the non-malleability with respect to $\omega(1)$-round protocols of NMCom (along with the strongly $\mathcal{WI}$ property of Stage 6) that, the view and the value A commits to in Stage 4 are indistinguishable. Therefore,

Claim 10 *For every function $i : N \to N$, the following ensembles are computationally indistinguishable:*

- $\left\{\mathsf{hyb}^{i(n),3}(n, x_1, \ldots, x_m, z)\right\}_{n\in N, x_1,\ldots,x_m\in(L\cap\{0,1\}^n)^m, z\in\{0,1\}^*}$
- $\left\{\mathsf{hyb}_+^{i(n),3}(n, x_1, \ldots, x_m, z)\right\}_{n\in N, x_1,\ldots,x_m\in(L\cap\{0,1\}^n)^m, z\in\{0,1\}^*}$

A formal proof of this claim will appear in the full version.

Completing Theorem 1 and Theorem 2. Above we constructed a $\tilde{O}(\log n)$-round $\mathcal{CNMZK}$ proof based on collision-resistant hash-functions. We obtain a $\tilde{O}(\log n)$-round $\mathcal{CNMZK}$ argument from one-way functions, simply by replacing the Stage 1 CECom_{sh} commitment with protocol CECom_{sb}. Note that the resulting protocol is still sound since because the Stage 2 commitment by the prover (CECom_{sb}) is statistically binding and "extractable".[5]

Furthermore, to obtain a poly(n)-round $\mathcal{CNMZK}$ proof based on one-way functions, we use the same protocol $\mathsf{CNMZKProof}$, except that we construct the Stage 1 CECom_{sh} using the public-coin statistically hiding commitment from one-way functions by Haitner et. al. [HNO$^+$09]. It follows using essentially the same security proof as for $\mathsf{CNMZKProof}$ that this protocol is $\mathcal{CNMZK}$; the difference lies in how to bound the "binding failure" However, as in the main proof, this can be bound using the analysis of [PTV08] since the commitment of [HNO$^+$09] is public-coin.

References

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[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.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC '00*, pages 235–244, 2000.

[CKPR01] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires $\tilde{\omega}(\log n)$ rounds. In *STOC '01*, pages 570–579, 2001.

[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.

[5] Given a prover that breaks soundness, we may break the computationally hiding property of the Stage 1 verifier CECom_{sb} by rewinding the prover and extracting the committed value of the Stage 2 prover CECom_{sb}.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, 1991.

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

[HNO$^+$09] Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 39(3):1153–1218, 2009.

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

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

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

[LPV09] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. A unified framework for concurrent security: universal composability from stand-alone non-malleability. In *STOC '09*, pages 179–188, 2009.

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

[MP06] Silvio Micali and Rafael Pass. Local zero knowledge. In *STOC '06*, pages 306–315, 2006.

[OPV08] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Constant-round concurrent non-malleable zero knowledge in the bare public-key model. In *ICALP (2)*, pages 548–559, 2008.

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

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *EUROCRYPT*, pages 160–176, 2003.

[PPS$^+$08] Omkant Pandey, Rafael Pass, Amit Sahai, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Precise concurrent zero knowledge. In *Eurocrypt '08*, pages 397–414, 2008.

[PR03] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *FOCS*, pages 404–, 2003.

[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`.

[RK99] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In *Eurocrypt '99*, pages 415–432, 1999.

[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.