Non-malleable Zero-Knowledge Arguments with Lower Round Complexity

Zhenbin Yan∗ Yi Deng†

Abstract

Round complexity is one of the fundamental problems in zero-knowledge proof systems. Non-malleable zero-knowledge (NMZK) protocols are zero-knowledge protocols that provide security even when man-in-the-middle adversaries interact with a prover and a verifier simultaneously. It is known that the first constant-round public-coin NMZK Arguments for NP can be constructed by assuming the existence of collision-resistant hash functions (Pass and Rosen STOC’05) and has relatively high round complexity; the first four-round private-coin NMZK Arguments for NP can be constructed in the plain model by assuming the existence of one-way functions (Goyal, Richelson, Rosen and Vald FOCS’14 and Ciampi, Ostrovsky, Siniscalchi and Visconti TCC’17).

In this paper, we present a six-round public-coin NMZK argument of knowledge system assuming the existence of collision-resistant hash functions and a three-round private-coin NMZK argument system from multi-collision resistance of hash functions assumption in the keyless setting.

∗State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China. School of Cyber Security, University of Chinese Academy of Sciences, China. Email: yanzhenbin@iie.ac.cn

†State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China. School of Cyber Security, University of Chinese Academy of Sciences, China. Email: deng@iie.ac.cn
1 Introduction

The fundamental notion of zero-knowledge was introduced by Goldwasser et al. [GMR89]. We say an interactive proof protocol for NP is zero-knowledge if the prover can prove a statement $x \in L$ is true without revealing other useful information. With such an intriguing nature, zero-knowledge proof has played a central role in the design and study of cryptographic protocols.

The notion of non-malleable zero knowledge (NMZK) was first introduced and achieved by Dolev et al. [DDN00], considering the execution of zero-knowledge proofs in the setting where a man-in-the-middle adversary interacts with an honest prover in the left session and an honest verifier in the right session. Pass and Rosen [PR05b, PR05a] constructed the first constant-round public-coin NMZK arguments based on the existence of collision-resistant hash functions. Their protocols rely on the non-black-box techniques used by Barak [Bar01] to obtain constant-round public-coin ZK arguments for NP. They first constructed a public-coin NMZK arguments with tags of length $O(\log(n))$ and then executed it in parallel to implement a 10-round public-coin NMZK arguments with tags of length $n$. We note that almost all known techniques for achieving non-malleable zero knowledge have relied crucially on non-malleable commitments. However, Pass and Rosen [PR05b, PR05a] reversed the roles. They first achieved a constant-round NMZK protocol and then used it to achieve non-malleable commitment schemes and concurrent non-malleable commitment schemes.

For the case of private-coin zero-knowledge, Goyal et al. [GRRV14] used an algebraic approach and achieved the first 4-round NMZK argument based on the minimal assumption that one way functions (OWFs) exist. Very recently, Ciampi et al. [COSV17a] achieved a delayed-input 4-round NMZK argument based on OWFs. In this work, we continue along this line of research and mainly focus on the following two problems:

Problem 1. We know that public-coin interactive proof protocol has many advantages, such as better resilience to leakage and so on. The round complexity of the original public-coin ZK protocols of [Bar01] is six rounds. In this paper, we focus on how to reduce the round complexity of the current known public-coin non-malleable zero-knowledge protocols, (i.e., [PR05b, PR05a]). In particular, we provide a construction of such protocol in six rounds based on standard collision-resistant hash functions assumptions. This demonstrates that, like private-coin zero-knowledge arguments, public-coin non-malleable zero-knowledge arguments can achieve the same round complexity as public-coin zero-knowledge arguments without the cost of additional complexity assumptions.

Theorem 1.1. Assuming the existence of collision-resistant hash functions, there exists a 6-round public-coin non-malleable zero-knowledge argument of knowledge system for NP.

As a corollary of our first result, using Barak et al.’s transformation [BGGL01], we can modify our construction into a resettably-sound non-malleable zero-knowledge arguments. This can be done by having the verifier V samples a seed $s$ for a pseudorandom function (PRF) $f_s$ at the beginning of the protocol, and generates each of its message by applying $f_s$ to the current history message it received. Thus, from [BGGL01] we can get the following result.

Theorem 1.2. Assuming the existence of collision-resistant hash functions, there exists a 6-round non-malleable zero-knowledge argument system for NP that is also resettably-sound.

Problem 2. Ever since the introduction of zero-knowledge, an extensive amount of research has been dedicated towards minimizing their round complexity. In the negative direction, two-round ZK arguments for NP were ruled out by Goldreich and Oren [GO94] and three-round black-box ZK arguments for NP were ruled out by Goldreich and Krawczyk [GK90]. In the positive direction, three-round ZK arguments with non-black-box simulators were studied in [BBK+16, BCPR16] et al., and
very recently, Bitansky, Kalai, and Paneth [BKP18] constructed general three-round ZK arguments for non-uniform provers and verifiers based on keyless multi-collision-resistant hash functions.

In this paper, we further study how to construct the 3-round non-malleable zero-knowledge arguments for NP based on the multi-collision resistance of hash functions. We present the following results:

**Theorem 1.3.** Assuming keyless multi-collision resistant hash functions, LWE and DDH (or QR or $N^{th}$ residuosity), all quasi-polynomially hard, there exists 3-round non-malleable zero-knowledge arguments for NP.

1.1 Technique Overview

**Public-coin NMZK argument.** Recall that due to the impossibility of constant-round public-coin zero-knowledge proofs with black-box simulator proved by Goldreich and Krawczyk [GK90, GK96], all currently known public-coin zero-knowledge protocol such as [Bar01, CLP13a, CLP13b, PRT13, PTW09] et al. are constructed by using Barak’s non-black-box simulation technique [Bar01]. In order to reduce the round complexity, our protocol will try to combine a variant of Barak’s protocol (i.e., the “special-purpose” universal argument [PR05b]) and a 3-round public-coin extractable non-malleable commitment scheme [GPR16] in parallel. We let the prover commit to a witness of the statement by using a non-malleable commitment scheme and proves that it either committed to a valid witness or a valid trapdoor by using the “special-purpose” universal argument protocol.

The “special-purpose” universal argument [PR05b] consists of three stages. The first stage is a commitment-challenge slot, the second stage is an “encrypted” universal argument to argue that the prover has a trapdoor, and the last stage is a 3-round public-coin WIPOK to prove that either it knows a witness for statement $x \in L$ or the opening of the transcript in stage two is an accepting proof for knowing the trapdoor. We observe that if we directly combine the 3-round of the WIPOK subprotocol with a 3-round non-malleable commitment subprotocol in parallel, then there is a subtle issue that we cannot deal with. We explain informally the difficulties that we encounter in the following.

Recall that the definition of non-malleable zero-knowledge requires the existence of a simulator-extractor $SE$ that can simulate the view of a man-in-the-middle adversary $A$ while simultaneously extracting the witnesses of the statement proved by the adversary in the right interaction. On the one hand, in order to prove the zero-knowledge property, we need the non-malleable commitment to commit to a witness for statement $x \in L$ in real execution or a trapdoor in the simulator execution. On the other hand, in order to prove the non-malleable property, we need to argue that when the simulator commits a trapdoor (instead of witnesses) in the left interaction, the man-in-the-middle adversary $A$ still cannot change its committed values in the right interaction. The usual approach is to design a series of hybrids and reduce the security to the witness indistinguishability of the WIPOK on the left. However, when we extract the witnesses on the right using the rewinding method, the WIPOK subprotocol (i.e. [Blu86]) on the left side will also be rewound, because the adversary can use a scheduling to make the two sides execute in parallel. Thus, we cannot use the simulator-extractor to break the witness indistinguishability of the WIPOK and arrive at a contradiction.

We bypass this obstacle by using the public-coin non-malleable commitment scheme to commit the witness twice in sequence. More specifically, we execute the two non-malleable commitments in five rounds where the messages in the third round of the first commitment and the first round of the second commitment are sent in one round. We let the five rounds non-malleable commitments execute in parallel with the six rounds “special-purpose” universal argument from its second round, and keep the round complexity at six (see Fig. 1). Using this special structure, no matter what
scheduling strategy the adversary uses, we can design a series of hybrids to complete the security proof. Because the simulator now has the freedom to choose which non-malleable commitment to extract the witness. (For the details see Sec. 3.1).

**Three-round NMZK argument.** Due the lower bound of the three-round black-box zero-knowledge showed by Goldreich and Krawczyk \[GK90, GK96\], three-round zero-knowledge protocols necessarily require using the code of the adversary in a non-black-box way. But Barak’s non-black-box technique \[Bar01\] does not apply to this question, because the two-round trapdoor statement generation steps must be fixed before the three-round WI universal argument and its round complexity exceeds three rounds. Another problem is that the protocol is only three rounds, we can not invoke the non-malleable commitment twice in sequence as before to bypass the reduction security to the WI subprotocol. In order to squeeze the round complexity, here we consider two main components to achieve our three rounds NMZK argument.

The first component is the variant of the two-message memory delegation technique \[CKLR11\] based on the LWE assumption. This approach first appeared in Bitansky et al. \[BBK^+16, BKP18\] to construct three-round zero-knowledge argument based on the existence of *keyless multi-collision resistant hash functions* and LWE assumptions against quasipolynomial-time attackers. Roughly speaking, the variant of the *verifiable memory delegation protocol* consists of three rounds. In the first round, the prover provides the verifier a short commitment to the digest \(d\) of a large memory \(D\). In the second round, the verifier sends a query \(q\) to the prover and holds a secret state \(vst\) itself. The verifier can then delegate any arbitrary deterministic computation \(M\) to be executed over the memory \(D\). In the third round, the prover responds with the Turing machine \(M\), the computation’s output \(y\), the committed values \(d\) and \(t\), as well as a short proof \(\pi\) for the correctness that \(\exists D\) such that \(M(D) = y\) within \(t\) steps where \(d\) is the digest of the memory \(D\). The verifier can verify the computation by using its secret state \(vst\) in time that is independent of that of the delegated computation and the size of the memory. Inspired by Barak’s non-black-box simulation idea when considering the construction of three-round zero-knowledge arguments, Bitansky et al. \[BBK^+16, BKP18\] see the trapdoor (i.e. the verifier’s code \(V^*\)) as the memory. The simulator can then prove that \(M(V^*) = r\) in one round by using the *verifiable memory delegation protocol* above. In particular, the simulator can construct a Turing machine \(M\) such that on input the memory \(D = V^*\), it parses \(V^*\) as a Turing machine which on input \(c\) will output the challenge string \(r\).

The second component is the *delayed-input statistical witness indistinguishable arguments* (SWI) in two rounds. Very recently, Kalai, Khurana and Sahai \[KKS18\] first achieved this protocol based on *quasi-polynomial hardness of two-message oblivious transfer*, which in turn can be based on the DDH or QR or \(N^{th}\) residuosity assumption against quasi-polynomial attacker. Such protocol has an advantage that the witness indistinguishable arguments will not leak the information about which witness is used by the simulator, because it is statistically secure. Therefore, when the simulator changes the witness used in the SWI from the real witness to the trapdoor on the left side, it will not affect the values of the adversary commitment on the right side. We note that the assumption of using statistical WI is stronger than using standard WIPOK, i.e., the Blum \[Blu86\] protocol or the FLS \[FLS99\] protocol before. However, this is not an issue, because our first component has been based on the quasi-polynomial hard assumption.

Thus, in our non-malleable zero-knowledge arguments, we can combine the *two-round SWI* \[KKS18\], the *three-round verifiable memory delegation protocol* \[BKP18\] and the *three-round public-coin extractable non-malleable* commitment scheme \[GPR16\] in parallel. More specifically, the prover runs the two-round SWI protocol to prove that either it knows the opening of the non-malleable commitment to \(w\) such that \(R(x,w) \in R_L\) or knows a short proof \(\pi\) for the correctness of the delegate computation \(M(V^*) = r\).
It is easy to see that the zero-knowledge property of the above protocol can be directly obtained from the original protocol [BKP18]. The non-malleable property can be obtained from the non-malleable security of the NMCom, the statistical witness-indistinguishable of SWI and the computational hiding of the non-interactive primitive Com. In particular, there are only three rounds in the protocol. The malicious scheduling the man-in-the-middle adversary can be used only two types, i.e., the synchronous scheduling where he lets the right interaction run in parallel with the execution of the left or the sequential scheduling where he lets the right interaction complete after the execution of the left. We can give a series of hybrid proofs separately to argue the non-malleable property. (For the details see Sec. 3.2).

1.2 Related Work

The round complexity of non-malleable zero-knowledge and the non-malleable commitment are usually studied in parallel. In particular, the former relies heavily on the latter. Pass [Pas13, Pas16] showed a lower bounds of the black-box reductions from two-round non-malleable commitment to any standard intractability assumptions. Since then, Goyal et al. [GRRV14, GPR16] constructed a four-round non-malleable commitment based on the existence of one-way function and a three-round protocol using quasi-polynomially hard injective one-way functions. Recently, Ciampi et al. [COSV16, COSV17b] showed a transformation from any three-round non-malleable commitment ([GPR16]) to three-round concurrent non-malleable commitment and constructed a four-round concurrent non-malleable commitment based on the existence of one-way function. Recently, Lin et al. [LPS17] demonstrates the existence of a two-round concurrent non-malleable commitment assuming the existence of non-interactive commitments, ZAPs, CRHF and a time-lock puzzle with sub-exponential security. Khurana et al. [KS17, Khu17] showed a two-round non-malleable commitment using sub-exponentially hard one-way permutations, sub-exponential ZAPs, and sub-exponential DDH and a three-round concurrent non-malleable commitment assuming the existence of polynomial hardness of DDH assumption or Quadratic Residuosity or $N^{th}$ Residuosity, together with ZAPs. Very recently, Goyal and Richelson [GR19] implemented the first construction of three-round non-malleable commitments from the injective one-way functions assumption by relying on a novel technique, which is called bidirectional Goldreich-Levin extraction technique [GL89].

The notion of multi-collision resistance was studied concurrently and independently by Berman et al. [BDRV18], Bitansky et al. [BKP18] and Komargodski et al. [KNY18]. Very recently, Bitansky and Lin [BL18] showed a fully-concurrent one-message non-malleable commitments against all efficient non-uniform adversaries assuming multi-collision-resistant keyless hash functions (and injective OWFs, NIWI and time-lock puzzles) with sub-exponential security.

1.3 Organizations

The rest of this paper is organized as follows. In Sect. 2, we give the basic definitions used throughout the paper, including the definition of non-malleable commitment, non-malleable zero-knowledge, keyless multi-collision resistant hash functions, memory delegation, and so on. Next, we describe and analyze the six-round public-coin non-malleable zero-knowledge argument of knowledge in Sect. 3.1 and the three-round non-malleable zero-knowledge in Sect. 3.2.
2 Preliminary

2.1 Notations

Let \( \mathbb{N} \) denote the set of all positive integers, for any integer \( n \in \mathbb{N} \), let \([n]\) denote the set \( \{1, 2, \cdots, n\} \), and let \( \{0, 1\}^n \) denote the set of \( n \)-bit long strings; furthermore, let \( \mu \) denote a negligible function, if for every positive polynomial \( p \) and all sufficiently large \( n \), it holds that \( \mu(n) < 1/p(n) \). We assume familiarity with interactive Turing machines and interactive protocols. Let \( \text{PPT} \) denotes \emph{probabilistic polynomial time} Turing machines, and denote by \((A(y), B(z))(x)\) an interactive protocol on the common input \( x \), where \( A \) with a private input \( y \), \( B \) with a private input \( z \), and the random tape of each machine is \emph{uniformly} and \emph{independently} chosen.

2.2 Statistically Binding Commitments

The hiding property of a commitment scheme is that the sender commits to a value while keeping it secret from the receiver; the binding property is that the commitment can only be opened to a single value as determined during the commitment protocol. In the \emph{statistically} (resp. \emph{perfectly}) binding commitments, the binding property holds against \emph{unbounded} adversaries, while the hiding property only holds against \emph{computationally bounded} (\emph{non-uniform}) adversaries.

The \emph{non-interactive} perfectly-binding commitment schemes can be constructed using any one-to-one one-way function (see Section 4.4.1 of \cite{Gol01}). The \emph{two-message} statistically binding commitment schemes can be obtained from any one-way function \cite{Nao91, HILL99}.

2.3 Witness Indistinguishability

\textbf{Definition 1.} (Witness Indistinguishability) An interactive protocol \((P, V)\) for \( L \in \text{NP} \) is \emph{witness indistinguishable} for \( R_L \) if for every \( \text{PPT} \) adversarial verifier \( V^* \) and for every two sequences \( \{w^1_x\}_{x \in L} \) and \( \{w^2_x\}_{x \in L} \) such that \((w^1_x, w^1_x) \in R_L(x)\), the following ensembles are computationally indistinguishable:

- \( \{ \text{View}_{V^*} (P(x, w^1_x) \leftrightarrow V^*(x)) \}_{x \in L} \)
- \( \{ \text{View}_{V^*} (P(x, w^2_x) \leftrightarrow V^*(x)) \}_{x \in L} \)

\textbf{Definition 2.} (Proof of Knowledge \cite{FS90, BG92}) An interactive proof (or argument) system \((P, V)\) for an \text{NP} language \( L \) with witness relation \( R_L \) is said to be \emph{proof (or argument) of knowledge}, if there exists a polynomial \( q \), a negligible function \( \mu \), and a probabilistic oracle machine \( E \) (also called the knowledge extractor), such that for every interactive machine \( P^* \) (or \( \text{PPT} \) \( P^* \)) and every \( x \in L \), the following holds: if \( \Pr[(P^*, V)(x) = 1] > \mu(|x|) \), then \( E^{P^*}(x) \) can output a witness for \( x \) with oracle access to \( P^*(x) \), and the running time of \( E \) is bounded by \( \Pr[(P^*, V)(x) = 1] - \mu(|x|) \).

\textbf{Definition 3.} (Special soundness) A three-round interactive protocol \((P, V)\) for \( L \in \text{NP} \) with witness relation \( R_L \) is \emph{special-sound}, if there exist two accepting transcripts \((\alpha, \beta, \gamma)\) and \((\alpha, \beta', \gamma')\) such that the first message are the same but the challenges \((\beta, \beta')\) are different, then there is a deterministic polynomial time algorithm which can extract the witness from the two transcripts.

Three-round special-soundness public-coin \( \text{WIPOK} \) protocol \cite{Blu86, FLS99} can be constructed from any \( 1\)-\( 1 \) one-way functions (or, \emph{four-round} protocol can be obtained from \emph{one-way functions}). Two-round statistically witness indistinguishable arguments \cite{KKS18} can be obtained assuming the existence of a \emph{quasi-poly secure} \( \text{OT} \), which can in turn be instantiated based on \emph{quasi-poly hardness} of the \emph{DDH} assumption \cite{NP01}, or based on the \emph{quasi-poly hardness} of \( QR \) or the \( N' \)th residuosity assumption \cite{HK12}. 

\null

5
2.4 Non-malleable Commitment [LP11]

A tag-based commitment scheme \( \langle C, R \rangle \) is a commitment scheme where the committer and the receiver receive a tag \( t \in \{0, 1\}^n \) (also called id) as common input. Consider a man-in-the-middle adversary \( \mathcal{A} \) that, on auxiliary input \( z \), participates in a left and a right interaction. On the left, \( \mathcal{A} \) interacts with \( C \), receiving a commitment to the value \( v \), using identity \( \text{id} \) of its choice. On the right, \( \mathcal{A} \) interacts with \( R \) attempting to commit to a related value \( \tilde{v} \), using identity \( \tilde{\text{id}} \) of its choice. If \( \tilde{\text{id}} = \text{id} \) or the right commitment is invalid, or undefined, its value \( \tilde{v} \) is set to \( \perp \). Let \( \text{nmc}^{\mathcal{A}}(v, z) \) denote a random variable that describes the value \( \tilde{v} \) and the view of \( \mathcal{A} \) in the above experiment.

**Definition 4.** (Non-Malleable Commitment) A commitment scheme \( \langle C, R \rangle \) is said to be non-malleable if for every polynomial \( p(\cdot) \), and every PPT man-in-the-middle adversary \( \mathcal{A} \), the following ensembles are computationally indistinguishable.

- \( \{\text{nmc}^{\mathcal{A}}(v, z)\}_{v, v' \in \{0, 1\}^n, z \in \{0, 1\}^*} \)
- \( \{\text{nmc}^{\mathcal{A}}(v', z)\}_{v, v' \in \{0, 1\}^n, z \in \{0, 1\}^*} \)

2.5 Honest Extractable Commitments [PW09, COSV17a]

**Definition 5.** (Honest Extractable Commitment Scheme) A statistically (resp. perfectly) binding commitment scheme \( \langle C, R \rangle \) is an honest extractable commitment scheme if there exists an expected PPT extractor \( \text{ExtCom} \) which given oracle access to any honest sender \( C \) can output a pair \((\tau, m)\) such that the following two properties hold:

- **Simulatability:** \( \tau \) is identically distributed to the view of \( C \) (when interacting with an honest \( R \)) in the commitment phase.
- **Extractability:** the probability that there exists a decommitment of \( \tau \) to a message \( m' \), where \( m' \neq m \) is negligible (resp. 0).

For man-in-the-middle adversary \( \mathcal{A} \) we say it is synchronous if it aligns the left and the right session. That is whenever \( \mathcal{A} \) receive the \( i \)-th round message on the left, it directly sends the \( i \)-th round message on the right, and vice versa. The three-round public-coin synchronous honest-extractable non-malleable commitment scheme [GPR16] can be obtained from 1-1 one-way functions (or, four-round protocol can be obtained from one-way functions), which is extractable w.r.t. honest sender and non-malleable against synchronous adversaries.

2.6 Non-malleable Zero-knowledge [LPTV10]

Let \( (P, V) \) be an interactive protocol for a language \( L \). Consider a PPT man-in-the-middle adversary \( \mathcal{A} \) that, given the common input \( x \) and an auxiliary input \( z \in \{0, 1\}^* \). On the left, \( \mathcal{A} \) acts as a verifier \( V^* \) to interact with \( P \) using the common input \( x \) and id, and the prover \( P \) will be given a valid witness \( w \in R_L(x) \). On the right, \( \mathcal{A} \) acts as a prover \( P^* \) that, on common input \( \tilde{x} \) to prove the validity using \( \text{id} \). During the experiment, the statement \( \tilde{x} \) and the tags \( \text{id}, \tilde{\text{id}} \) are all chosen by the adversary \( \mathcal{A} \). Let \( \text{view}_{\mathcal{A}}(1^n, x, z) \) denotes the random variable that describes the view of \( \mathcal{A} \) in the above experiment. Loosely speaking, an interactive proof is a non-malleable zero-knowledge protocol, if for all man-in-the-middle adversary \( \mathcal{A} \), there exists a PPT machine (called the simulator-extractor) that can simulate both the left and the right interaction for \( \mathcal{A} \), while outputting a witness for the statement proved by the adversary in the right interaction.
Definition 6. An interactive protocol \((P, V)\) for \(L \in \text{NP}\) is said to be **non-malleable zero-knowledge** if for every \(n \in \mathbb{N}\), and every PPT man-in-the-middle adversary \(A\), there exists a PPT machine \(SE\) such that:

1. The following ensembles are computationally indistinguishable:
   
   \[
   \{\text{view}_A(1^n, x, z)\}_{n \in \mathbb{N}, x \in L \cap \{0,1\}^n, z \in \{0,1\}^n} \quad \text{and} \quad \{S(1^n, x, z)\}_{n \in \mathbb{N}, x \in L \cap \{0,1\}^n, z \in \{0,1\}^n}
   \]

   where \(S(1^n, x, z)\) is the first output of \(SE(1^n, x, z)\).

2. Let \(\bar{x}\) be the statements to be proved in the right interaction and \((\text{view}, \bar{w})\) denote the output of \(SE(1^n, x, z)\). Then if the right interaction is accepting and \(\hat{id} \neq id\), it holds that \(\bar{w}\) is a valid witness such that \(R_L(\bar{x}, \bar{w}) = 1\).

2.7 Resettably-sound arguments [BGGL01]

Definition 7. (Resettably-sound arguments). Let \((P, V)\) is an interactive proof protocol for \(L \in \text{NP}\). A resetting attack of a cheating prover \(P^*\) is defined as follows:

1. Let \(t = \text{poly}(n)\), uniformly select and fix \(t\) random-tapes \(r_1, \cdots, r_t\) for \(V\), resulting in deterministic strategies \(V^{(j)}(x) = V_{x, r_j}\), defined by \(V_{x, r_j}(\alpha) = V(x, r_j, \alpha)^*\), where \(x \in \{0,1\}^n\) and \(j \in [t]\). Each \(V^{(j)}(x)\) is called an incarnation of \(V\).

2. \(P^*\) is allowed to initiate \(\text{poly}(n)\)-many interactions with the \(V^{(j)}(x)\). The activity of \(P^*\) proceeds in rounds. In each round, \(P^*\) chooses \(x \in \{0,1\}^n\) and \(j \in [t]\), defines \(V^{(j)}(x)\), and conducts a complete session with it.

We say that \((P, V)\) is a **resettably-sound** argument if for every polynomial-size resetting attack, the probability that in some session the corresponding \(V^{(j)}(x)\) has accepted and \(x \notin L\) is negligible.

2.8 Weak \((K, \gamma)\)-Collision Resistance [BKP18]

Definition 8. Let \(K(\cdot, \cdot)\) be a function and \(\lambda \in \mathbb{N}\) be a security parameter. We say that \(H\) is weakly \((K, \gamma)\)-collision-resistant if for any probabilistic \(\gamma^{O(1)}\)-time \(A(\text{possibly } \gamma = \lambda^{\omega(1)})\) and any sequence of polynomial-size advice \(\{z_\lambda\}_{\lambda \in \mathbb{N}}\), there is a negligible function \(\mu\), such that for any \(\lambda \in \mathbb{N}\), letting \(K = K(\lambda, |z_\lambda|)\),

\[
\Pr \left[ \begin{array}{c}
\forall i \neq j : X_i \neq X_j \\
Y_1 = \cdots = Y_K
\end{array} \right] \begin{array}{c}
hk \leftarrow H.\text{Gen}(1^\lambda) \\
(X_1, \cdots, X_K) \leftarrow A(hk; z_\lambda) \\
\forall : Y_i = H.\text{hash}(hk, X_i)
\end{array} \leq \mu(n).
\]

2.9 Weak Memory Delegation [BKP18]

Definition 9. Let \(\lambda \in \mathbb{N}\) be a security parameter, a two-message memory delegation scheme consists of algorithms \(\text{MD} = (\text{MD.\text{Gen}}, \text{MD.\text{Mem}}, \text{MD.\text{Query}}, \text{MD.\text{Prove}}, \text{MD.\text{Ver}})\) satisfies:

*Here, \(V(x, r_j, \alpha)\) denotes the message sent by the strategy \(V\) on common input \(x\), random-tape \(r_j\), after seeing the message-sequence \(\alpha\).*
Weak Soundness for Computation-Time Bound

Correctness: There exists a universal polynomial $p(\cdot)$ such that for every security parameter $\lambda \in \mathbb{N}$, every $(M, t, y) \in \{0, 1\}^\lambda$, and every $D$ such that $M(D)$ outputs $y$ within $t$ steps, and $|D| \leq t \leq 2^\lambda$:

$$\Pr \left[ \text{MD.Ver}(pp, d, (M, t, y), vst, \pi) = 1 \middle| \begin{array}{l}
pp \leftarrow \text{MD.Gen}(1^\lambda) \\
d \leftarrow \text{MD.Mem}(pp, D) \\
(q, vst) \leftarrow \text{MD.Query}(1^\lambda) \\
\pi \leftarrow \text{MD.Prove}(pp, D, (M, t, y), q)
\end{array} \right] = 1,$$

where the prover $\text{MD.Prove}(pp, D, (M, t, y), q)$ runs in time $p(\lambda, t)$ and the verifier $\text{MD.Ver}(pp, d, (M, t, y), vst, \pi)$ runs in time $p(\lambda)$.

Weak Soundness for Computation-Time Bound $\bar{t}(\lambda)$: For every pair of PPT adversaries $(A_1, A_2)$ and polynomial-size advice $\{z_\lambda\}_{\lambda \in \mathbb{N}}$, there is a negligible function $\mu$, such that for every $t(\lambda) \leq \bar{t}^O(1)$, any ensemble of samplable entropic distributions $\{Y_\lambda\}_{\lambda \in \mathbb{N}}$ such that the min-entropy of $Y_\lambda$ is $\Omega(\lambda)$, letting $K = K(\lambda, |z_\lambda|, t)$,

$$\Pr \left[ \text{MD.Ver}(pp, d, (M, t, y), vst, \pi) = 1 \middle| \begin{array}{l}
pp \leftarrow \text{MD.Gen}(1^\lambda) \\
(d, M, st) \leftarrow A_1(pp; z_\lambda) \\
(q, vst) \leftarrow \text{MD.Query}(1^\lambda) \\
y \leftarrow Y_\lambda \\
\pi \leftarrow A_2(q, y; st)
\end{array} \right] \leq \mu(\lambda).$$

2.10 1-Hop Homomorphic Encryption [GHV10]

Definition 10. A scheme $(\text{Enc}, \text{Eval}, \text{Dec})$, where $\text{Enc}$, $\text{Eval}$ are probabilistic and $\text{Dec}$ is deterministic, is a semantically-secure, circuit-private, 1-hop homomorphic encryption scheme if it satisfies the following properties:

Perfect correctness: For any $n \in \mathbb{N}$, $x \in \{0, 1\}^n$ and circuit $C$:

$$\Pr[(ct, sk) \leftarrow \text{Enc}(x) : \hat{ct} \leftarrow \text{Eval}(ct, C) \land \text{Dec}_{sk}(\hat{ct}) = C(x)] = 1.$$

Semantic security: For any non-uniform PPT $A \in \{A_n\}_{n \in \mathbb{N}}$, and any pair of inputs $x_0, x_1 \in \{0, 1\}^{\text{poly}(n)}$ of equal length:

$$\Pr[b \leftarrow \{0, 1\}, ct \leftarrow \text{Enc}(x_b) : A_n(ct) = b] \leq \frac{1}{2} + \text{negl}(n).$$

Circuit privacy: Let $\mathcal{E}(x) = \text{Supp}(\text{Enc}(x))$ be the set of all legal encryptions of $x$, $\mathcal{E}_n = \bigcup_{x \in \{0, 1\}^n} \mathcal{E}(x)$ be the set legal encryptions for strings of length $n$, and $\mathcal{C}_n$ be the set of all circuits on $n$ input bits. There exists a (possibly unbounded) simulator $S_{\text{1hop}}$ such that:

$$\{C, \text{Eval}(c, C)\}_{n \in \mathbb{N}, C \in \mathcal{C}_n, c \in \{0, 1\}^n, c \in \mathcal{E}(x)} \approx \{C, S_{\text{1hop}}(c, C(x), |C|)\}_{n \in \mathbb{N}, C \in \mathcal{C}_n, x \in \{0, 1\}^n, c \in \mathcal{E}(x)};

$$

$$\{C, \text{Eval}(c, C)\}_{n \in \mathbb{N}, C \in \mathcal{C}_n, c \in \mathcal{E}(x)} \approx \{C, S_{\text{1hop}}(c, \perp, |C|)\}_{n \in \mathbb{N}, C \in \mathcal{C}_n, c \in \mathcal{E}(x)}.$$

Theorem 2.1. [BKP18] For any (arbitrary small) $\tau(\lambda) = \omega(\log \lambda)$, there exists $\bar{t}(\lambda) = \lambda^{\omega(1)}$ such that assuming a weakly $(K, \gamma)$-collision-resistant hash, for $K(\lambda, |z_\lambda|) = \text{poly}(\lambda, |z_\lambda|)$ and $\gamma(\lambda) = \lambda^\tau$, and quasi-$\text{poly}(\lambda)$-secure fully-homomorphic encryption, there exists a two-message memory-delegation scheme with weak soundness for computation-time bound $\bar{t}$. 

8
3 The Protocol

3.1 6-round Public-Coin Non-malleable Zero-Knowledge

In this section we give our construction of the 6-round public-coin non-malleable zero-knowledge protocol. We use the following building blocks:

- 2-round statistically binding commitment scheme: Com.
- 4-round public-coin honest-extractable non-malleable commitment scheme: NMCom.
- 4-round “delay-input” special-soundness public-coin WIPOK be instantiated with the FLS protocol [FLS99]: sWI.
- 6-round “special purpose” universal argument: sUA.

Consider a language \( L \in \text{NP} \) and a security parameter \( n \), and let the prover and verifier receive a common input \( x \in \{0,1\}^n \), \( \text{id} \in \{0,1\}^n \). The auxiliary input to the prover is a \( \text{NP} \) witness \( w \) such that \( R_L(x,w) = 1 \). Because the 2-round statistically binding commitment scheme Com will be used as the basic tool for the subprotocol of NMCom, sWI and sUA. For simplicity, we omit the first round messages of Com used for NMCom, sWI and sUA, which are public coins. Let \( \{w_1, w_2, w_3\} \) be the transcript of the sWI, \( (nm_1, nm_2, nm_3) \) be the transcript of the commitment to the witness \( w \) computed using NMCom under tag \( \text{id} \), and \( (\beta, \gamma, \delta) \) be the transcript of the “encrypted” UA of the sUA.

Now, we start by describing a variant of Barak’s relation, which we denote by \( R_{\text{Sim}} \). Let \( H_n \) be a family of hash functions and \( h \in H_n: \{0,1\}^* \rightarrow \{0,1\}^n \). The relation \( R_{\text{Sim}} \) and the language \( L_{\text{ua}} \) are described in Fig. 2. Roughly speaking, we say \((h,c,r),(M,\rho_1,y)\) \( \in R_{\text{Sim}} \) iff \( M \in \{0,1\}^{n^{|h|}} \), \( \rho_1 \in \{0,1\}^{poly(n)} \), \( y \leq |r| - n \), and \( c = \text{Com}(h(M),\rho_1) \) such that \( M(y) = r \) within \( n^{|h|} \) steps. We say the “encrypted” UA transcript \((h,c,r,\beta,\gamma,\delta) \in L_{\text{ua}} \), iff there exist \((M,\rho_1,y,\beta,\rho_2,\delta,\rho_3) \) such that \( c = \text{Com}(h(M),\rho_1) \), \( \beta = \text{Com}(\beta,\rho_2) \), \( \delta = \text{Com}(\delta,\rho_3) \) and \((h,\beta,\gamma,\delta) \) is an accepting transcript of universal argument proving the statement: \((h,c,r),(M,\rho_1,y)\) \( \in R_{\text{Sim}} \).

Next, we give our 6-round NMZK argument protocol description Fig. 3, and an high-level description of our 6-round public-coin NMZK argument is described in Fig. 1. In stage one, the prover and the verifier generate the trapdoor statement \((h,c,r)\) by running Barak’s protocol and compute the first non-malleable commitment to the witness \( w \) under the identity \( \text{id} \). More specifically, in the first round, the verifier sends a random hash function \( h \xleftarrow{\text{R}} H \) where \( h: \{0,1\}^* \rightarrow \{0,1\}^n \) and random coins \( \rho \in \{0,1\}^{poly(n)} \) which will be used as the first messages of the commitments Com for NMCom, sWI and sUA. In the second round, the honest prover computes \( c = \text{Com}(0^n,\rho_1) \) using Com and the first round message \( nm_1^1(w,s_1) \) using NMCom and sends \( c, nm_1 \) to \( V \). In the third round, the verifier computes \( r \xleftarrow{\text{R}} \{0,1\}^{2n} \) and the second round message \( nm_2^2 \xleftarrow{\text{R}} \{0,1\}^n \) using NMCom, and sends \( r, nm_2^1 \) to \( P \).

In stage two, \( P \) and \( V \) run the three round sWI, NMCom and “encrypted” sUA in parallel, where \( P \) will compute a second non-malleable commitment to the witness \( w \) under the identity \( \text{id} \). More specifically, in the fourth round, the honest prover computes the third round message \( nm_3^3(w,s_2) \) to complete the first non-malleable commitment, the first round message \( nm_1^2(w,s_3) \) of the second non-malleable commitment, the first “encrypted” UA message \( \hat{\beta} = \text{Com}(0^n,\rho_2) \) and the first sWI message \( w_1 \) and sends \( (\hat{\beta},w_1, nm_1^1, nm_2^2) \) to \( V \). In the fifth round, the verifier computes three public-coins \( nm_1^2, \gamma, w_2 \xleftarrow{\text{R}} \{0,1\}^n \) and sends \( (\gamma, w_2, nm_1^2) \) to \( P \). In the sixth round, the honest prover computes the third round message \( nm_3^2(w,s_4) \) to complete the second non-malleable
• \((h, c, r, \hat{\beta}, \gamma, \hat{\delta})\) is the transcript computed using the “special purpose” universal argument.

• \((\text{nm}_1^1, \text{nm}_2^1, \text{nm}_3^1)\) is the transcript of the first commitment to the witness \(w\) computed using the honest-extractable non-malleable commitment scheme \(\text{NMCom}\) under the identity \(id\).

• \((\text{nm}_1^2, \text{nm}_2^2, \text{nm}_3^2)\) is the transcript of the second commitment to the witness \(w\) computed using the honest-extractable non-malleable commitment scheme \(\text{NMCom}\) under the identity \(id\).

• \((\text{wi}_1, \text{wi}_2, \text{wi}_3)\) is the transcript of the sWI proving the knowledge of the witness \(w\) for \(x \in L\) and the opening of the two non-malleable commitments to \(w\) or the knowledge of the opening of the “encrypted” \(\text{UA}\), i.e., \((\hat{\beta}, \gamma, \hat{\delta})\), which is an accepting \(\text{UA}\) proof for the statement \((h, c, r) \in L_{\text{sim}}\).

Figure 1: 6-round public-coin NMZK

\[
\begin{array}{c|c|c}
P_{\text{nmzk}} & h & \rho \\
\hline
& c & \text{nm}_1^1 \\
\hline
& r & \text{nm}_2^1 \\
\hline
\hat{\beta} & \text{nm}_1^1 & \text{nm}_2^1 & \text{wi}_1 \\
\gamma & \text{nm}_2^2 & \text{wi}_2 \\
\hat{\delta} & \text{nm}_3^2 & \text{wi}_3 \\
\end{array}
\]

commitment, \(\hat{\delta} = \text{Com}(0^n, \rho_3)\) to complete the “encrypted” \(\text{UA}\), \(\text{wi}_3\) to complete the sWI and sends \((\hat{\delta}, \text{wi}_3, \text{nm}_3^2)\) to \(V\).

Finally, the verifier accepts if the transcript \((\text{wi}_1, \text{wi}_2, \text{wi}_3)\) is an accepting proof for the OR statements: \(\exists (w, \text{dec}_1, \text{dec}_2)\) s.t. \(R_L(x, w) = 1\) and \((w, \text{dec}_1)\) is the decommitment for the transcript \((\text{nm}_1^1, \text{nm}_2^1, \text{nm}_3^1)\) and \((w, \text{dec}_2)\) is the decommitment for \((\text{nm}_1^2, \text{nm}_2^2, \text{nm}_3^2)\) or \(\exists (M, \rho_1, y, \beta, \rho_2, \delta, \rho_3)\) s.t. \((h, c, r, \hat{\beta}, \gamma, \hat{\delta}) \in L_{\text{ua}}\).

**Theorem 3.1.** Assuming the existence of collision-resistant hash functions, the protocol in Fig 3 is a 6-round public-coin non-malleable zero-knowledge argument of knowledge system for \(NP\).

**Proof 1.** The definition of the language in Fig 2 is essentially the same as the definition in [PR05b], so the protocol in Fig 3 is also a public-coin zero-knowledge argument. The completeness and soundness follow quite naturally.

**Completeness.** Roughly speaking, the completeness can be obtained from the correctness of \(\text{NMCom}\) and the completeness of \(\text{sWI}\).

**Soundness.** The soundness of protocol can be obtained from the binding property of the \(\text{NMCom}\) and the soundness of \(\text{sWI}\) used in stage two. In particular, assume \(x \notin L\), if there exists a PPT cheating \(P^*\) which can convince the verifier in stage two. From the definition of \(L_{\text{ua}}\), there must exist an accepting universal argument transcript \((h, \beta, \gamma, \delta)\) which can prove the rela-
Relation $R_{sim}$:

For a triplet $\langle h, c, r \rangle \in \mathcal{H}_n \times \{0,1\}^n \times \{0,1\}^{2n}$, we say a relation $R_{sim}(\langle h, c, r \rangle, (M, \rho_1, y)) = 1$, iff $M \in \{0,1\}^{n^{\omega(1)}}, \rho_1 \in \{0,1\}^{poly(n)}, |y| \leq |r| - n$;

$c = \text{Com}(h(M), \rho_1)$ such that $M(y) = r$ within $n^{\omega(1)}$ steps.

Language $L_{ua}$:

We say $\langle h, c, r, \beta, \gamma, \delta \rangle \in L_{ua}$, iff there exist $(M, \rho_1, y, \beta, \rho_2, \delta, \rho_3)$ such that

- $M \in \{0,1\}^{n^{\omega(1)}}, \rho_1, \rho_2, \rho_3 \in \{0,1\}^{poly(n)}, |y| \leq |r| - n$;

- $c = \text{Com}(h(M), \rho_1), \beta = \text{Com}(\beta, \rho_2), \delta = \text{Com}(\delta, \rho_3)$;

- $\langle h, c, r, (M, \rho_1, y) \rangle \in R_{sim}$.

Figure 2: The languages used in public-coin NMZK

We say $\langle h, c, r, \beta, \gamma, \delta \rangle \in L_{ua}$ is true except with negligible probability. That is, there exists a PPT machine $M$ on input a short bit string $y$ (of length bounded in $n$) which can predict the challenge message $r$ (length of $2n$). However, this is information theoretically impossible. Thus, we reach a contradiction through violating the soundness of Barak's protocol.

**NMZK.** Now we sketch how to build the simulator-extractor to prove the non-malleable zero-knowledge. First, we construct a PPT simulator $S$ that simulates the view of $A$ but does not extract witnesses in the right session. Then, we construct a PPT simulator-extractor $SE$ via intermediate simulator $S$ that simulates the view of $A$ and extracts the witness from the extractable non-malleable commitment $NMCom$.

More specifically, $S$ internally invokes $A$ and interacts with $A$ as honest prover and honest verifier in the following way. To simulate the view in the right interaction, $S$ simply follows the honest verifier strategy. To simulate the view in the left interaction, $S$ commits a dummy string (i.e., $0^n$) by invoking $NMCom$ twice to generate the transcripts $(nm_1^1(0^n), nm_1^2, nm_1^3(0^n))$ and $(nm_1^2(0^n), nm_2^1, nm_2^3(0^n))$ and uses the code description of the adversary $A$ as the fake witness in a straight-line manner to generate the transcript of the “special purpose” universal argument $\langle h, c, r, \beta, \gamma, \delta \rangle$ and $\langle wi_1, wi_2, wi_3 \rangle$. We denote the code description of $A$ as $M$ and the code description of the honest verifier as $V$. In the first stage, upon receiving the hash function $h$ from the adversary $A$, $S$ sends $c = \text{com}(h(M, V), \rho_1)$ and $nm_1^1(0^n)$ to $A$ and receives the corresponding responses $r, nm_1^2$. In the second stage, because $S$ has the witness for statement $\langle h, c, r \rangle$, it can complete the encrypted universal argument proof and special WI as follows:

- The simulator $S$ completes the first non-malleable commitment by computing $nm_1^3(0^n)$ and executes the second non-malleable commitment by computing $nm_2^1(0^n)$. Next, $S$ invokes the underlying PCP system to generate the PCP proof and then constructs the Markle tree $HT$ of this proof under the hash function $h$. Denote the root of the Markle tree as $\beta$, and then it computes $\beta = \text{Com}(\beta, \rho_1)$. Finally, $S$ acts as an honest prover to compute $wi_1$ and sends $(\beta, wi_1, nm_3^1, nm_3^2)$ to $A$.

- Upon receiving the fifth message $\langle \gamma, wi_2, nm_1^3 \rangle$ from $A$, $S$ invokes PCP system on input $\langle (h, c, r), \gamma \rangle$
Finally, the simulator $S$ sim-view will compute a second non-malleable commitment to the witness $P$.

Stage one: $P$ and $V$ generate the trapdoor statement by running Barak’s protocol and compute the first non-malleable commitment to the witness $w$ under the identity $id$.

1. The verifier computes $h \leftarrow R H_n, \rho \leftarrow \{0,1\}^{poly(n)}$ and sends $h, \rho$ to $P$.
2. The prover computes $c = Com(0^n, \rho_1)$ using $Com$ and the first round message $nm_1^1(w, s_1)$ using $NMCom$, and sends $c, nm_1^1$ to $V$.
3. The verifier computes $r \leftarrow R \{0,1\}^{2n}$ and the second round message $nm_2^1 \leftarrow R \{0,1\}^n$ using $NMCom$, and sends $r, nm_2^1$ to $P$.

Stage two: $P$ and $V$ run the three round $sWI$, $NMCom$ and “encrypted” $sUA$ in parallel, where $P$ will compute a second non-malleable commitment to the witness $w$ under the identity $id$.

4. The prover computes
   - the third round message $nm_3^1(w, s_2)$ to complete the first non-malleable commitment and the first round message $nm_2^1(w, s_3)$ using $NMCom$ under the identity $id$,
   - $\hat{\beta} = Com(0^n, \rho_2)$ using $Com$,
   - $wi_1$ using $sWI$,
   and sends $(\hat{\beta}, wi_1, nm_3^1, nm_1^1)$.
5. The verifier computes $nm_1^2 \leftarrow R \{0,1\}^n, \gamma \leftarrow R \{0,1\}^n, wi_2 \leftarrow R \{0,1\}^n$ and sends $(\gamma, wi_2, nm_1^2)$.
6. The prover computes
   - the third round message $nm_3^2(w, s_4)$ to complete the second non-malleable commitment,
   - $\hat{\delta} = Com(0^n, \rho_3)$ using $Com$ and $wi_3$ using $sWI$,
   and sends $(\hat{\delta}, wi_3, nm_3^2)$.

$V$ accepts if the transcript $(wi_1, wi_2, wi_3)$ is an accepting proof for the following statements:

- $\exists (w, dec_1, dec_2) s.t. R_L(x, w) = 1$ and $(w, dec_1)$ is the decommitment for the transcript $(id, nm_1^1, nm_2^1, nm_3^1)$ and $(w, dec_2)$ is the decommitment for the transcript $(id, nm_1^2, nm_2^2, nm_3^2)$ or
- $\exists (M, \rho_4, y, \beta, \rho_2, \delta, \rho_5) s.t. (h, c, r, \hat{\beta}, \gamma, \hat{\delta}) \in L_{ua}$.

Figure 3: 6-round public-coin non-malleable zero-knowledge argument

to generate $PCP$ queries $Q$, and computes $PCP$ answers $\delta = \{q, \sigma_q, auth_q(\sigma)\}_{q \in Q}$, where $\sigma_q$ is the $q$-th bit of $\sigma$ and $auth_q(\sigma)$ is the certificate path of $HT$ for $\sigma_q$. Then it computes $\hat{\delta} = Com(\delta, \rho_2)$. Next, $S$ completes the second non-malleable commitment by computing $nm_3^2(0^n)$ and computes $wi_3$ such that $(wi_1, wi_2, wi_3)$ is a proof for the statement $(h, c, r, \hat{\beta}, \gamma, \hat{\delta}) \in L_{ua}$.

Now, $S$ sends $(\delta, wi_3, nm_3^2)$ to $A$.

Finally, the simulator $S$ outputs the view of the adversary $A$. We denote the simulated view as $\{sim-view_A(1^n, x, z)\}_{n \in N, x \in L \cap (0,1)^n, z \in \{0,1\}^*}$ and the real view as $\{real-view_A(1^n, x, z)\}_{n \in N, x \in L \cap (0,1)^n, z \in \{0,1\}^*}$.
**Analysis of the Simulator.** From the language $L_{ua}$ defined in Fig 2, we know that if the simulator can non-black box access the adversary $A$, then it uses the code description of the adversary $A$ as witness can complete the proof for the statement in $L_{ua}$ in stage two. Thus, the correctness of $S$ can be directly obtained from the completeness of $sUA$, $NMCom$ and $sWI$.

The computational indistinguishability of $\{\text{real-view}_A\}$ and $\{\text{sim-view}_A\}$ can be obtained from the computational-hiding of the $Com$ and $NMCom$ and the witness-indistinguishability of the $sWI$. Roughly speaking, we can consider the following hybrid experiments. Let $Hyb_0$ denotes the real experiment which outputs the real view $\{\text{real-view}_A\}$ and $Hyb_3$ denotes the simulated experiment by the simulator $S$ which outputs the simulated view $\{\text{sim-view}_A\}$.

We define $Hyb_1$ in the same way as the $Hyb_0$ except that the simulator $S$ use both the witness $w$ and the adversary’s code $M$ to complete the protocol execution. More specially, in $Hyb_1$, $S$ uses the fake witness $\tilde{w}$ to complete the “special-purpose” universal argument in the left interaction where $c = \tilde{Com}(\tilde{h}(M,V),\rho_1), \tilde{\beta} = \tilde{Com}(\tilde{\beta},\rho_2)$ and $\delta = \tilde{Com}(\delta,\rho_3)$. However, in stage two $S$ still prove the OR statement using the witness $w$ and $dec_1$ and $dec_2$.

Thus, the computational indistinguishability of the output of $Hyb_0$ and $Hyb_1$ can be followed from the computational-hiding of the $Com$.

We define $Hyb_2$ in the same way as the $Hyb_1$ except that in stage two $S$ prove the OR statement using the witness $(h,(M,V),\rho_1,c,\beta,\rho_2,\delta,\rho_3)$. Thus, the computational indistinguishability of the output of $Hyb_1$ and $Hyb_2$ can be followed from the witness-indistinguishability of the $sWI$.

The only difference between the output of $Hyb_2$ and $Hyb_3$ is that, in the former the transcripts $(nm_1, nm_2, nm_3)$ and $(\tilde{nm}_1, \tilde{nm}_2, \tilde{nm}_3)$ are generated using the value $w$, and in the latter, the transcripts $(nm_1, nm_2, nm_3)$ and $(\tilde{nm}_1, \tilde{nm}_2, \tilde{nm}_3)$ are generated using $0^n$.

Thus, the computational indistinguishability of the output of $Hyb_2$ and $Hyb_3$ can be followed from the computational-hiding of the $NMCom$.

Combining the above, we can argue that the output of $Hyb_0$ and $Hyb_3$ are computationally indistinguishable, that is $\{\text{real-view}_A\} \approx \{\text{sim-view}_A\}$.

**Simulator-Extractor SE.** We use $SE$ to simulate the view of $A$ by executing $S$ as the first part of its output. Now considering the right interaction is accepted and $\tilde{id}$ is different from $id$ in the left interaction, we will show that the extracted witness is indeed the witnesses of the statement proved in the right interaction.

Observe that in the experiment $Hyb_0$, the simulator $S$ holds the real witnesses of the left interaction and just acts as an honest prover in the interaction and an honest verifier in the right interaction. Then following from the soundness of the $WIPOK$ and the honest-extractable property of the $NMCom$, we can conclude that for any accepting right interaction and the right $\tilde{id}$ different from the left $id$, $A$ commits successfully a real witness in the $NMCom$ except with negligible probability. That is $A$ never cheats in $Hyb_0$, and the simulator-extractor $SE$ can extract the witness by rewinding the first (or the second) non-malleable commitment from the third round to the second round (or the sixth round to the fifth round) except with negligible probability. In order to prove the non-malleable zero-knowledge, we need to prove that $A$ never cheats in $Hyb_1$ and $Hyb_2$ also, which means that $A$ will commit the same witness $\tilde{w}$ in the non-malleable commitments on the right.

Recall that the adversary $A$ controls the message executing in the two sides and there are many adversarial schedules. According to when the adversary $A$ sends the fourth round message on the right, we divide the schedules into two cases and prove them separately.

**Schedule 1:** The adversary $A$ sends the fourth round message on the right before it receives the sixth round message on the left. In such schedule, we observe that the fifth round messages on the left are sent by the adversary $A$ to the prover. So we only need to focus on the first four rounds
messages of the left which may interrupt the right side.

Recall that the only difference of the message received in the first four rounds of the left side between $Hyb_0$ and $Hyb_1$ is that, in the former, the commitments $c = \text{Com}(0^n, \rho_1), \hat{\beta} = \text{Com}(0^n, \rho_2)$, in the latter the commitments $c = \text{Com}(h(M, V), \rho'_1), \hat{\beta} = \text{Com}(\beta, \rho'_2)$. From Naor's commitment scheme $\text{Com}$, we know that the second round message (i.e., $c, \hat{\beta}$) is generated by the prover using a pseudo-random generator. Even if we rewind this commitment, we still can't obtain its commitment value. So we can conclude that the right rewinding on the first non-malleable commitment does not interrupt the security of the computational hiding commitment scheme $\text{Com}$. Thus, we can prove that $A$ never cheats in $Hyb_1$ except with negligible probability, otherwise we can break the computational hiding property of $\text{Com}$.

Next, the only difference of the message received in the first four rounds of the left side between $Hyb_1$ and $Hyb_2$ is the message $w_{i_1}$. From the delay-input $sWI$, we know that $w_{i_1}$ are just the commitments generated by $\text{Com}$ which is independent of the statement to be proved. Thus, we can prove that $A$ never cheats in $Hyb_2$ except with negligible probability, otherwise we can break the computational hiding property of $\text{Com}$.

Next, recall that the only difference of the message received in the first four rounds of the left sides between $Hyb_2$ and $Hyb_3$ is that, in the former, the messages $nm_1^1, nm_1^3, nm_1^2$ are generated using the value $w$, and in the latter the messages $nm_1^1, nm_3^1, nm_2^1$ are generated using the value $0^n$. Here, we further consider the two types of schedules.

- The first type is that the third and the fourth round messages (which contains $\tilde{nm}_2^1$ and $\tilde{nm}_1^1$) on the right cover the third and fourth round messages (which contains $nm_3^1$ and $nm_1^1$) on the left but do not cover the second round messages (which contains $nm_1^1$) on the left. In such condition, the advantage of the adversary $A$ is equal to the advantage of the adversary $A$ who execute the right four rounds messages in parallel with the left four rounds messages. In such case we can see the adversary $A$ as a synchronous adversary. Because the three-round non-malleable commitment we use is a non-malleable against a synchronizing adversary (see \cite{GPR16}), we can argue that if in $Hyb_2$ the adversary commits the values $\tilde{w}$ in the transcript $(\tilde{nm}_1^1, \tilde{nm}_3^1, \tilde{nm}_2^1)$, then in $Hyb_3$ the adversary will also commit the same value $\tilde{w}$ in the transcript $(nm_1^1, nm_2^1, nm_3^1)$ except with negligible probability, otherwise we can break the non-malleable property of the $\text{NMCom}$ (we note that this idea was first used in \cite{COSV17}).

- The second type includes the schedules other than the above. That is the right four rounds messages are executed not in parallel with the left four rounds messages. In such case, we can reduce the security to the computational-hiding of the non-malleable commitment scheme. Because when we rewind the right three round non-malleable commitment to extract the commitment value $\tilde{w}$, the left three round non-malleable commitment will not be rewound and its computational-hiding is preserved. If the adversary $A$ cheats in $Hyb_3$, which means that in $Hyb_2$ the simulator-extractor $SE$ can not extract the real witness except with negligible probability. Recall that the adversary $A$ never cheats in $Hyb_2$, which means $SE$ can extract the real witness except with negligible probability. So in $Hyb_2$ and $Hyb_3$ the commitment value which we extract are different with noticeable probability. Because the output of $Hyb_2$ and $Hyb_3$ are computationally indistinguishable and $SE$ is an PPT machine, thus we can obtain a contradiction.

**Schedule 2:** The adversary $A$ sends the fourth round message on the right after it received the sixth round message on the left. This case can be easily proved. Because in such condition, the second non-malleable commitment is fully executed after the execution of the left protocol. As before, we can reduce the security to the computational-hiding of the non-malleable commitment.
scheme \textit{NMCom} and the non-interactive commitment \textit{Com} and the witness-indistinguishability of the \textit{sWI}. More specifically, for $\text{Hyb}_0$ and $\text{Hyb}_1$, if the adversary $A$ cheats in $\text{Hyb}_1$, then we can break the computational-hiding of the non-interactive commitment \textit{Com}. For $\text{Hyb}_1$ and $\text{Hyb}_2$, if the adversary $A$ cheats in $\text{Hyb}_2$, then we can break the witness-indistinguishability of the \textit{sWI}. For $\text{Hyb}_2$ and $\text{Hyb}_3$, if the adversary $A$ cheats in $\text{Hyb}_3$, then we can break the computational-hiding of the non-malleable commitment scheme \textit{NMCom}.

Put the three above together, we obtain that the extractor does not break the security of the left protocol no matter for the simulator or for the honest prover on the left. Because for the \text{sim} against polynomially adversaries, we should use the “error-correcting code” \textit{ECC} (i.e., with constant distance and with polynomial-time encoding and decoding) technique first appeared in \cite{BG08}. More specifically, we replace the commitment of the form $\text{Com}(\cdot)$ with $\text{Com}(h(\text{ECC}(M)))$ in stage 1. The language relation $R_{\text{sim}}$ we define in Fig. 2 should modify as follows: $R_{\text{sim}}((h,c,r),(M,s,y)) = 1$ iff $c = \text{Com}([\text{ECC}(M)], h(\text{ECC}(M)))$ and $M(y) = r$, where $h(\text{ECC}(M))$ is the root of the markle tree of the \text{ECC}(M) under the hash function $h$.

\subsection{3.2 3-round Non-malleable Zero-Knowledge}

In this section we give our construction of the 3-round non-malleable zero-knowledge argument protocol. We use the following building blocks:

- Non-interactive perfectly-binding commitment scheme: \textit{Com}.
- 3-round public-coin honest-extractable non-malleable commitment scheme: \textit{NMCom}.
- 2-round delayed-input statistical witness indistinguishable arguments: \textit{SWI}.
- 2-round weak memory delegation scheme for quasi-polynomial bounded computations: \textit{MD}.
- Semantically secure and circuit-private, 1-hop homomorphic encryption scheme: (\textit{Enc}, \textit{Eval}, \textit{Dec}).

For our propose, we introduce the following notation \cite{BKP18, BBK16} which was used to achieve the variant of the two-message memory delegation technique \cite{CKLR11} described in the introduction.

- For a well-formed string $\text{mes} = (c, \text{nm})$, denote by $M_{\text{mes}}$ a Turing machine as follows: On input the memory $D = V^*$, $M_{\text{mes}}$ parses $V^*$ as a Turing machine, runs $V^*$ on input $\text{mes}$, parses the result as $(r, \text{nm}_2, q, \text{ct}_{\text{st}}, \text{wi}_1)$, and outputs $r$.

- For a well-formed string $\text{param} = (\text{mes}, d, t, r, \pi, q)$, denote by $C_{\text{param}}$ a circuit with \text{param} hard-coded as follows: on input a verification state $\text{vst}$, $C_{\text{param}}(\text{vst})$ outputs

  - 1 if $\text{MD} \text{.Ver}(d, M_{\text{mes}}, t, r, \text{vst}, \pi) = 1$ or $(q, \text{vst}) \neq \text{MD} \text{.Query}(1^n)$. 

\begin{itemize}
  
\item
\end{itemize}
– 0 otherwise.

• Denote by \( C_1 \) a circuit of the same size as the circuit \( C_{\text{param}} \) that always returns 1.

\[
P_{\text{nmzk}} \xrightarrow{c, nm_1} V_{\text{nmzk}}
\[
\xrightarrow{r, nm_2, q, ct_{\text{vst}}, wi_1}
\[
\xrightarrow{nm_3, \hat{ct}, wi_2}
\]

• \((c, r)\) is the transcript of the “encrypted” two-message memory delegation, where \( c \) is a commitment to \((0^n, 0^{\log n})\), and \( r \) is the challenge message.

• \((nm_1, nm_2, nm_3)\) is the transcript of the commitment to the witness \( w \) computed using the honest-extractable non-malleable commitment \( \text{NMCom} \) under the identity \( id \).

• \((wi_1, wi_2)\) is the transcript of the SWI proving the knowledge of the opening of the non-malleable commitment to \( w \) such that \( R(x, w) \in R_L \) or the knowledge of \((d, t, \rho, \pi)\) such that \( c = \text{Com}(d, t; \rho) \) and \( \hat{ct} = \text{Eval}(C_{\text{param}}, ct_{\text{vst}}) \).

Figure 4: 3-round non-malleable zero-knowledge for NP

The formal description of 3-round non-malleable zero-knowledge argument of knowledge protocol is described in Fig. 5 and an high-level description of this protocol is described in Fig. 4.

**Theorem 3.2.** Assuming keyless multi-collision resistant hash functions, LWE and DDH (or QR or \( N^{\text{th}} \) residuosity), all quasi-polynomially hard, there exist 3-round non-malleable zero-knowledge arguments for \( \text{NP} \).

**Proof 2.** The completeness can be obtained from the correctness of \( \text{NMCom} \) and the completeness of SWI. The soundness and zero-knowledge essentially can be proved in the same way as the protocol in [BKP18]. For completeness, we briefly describe the proof, and we refer the reader to [BKP18] for more detail about this part.

**Soundness.** Assuming that \( x \notin L \), in order to pass the SWI with respect to an evaluated cipher \( \hat{ct} \) that decrypts to 1, the prover must know a digest \( d \), a time bound \( t \), and proof \( \pi \), such that \( C_{\text{param}}(\text{vst}) = 1 \). This, by definition, means that \((d, t, \pi)\) are such that the delegation verifier \( \text{Ver} \) is convinced that the digest \( d \) corresponds to a machine \( V^* \) such that \( V^*(c, nm_1) = r \). Intuitively, this implies that the prover manages to commit to a program that predicts the random string \( r \) before it was ever sent, which is unlikely. Specifically, we can construct an attacker from this prover which can sample a digest \( d \) and a computation \( M \) such that with noticeable probability \( \epsilon \) over a random output \( r \in \{0, 1\}^n \), it successfully produces a convincing proof consistently with \((M, r)\), which is enough to break the weak soundness of memory delegation in Definition 2.9. Here we will also rely on the semantic security of the encryption scheme to claim that the encrypted verification state \( ct_{\text{vst}} \) is hiding.

**NMZK.** Now we sketch how to build the simulator-extractor to prove the non-malleable zero-knowledge. First, we construct a PPT simulator \( S \) that simulates the view of \( A \) but does not extract
Common input: $x \in L$ and identity id $\in \{0,1\}^n$.

Auxiliary input to $P$: $w \in R_L(x)$.

$P$ and $V$ run the three round SWI, NMCom and “encrypted” MD in parallel.

1. The prover computes
   - $c = \text{Com}(0^n, 0^{\log \gamma(n)}; \rho)$, where $\gamma = n^{\log \log(n)}$,
   - the first round message $\text{nm}_1(w, s_1)$ using NMCom,
   and sends $(c, \text{nm}_1)$ to $V$.

2. The verifier computes
   - $r \leftarrow \{0,1\}^n$,
   - the second round message $\text{nm}_2 \leftarrow \{0,1\}^n$ using NMCom,
   - $(q, \text{vst}) \leftarrow \text{MD.Query}(1^n)$, where we assume w.l.o.g that vst consists of the coins of MD Query,
   - $(\text{ct}_{\text{vst}}, \text{sk}) \leftarrow \text{Enc}_{\text{sk}}(\text{vst})$, an encryption of the verification state,
   - the first round message $\text{wi}_1$ using the SWI,
   and sends $(r, \text{nm}_2, q, \text{ct}_{\text{vst}}, \text{wi}_1)$ to $P$.

3. The prover computes
   - the third round message $\text{nm}_3(w, s_2)$ to complete the non-malleable commitment,
   - $\hat{\text{ct}} \leftarrow \text{Eval}(C_1, \text{ct}_{\text{vst}})$, an evaluation of the constant one function,
   - the second round message $\text{wi}_2$ using the SWI,
   and sends $(\text{nm}_3, \hat{\text{ct}}, \text{wi}_2)$ to $V$.

$V$ accepts if $\text{Dec}_{\text{sk}}(\hat{\text{ct}}) = 1$ and the transcript $(\text{wi}_1, \text{wi}_2)$ is an accepting proof for the following statements:
   - $\exists (w, \text{dec} \in \{0,1\}^{\text{poly}(n)})$ s.t. $R_L(x, w) = 1$ and $(w, \text{dec})$ is the decommitment for the transcript $(\text{id}, \text{nm}_1, \text{nm}_2, \text{nm}_3)$ or
   - $\exists (d, \pi, \rho \in \{0,1\}^n, t < \gamma(n))$ s.t. $c = \text{Com}(d, t; \rho)$ and $\hat{\text{ct}} = \text{Eval}(C_{\text{param}}, \text{ct}_{\text{vst}})$ where $\text{param} = (\text{mes} = (c, \text{nm}_3), d, t, r, \pi, q)$.

Figure 5: 3-round non-malleable zero-knowledge argument of knowledge for NP witnesses in the right session. Then, we construct a PPT simulator-extractor $SE$ via intermediate simulator $S$ that simulates the view of $A$ and extracts the witnesses from the extractable non-malleable commitment $\text{NMCom}$.

More specifically, $S$ internally invokes $A$ and interacts with $A$ as honest prover and honest verifier in the following way. To simulate the view in the right interaction, $S$ simply follows the honest verifier strategy. To simulate the view in the left interaction, $S$ commits a dummy string (i.e., $0^n$) by invoking $\text{NMCom}$ to generate the transcripts $(\text{nm}_1(0^n), \text{nm}_2, \text{nm}_3(0^n))$, and uses the code description of the adversary $A$ as the fake witness in a straight-line manner to generate the transcript of the $(c, r, \text{q}, \text{ct}_{\text{vst}}, \hat{\text{ct}})$ and $(\text{wi}_1, \text{wi}_2)$.

More specifically, for a statement $x \in L$ where $|x| = n$, we denote the code description of $A$
as $M$ and the code description of the honest verifier as $V$, a polynomial bound $t(n) = n^{O(1)}$ on its running time. The simulator $S$ operates on the left side as follows:

- In the first round, $S$ computes the digest $d = \text{MD} \cdot \text{Mem}(1^n, (M, V))$ and then computes a commitment $c = \text{Com}(d, t; \rho)$ to the digest $d$ and $M$'s running time $t$ using random coins $\rho \xleftarrow{R} \{0,1\}^n$. Next, it computes the first message of the $\text{NMCom} \ nm_1 = nm_1(0^n)$ to a dummy string $0^n$ and sends $(c, nm_1)$ to $A$. Here, we set $\text{mes} = (c, nm_1)$.

- In the third round, upon receiving the responses $(r, nm_2, q, ct_{\text{vst}}, wi_1)$ from $A$, $S$ computes the proof $\pi = \text{MD} \cdot \text{Prov}(1^n, (M, V), (\text{mes}, t, r), q)$ for the memory computation that $\text{mes}$ on input the memory $(M, V)$ output $r$ within $t$ steps. Here, we set $\text{param} = (\text{mes} = (c, nm_1), d, t, r, \pi, q)$. Now, $S$ computes $ct_1 = \text{Eval}(C_{\text{param}}, ct_{\text{vst}})$ and the third message of the $\text{NMCom} nm_3 = nm_3(0^n)$ to the dummy string $0^n$. Next, $S$ computes the second $\text{SWI}$ message $wi_2$ for the statement that it knows the opening of the non-malleable commitment to $w$ such that $R(x, w) \in R_L$ or $c = \text{Com}(d, t; \rho)$ and $ct = \text{Eval}(C_{\text{param}}, ct_{\text{vst}})$ using the witness $(d, t, \rho, \pi)$. Then, $S$ sends $(nm_3, ct, wi_2)$ to $A$.

Finally, the simulator $S$ outputs the view of the adversary $A$. We denote the simulated view as $\{\text{sim-view}_A(1^n, x, z)\}_{n \in N, x \in L \cap (0,1)^n, z \in \{0,1\}^*}$ and the real view as $\{\text{real-view}_A(1^n, x, z)\}_{n \in N, x \in L \cap (0,1)^n, z \in \{0,1\}^*}$.

**Analysis of the Simulator.** From the definition of the Turing machine $M_{\text{mes}}$ and the circuit $C_{\text{param}}$, if the simulator can non-black box access the adversary $A$, then it uses the code description of the adversary $A$ to predict the challenge string $r$ except with negligible probability. Therefore, the correctness of $S$ can be directly obtained from the correctness of $\text{MD}$, the perfect correctness of the 1-hop homomorphic encryption scheme and the completeness of the $\text{SWI}$.

The computational indistinguishability of $\{\text{real-view}_A\}$ and $\{\text{sim-view}_A\}$ can be obtained from the computational-hiding of the $\text{Com}$ and $\text{NMCom}$ and the witness-indistinguishability of the $\text{SWI}$. Roughly speaking, we consider a simulator $S$ that it uses both the witness $w$ and the adversary's code $M$ to complete the protocol. Now, we define the hybrid experiments required to prove the zero-knowledge.

Let $\text{Hyb}_0$ denotes the real experiment which outputs the real view $\{\text{real-view}_A\}$. In particular, for $x \in L$ and $(x, w) \in R_L$, the simulator acts as an honest prover who uses the witness $w$ on the left to interactive with the adversary $A$, and acts as an honest verifier in the right to interactive with $A$.

The hybrid $\text{Hyb}_1$ is identical to $\text{Hyb}_0$ except that the simulator $S$ uses the trapdoor $M$ to compute the commitment $c = \text{Com}(d, t; \rho)$ to the digest $d$ and $M$'s running time $t$. It is easy to see that the computational indistinguishability of the output of $\text{Hyb}_0$ and $\text{Hyb}_1$ can be followed from the computational-hiding of the $\text{Com}$.

The hybrid $\text{Hyb}_2$ is identical to $\text{Hyb}_1$ except that the simulator $S$ computes $ct$ by invoking $\text{Eval}(C_{\text{param}}, ct_{\text{vst}})$ instead of $\text{Eval}(C_1, ct_{\text{vst}})$. The computational indistinguishability of the output of $\text{Hyb}_2$ and $\text{Hyb}_1$ can be followed from the circuit privacy of the 1-hop homomorphic encryption. More specifically, if the query is inconsistent with the query $q$, i.e., $(q, vst^*) \neq \text{MD} \cdot \text{Query}(1^n)$, then by the definition in the beginning it holds that $\text{Eval}(C_{\text{param}}, vst^*) = 1$. Thus, for an illegal encryption $ct^*$, by the circuit privacy, we have that $\text{Eval}(C_{\text{param}}, ct^*) \approx S_{1 \text{hop}}(ct^*, \bot, |C_{\text{param}}|) \equiv S_{1 \text{hop}}(ct^*, \bot, |C_1|) \approx S_{1 \text{hop}}(ct_{\text{vst}},\bot,|C_1|) \approx \text{Eval}(C_1, ct_{\text{vst}})$. Otherwise, for any $vst$ which is consistent with the query $q$, by the perfect completeness of the delegation scheme it holds that $\text{Eval}(C_{\text{param}}, vst) = 1$. Also, by the circuit privacy, we have that $\text{Eval}(C_{\text{param}}, ct_{\text{vst}}) \approx S_{1 \text{hop}}(ct_{\text{vst}},\text{Eval}(C_{\text{param}}, vst),|C_{\text{param}}|) \equiv S_{1 \text{hop}}(ct_{\text{vst}},C_1,|C_1|) \approx \text{Eval}(C_1, ct_{\text{vst}})$.
The hybrid $Hyb_3$ is identical to $Hyb_2$ except that the simulator $S$ uses the witness $(d, t, \rho, \pi)$ instead of $w$ to complete the SWI on the left. It is easy to see that this hybrid is statistically witness indistinguishable from $Hyb_2$.

The hybrid $Hyb_4$ is identical to $Hyb_3$ except that the simulator $S$ computes the non-malleable commitment $(nm_1, nm_2, nm_3)$ to a dummy string $0^n$ instead of the witness $w$. Thus, the computational indistinguishability of the output of $Hyb_4$ and $Hyb_3$ can be followed from the computational-hiding of the NMCom.

Combining the above, we can argue that the output of $Hyb_0$ and $Hyb_4$ are computationally indistinguishable. The hybrid experiment $Hyb_4$ is identical to with the simulator $S$ we describe above, therefore we get $\{\text{real-view}_A\} \approx \{\text{sim-view}_A\}$.

**Simulator-Extractor SE.** We use SE to simulate the view of $A$ by executing $S$ as the first part of its output. Now considering the right interaction is accepted and $\hat{id}$ is different from $id$ in the left interaction, we will show that the extracted witness is indeed the NP witnesses of the statement proved in the right interaction.

Observe that in the experiment $Hyb_0$, the simulator $S$ holds the real witnesses of the left interaction and just acts as an honest prover in the interaction and an honest verifier in the right interaction. Then following from the soundness of the SWI and the honest-extractable property of the NMCom, we can conclude that for any accepting right interaction and the right $\hat{id}$ different from the left $id$, $A$ commits successfully a real witness in the NMCom except with negligible probability. That is $A$ never cheats in $Hyb_0$ and the simulator-extractor SE can extract the witness by rewinding the non-malleable commitment from the third round to the second round except with negligible probability. In order to prove the non-malleable zero-knowledge, we need to prove that $A$ never cheats in the hybrid experiments from $Hyb_1$ to $Hyb_4$, which means that $A$ will commit the same witness $\tilde{w}$ in the non-malleable commitments on the right.

Recall that the adversary $A$ controls the message scheduling in the two sides due to the left and right protocol are both three rounds, so there are only two different type of adversarial schedules:

**Schedule 1:** The left protocol and the right protocol are executed in parallel. In this schedule, we can see $A$ as a synchronous adversary. We now give a series of hybrid proofs to argue the non-malleable property.

- The only difference of the message received on the left side between $Hyb_0$ and $Hyb_1$ is that, in the former $c = \text{Com}(0^n, 0^{\log n}; \rho)$ and in the latter $c = \text{Com}(d, t; \rho')$. We know that any non-malleable commitment scheme is non-malleable w.r.t to any non-interactive primitives, which means that the right rewinding on the non-malleable commitment does not interrupt the security of the non-interactive computational hiding commitment scheme $\text{Com}$. Thus, we can prove that $A$ never cheats in $Hyb_1$ except with negligible probability, otherwise we can break the computational hiding property of $\text{Com}$.

- The only difference of the message received on the left side between $Hyb_1$ and $Hyb_2$ is that, in the former $\hat{c}t = \text{Eval}(C_1, ct_{\text{ext}})$ and in the latter $\hat{c}t = \text{Eval}(C_{\text{param}}, ct_{\text{ext}})$. The same reason as before, we can argue that $A$ never cheats in $Hyb_2$ except with negligible probability from the circuit privacy of the non-interactive 1-hop homomorphic encryption.

- The only difference of the message received on the left side between $Hyb_2$ and $Hyb_3$ is that, in the former the adversary use the witness $w$ in SWI, and in the latter the adversary use the witness $(d, t, \rho, \pi)$ in SWI instead. Since SWI is statistically witness indistinguishable, we get that in $Hyb_3$ the adversary will also commit the same value $\tilde{w}$ in $(\tilde{nm}_1, \tilde{nm}_2, \tilde{nm}_3)$ except with negligible probability.
• The only difference of the message received on the left side between $Hyb_3$ and $Hyb_4$ is that, in the former the adversary commits the values $w$ in the transcript $(nm_1, nm_2, nm_3)$, and in the latter the adversary commits the values $0^n$ instead. Because the three-round non-malleable commitment we use is non-malleable against a synchronizing adversary, we can argue that if in $Hyb_3$ the adversary commits the values $\bar{w}$ in the transcript $(\bar{nm}_1, \bar{nm}_2, \bar{nm}_3)$, then in $Hyb_4$ the adversary will also commit the same value $\bar{w}$ in $(\bar{nm}_1, \bar{nm}_2, \bar{nm}_3)$ except with negligible probability, otherwise we can break the non-malleable property of the $NMCom$.

**Schedule 2:** The adversary $A$ sends the first round message on the right after it receives the third round message on the left. In such condition, the right protocol is fully executed after the execution of the left protocol, and the simulator-extractor can extract the right witness $\bar{w}$ using rewinding approach without interfering the left execution. Therefore, we can argue the adversary $A$ never cheats in $Hyb_1$ by reducing the security to the computational-hiding of $NMCom$, never cheats in $Hyb_2$ by reducing the security to the circuit privacy of the non-interactive 1-hop homomorphic encryption, never cheats in $Hyb_3$ by reducing the security to the witness-indistinguishability of SWI, except with negligible probability and never cheats in $Hyb_4$ by reducing the security to the computational-hiding of $Com$.

Put the above together, we obtain that the simulator-extractor does not break the security of the left protocol no matter for the simulator or for the honest prover on the left. Because for the simulator $S$ we have $\{\text{sim-view}_A(1^n, x, z)\} \approx \{\text{real-view}_A(1^n, x, z)\}$, thus we can conclude that for the simulator-extractor $SE$ it holds that $\{\text{sim-view}_A(1^n, x, z, \bar{w})\} \approx \{\text{real-view}_A(1^n, x, z, \bar{w})\}$ for any right interaction that is accepting and uses a different identity from the left interaction.

Combining the above analysis together, we complete the proof of three-round non-malleable zero-knowledge property. $\square$

### 4 Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61932019, No. 61379141 and No. 61772521), Key Research Program of Frontier Sciences, CAS (Grant No. QYZDB-SSW-SYS035), and the Open Project Program of the State Key Laboratory of Cryptology.

### References


[COSV17b] Michele Ciampi, Rafail Ostrovsky, Luisa Siniscalchi, and Ivan Visconti. Four-round concurrent non-malleable commitments from one-way functions. In Advances in Cryptology


