Yan *et al. Cybersecurity* (2018) 1:12
https://doi.org/10.1186/s42400-018-0014-7

Cybersecurity

RESEARCH Open Access

Concurrent non-malleable zero-knowledge and simultaneous resettable non-malleable zero-knowledge in constant rounds

Zhenbin Yan[1,2], Yi Deng[1,2*] and Yiru Sun[1,2]

Abstract

Concurrent non-malleable zero-knowledge (CNMZK) considers the *concurrent* execution of zero-knowledge protocols in a setting even when adversaries can simultaneously corrupt *multiple* provers and verifiers. As far as we know, the round complexity of all the constructions of CNMZK arguments for **NP** is at least $\omega(\log n)$. In this paper, we provide the first construction of a *constant-round concurrent non-malleable zero-knowledge argument* for every language in **NP**. Our protocol relies on the existence of families of *collision-resistant hash functions*, *one-way permutations* and *indistinguishability obfuscators*. As an additional contribution, we study the composition of two central notions in zero knowledge, the *simultaneously resettable zero-knowledge* and *non-malleable zero-knowledge*, which seemingly have *stronger* proved security guarantees. We give the first construction of a *constant-round simultaneously-resettable non-malleable zero-knowledge*. To the best of our knowledge, this is the first study to combine the two security concepts described above together in the zero-knowledge protocols.

Keywords: Zero-knowledge, Concurrent non-malleable zero-knowledge, Simultaneously resettable zero-knowledge, Concurrent security computation

Introduction

Zero-knowledge proof systems were introduced by Goldwasser, Micali and Rackofi in (1989). Informally, an interactive proof protocol is zero-knowledge if the prover can convince the verifier that a statement is true without revealing any information other than the fact itself. With such an intriguing nature, zero-knowledge proof has played a central role in the design and study of cryptographic protocols. The notion of *concurrent* zero knowledge(CZK) was first introduced by Dwork, Naor and Sahai (1998) to consider that many copies of the zero-knowledge protocol are executed simultaneously in an *asynchronous* network, where messages from different copies may be arbitrarily interleaved by the verifier. The notion of *non-malleable* zero knowledge(NMZK) was first introduced by Dolev, Dwork and Naor (2000) to consider the execution of zero-knowledge protocol in the setting where the *man-in-the-middle* adversary interacts with an honest prover in the left session and an honest verifier in the right session.

Concurrent Non-malleable Zero-Knowledge. By combining the *concurrent zero-knowledge* with the security against *man-in-the-middle* adversaries, Barak, Prabhakaran and Sahai (2006) introduced a stronger form of zero knowledge referred to as *concurrent non-malleable zero knowledge* (CNMZK). In such protocol, the adversary can complete control over the communication channel and participate in an *unbounded* number of concurrent executions. It guarantees that the proofs in the left sessions does not help the adversary to give proofs in the right sessions.

After the original protocol by (Barak et al. 2006), various other *concurrent non-malleable* ZK protocols have been obtained (Ostrovsky et al. 2008, 2010; Lin et al. 2010; Lin and Pass 2011; Orlandi et al. 2014; Kiyoshima 2015). Lin, Pass, Tseng and Venkitasubramaniam (2010) focused on

*Correspondence: deng@iie.ac.cn
[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

enhancing the soundness property by combining the notation of *robust* non-malleable commitments introduced by Lin et al. (2009) with the *concurrently extractable* commitments (CECom) introduced by Micciancio et al. (2006). They showed a poly(n)-round CNMZK *proof* for all of **NP** based on *one way function* assumption and a $\widetilde{O}(log(n))$-round protocol based on the existence of *collision resistant hash-functions*(CRHFs). Recently, Orlandi et al. (2014) achieved the first *statistical* CNMZK *argument system*. In their protocol, they used a special kind of commitment scheme called "*mixed non-malleable commitment*" scheme based on the DDH assumptions. Very recently, Kiyoshima (2015) achieved a poly(n) rounds *statistical* CNMZK *argument system* only assuming the existence of *one-way functions*. In their protocol, instead of using a non-malleable commitment to commit the real witness (see (Barak et al. 2006; Lin et al. 2010)), they used a constant-round *k-robust one-one CCA-secure* commitment (Canetti et al. 2010; Lin and Pass 2012; Kiyoshima 2014; Goyal et al. 2015) to commit a random string (e.g., 0^n).

However, we observe that the round complexity of all the above protocols based on the standard assumptions is at least $\widetilde{O}(\log n)$ rounds. Indeed, in the standard model without set-up assumptions, Canetti, Kilian, Petrank and Rosen (2001) based on earlier works by (Kilian et al. 1998; Rosen 2000) have showed that any *black-box concurrent* zero-knowledge protocol require at least $\widetilde{\Omega}(\log n)$ rounds. It can be observed that the lower bound also holds for the *black-box concurrent non-malleable* zero-knowledge protocol. A breakthrough work was made by Barak in (2001), he proposed the first *non-black-box* simulation techniques and constructed the first *constant-round bounded* CZK *argument system* assuming the existence of CRHFs. Recently, Pandey, Prabhakaran and Sahai (2015) showed a new *non-black-box* simulation technique independent of the PCP theorem and constructed a 4-round CZK *argument system* based on the existence of CRHFs and *differing-input obfuscation* (**diO**)(Barak et al. 2001; Boyle et al. 2014; Ishai et al. 2015). Very recently, Chung, Lin and Pass (2015) achieved *constant-round* CZK with *non-uniform* soundness assuming the existence of CRHFs, OWP and iO (Barak et al. 2001; Garg et al. 2013) for **P**/poly. We stress that Ostrovsky, Persiano and Visconti in (2008) have showed a *constant-round concurrent non-malleable* zero-knowledge argument system for **NP** in the Bare Public-Key model. However, in this model each verifier have to register the *public key* in a public file during a *preprocessing stage* and the *secret key* is known only to itself. Thus, one natural question we ask in this work is:

Whether a constant rounds concurrent non-malleable zero-knowledge protocol in the standard model can be obtained?

Simultaneous Resettable Zero-Knowledge. The notion of *resettable zero-knowledge* (rZK) was first introduced by Canetti, Goldreich, Goldwasser and Micali (2000). It requires the zero-knowledge condition holds even when the verifier can reset the prover to reuse the previous randomness. From the definition, we can see that the security of *resettable zero-knowledge* is stronger than that of *concurrent zero-knowledge*, because a resetting verifier could emulate any *concurrent attack* in the CZK protocol. Subsequently, Barak, Goldreich, Goldwasser and Lindell (2001) introduced the notion of *resettably-sound* zero-knowledge (rsZK). It requires the *soundness* condition holds even when the prover can reset the verifier to use the *same* random tape in multiple concurrent executions. Following the two works above, a number of works have investigated the resettable security in zero-knowledge protocols (Deng et al. 2009; Cho et al. 2012; Garg et al. 2012; Chung et al. 2013b, 2014; Bitansky and Paneth 2015; Ostrovsky et al. 2015), which focused on either reducing the complexity assumptions or reducing the round complexity and so on. Recently, Chung et al. (2013a) presented a construction of the *simultaneous resettable* zero-knowledge protocol with polynomial rounds based on the minimal assumption of *one-way functions*. Very recently, Chongchitmate et al. (2017) showed a *constant-round simultaneous resettable zero-knowledge argument system* based on the work of Chung et al. (2015). Thus, another question in this work is:

Whether a constant rounds interactive protocol can be both simultaneous resettable zero-knowledge and non-malleable zero-knowledge ?

Our results

In this paper, we combine the forementioned approaches and answer the above question positively. In the main result, we construct the first *constant-round non-malleable concurrent* zero-knowledge argument system.

Theorem 1 *Assuming the existence of collision-resistant hash functions, one-way permutations and **iO** for **P**/poly (with slightly super-polynomial security), there exists a constant-round concurrent non-malleable zero-knowledge argument system for **NP**.*

Our additional contribution is that by combining our CNMZK argument system with the approach of (Chongchitmate et al. 2017) and (Deng et al. 2009), we get the first *constant-round simultaneously resettable and non-malleable* zero-knowledge protocol.

Theorem 2 *Assuming the existence of collision-resistant hash functions, one-way permutations and **iO** for **P**/poly*

(with slightly super-polynomial security), there exists a constant-round simultaneously resettable and non-malleable zero-knowledge argument system for ***NP****.*

Our techniques

Below, we first recall the techniques in (Barak 2001; Chung et al. 2015; Kiyoshima 2015) and then give an overview of our construction approach.

Barak's protocol. Barak's non-black-box zero-knowledge argument system consists of three stages. In stage 1, the verifier V chooses a hash function $h \xleftarrow{R} \mathcal{H}$ and sends it to the prover P, where $\mathcal{H}$ is a *collision-resistant hash function family*. In stage 2, P sends a commitment $c \leftarrow \mathsf{Com}(0^n, \rho)$ to V, where Com is a *statistically binding* commitment scheme; then V responds with a random string $r \in \{0,1\}^{2n}$ to P. In stage 3, P and V start a *witness-indistinguishable universal argument*(WIUA) system where P proves to V that there exists $x \in L$ or $(h, c, r) \in \Lambda$. The language Λ is defined as $(h, c, r) \in \Lambda$ iff there exists a program Π such that $c = \mathsf{Com}(h(\Pi), \rho)$ and Π on input c can output r within $n^{\log\log n}$ steps.

The soundness of Barak's protocol follows from the fact that even if a malicious prover P^* tries to commit to some program Π (instead of committing to 0^n), with a high probability, the output of $\Pi(c)$ will be different from the string r sent by V for every string $r \in \{0,1\}^{2n}$. To prove zero knowledge, just use the code of the malicious verifier V^* as trapdoor in stage 2. By the definition of the language Λ, it must holds that $c = \mathsf{Com}(h(\Pi)) = \mathsf{Com}(h(V^*))$ and $\Pi(c) = V^*(c) = r$.

Chung et.al's constant-round CZK protocol. In (Chung et al. 2013), Chung et.al presented a **P**-certificates assumption for the language $L_c \in \mathbf{P}$ where $L_c = \{(M, x, y) : M(x) = y \text{ within } |x|^c \text{ steps}\}$. In a **P**-certificate system, an *efficient* prover can generate a *short* certificate π of a *fixed* polynomial length (*independent* of the *running-time* and *size* of M) for a tuple (M, x, y) in a *prior bounded* polynomial time in $|x|^c$. By using π the verifier can check the validity of the *deterministic* polynomial-time computation $M(x) = y$ in some *fixed* polynomial time (*independent* of the *running-time* of M). Such proof system has two salient features, i.e., the "non-interactivity" and "succinctness", which guarantee the simulator can *reuse the same certificate* in many nested sessions and *amortize* the cost of generating WIUA proof. We stress that this is essentially to overcome the exponentially blow-up problem in the running time of the concurrent simulation. Based on the Barak's non-black-box zero-knowledge protocol, they modified the part of the stage 3 and defined a new language Λ. More specifically, they defined that a statement $(h, c, r) \in \Lambda$ iff there exists a program M, a certificate π, a vector $\lambda = ((1, \pi_1), (2, \pi_2) \cdots)$ and a vector $\vec{m}$ such that $c = \mathsf{Com}(h(M))$, π is a proof for $M(\lambda) = r$ and each π_j certifies that $M(\lambda_{<j})$ outputs m_j in its j-th communication round (where $\lambda_{<j} = ((1, \pi_1), (2, \pi_2) \cdots (j-1, \pi_{j-1}))$).

The soundness can be obtained as follows. Roughly speaking, from the *statistically binding* property of the Com, for every commitment c (i.e., m_1), there exists a prior fixed *deterministic* polynomial-time program M. By the *unique* certificate property of the **P**-certificate, we can infer that the certificate π_1 for $M(\cdot) = m_1$ is also *uniquely* defined. Due to the same analysis, we can conclude that for every $j > 1$, m_j is *uniquely* defined. Thus, also the *unique* (accepting) certificate π_j certifying $M(\lambda_{<j}) = m_j$. That is, there is a *unique* valid vector λ for program M, so there exists a single r satisfied the computation $M(\lambda) = r$. From the soundness of the previous Barak's protocol (Barak 2001), we can obtain that, with a high probability, the string r sent by V will be different from $M(\lambda)$ for every string $r \in \{0,1\}^{4n}$.

To prove the zero-knowledge, the key difference from Barak's protocol is that each certificate π_i generated during construct the WIUA proofs in stage 3 of a session, can be reused as a part of the input witness $\lambda = ((1, \pi_1), (2, \pi_2) \cdots)$ for the subsequent sessions that contains this session. Thus, *the only expensive part* of the generation of the WIUA in each session is *the generation of the* ***P****-certificates* π, which can be generated in a *prior bounded polynomial time* for the following reasons. Recall that when arriving at the point of stage 3, the simulator S has emulated the partial execution of M and outputted the message r. We assume that the time spent in this part is bounded by $|x|^c$ for some constant $c \in \mathbb{N}$, where x is the statement $M(\lambda) = r$. Then the *certificate* π for this part computation can be implemented in polynomial time in $|x|^c$ by the **P**-certificates system. So the whole simulation can be finished in polynomial time, we refer the reader to (Chung et al. 2015) for more detail about this part.

Our Approach on CNMZK. Our protocol attempts to combine the constant-round CZK techniques and the previous CNMZK techniques together. Compared with the work of (Kiyoshima 2015; Lin et al. 2010), we use the *non-black-box* techniques to reduce the round complexity.

Recall that the definition of *standalone* NMZK requires the existence of a simulator-extractor SE that can simulate the view of a *man-in-the-middle* adversary $\mathcal{A}$ while simultaneously extracting the witnesses for the statements proved by the adversary in the right interaction. On the high level, in order to satisfy this definition, the traditional method is that the verifier commits a trapdoor in the first stage, and then the prover uses a *non-malleable* commitment to commit the real witness, finally the prover uses the WIAOK protocol to prove that it either committed a real witness or known the trapdoor. So when considering the CNMZK protocols, intuitively, we need the prover to use a *concurrent non-malleable* commitment scheme (Pass and Rosen 2005; Lin et al. 2008, 2017; Ciampi et al.

2016; Khurana and Sahai 2017) to commit the real witness. However, we note that this is not necessary, as described in (Barak et al. 2006), since we only need to prove that the adversary still commits the real witness in each session rather than all the right sessions together. That is stand-alone non-malleable commitment is sufficient for our purpose.

By the definition of CNMZK, the crux of the proof is to show that even during simulation, when the simulator commits a fake witness (instead of real witnesses) in left interactions, the *man-in-the-middle* adversary $\mathcal{A}$ still cannot change its committed values in right interactions. The most delicate part of the proof is that we need to consider the mutual influence on the both sides of the rewinds when extract the trapdoors in the left and the witnesses in the right. That is we should carefully design a series of hybrids to argument the rewinds do not affect the reduction of the *concurrent non-malleability* of our zero-knowledge protocol to *(non-concurrent) non-malleability* of the commitment scheme.

In the previous protocol (Lin et al. 2010), they used a special skill to reduce the difficulty of the proof. More specifically, the prover first uses a non-malleable commitment scheme with a *robust* property to commit to a witness *w twice* (sequentially), and then they designed a series of hybrids to show that the adversary must commit the valid witness (except with a negligible probability) in each case. Otherwise, they can use the adversary to break the non-malleable property with respect to itself or the non-malleable property w.r.t. *k*-round protocols. In the protocol (Kiyoshima 2015), because their goal is to implement a *statistically* CNMZK argument system, instead of using a non-malleable commitment to commit the witness, they commit a random string (e.g., 0^n). Thus, in their simulation-extractability proof, they can not directly use the *extractability* of the commitment scheme, instead they have to rewind the sWIAOK proof to extract the witness in the right. Their proof strategy is that assume there exists an adversary which can extract a fake witness in the right, then they can give a series of indistinguishable hybrids to show that even the simulator in the right interaction (act as an honest verifier) just send a commitment with the value 0^n, the adversary still can extract this fake witness, this is a contradiction.

Because our goal is to construct the *constant-round concurrent non-malleable* zero-knowledge protocol, so the non-malleable commitment scheme should be constant rounds, here we use the *constant-round* 4*-robust one-one CCA-secure* commitment scheme which first appeared in (Kiyoshima 2015) based Canetti et al. (2010). Such commitment scheme can be based on the minimum assumption of the existence of *one way functions*. The difference from (Kiyoshima 2015) is that our protocol use the CCA-secure commitment scheme to commit the witness not the *random* string.

More specifically, the commitment scheme we use has a salient feature, i.e., its security can be guaranteed even the adversaries have access to the *committed-value oracle* in the right. This advantage brings us the convenience in designing the hybrids since we need not consider the impact on the left side when we do oracle access to the committed-value oracle in the right sessions. Indeed, in our final proof, we use an opposite argument which is essentially the same. Roughly speaking, we consider the following hybrids $\mathsf{SE}_i^{\mathcal{O}}$ and $\mathsf{SE}_{i+}^{\mathcal{O}}$, where the former simulator-extractor SE uses the "fake" witness in the *i*-th left session and the later simulator-extractor SE uses the real witness in the *i*-th left session, while allowing both SE to access the committed oracle $\mathcal{O}$. If the adversary $\mathcal{A}$ can convince the verifier accept a right session and uses a different identity from all the left sessions, then from the soundness of the WIAOK and the binding property of the commitment, the *one-one CCA* commitment of this right session must commit a right witness *except with a negligible probability*. Now we can forward it to the external *committed-value oracle* and obtain its commit value. Next assume there exists an adversary $\mathcal{A}$ which can distinguish the two simulator-extractor $\mathsf{SE}_i^{\mathcal{O}}$ and $\mathsf{SE}_{i+}^{\mathcal{O}}$, then we can use such adversary to break the *witness indistinguishability* of the 4-round WISSP or the *k-robust CCA security*. This gives a contradiction, thus each hybrids $\mathsf{SE}_i^{\mathcal{O}}$ and $\mathsf{SE}_{i+}^{\mathcal{O}}$ are indistinguishable and we can claim that our protocol is *concurrent non-malleable zero-knowledge argument*. The more details proofs are given in "Constant-round Concurrent Non-malleable Zero-Knowledge" section. Since we only add a *constant-round* commitment on the original *constant-round* CZK, the whole protocol also a *constant-round* protocol, so we can draw the conclusion given in Theorem 1.

Towards Simultaneously-Resettable NMZK. Let us turn to the second question namely the *simultaneously-resettable non-malleable zero-knowledge argument system*. The formal definition is somewhat complicated and will be given in the "Simultaneously-Resettable and Non-Malleable Zero-Knowledge" section. Roughly speaking, the protocol need to satisfy the *non-malleable* security even if the *man-in-the-middle* adversary $\mathcal{A}$ can reset the prover to have several interactions in the left, at the same time, $\mathcal{A}$ can reset the verifier to have multiple interactions in the right. Thus, all the previous protocols will not satisfy our new security requirements, our solution is to enhance the recently result of Chongchitmate et al. (2017) in the following.

In (Chongchitmate et al. 2017) they given a *constant-round simultaneously-resettable zero-knowledge argument system*. More specifically, they first gave a transformation

from any ℓ-round CZK *argument system* to $O(\ell)$-round *resettable zero-knowledge argument*. Then they can achieve a *resettably-sound concurrent* zero-knowledge argument(rsCZK) by plugging a *constant-round* rZK into a *constant-round* CZK system. Finally, following the general transformation of (Deng et al. 2009), they obtained a *simultaneously-resettable* ZK protocol. We stress that, to the best of our knowledge, this transformation is the most direct route to achieve *simultaneously-resettable* zero-knowledge argument system (see also (Bitansky and Paneth 2015; Chung et al. 2013a; Canetti et al. 2013)). In this paper, we observe that this construction actually preserves non-malleability: If the original protocol is a *constant-round concurrent non-malleable* zero-knowledge argument system, then the new one is a *constant-round resettably-sound concurrent non-malleable* zero-knowledge argument. Further, by applying a combination of the transformations in (Deng et al. 2009), we can achieve a *constant-round simultaneously-resettable* NMZK, thus we can draw the conclusion given in Theorem 2.

Organization

The rest of this paper is organized as follows. Some necessary preliminaries and security notion are given in "Preliminary" section. The concrete construction and the security analysis for *constant-round* CNMZK *argument system* are described in "Constant-round Concurrent Non-malleable Zero-Knowledge" section. Finally, we show how to use our CNMZK *argument system* to construct the *constant-round simultaneously-resettable non-malleable* ZK *argument system* in "Simultaneously-Resettable and Non-Malleable Zero-Knowledge" section.

Preliminary

k-robust (one-one) CCA-secure Commitment Schemes (Canetti et al. 2010)

A *tag-based* commitment scheme $\langle C, R\rangle$ is a commitment scheme where the committer and the receiver receive a *tag* $\in \{0,1\}^n$ (also called id) as *common input*. An adversary $\mathcal{A}^{\mathcal{O}}$ can interact with a committed value oracle $\mathcal{O}$ as a committer by using identities adaptively in many sessions. At the end of each session, if the session is *valid*, the oracle $\mathcal{O}$ *reveals* the *unique* committed value of that session to $\mathcal{A}$; otherwise, it sends $\perp$. Consider the following probabilistic experiment $\mathsf{IND}_b\left(\langle C, R\rangle, \mathcal{A}^{\mathcal{O}}, 1^n, z\right)$. The oracle adversary $\mathcal{A}^{\mathcal{O}}$ is allowed to adaptively choose an id and a pair of values $\left(v^0, v^1\right) \in \{0,1\}^n$ as the *challenge* messages. When the adversary $\mathcal{A}^{\mathcal{O}}$ receives a commitment to v_b, it guess a bit b' as the output of the experiment. The additional constraint is that if during the execution the adversary $\mathcal{A}$ interacts with $\mathcal{O}$ using the *challenge* identity id, then the experiment outputs $\perp$.

Definition 1 *We say a tag-based commitment scheme $\langle C, R\rangle$ is CCA-secure w.r.t. the committed-value oracle $\mathcal{O}$, if for every PPT oracle machine $\mathcal{A}$, the following ensembles are computationally indistinguishable:*

- $\left\{\mathsf{IND}_0\left(\langle \mathsf{C}, \mathsf{R}\rangle, \mathcal{A}^{\mathcal{O}}, \mathsf{n}, \mathsf{z}\right)\right\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$
- $\left\{\mathsf{IND}_1\left(\langle \mathsf{C}, \mathsf{R}\rangle, \mathcal{A}^{\mathcal{O}}, \mathsf{n}, \mathsf{z}\right)\right\}_{n\in\mathbb{N}, z\in\{0,1\}^*}$

Additionally, if $\langle C, R\rangle$ is CCA-secure only against adversaries that start a *single* session with $\mathcal{O}$, then we say that $\langle C, R\rangle$ is *one-one* CCA-secure.

The notion of non-malleability w.r.t. arbitrary k-round protocols is introduced in (Lin and Pass 2009), which considers the *man-in-the middle* adversaries can participate arbitrary k-round protocols in the left when running the commitment scheme in the right. Roughly speaking, we say $\langle C, R\rangle$ is k-robust w.r.t $\mathcal{O}$ if the (joint) output of every k-round interaction with an adversary having access to the oracle $\mathcal{O}$, can be simulated without the oracle.

Definition 2 *Let $\langle C, R\rangle$ be a tag-based commitment scheme and $\mathcal{O}$ be the committed-value oracle. For any constant $k \in \mathbb{N}$, we say that $\langle C, R\rangle$ is k-robust w.r.t. $\mathcal{O}$ if there exists a PPT oracle machine S such that for any PPT adversary $\mathcal{A}$ and any k-round PPT interactive Turing machine B, the following are computationally indistinguishable:*

- $\{\mathsf{output}_{B,\mathcal{A}^{\mathcal{O}}}[B(1^n, x, y)] \leftrightarrow \mathcal{A}^{\mathcal{O}}(1^n, x, z)\}_{n\in\mathbb{N}, x,y,z\in\{0,1\}^*}$
- $\{\mathsf{output}_{B,S^{\mathcal{A}}}[B(1^n, x, y)] \leftrightarrow S^{\mathcal{A}}(1^n, x, z)]\}_{n\in\mathbb{N}, x,y,z\in\{0,1\}^*}$

In our protocol, we use the *constant-round* 4-robust *one-one* CCA-secure commitment scheme (namely $\mathsf{CCACom}^{1:1}$) which first appeared in (Kiyoshima 2015) and can be constructed from one-way functions based on the result of (Goyal et al. 2015).

Forward-secure PRG (Bellare and Yee 2003; Chung et al. 2013)

Definition 3 *(Forward-secure Pseudorandom Generator) We say a polynomial-time computable function is a forward secure pseudorandom generator (fsPRG) if the following properties hold:*

Consistency: *For every $n, \ell \in \mathbb{N}$, $s \in \{0,1\}^n$, if $\mathsf{fsPRG}(s, \ell) = ((s_\ell, s_{\ell-1}, \cdots, s_1), (\rho_\ell, \rho_{\ell-1}, \cdots, \rho_1))$, then $\mathsf{fsPRG}(s_\ell, \ell - 1) = ((s_{\ell-1}, \cdots, s_1), (\rho_{\ell-1}, \cdots, \rho_1))$.*

Forward Security: *For every polynomial $p(n)$, the following ensembles are computationally indistinguishable:*

- $\{s \leftarrow U_n, (\vec{s}, \vec{\rho}) \leftarrow \mathsf{fsPRG}(s, \ell) : s_t, \vec{\rho}_{\le t}\}_{n\in\mathbb{N}, \ell\in[p(n)], t\in[\ell]}$

– $\{s_t \leftarrow U_n, \vec{\rho} \leftarrow (U_n)^{\ell} : s_t, \vec{\rho}_{\leq t}\}_{n\in\mathbb{N},\ell\in[p(n)],t\in[\ell]}$

where U_n is the uniform distribution over $\{0,1\}^n$, and $\vec{\rho}_{\leq t} = (\rho_t, \rho_{t-1}, \cdots, \rho_1)$.

From the definition above, if the seed s_t is exposed then the later sequence $(\rho_{t+1}, \rho_{t+2}, \cdots)$ are also exposed, but the earlier sequence $\rho_1, \cdots, \rho_t$ remain pseudorandom. The existence of a fsPRG is implied by any (traditional) PRG, thus it is also implied by the existence of one-way functions (Håstad et al. 1999).

P-certificates in the delegatable CRS model (Chung et al. 2015)

For every constant $c \in \mathbb{N}$, consider the language $L_c \in \mathbf{P}$ such that $L_c = \{(M,x,y) : M(x) = y \text{ within } |x|^c \text{ steps}\}$, let $T_M(x)$ denotes the running time of M on input x.

Definition 4 *A tuple of PPT algorithms (Setup, PreGen, CRSGen, P_{cert}, V_{cert}), is a **P**-certificate system in the delegatable CRS model if there exist polynomials $\ell_d, \ell_\kappa, \ell_{CRS}$ and ℓ_π, such that the following holds:*

Syntax and Efficiency: *for every $c \in \mathbb{N}$ and every $q = (M,x,y) \in L_c$, the verification of the statement proceed as follows:*

- ***1)*** CRS SETUP: $(PP, K) \xleftarrow{\$} Setup(1^n, c)$, *where PP the public parameter and K the key;*
- ***2)*** CRS PREPROCESSING: $d = PreGen(PP, q)$ *where $|d|$ is bounded by ℓ_d;*
- ***3)*** CRS GENERATION: $\kappa \xleftarrow{\$} CRSGen(PP, K, q)$ *and $CRS = (PP, \kappa)$, where $|k|$ is bounded by ℓ_κ and $|CRS|$ is bounded by ℓ_{CRS}*
- ***4)*** PROOF GENERATION: $\pi \xleftarrow{\$} P_{cert}(1^n, c, q, CRS)$, *where $|\pi|$ is bounded by l_π and P_{cert} runs in time poly $(1^n, |x|, min\,(T_M(x), |x|^c))$*
- ***5)*** PROOF VERIFICATION: $b = V_{cert}(1^n, c, CRS, q, \pi)$, *where V_{cert} runs in time poly$(k, |q|)$. Additionally, if the verification procedure V_{cert} is independent of the statement q and the language index c, then we say that the verification algorithm is simple.*

(Perfect) Completeness: *For every $c, c' \in \mathbb{N}$, there exists a negligible function μ such that for every $q = (M,x,y) \in L_c$ such that $|q| \leq k^{c'}$, the probability that V_{cert} outputs 1 is 1.*

Selective Strong Soundness: *There exists a super-polynomial function $T(n) = n^{\omega(1)}$ and a super-constant function $C(n) = \omega(1)$ such that for every probabilistic algorithm P^* with running-time bounded by T(n), there exists a negligible function $\mu(n)$, such that, for every $n \in \mathbb{N}$ and $c \leq C(n)$,*

$$\Pr\left[\begin{array}{l}(q, st) \xleftarrow{\$} P^*(1^n, c)\\ CRS \xleftarrow{\$} Gen(1^n, c) : V_{cert}(1^n, c, CRS, q, \pi) = 1 \wedge q \notin L_c\\ \pi \xleftarrow{\$} P^*(st, CRS)\end{array}\right] \leq \mu(n)$$

Unique certificate: *We say that a **P**-certificate system is unique if for every $c \in \mathbb{N}$, string $CRS \in \{0,1\}^*$ and $q \in \{0,1\}^*$, there exists at most one string π such that $V_{cert}(1^n, c, CRS, q, \pi) = 1$.*

Theorem 3 *(Chung et al. 2015) Assume the existence of an $i\mathcal{O}$ for **P**/poly and an injective pseudo-random generator, then there exists a **P**-certificate system for **NTIME**$(n^{w(1)})$ with (strong) soundness, uniqueness in delegatable CRS Model and the verification algorithm is simple.*

Concurrent non-malleable zero-knowledge arguments (Barak et al. 2006; Lin et al. 2010; Kiyoshima 2015)

Let (P, V) be an interactive protocol for a language L, n be the security parameter and m be a polynomial. Consider a PPT *man-in-the-middle* adversary $\mathcal{A}$ given the *common input* $(x_1, \cdots, x_m)$ and an *auxiliary input* $z \in \{0,1\}^*$. On the left, the adversary $\mathcal{A}$ acts as a verifier V^* to interact with m independent copies of P using $(\mathsf{id}_1, \cdots, \mathsf{id}_m)$, and each copy of prover P will be given a valid witness $w_i \in R_L(x_i)$. On the right, the adversary $\mathcal{A}$ acts as a prover P^* that, on common input $(\widetilde{x}_1, \cdots, \widetilde{x}_m)$ to prove the validity of each statement using $\left(\widetilde{\mathsf{id}}_1, \cdots, \widetilde{\mathsf{id}}_m\right)$. During the experiment, the statements proved in the right interactions and the identities in both the left and right interactions are all chosen by the adversary $\mathcal{A}$, and the messages of the left sessions can be scheduled by the adversary $\mathcal{A}$ without any restriction. Let $\mathsf{view}_{\mathcal{A}}(1^n, x_1, \cdots, x_m, z)$ denotes the random variable that describes the view of $\mathcal{A}$ in the above experiment. Loosely speaking, an interactive proof is a *concurrent 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 interactions for $\mathcal{A}$, while outputting a witness for each statement proved by the adversary in the right interactions.

Definition 5 *An interactive protocol (P, V) for $L \in$ **NP** is said to be concurrent non-malleable zero-knowledge if for every $n \in \mathbb{N}$, every polynomial m, and every PPT man-in-the-middle adversary $\mathcal{A}$ that participates in at most $m(n)$ concurrent executions, there exists a PPT machine SE such that:*

1. *The following ensembles are computationally indistinguishable:*

   – $\{\mathsf{view}_{\mathcal{A}}(1^n, x_1, \cdots, x_m, z)\}_{n\in\mathbb{N},x_1,\cdots,x_m\in L\cap\{0,1\}^n z\in\{0,1\}^n}$
   – $\{\mathsf{S}(1^n, x_1, \cdots, x_m, z)\}_{n\in\mathbb{N},x_1,\cdots,x_m\in L\cap\{0,1\}^n,z\in\{0,1\}^n}$

   where $\mathsf{S}(1^n, x_1, \cdots, x_m, z)$ is the first output of $\mathsf{SE}(1^n, x_1, \cdots, x_m, z)$.

2. *Let* $(\widetilde{x}_1, \cdots, \widetilde{x}_m)$ *be the statements to be proved in the right interactions and* (view, $\{\widetilde{w}_i\}_{i\in m}$) *denote the outputs of* $\mathsf{SE}(1^n, x_1, \cdots, x_m, z)$. *For every* $i \in [m]$, *if the* i*-th right interaction is accepting and* $\widetilde{id}_i \neq id_j$ *for all* $j \in [m]$, *then* $\widetilde{w}_i$ *is a valid witness such that* $R_L(\widetilde{x}_i, \widetilde{w}_i) = 1$.

Indistinguishability obfuscation (Barak et al. 2001)

Definition 6 *(Indistinguishability obfuscation) A PPT algorithm* $i\mathcal{O}$ *is said to be an indistinguishability obfuscator for a collection of polynomial size circuits* $\mathcal{C} = \cup_{n\in\mathbb{N}}\mathcal{C}_n$, *if it satisfies:*

1. *Functionality: For any* $C \in \mathcal{C}$,

$$\Pr_{i\mathcal{O}}[\forall x : i\mathcal{O}(C)(x) = C(x)] = 1 .$$

2. *Indistinguishability: For any poly-size distinguisher* $\mathcal{D}$ *there exists a negligible function* μ, *such that for any* $n \in \mathbb{N}$, $C_1, C_2 \in \mathcal{C}_n$ *of the same size and functionality*

$$\left|\Pr_{i\mathcal{O}}[D(i\mathcal{O}(C_1)) = 1] - \Pr_{i\mathcal{O}}[D(i\mathcal{O}(C_2)) = 1]\right| \leq \mu(n).$$

Resettable zero-knowledge (Canetti et al. 2000)

Let (P, V) be an interactive proof system for a language L, z be an *auxiliary input* received by V^*, $t = poly(n)$, $\overline{x} = x_1, x_2, \cdots, x_t \in L \cap \{0,1\}^n$ be a sequence of *common inputs* and $\overline{w} = w_1, w_2, \cdots, w_t$ be the corresponding witnesses such that $(x_i, w_i) \in R_L$ for $i = 1, \cdots, t$. The distribution $\{\mathsf{view}^{P(\overline{w})}_{V^*(z)}(\overline{x})\}$ is the view of V^* that defined as follows:

1. Randomly select and fix t random tapes $r_1, r_2, \cdots, r_t$ for P, resulting in deterministic strategies $P^{(i,j)} = P_{x_i,w_i,r_j}$, defined by $P_{x_i,w_i,r_j}(\alpha) = P(x_i, w_i, r_j, \alpha)$ [1], for $i, j \in [t]$.
2. A resetting verifier V^* is allowed to run $poly(n)$-many sessions with the $P^{(i,j)}$. V^* can send arbitrary messages to each of the $P^{(i,j)}$ and obtain the responses of $P^{(i,j)}$ to such message.
3. Once V^* decides it is done interacting with the P, it produces its view of these interactions.

The distribution $\{\mathsf{S}_{V^*(z)}(\overline{x})\}$, indexed by a sequence of *common inputs* $\overline{x} = x_1, x_2, \cdots, x_{poly(n)} \in L \cap \{0,1\}^n$, is the output of an *expected* PPT machine S that interacts with V^* on *common inputs* $\overline{x}$.

Definition 7 (Resettable Zero-knowledge) *We say that* (P, V) *is resettable zero-knowledge if for every PPT adversary* V^* *there exists an expected PPT simulator* S_{V^*} *such that the for all pairs* $(\overline{x}, \overline{w}) \in R_L$, *the ensembles* $\left\{\mathsf{view}^{P(\overline{w})}_{V^*(z)}(\overline{x})\right\}$ *and* $\{\mathsf{S}_{V^*(z)}(\overline{x})\}$ *are computationally indistinguishable*

Theorem 4 *(Chongchitmate et al. 2017) Assuming the existence of one-way functions, then any* ℓ*-round concurrent zero-knowledge argument system can be transformed into a* $O(\ell)$*-round resettable zero-knowledge argument system.*

Resettably-sound arguments (Barak et al. 2001)

Definition 8 *(Resettably-sound arguments). Let* (P, V) *is an interactive proof protocol for* $L \in$ **NP**. *A resetting attack of a cheating prover* P^* *is defined as follows:*

1. *Let* $t = poly(n)$, *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* $V_{x,r_j}(\alpha) = V(x, r_j, \alpha)$ [2], *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* $poly(n)$*-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.*

Theorem 5 *(Chung et al. 2014) Assume the existence of one-way functions, then there exists a 4-round resettably-sound zero-knowledge argument of knowledge for every language in* ***NP*** .

Theorem 6 *(Deng et al. 2009, Chongchitmate et al. 2017) Assuming the existence of ZAPs (i.e., 2-round resettably-sound resettable witness-indistinguishable proof systems) and family of pseudorandom functions, then there exists a transformation from a* ℓ*-round resettably-sound concurrent zero-knowledge argument to a* $O(\ell)$*-round resettably-sound resettable zero-knowledge argument.*

Constant-round concurrent non-malleable zero-knowledge

Our protocol

In this section, we give our construction of the *constant-round concurrent non-malleable zero-knowledge* argument system. We use the following building blocks:

1. Two-round *statistically binding* commitment scheme: Com
2. $O(1)$-round 4*-robust one-one CCA-secure* commitment scheme: $\mathsf{CCACom}^{1:1}$

3. Four-round *special-sound witness indistinguishability* proofs: WISSP
4. $O(1)$-round *witness indistinguishability universal argument*: WIUA
5. Four-round **P**-*certificates in the delegatable CRS Model*: PC

Now consider a language $L \in \mathbf{NP}$ and a security parameter n. Let the prover and verifier receive a *common inputs* $x \in \{0,1\}^n$, $\mathsf{id} \in \{0,1\}^n$. The auxiliary input to the prover is a **NP** witness w such that $R_L(x,w) = 1$. Let $m(n)$ be a polynomial that upper bounds the number of *concurrent* sessions, and D be a super-constant bounded by $\log\log\log n$. Then, our protocol proceeds in five stages as follows:

- In stage 1, the prover P computes $c_1 = \mathsf{Com}(0^n, \rho_1)$ and sends it to V; V responds with a string $r \xleftarrow{R} \{0,1\}^{4n}$.
- In stage 2, the prover P computes $c_2 = \mathsf{Com}(0^n, \rho_2)$ and sends it to V. P and V run a WIUA system where P proves to V that there exists $(M, \rho_1, \mathcal{O}_\pi, (j, s_j), \rho_2)$ s.t., $(h, c_1, c_2, r) \in \Lambda_1$ or exists w s.t., $(x,w) \in R_L$. In more detail, in the simulation phase, P proves that $c_1 = \mathsf{Com}(h(M))$ for a program M and $c_2 = \mathsf{Com}(h(\mathsf{q}))$ for $\mathsf{q} = ((M, \mathcal{O}_\pi), (j, s_j), r)$. The statement q represents that the oracle program $M^{\mathcal{O}_\pi}$ on input (j, s_j) can output a message r. The oracle $\mathcal{O}_\pi$ stores all the CRS and proof pairs $\{(\mathsf{CRS}_i, \pi_i)\}$ that generated by the **P**-certificate system in the current history(see the definition in Table 1).
- In stage 3, the verifier V invokes the algorithm PC.setup to generate (PP,K) and sends the public parameter PP to P. The prover P computes $c_3 = Com(0^n, \rho_3)$ and sends it to V. P and V run a WIUA system where P proves to V that there exists $(M, \mathcal{O}_\pi, (j, s_j), d, \rho_2, \rho_3)$ s.t., $(h, \mathsf{PP}, c_2, c_3, r) \in \Lambda_2$ or exists w s.t., $(x,w) \in R_L$. In more detail, in the simulation phase, P proves that $c_2 = Com(h(\mathsf{q}))$ and $c_3 = \mathsf{Com}(d, \rho_3)$ where $d = \mathsf{PC.PreGen}(\mathsf{PP}, \mathsf{q})$.
- In stage 4, the verifier V sends an obfuscation *algorithm* $\hat{\mathcal{P}}_{\mathsf{CRSGen}}$ to P and gives a ZK argument of the statement $(\mathsf{PP}, c_3, \hat{\mathcal{P}}_{\mathsf{CRSGen}}) \in \Lambda_3$. In more detail, V proves that there exists $(\mathsf{K}, \mathcal{P}^{c_3,\mathsf{PP},\mathsf{K},\rho_{\mathsf{CRSGen}}}, \rho_{\mathsf{Setup}}, \rho_{\mathsf{CRSGen}}, \rho_{i\mathcal{O}})$ such that $(\mathsf{PP}, \mathsf{K}) = \mathsf{PC.Setup}(1^n, D, \rho_{\mathsf{Setup}})$ and $\hat{\mathcal{P}}_{\mathsf{CRSGen}} = i\mathcal{O}(\mathcal{P}^{c_3,\mathsf{PP},\mathsf{K},\rho_{\mathsf{CRSGen}}}, \rho_{i\mathcal{O}})$. The detailed descriptions of the circuit $\mathcal{P}$ and $\mathcal{Q}$ are given in Table 1.
- In stage 5, the prover P computes $c_4 = \mathsf{CCACom}_{\mathsf{id}}^{1:1}(w, \rho_4)$ under identity id, $c_5 = \mathsf{Com}(0^n, \rho_5)$ and sends them to V. P and V runs a WISSP system where P proves to V that there exists (d, ρ_3, π) s,t., $(\mathsf{PP}, \hat{\mathcal{P}}_{\mathsf{CRSGen}}, c_5) \in \Lambda_4$ or exists w s.t., $c_4 = \mathsf{CCACom}_{\mathsf{id}}^{1:1}(w, \rho_4)$ and $(x,w) \in R_L$. In more detail, in the simulation phase, P proves that $\kappa = \hat{\mathcal{P}}_{\mathsf{CRSGen}}(d, \rho_3)$, $c_5 = \mathsf{Com}((\mathsf{PP}, \kappa), \rho_5)$ and $\mathsf{PC.V}_{\mathsf{cert}}(1^n, (\mathsf{PP}, \kappa), \pi) = 1$.

Table 1 The languages used in CNMZK

Oracle $\mathcal{O}_\pi$:

$$\mathcal{O}_\pi(\mathsf{CRS}_i) = \begin{cases} \pi_i & \text{if there exists unique } \pi_i \text{ stored in oracle } \mathcal{O}_\pi \text{ and} \\ & \mathsf{PC.V}_{\mathsf{cert}}(1^n, \mathsf{CRS}_i, \pi_i) = 1 \\ \perp & \text{otherwise} \end{cases}$$

Circuit $\mathcal{P}^{n,c_3,\mathsf{PP},\mathsf{K},\rho_{\mathsf{CRSGen}}}$:

$$\mathcal{P}(d, \rho_3) = \begin{cases} \kappa & \text{if } c_3 = \mathsf{Com}(d, \rho_3), \text{then set } \kappa := \mathsf{PC.CRSGen}(\mathsf{PP}, \mathsf{K}, d, \rho_{\mathsf{CRSGen}}) \\ \perp & \text{otherwise} \end{cases}$$

Equivalent Circuit $\mathcal{Q}^{n,c_3,\kappa}$:

$$\mathcal{Q}(d, \rho_3) = \begin{cases} \kappa & \text{if } c_3 = \mathsf{Com}(d, \rho_3) \\ \perp & \text{otherwise} \end{cases}$$

Language Λ_1 :

We say $(h, c_1, c_2, r) \in \Lambda_1$, iff there exist $(M, \rho_1, \mathcal{O}_\pi, (j, s_j), \rho_2)$ such that

- $\rho_1, \rho_2, s_j \in \{0,1\}^n, j \in [m], M, \mathcal{O}_\pi \in \{0,1\}^{n^{\log\log n}}$;
- $c_1 = \mathsf{Com}(h(M), \rho_1)$;
- $c_2 = \mathsf{Com}(h(\mathsf{q}), \rho_2)$ where $\mathsf{q} = ((M, \mathcal{O}_\pi), (j, s_j), r)$.

Language Λ_2 :

We say $(h, \mathsf{PP}, c_2, c_3, r) \in \Lambda_2$, iff there exist $(M, \mathcal{O}_\pi, (j, s_j), d, \rho_2, \rho_3)$ such that

- $d, s_j, \rho_2, \rho_3 \in \{0,1\}^n, j \in [m], M, \mathcal{O}_\pi \in \{0,1\}^{n^{\log\log n}}$;
- $c_2 = \mathsf{Com}(h(\mathsf{q}), \rho_2)$ where $\mathsf{q} = ((M, \mathcal{O}_\pi), (j, s_j), r)$;
- $c_3 = \mathsf{Com}(d, \rho_3)$ where $d = \mathsf{PC.PreGen}(\mathsf{PP}, \mathsf{q})$.

Language Λ_3 :

We say $(\mathsf{PP}, c_3, \hat{\mathcal{P}}_{\mathsf{CRSGen}}) \in \Lambda_3$, iff there exist $\left(\mathsf{K}, \mathcal{P}^{c_3,\mathsf{PP},\mathsf{K},\rho_{\mathsf{CRSGen}}}, \rho_{\mathsf{Setup}}, \rho_{\mathsf{CRSGen}}, \rho_{i\mathcal{O}}\right)$ such that

- $\mathsf{K}, \rho_{\mathsf{Setup}}, \rho_{\mathsf{CRSGen}}, \rho_{i\mathcal{O}} \in \{0,1\}^n$;
- $(\mathsf{PP}, \mathsf{K}) = \mathsf{PC.Setup}\left(1^n, D, \rho_{\mathsf{Setup}}\right)$;
- $\hat{\mathcal{P}}_{\mathsf{CRSGen}} = i\mathcal{O}\left(\mathcal{P}^{c_3,\mathsf{PP},\mathsf{K},\rho_{\mathsf{CRSGen}}}, \rho_{i\mathcal{O}}\right)$.

Language Λ_4 :

We say $(\mathsf{PP}, \hat{\mathcal{P}}_{\mathsf{CRSGen}}, c_5) \in \Lambda_4$, iff there exist (d, ρ_3, π) such that

- $d, \rho_3, \pi \in \{0,1\}^n$;
- $\kappa = \hat{\mathcal{P}}_{\mathsf{CRSGen}}(d, \rho_3)$;
- $c_5 = \mathsf{Com}((\mathsf{PP}, \kappa), \rho_5)$;
- $\mathsf{PC.V}_{\mathsf{cert}}(1^n, (\mathsf{PP}, \kappa), \pi) = 1$.

The formal protocol CNMZK is described below in Table 1 and Table 2.

Completeness and soundness

Completeness. The completeness of the protocol can be directly obtained from the construction in Table 2. More specifically, for any $x \in L$, $w \in R_L(x)$ and $\mathsf{id} \in \{0,1\}^n$,

Table 2 Constant-round concurrent non-malleable zero-knowledge argument

Common input: $x \in L$ and identity $\mathsf{id} \in \{0,1\}^n$.

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

Stage 1: P and V runs a generation protocol as (Barak 2001)

$P \longleftarrow V$: $h \xleftarrow{R} \mathcal{H}_n$

$P \longrightarrow V$: $c_1 = \mathsf{Com}(0^n, \rho_1)$

$P \longleftarrow V$: $r \xleftarrow{R} \{0,1\}^{4n}$

Stage 2: P runs a WIUA using its auxiliary input w

$P \Longleftrightarrow V$: P sends c_2 to V and gives a WIUA argument of the statement $x \in L$ or $(h, c_1, c_2, r) \in \Lambda_1$, where $c_2 = \mathsf{Com}(0^n, \rho_2)$.

Stage 3: P runs a WIUA again upon receiving the public parameter PP

$P \longleftarrow V$: V invokes the PC.Setup *algorithm* to generate (PP,K) and sends PP to P.

$P \Longleftrightarrow V$: P sends c_3 to V and gives a WIUA argument of the statement $x \in L$ or $(h, \mathsf{PP}, c_2, c_3, r) \in \Lambda_2$, where $c_3 = \mathsf{Com}(0^n, \rho_3)$.

Stage 4: V *delegates* P to generate the CRS

$P \Longleftrightarrow V$: V sends the *algorithm* $\hat{\mathcal{P}}_{\mathsf{CRSGen}}$ to P and gives a ZK argument of the statement $(\mathsf{PP}, c_3, \hat{\mathcal{P}}_{\mathsf{CRSGen}}) \in \Lambda_3$, where $\hat{\mathcal{P}}_{\mathsf{CRSGen}} = i\mathcal{O}(\mathcal{P}^{c_3,\mathsf{PP},\mathsf{K},\rho_{\mathsf{CRSGen}}}, \rho_{i\mathcal{O}})$.

Stage 5: P runs a WISSP using its auxiliary input w

$P \Longleftrightarrow V$: P sends (c_4, c_5) to V and gives a WISSP argument of the statement that $(x, w) \in R_L(x)$ or $(\mathsf{PP}, \hat{\mathcal{P}}_{\mathsf{CRSGen}}, c_5) \in \Lambda_4$, where $c_4 = \mathsf{CCACom}^{1:1}_{\mathsf{id}}(w, \rho_4)$ and $c_5 = \mathsf{Com}((\mathsf{PP}, \kappa), \rho_5)$.

from the completeness of the WIUA system in stage 2 and stage 3, the completeness of the ZK argument system in stage 4 and the completeness of WISSP system in stage 5, we have that $\Pr[P(w), V(z)(x, \mathsf{id})] = 1$.

Soundness. The soundness of protocol follows from (1) the binding property of the commitments c_1, c_2, c_3 in stage 1, 2 and 3; (2) the hiding property of $i\mathcal{O}$ for the circuit $\mathcal{P}$ in stage 4; (3) the *selective strong* soundness of **P**-certificates and (4) the *special-soundness* of WISSP used in stage 5. Roughly speaking, assume that the statement $x \notin L$. Consider the point where the prover has given the commitment c_3 and is now expecting the verifier message $\hat{\mathcal{P}}_{\mathsf{CRSGen}}$. Because at this point, $c_1, c_2, c_3, \mathsf{PP}, \mathsf{K}$ are determined, the two circuit $\mathcal{P}$ and $\mathcal{Q}$ described in Table 1 are functional equivalent. We assume that, w.l.o.g, $\mathcal{P}$ and $\mathcal{Q}$ have the same polynomial size in n, then from the security definition of $i\mathcal{O}$, we can infer that the secret key K is hiding in the obfuscation circuit $\hat{\mathcal{P}}$. Otherwise, we can use the adversary to distinguish the circuit $i\mathcal{O}(\mathcal{P})$ and $i\mathcal{O}(\mathcal{Q})$, which leads to a contradiction. Next in stage 5, if there exists a PPT cheating P^* who can convince the verifier, then from the definition of **P**-certificate system, there must exist an accepted **P**-certificate π argument of the statement q is true based on $\mathsf{CRS} = (PP, \kappa)$ *except with negligible probability*. That is there exists an PPT machine M on input a short bit string (j, s) (of length bounded in $3n$) can predict the challenge message r (length of $4n$). However, this is *information theoretically* impossible. Thus, we reach a contradiction through violate the soundness of Barak's protocol.

Next, we describe the construction of our simulator-extractor SE in "Our simulator-extractor" section and show its correctness satisfies the definition of CNMZK in "The view generated by the simulator" section and "The witnesses output by the extractor" section.

Our simulator-extractor

Recall that the definition of CNMZK requires the existence of a simulator-extractor SE that can simulate the view of a *man-in-the-middle* adversary $\mathcal{A}$ while extracting a witness in every accepted right session. Below, we sketch how to build a simulator-extractor. First, we construct a PPT simulator S that simulates the view of $\mathcal{A}$ but does not extract witnesses in the right seasons. Then, we construct a PPT simulator-extractor SE via the intermediate simulator S, which can simulate the view of $\mathcal{A}$ and extract the witnesses by the committed value oracle.

Simulator S On a high level, S internally invokes $\mathcal{A}$ and interacts with $\mathcal{A}$ as honest prover and honest verifier in the following way. To simulate the view of each session in the right interactions, S simply follows the honest verifier strategy. To simulate the view of each session in the left interactions, S uses the description of the adversary $\mathcal{A}$ as the fake witness and reused the previous generated **P**-certificates if necessary in a *straight-line* manner. The formal description of this process will be given below. Finally, S outputs the view of the adversary $\mathcal{A}$.

Simulator SE On a high level, SE simulates the view of $\mathcal{A}$ by executing S as the first part of its output. For each $i \in [m]$, if the i-th right session is accepted and $\widetilde{\mathsf{id}}_i$ is different from id_j for all $j \in [m]$, SE extracts a witness from the session i by oracle access to the *one-session committed-value oracle* $\mathcal{O}_{\mathsf{cca}}$ of $\mathsf{CCACom}^{1:1}$.

The view generated by the simulator

In this section, we show that the view generated by S is indistinguishable from the real view of $\mathcal{A}$:

Lemma 1 *The following ensembles are computationally indistinguishable:*

- $\{view_{\mathcal{A}}(1^n, x_1, \cdots, x_m, z)\}_{n \in \mathbb{N}, x_1, \cdots, x_m \in L \cap \{0,1\}^n, z \in \{0,1\}^n}$
- $\{S(1^n, x_1, \cdots, x_m, z)\}_{n \in \mathbb{N}, x_1, \cdots, x_m \in L \cap \{0,1\}^n, z \in \{0,1\}^n}$

Proof To simplify the exposition, w.l.o.g, we assume that the *man-in-the-middle* adversary $\mathcal{A}$ is a *deterministic* Turing machine with a *non-uniform* advice. Let $N = c \cdot m$

denote the total number of messages between the simulator S and $\mathcal{A}$, where c is the rounds of our CNMZK protocol and m is the total number of concurrent sessions bounded by a polynomial.

We invoke the *forward-secure pseudorandom generator* to generate the random-tape we needed. Let $\mathsf{fsPRG}(s,N) = ((s_N,\cdots,s_1),(\rho_N,\cdots,\rho_1))$, where $s \in \{0,1\}^n$ is the random seed and each $\rho_j \in \{0,1\}^n$ is the randomness used to generate the j-th prover message in the left side.

We use three tables $\mathcal{V},\mathcal{O}_\pi,\mathcal{T}$. $\mathcal{V}$ stores the commitment values in the simulation of the left interaction. $\mathcal{O}_\pi$ stores all the CRS and proof pairs $\{(\mathsf{CRS}_i,\pi_i)\}$ generated by the **P**-certificate system in the current history. $\mathcal{T}$ stores the messages simulated so far in both left and right sides. We initialize $\mathcal{O}_\pi,\mathcal{T}$ to be empty and add the code descriptions of the simulator S and $\mathcal{A}$ to table $\mathcal{V}$. Next we give a detailed description of the program $\mathsf{S}(1^n,x_1,\cdots,x_m,\mathcal{A},\mathcal{V},\mathcal{O}_\pi,\mathcal{T},s,N)$:

In each right session, S interacts with $\mathcal{A}$ simply by following the *honest* verifier strategy described in our protocol 2. It can generate its random coins by using the PRG on a random seed in this part of the execution. In each left session, do as follows:

Simulate Stage 1 Upon receiving a hash function h_i in session i, S provides a commitment c_i^1 to $M_i((\cdot,\cdot),\mathcal{A},\mathcal{T})$, where M_i is an interactive Turing machine with the code description of S and $\mathcal{A}$ plus the current state of them. Here the first two parameters of M_i will be given when M_i is used as the witness to construct the statement q_i in stage 2.

Simulate Stage 2 Upon receiving a challenge r_i in session i during the j-th communication round, S retrieves the committed value M_i and provides a commitment c_i^2 to the trapdoor statement $\mathsf{q}_i = ((M_i,\mathcal{O}_\pi),(j,s_j),r_i)$, where s_j is the random seed used by fsPRG in the j-th round. According to our previous definition, the oracle program $M^{\mathcal{O}_\pi}$ on input (j,s_j) can recover all the previous randomness and any oracle queries $\{\mathsf{CRS}_i\}$ that $M^{\mathcal{O}_\pi}$ makes before it outputs r can be answered using the current $\mathcal{O}_\pi$. Thus, the simulator S can use $(M_i,\mathcal{O}_\pi,(j,s_j))$ and the corresponding randomness to finish the WIUA for the statement $(h_i,c_i^1,c_i^2,r_i) \in \Lambda_1$.

Simulate Stage 3 Upon receiving a challenge PP_i in session i during the j-th communication round, S provides a commitment c_i^3 to the digest d_i, where $\mathsf{d}_i = \mathsf{PC.PreGen}(\mathsf{PP}_i,\mathsf{q}_i)$. Now we can make S use the fake witnesses $(M_i,\mathcal{O}_\pi,(j,s_j),d_i)$ and the corresponding randomness to finish the WIUA for the statement $(h_i,\mathsf{PP}_i,c_i^2,c_i^3,r_i) \in \Lambda_2$.

Simulate Stage 4 Upon receiving an obfuscated program $\hat{\mathcal{P}}_{\mathsf{CRSGen}}$ in session i during the j-th communication round, S interacts with $\mathcal{A}$ as an honest verifier to finish the ZK argument part.

Simulate Stage 5 Upon receiving the last message from $\mathcal{A}$ in Stage 4 of session i, S computes $\kappa_i = \hat{\mathcal{P}}_{\mathsf{CRSGen}}(d,\rho_3)$ and $\pi_i = \mathsf{PC.P}_{\mathsf{cert}}(\mathsf{q}_i,\mathsf{CRS}_i)$. Now for the $\mathsf{CRS}_i = (\mathsf{PP}_i,\kappa_i)$, S checks if $\mathsf{PC.V}_{\mathsf{cert}}(1^n,\mathsf{CRS}_i,\pi_i) = 1$ and extends the pair (CRS_i,π_i) to the oracle $\mathcal{O}_\pi$, otherwise it will abort. Next, S provides a commitment c_i^4 to a dummy string i.e.,0^n and a commitment c_i^5 to CRS_i. Thus, S has all the witnesses $(\mathsf{d}_i,\rho_{\mathsf{d}_i},\pi_i)$ for the statement $\left(\mathsf{PP}_i,\hat{\mathcal{P}}_{\mathsf{CRSGen}},c_i^5\right) \in \Lambda_4$, it can finish the WISSP in stage 5.

Finally, the simulator will output all the messages of the both interactive sides stored in the table $\mathcal{T}$.

Correctness of the simulation. We observe the correctness of S. By our construction, the only place where abort is likely to happen is when the simulator computes an unaccepted certificate π_i for CRS_i based on a true statement q_i in stage 5. However, the only difference of the **P**-certificates system used in our protocol is that, instead of sending κ in directly, the verifier first send the indistinguishability obfuscation of the GenCRS algorithm and then give a ZK argument to prove their correctness. Thus, from the *perfect correctness* of the indistinguishability obfuscator, the *completeness* of zero-knowledge argument and the *perfect completeness* of our **P**-certificates system, it suffices to show that for a true statement q_i, the probability of $V_{\mathsf{cert}}(1^n,\mathsf{CRS}_i,\pi_i) \neq 1$ is only *negligible*. So the probability of simulator output abort is also *negligible*.

Indistinguishability of the simulation. Now we use the hybrid argument to show the indistinguishability of the simulation, consider $2N$ hybrid experiments as follows. Experiment Hyb^i, $0 \le i \le N$: the first i communication rounds are simulated by simulator S with the pseudorandomness and fake witness, and all the later communication round $j > i$ are simulated by simulator S with true randomness and the true witnesses. We also define hybrid Hyb_+^i that proceed identically as Hyb^i except that it simulates the i-th round following the honest prover strategy using the real witness.

Claim *The output of Hyb_+^i and Hyb^i are computationally indistinguishable.* □

Proof Because Hyb_+^i and Hyb^i differs only which witness (fake or real) is used in the i-round of the left interaction. If in the i-th round the prover message is a commitment to a witness, indistinguishability of Hyb_+^i and Hyb^i follows directly by the hiding property of the commitment scheme. If in the i-th round the prover message is a message of the WIUA or WISSP subprotocol, indistinguishability of Hyb_+^i and Hyb^i follows directly by the witness indistinguishability property of the WIUA or WISSP. □

Claim *The output of* Hyb^i_+ *and* Hyb^{i+1} *are computationally indistinguishable.* □

Proof Because Hyb^i_+ and Hyb^{i+1} differs only which randomness (true or pseudo) is used in the i-round of the left interaction. The indistinguishability of Hyb^i_+ and Hyb^{i+1} follows directly from the forward security of the PRG. □

Finally, it is easy to see that the output of Hyb^N is identical to the output of S, and the output of Hyb^0 is identical to the real view $\mathsf{view}_{\mathcal{A}}$. Because there are at most polynomial hybrids in this experiment, we can conclude that the output of S is indistinguishable from the output of the real interaction.

Combining the above, the Lemma 1 follows.

The witnesses output by the extractor

Proof Our simulator-extractor SE in "Our Simulator-Extractor" section allows the extractor to access the decommitment oracle. We note that this is allowed for the reason of a k-robust CCA-secure commitment scheme used in our protocol. From the definition 2, we know that, for any constant-round k, the joint output of every k-round interaction, with an adversary (here it means the SE) having access to the oracle $\mathcal{O}_{\mathsf{cca}}$, can be simulated without the oracle in polynomial time. That is, the simulator-extractor SE access to the oracle does not help it in participating in any k-round protocols. But allowing the simulator-extractor SE to access the oracle has the following benefits, we only need to pay attention to the impact of the hybrid experiment on SE when switching on the left witness from real to fake, without any further analysis of the interference from the right rewinding. So in the following, we just need to analyze whether such simulator-extractor can output the witness.

Consider the series of hybrids, we define SE^i (SE^i_+) the same as SE except that the execution of S is replaced with that of Hyb^i (Hyb^i_+). Then, by the definition of CNMZK, we need to argument that, in the experiment SE^N (which identical to SE), for any PPT *man-in-the-middle* adversary $\mathcal{A}$ and every $x_1, \cdots, x_m \in \{0,1\}^n \cap L$, such that for each right interaction that is accepted and uses a different identity from all left interactions, the simulator-extractor SE does extract a valid witness of the statement proved.

Observe that in the experiment SE^0_+, the simulator S holds all the real witnesses of the left sessions and just acts as an honest prover in each left interaction and an honest verifier in each right interaction. Then following from the soundness of our protocol, we can conclude that in every accepted right interaction, $\mathcal{A}$ commits a real witness in the $\mathsf{CCACom}^{1:1}$ successfully *except with negligible probability*. In other words, $\mathcal{A}$ never cheats in SE^0_+, so the simulator-extractor can extract the witness with the help of the committed value oracle *except with negligible probability*.

Next, we observe the experiment SE^N which based on the definition of Hyb^N. Now we assume that there exists a polynomial function p such that $\mathcal{A}$ cheats in one of the right sessions in the experiment SE^N with probability $1/p(n)$. In other words, there exists a right session which is accepted and uses a different identity from all the left interactions such that $\mathcal{A}$ fails to commit to a valid witness in Stage 5 with probability $1/p(n)$. Then SE^N can not extract the witness from this right session with probability $1/p(n)$ as well. However, we have that SE^0_+ can extract the witness from this right session *except with negligible probability*. Thus, from an average argument, there must exist an i such that the probability of cheating differ by at least a polynomial amount in the hybrids SE^i and SE^i_+ or in the hybrids SE^i_+ and SE^{i+1}. Therefore, there is a gap between A's chance of committing the valid witness on the right in SE^i_+ and SE^{i+1} or there is a gap between A's chance of committing the valid witness on the right in SE^i and SE^i_+. We analyze these two cases as follows:

In the first case, the only difference between SE^i_+ and SE^{i+1} is which randomness (true or pseudo) is used in the i-round of the left interaction. Therefore, they are computationally indistinguishable from claim 2. In the second case, the only difference between SE^i and SE^i_+ is which witness (fake or real) is used in the i-th round. The former, in stage 5, uses a dummy string 0^n as the committed value of $\mathsf{CCACom}^{1:1}$ followed with an WISSP for knowing the fake witness instead of the witness w_i of x_i. The latter, in stage 5, acts as an honest prover holding a real witness w_i of x_i. If the gap is due to the committed value of $\mathsf{CCACom}^{1:1}$, then we can use this gap to break the security of the non-malleable w.r.t itself. If the gap is due to the witness used in the four-round WISSP of the left session, then we can use this gap to break the *4-robustness CCA-secure* of $\mathsf{CCACom}^{1:1}$.

Thus, we reach a contradiction, in the experiment SE^N, $\mathcal{A}$ must commit to a valid witness in Stage 5 *except with negligible probability*. We know that the output of SE^N is identical to the output of SE, hence the simulation-extractability of protocol 2 follows.

Combining "The View Generated by the Simulator" section and "The Witnesses Output by the Extractor" section, the concurrent non-malleable zero-knowledge property follows. This completes the proof of Theorem 1. □

Simultaneously-resettable and non-malleable zero-knowledge

From concurrent NMZK to resettable NMZK

In (Chongchitmate et al. 2017), Chongchitmate et al. gave a transformation from any *constant-round concurrent* ZK to a *constant-round resettable* ZK based on (Barak et al. 2001; Deng et al. 2009). We observe that this transformation essentially preserves the non-malleability. That is,

if the original protocol is a *constant-round concurrent* NMZK, then the new protocol will be a *constant-round resettable* NMZK. We provide the details of the transformation in Table 3, which are taken almost verbatim from (Chongchitmate et al. 2017), except that we require the prover and the verifier to have a extra common id. Then we give a proof about the non-malleability for this new protocol.

Lemma 2 *Protocol rNMZK in Table 3 is a constant-round resettable non-malleable ZK argument system.*

Proof The proof of the completeness and soundness conditions are similar to our proof in "The View Generated by the Simulator" section, and are omitted. The proof of the *resettable* zero-knowledge can be directly obtained from the Theorem 4, because the protocol CNMZK itself is a *constant-round* CZK protocol. Next, we give the analysis of non-malleability.

Roughly speaking, for a *man-in-the-middle* adversary $\mathcal{A}$ with an extra power of resetting attack, we need to prove that the view of $\mathcal{A}$ in the real interaction can be simulated by a simulator without all the witnesses of the left sessions, and there exists an extractor that can extract the witnesses in every accepting right session from this simulated view. More specifically, we first construct a simulator and give an extractor based on this simulator as the previous section. Then, we reduce the security to the underlying assumptions by using a series of hybrids.

Let $\mathsf{H}_0 = \{\mathsf{real\text{-}view}_{f_s}, \{\widetilde{w}_i\}_{i\in[m]}\}$ denote the combined view of $\mathcal{A}$ in the real experiment of the protocol rNMZK and the values extracted by the committed value oracle. Then, following from the soundness of the protocol rNMZK that, *except with negligible probability*, in every accepting right interaction, $\mathcal{A}$ commits to a real witness in stage 5 and the extracted value is a real witness as well.

Next, we modify the protocol rNMZK into a protocol rNMZK_F by replacing the pseudorandom function f_s with a truly random function $F : \{0,1\}^* \to \{0,1\}^{l(n)}$. Let $\mathsf{H}_1 = \{\mathsf{real\text{-}view}_F, \{\widetilde{w}_i\}_{i\in[m]}\}$ denote the combined view of $\mathcal{A}$ in the real experiment of the protocol rNMZK_F and the values extracted the committed value oracle. It then follows from the security of pseudorandom function that, the view and the value extracted from oracle are computationally indistinguishable in H_0 and H_1. Otherwise, we can use the adversary to break the indistinguishability between the pseudorandom function family and truly random function family.

Next, we construct our simulator $\hat{\mathsf{S}}$ based on the simulator S in the "The View Generated by the Simulator" section. We need $\hat{\mathsf{S}}$ to be able to emulate the execution for the *man-in-the-middle* and *resetting* adversary $\mathcal{A}$ in the protocol rNMZK_F. For the adversary $\mathcal{A}$, we divide its *resetting* attack in the left into two cases. The first case is that the new first message m_0 sent by $\mathcal{A}$ is different from all the first messages in the previous sessions on the left. Because our protocol rNMZK_F uses the truly random function F, in such case, we can see it as a new session, and simulator $\hat{\mathsf{S}}$ just does the simulation of the left and right interactions in the same manner as S. Additionally, when executing the part of resettably-roundness ZK protocol, the simulator $\hat{\mathsf{S}}$ will act as an honest verifier on the left. The second case is that the new first message m_0 sent by $\mathcal{A}$ has been sent in a previous session, and then the simulator $\hat{\mathsf{S}}$ just resends the responses from its history records of the corresponding session. This is because, for a fixed truly random function F, the transcript of the whole session are fixed when the message m_0 is fixed. Otherwise, we can use this experiment to break the binding property of the commitment scheme Com.

Let $\mathsf{sim\text{-}view}_F$ be the view of $\mathcal{A}$ in the simulated experiment of the protocol rNMZK_F by the simulator $\hat{\mathsf{S}}$, $\{\widetilde{w}_i\}_{i\in[m]}$ be the values extracted by the committed value oracle. It is easy to see that the $\{\mathsf{sim\text{-}view}_F\}$ and $\{\mathsf{real\text{-}view}_F\}$ are computationally indistinguishable, otherwise we can use this experiment to break the *concurrent* zero-knowledge of the protocol CNMZK. Now denote $\mathsf{H}_2 = \{\mathsf{sim\text{-}view}_F, \{\widetilde{w}_i\}_{i\in[m]}\}$ as the combined view of $\mathcal{A}$ in the simulate and extract experiment of the protocol rNMZK_F. As before, we can construct a series of hybrids as "The Witnesses Output by the Extractor" section to argument that the view and the values are indistinguishable in H_2 and H_1 by reducing to the security of the 4-*robust one-one CCA-secure* commitment scheme $\mathsf{Com}^{1:1}$(the non-malleable w.r.t itself or the 4-round WISSP).

More specifically, suppose that when the adversay $\mathcal{A}$ complete the resetting attack against the prover, the total number of rounds of the left interactions is N' and w.l.o.g, we assume N' is bounded by a fixed polynomial. For each $i \in[N']$, define the simulator $\hat{\mathsf{S}}^i$ that the first i

Table 3 Constant-Round Resettable NMZK Argument(rNMZK)

The prover P and the verifier V on common input 1^n, x and id, and private input w for P:

1. V sends $m_0 = (\mathsf{Com}(r_1), \cdots, \mathsf{Com}(r_\ell))$ to P, where $\ell = O(1)$ is the number of rounds of the CNMZK protocol.

2. P chooses a random seed s for a pseudorandom function $f_s : \{0,1\}^* \to \{0,1\}^{l(n)}$ where $l(n)$ is the upper bound of the size of random bits that P needs in each round of the protocol CNMZK in Table 2

3. P and V run protocol CNMZK with the following modifications:

   – For each message m_i that V sends in the i-th round of CNMZK, V and P run (P_{rsZK}, V_{rsZK}) so that V proves to P that m_i is computed using random bits r_i that committed in m_0 in the first round.

   – For each message m'_i that P sends in the i-th round of CNMZK, P applies f_s to the transcript so far and uses the output as random bits to compute m'_i.

communication rounds are simulated by simulator $\hat{\mathsf{S}}$ with the pseudo-randomness and fake witness, and all the later communication round $j > i$ are simulated by simulator $\hat{\mathsf{S}}$ with true randomness and the true witnesses. We also define the simulator $\hat{\mathsf{S}}_+^i$ that proceed identically as $\hat{\mathsf{S}}^i$ except that it simulates the i-th round following the honest prover strategy using the real witness. Then, let us consider the following hybrid experiments. The experiment $\hat{\mathsf{H}}^i\left(\hat{\mathsf{H}}_+^i\right)$ is the same as H_2 except that the execution of $\hat{\mathsf{S}}$ is replaced with that of $\hat{\mathsf{S}}^i\left(\hat{\mathsf{S}}_+^i\right)$. It is easy to see that the output of $\hat{\mathsf{H}}^{N'}$ is identical to the output of H_2, and the output of $\hat{\mathsf{H}}^0$ is identical to the real view of H_1.

Now, assume there exists a polynomial function p, such that the resetting attacker $\mathcal{A}$ cheats in one of the right sessions in the experiment H_2 with probability $1/p(n)$. We mean that there exists a right session that is accepted and uses a different identity from all the left interactions, $\mathcal{A}$ fails to commit to a valid witness in Stage 5 with probability $1/p(n)$. Then H_2 can not extract the witness from this right session with probability $1/p(n)$ as well. However, we have that H_1 can extract the witness from this right session *except with negligible probability*. Thus, from an average argument, there must exist an i such that the probability of cheating differ by at least a polynomial amount in the hybrids $\hat{\mathsf{H}}^i$ and $\hat{\mathsf{H}}_+^i$ or in the hybrids $\hat{\mathsf{H}}_+^i$ and $\hat{\mathsf{H}}^{i+1}$.

The same analysis as before, the only difference between $\hat{\mathsf{H}}_+^i$ and $\hat{\mathsf{H}}^{i+1}$ is which randomness (true or pseudo) is used in the i-round of the left interaction, hence the two ensembles are computationally indistinguishable. On the other hand, the only difference between $\hat{\mathsf{H}}^i$ and $\hat{\mathsf{H}}_+^i$ is which witness (fake or real) is used in the i-th round. The former, uses a dummy string 0^n as the committed value of $\mathsf{CCACom}^{1:1}$ followed with an WISSP for knowing the fake witness instead of the witness w_i of x_i; the latter, acts as an honest prover holding a real witness w_i of x_i. If the gap is due to the committed value of $\mathsf{CCACom}^{1:1}$, then we can use this gap to break the security of the non-malleable w.r.t itself. If the gap is due to the witness used in the four-round WISSP of the left session, then we can use this gap to break the *4-robustness CCA-secure* of $\mathsf{CCACom}^{1:1}$. Hence, we obtain a contradiction.

Thus, we have that H_2 is computationally indistinguishable from H_1. Recall that in the beginning we have proved that $\mathsf{H}_1 \approx \mathsf{H}_0$, so we have that H_2 is also computationally indistinguishable from H_0. Combining the above, we obtain that the protocol in Table 3 is resettable non-malleable zero-knowledge.

This concludes the proof of Lemma 2. □

Towards constant-round simultaneously-resettable NMZK

Towards the *constant-round simultaneously-resettable* NMZK, we first transform the *constant-round* CNMZK protocol into a *constant-round resettably-sound* CNMZK (rsCNMZK), which is similar to the method of (Chongchitmate et al. 2017). More specifically, in each round, we let the verifier generate its randomness by using a pseudorandom function $f_s : \{0,1\}^* \to \{0,1\}^{l(n)}$ to his transcript so far. Additionally, we replace the ZK argument in stage 4 with a constant-round rNMZK argument constructed in Table 3.

The final step, to obtain our Theorem 2, we apply the transformation of (Deng et al. 2009) (Theorem 6) to our constant-round rsCNMZK protocol to obtain the constant-round simultaneous resettability NMZK. This step can be proved by using the same approach in "The View Generated by the Simulator" section based on the analysis of (Deng et al. 2009). Intuitively, on the one hand, a protocol with an extra resettably-sound property will not increase the power of the *man-in-the-middle* adversary on the right; on the other hand, for a *man-in-the-middle* adversary with resetting-attack on the left, we can construct a simulator-extractor to simulate its view and extract the witnesses in the right accepted session, otherwise we can use this experiment to break the 4-*robust one-one CCA-secure* commitment scheme $\mathsf{Com}^{1:1}$.

Combining "From Concurrent NMZK to Resettable NMZK" section and "Towards Constant-round Simultaneously-Resettable NMZK" section, the *constant-round simultaneously-resettable non-malleable zero-knowledge* protocol follows.

This completes the proof of Theorem 2. □

Conclusions

In this paper, we provide the first construction of a *constant-round concurrent non-malleable zero-knowledge argument* for every language in **NP** and give a detailed proof for our protocol. Furthermore, by studying the composition of the *simultaneously resettable zero-knowledge* and the *non-malleable zero-knowledge*, we give the first construction of a *constant-round simultaneously-resettable non-malleable zero-knowledge*. However, there is still an interesting question about how to design a *round-optimal concurrent non-malleable zero-knowledge argument*. Here we leave it as an open problem.

Endnotes

[1] Here, $P(x_i, w_i, r_j, \alpha)$ denotes the message sent by the strategy P on common input x_i, auxiliary input w_i and random-tape r_j, after seeing the message-sequence α.

[2] 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 α.

Acknowledgments

This work was supported in part by the National Natural Science Foun- dation of China (Grant No. 61772521), Key Research Program of Frontier Sciences, CAS (QYZDB-SSW-SYS035), and the Open Project Program of the State Key

Laboratory of Cryptology. The first author wants to thank Yiwen Gao for making useful comments on the paper.

Authors' contributions
All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 22 May 2018 Accepted: 7 September 2018
Published online: 29 September 2018

References

Barak B (2001) How to go beyond the black-box simulation barrier. In: 42nd Annual Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Las Vegas. pp 106–115. https://doi.org/10.1109/SFCS.2001.959885

Barak B, Goldreich O, Goldwasser S, Lindell Y (2001) Resettably-sound zero-knowledge and its applications. In: 42nd Annual Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Las Vegas. pp 116–125. https://doi.org/10.1109/SFCS.2001.959886

Barak B, Goldreich O, Impagliazzo R, Rudich S, Sahai A, Vadhan SP, Yang K (2001) On the (im)possibility of obfuscating programs. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Proceedings. Springer, Santa Barbara. pp 1–18. https://doi.org/10.1007/3-540-44647-8_1

Barak B, Prabhakaran M, Sahai A (2006) Concurrent non-malleable zero knowledge. In: 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), Proceedings. IEEE Computer Society, Berkeley. pp 345–354. https://doi.org/10.1109/FOCS.2006.21

Bellare M, Yee BS (2003) Forward-security in private-key cryptography. In: Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference, Proceedings. Springer, San Francisco. pp 1–18. https://doi.org/10.1007/3-540-36563-X_1

Bitansky N, Paneth O (2015) On non-black-box simulation and the impossibility of approximate obfuscation. SIAM J Comput 44(5):1325–1383

Boyle E, Chung K, Pass R (2014) On extractability obfuscation. In: Theory of Cryptography - 11th Theory of Cryptography Conference, TCC. Proceedings. Springer, San Diego. pp 52–73. https://doi.org/10.1007/978-3-642-54242-8_3

Canetti R, Goldreich O, Goldwasser S, Micali S (2000) Resettable zero-knowledge (extended abstract). In: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing. ACM, Portland. pp 235–244. https://doi.acm.org/10.1145/335305.335334

Canetti R, Kilian J, Petrank E, Rosen A (2001) Black-box concurrent zero-knowledge requires omega~(log n) rounds. In: Proceedings on 33rd Annual ACM Symposium on Theory of Computing, STOC. ACM, Heraklion. pp 570–579. http://doi.acm.org/10.1145/380752.380852

Canetti R, Lin H, Paneth O (2013) Public-coin concurrent zero-knowledge in the global hash model. In: Theory of Cryptography - 10th Theory of Cryptography Conference, TCC. Proceedings. Springer, Tokyo. pp 80–99. https://doi.org/10.1007/978-3-642-36594-2_5

Canetti R, Lin H, Pass R (2010) Adaptive hardness and composable security in the plain model from standard assumptions. In: 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Las Vegas. pp 541–550. https://doi.org/10.1109/FOCS.2010.86

Cho C, Ostrovsky R, Scafuro A, Visconti I (2012) Simultaneously resettable arguments of knowledge. In: Theory of Cryptography - 9th Theory of Cryptography Conference, TCC. Proceedings. Springer, Taormina. pp 530–547. https://doi.org/10.1007/978-3-642-28914-9_30

Chongchitmate W, Ostrovsky R, Visconti I (2017) Resettably-sound resettable zero knowledge in constant rounds. In: Theory of Cryptography - 15th International Conference, TCC, Proceedings, Part II. Springer, Baltimore. pp 111–138. https://doi.org/10.1007/978-3-319-70503-3_4

Chung K-M, Lin H, Pass R (2013) Constant-round concurrent zero knowledge from p-certificates. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Berkeley. pp 50–59. https://doi.org/10.1109/FOCS.2013.14

Chung K-M, Lin H, Pass R (2015) Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In: Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Proceedings, Part I. Springer, Santa Barbara. pp 287–307. https://doi.org/10.1007/978-3-662-47989-6_14

Chung K-M, Ostrovsky R, Pass R, Venkitasubramaniam M, Visconti I (2014) 4-round resettably-sound zero knowledge. In: Theory of Cryptography - 11th Theory of Cryptography Conference, TCC. Proceedings. Springer, San Diego. pp 192–216. https://doi.org/10.1007/978-3-642-54242-8_9

Chung K-M, Ostrovsky R, Pass R, Visconti I (2013a) Simultaneous resettability from one-way functions. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Berkeley. pp 60–69. https://doi.org/10.1109/FOCS.2013.15

Chung K-M, Pass R, Seth K (2013b) Non-black-box simulation from one-way functions and applications to resettable security. In: Symposium on Theory of Computing Conference, STOC'13. ACM, Palo Alto. pp 231–240. http://doi.acm.org/10.1145/2488608.2488638

Ciampi M, Ostrovsky R, Siniscalchi L, Visconti I (2016) Concurrent non-malleable commitments (and more) in 3 rounds. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Proceedings, Part III. Springer, Santa Barbara. pp 270–299. https://doi.org/10.1007/978-3-662-53015-3_10

Deng Y, Goyal V, Sahai A (2009) Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In: 50th Annual IEEE Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Atlanta. pp 251–260. https://doi.org/10.1109/FOCS.2009.59

Dolev D, Dwork C, Naor M (2000) Nonmalleable cryptography. SIAM J Comput 30(2):391–437

Dwork C, Naor M, Sahai A (1998) Concurrent zero-knowledge. In: Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, STOC. ACM, Dallas. pp 409–418. http://doi.acm.org/10.1145/276698.276853

Garg S, Gentry C, Halevi S, Raykova M, Sahai A, Waters B (2013) Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Berkeley. pp 40–49. https://doi.org/10.1109/FOCS.2013.13

Garg S, Ostrovsky R, Visconti I, Wadia A (2012) Resettable statistical zero knowledge. In: Theory of Cryptography - 9th Theory of Cryptography Conference, TCC. Proceedings. Springer, Taormina. pp 494–511. https://doi.org/10.1007/978-3-642-28914-9_28

Goldwasser S, Micali S, Rackoff C (1989) The knowledge complexity of interactive proof systems. SIAM J Comput 18(1):186–208. https://doi.org/10.1137/0218012

Goyal V, Lin H, Pandey O, Pass R, Sahai A (2015) Round-efficient concurrently composable secure computation via a robust extraction lemma. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC, Proceedings, Part I. Springer, Warsaw. pp 260–289. https://doi.org/10.1007/978-3-662-46494-6_12

Håstad J, Impagliazzo R, Levin LA, Luby M (1999) A pseudorandom generator from any one-way function. SIAM J Comput 28(4):1364–1396

Ishai Y, Pandey O, Sahai A (2015) Public-coin differing-inputs obfuscation and its applications. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC, Proceedings, Part II. Springer, Warsaw. pp 668–697. https://doi.org/10.1007/978-3-662-46497-7_26

Khurana D, Sahai A (2017) How to achieve non-malleability in one or two rounds. In: 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Berkeley. pp 564–575. https://doi.org/10.1109/FOCS.2017.58

Kilian J, Petrank E, Rackoff C (1998) Lower bounds for zero knowledge on the internet. In: 39th Annual Symposium on Foundations of Computer Science, FOCS '98. IEEE Computer Society, Palo Alto. pp 484–492. https://doi.org/10.1109/SFCS.1998.743499

Kiyoshima S (2014) Round-efficient black-box construction of composable multi-party computation. In: Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Proceedings, Part II. Springer, Santa Barbara. pp 351–368. https://doi.org/10.1007/978-3-662-44381-1_20

Kiyoshima, S (2015) An alternative approach to non-black-box simulation in fully concurrent setting. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC, Proceedings, Part I. Springer, Warsaw. pp 290–318. https://doi.org/10.1007/978-3-662-46494-6_13

Lin H, Pass R (2009) Non-malleability amplification. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC. ACM, Bethesda. pp 189–198. http://doi.acm.org/10.1145/1536414.1536442

Lin, H, Pass R (2011) Concurrent non-malleable zero knowledge with adaptive inputs. In: Theory of Cryptography - 8th Theory of Cryptography Conference, TCC. Proceedings. Springer, Providence. pp 274–292. https://doi.org/10.1007/978-3-642-19571-6_17

Lin H, Pass R (2012) Black-box constructions of composable protocols without set-up. In: Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference. Proceedings. Springer, Santa Barbara. pp 461–478. https://doi.org/10.1007/978-3-642-32009-5_27

Lin H, Pass R, Soni P (2017) Two-round and non-interactive concurrent non-malleable commitments from time-lock puzzles. In: 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS. IEEE Computer Society, Berkeley. pp 576–587. https://doi.org/10.1109/FOCS.2017.59

Lin H, Pass R, Tseng WD, Venkitasubramaniam M (2010) Concurrent non-malleable zero knowledge proofs. In: Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference. Proceedings. Springer, Santa Barbara. pp 429–446. https://doi.org/10.1007/978-3-642-14623-7_23

Lin H, Pass R, Venkitasubramaniam M (2008) Concurrent non-malleable commitments from any one-way function. In: Theory of Cryptography, Fifth Theory of Cryptography Conference, TCC. Springer, New York. pp 571–588. https://doi.org/10.1007/978-3-540-78524-8_31

Micciancio D, Ong SJ, Sahai A, Vadhan S (2006) Concurrent zero knowledge without complexity assumptions. In: Theory of Cryptography, Third Theory of Cryptography Conference, TCC, Proceedings. Springer, New York. pp 1–20. https://doi.org/10.1007/11681878_1

Orlandi C, Ostrovsky R, Rao V, Sahai A, Visconti I (2014) Statistical concurrent non-malleable zero knowledge. In: Theory of Cryptography - 11th Theory of Cryptography Conference, TCC. Proceedings. Springer, San Diego. pp 167–191. https://doi.org/10.1007/978-3-642-54242-8_8

Ostrovsky R, Pandey O, Visconti I (2010) Efficiency preserving transformations for concurrent non-malleable zero knowledge. In: Theory of Cryptography, 7th Theory of Cryptography Conference, TCC. Proceedings. Springer, Zurich. pp 535–552. https://doi.org/10.1007/978-3-642-11799-2_32

Ostrovsky R, Persiano G, Visconti I (2008) Constant-round concurrent non-malleable zero knowledge in the bare public-key model. In: Automata, Languages and Programming, 35th International Colloquium, ICALP, Proceedings. Springer, Reykjavik. pp 548–559. https://doi.org/10.1007/978-3-540-70583-3_45

Ostrovsky R, Scafuro A, Venkitasubramaniam M (2015) Resettably sound zero-knowledge arguments from owfs - the (semi) black-box way. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC, Procee6dings, Part I. Springer, Warsaw. https://doi.org/10.1007/978-3-662-46494-6_15

Pandey O, Prabhakaran M, Sahai A (2015) Obfuscation-based non-black-box simulation and four message concurrent zero knowledge for NP. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC, Proceedings, Part II. Springer, Warsaw. pp 638–667. https://doi.org/10.1007/978-3-662-46497-7_25

Pass R, Rosen A (2005) Concurrent non-malleable commitments. In: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), Proceedings. IEEE Computer Society, Pittsburgh. pp 563–572. https://doi.org/10.1109/SFCS.2005.27

Rosen A (2000) A note on the round-complexity of concurrent zero-knowledge. In: Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Proceedings. Springer, Santa Barbara. pp 451–468. https://doi.org/10.1007/3-540-44598-6_28