Statistical Concurrent Non-malleable Zero-knowledge from One-way Functions

Susumu Kiyoshima

NTT Secure Platform Laboratories
kiyoshima.susumu@lab.ntt.co.jp

Abstract. *Concurrent non-malleable zero-knowledge* (CNMZK) protocols are zero-knowledge protocols that are secure even when the adversary interacts with multiple provers and verifiers simultaneously. Recently, the first *statistical* CNMZK argument for $\mathcal{NP}$ was constructed by Orlandi el al. (TCC'14) under the DDH assumption.

In this paper, we construct a statistical CNMZK argument for $\mathcal{NP}$ *assuming only the existence of one-way functions*. The security is proven via black-box simulation, and the round complexity is $\mathsf{poly}(n)$. Under the existence of collision-resistant hash functions, the round complexity can be reduced to $\omega(\log n)$, which is essentially optimal for black-box concurrent zero-knowledge.

1 Introduction

Zero-knowledge (ZK) *proofs* and *arguments* are protocols that enable the prover to convince the verifier of the correctness of a mathematical statement while providing *zero additional knowledge*. This "zero additional knowledge" property is formalized by using the *simulation paradigm*: An interactive proof or argument is said to be zero-knowledge if for any adversarial verifier there exists a *simulator* that can output a simulated view of the adversary. In the original definition of the ZK property, the adversary interacts with a single prover at a time. Thus, the original definition guarantees the ZK property in the stand-alone setting.

Non-malleable zero-knowledge (NMZK) [6] and *concurrent zero-knowledge* (CZK) [7] are security notions that guarantee the ZK property in the concurrent setting. Specifically, NMZK guarantees the ZK property in the setting where the adversary concurrently interacts with a honest prover in the *left session* and a honest verifier in the *right session*, and CZK guarantees the ZK property in the setting where the adversary concurrently interacts with unbounded number of honest provers.

As a security notion that implies both NMZK and CZK, Barak et al. [1] proposed *concurrent non-malleable zero-knowledge* (CNMZK). CNMZK guarantees the ZK property in the setting where the adversary concurrently interacts with many provers in the left sessions and many verifiers in the right sessions. In particular, it guarantees that receiving proofs in the left session does not help the adversary to give proofs in the right sessions—that is, it guarantees that if the adversary can prove some statements in the right sessions while receiving proofs in the left sessions, the adversary could prove the same statements *even without receiving proofs in the left sessions*. In the definition of

CNMZK, this guarantee is formalized as the existence of a *simulator-extractor* that can simulate the adversary's view in the left and right sessions while extracting witnesses from the adversary in the simulated right sessions.

The first CNMZK argument was constructed by Barak et al. [1]. Subsequently, a computationally efficient construction was shown by Ostrovsky et al. [21]. The first CNMZK *proof* was constructed by Lin et al. [16], and a variant of their protocol was shown to be secure with adaptively chosen inputs by Lin and Pass [14]. Additionally, a CNMZK argument that is secure with "fully" adaptively chosen inputs was recently constructed by Venkitasubramaniam [26].

Very recently, Orlandi et al. [20] constructed the first *statistical* CNMZK argument—that is, a CNMZK argument such that the view simulated by the simulator-extractor is statistically indistinguishable from the adversary's view. Statistical CNMZK is clearly of great interest since it guarantees quite strong security in the concurrent setting. However, statistical CNMZK is hard to achieve, and the existing techniques of computational CNMZK protocols seem to be insufficient for constructing statistical CNMZK protocols (see Section 2.1).

On statistical CNMZK protocols, an important open question is what hardness assumption is needed for constructing them. The statistical CNMZK argument of Orlandi et al. [20] was constructed under the DDH assumption (or the existence of dense cryptosystems). Thus, it is already known that statistical CNMZK protocols can be constructed under standard assumptions. However, since it is known that the existence of one-way functions is sufficient for constructing both statistical ZK protocols and computational CNMZK protocols [10, 1], it is important to study the following question.

> *Can we construct statistical concurrent non-malleable zero-knowledge protocols by assuming only the existence of one-way functions?*

1.1 Our Result

In this paper, we answer the above question affirmatively.

Theorem 1. *Assume the existence of one-way functions. Then, there exists a statistical concurrent non-malleable zero-knowledge argument for $\mathcal{NP}$ with round complexity* $\mathsf{poly}(n)$*. Furthermore, if there exists a family of collision-resistant hash functions, the round complexity can be reduced to* $\omega(\log n)$*.*

The round complexity of our statistical CNMZK argument—$\mathsf{poly}(n)$ rounds when only the existence of one-way functions is assumed and $\omega(\log n)$ rounds when the existence of a family of collision-resistant hash functions is assumed—is the same as the round complexity of the known statistical CZK arguments [9]. Thus, our result closes the gap between statistical CNMZK arguments and statistical CZK arguments. Furthermore, since the security of our statistical CNMZK protocol is proven via black-box simulation, the logarithmic round complexity of our hash-function-based protocol is essentially tight due to the lower bound on black-box CZK protocols [3].

2 Techniques

2.1 Previous Techniques

Before explaining our technique, we explain the difficulty of constructing statistical CNMZK protocols by using the techniques of existing computational CNMZK protocols [1, 16].

We first recall the protocols of [1, 16]. The definition of CNMZK requires the existence of a simulator-extractor that simulates the adversary's view while extracting the witnesses for the statements proven by the adversary in the simulated view. To satisfy this definition, protocols need to satisfy the following properties: (i) the proofs in the left sessions can be simulated for the adversary; (ii) even when the adversary receives simulated proofs in the left sessions, the witnesses can be extracted from the adversary in the right sessions. In the protocol of [1, 16], the simulatability of the left sessions is guaranteed by requiring the verifier to commit to a random trapdoor by using a *concurrently extractable commitment scheme* CECom [17]. Since the committed values of CECom can be extracted by a rewinding extractor even in the concurrent setting, the proofs in the left sessions can be simulated by extracting the trapdoors from CECom. On the other hand, the witness-extractability of the right sessions is guaranteed by requiring the prover to commit to the witness with a non-malleable commitment scheme NMCom [6] and additionally designing the protocols so that the following hold.

1. When the adversary receives honest proofs in the left sessions, the committed value of the NMCom commitment is indeed a valid witness in every accepted right session.
2. When the proofs in the left sessions are switched to the simulated ones, the committed values of the NMCom commitments do not change in the right sessions due to the non-malleability of NMCom.

It follows from these that even when the adversary receives simulated proofs in the left sessions, the committed value of the NMCom commitment is a witness for the statement in every accepted right session. Therefore, the witnesses can be extracted in the right sessions by extracting the committed values of the NMCom commitments.

As mentioned above, the techniques of [1, 16] alone seem to be insufficient for constructing statistical CNMZK protocols. This is because the techniques of [1, 16] requires the prover to commit to the witness by using NMCom, which is only computationally hiding.[1] Since in the simulation the committed values of NMCom need to be switched to another values (e.g., 0^n) in the left sessions, the simulated view can be only computational indistinguishable from the real view.

Recently, Orlandi et al. [20] constructed a statistical CNMZK protocol by modifying the CNMZK protocol of [1] with *mixed non-malleable commitment scheme* $\mathsf{MXNMCom}$. $\mathsf{MXNMCom}$ is parametrized by a string and is either statistically hiding or non-malleable

[1] NMCom need to be *non-malleable w.r.t. commitment* [6], which roughly says that the committed value of the commitment that the man-in-the-middle adversary gives is independent of the committed value of the commitment that adversary receives. Since the definition of non-malleability w.r.t. commitment is meaningless when the committed values cannot be uniquely determined, NMCom cannot be statistically hiding.

depending on the string.[2] Very roughly speaking, Orlandi et al. circumvent the above problem by switching the parameter string of $\mathsf{MXNMCom}$ in the security proof—when proving the statistical indistinguishability of the simulation, the string is set so that $\mathsf{MXNMCom}$ is statistically hiding, and when proving the non-malleability, the string is set so that $\mathsf{MXNMCom}$ is non-malleable. The use of $\mathsf{MXNMCom}$, however, requires assumptions that are stronger than the existence of one-way functions (such as the DDH assumption or the existence of dense cryptosytems). Thus, the technique of Orlandi et al. cannot be used to construct statistical CNMZK protocols from one-way functions.

2.2 Our Technique

Since the techniques of [1, 16] cannot be used for statistical CNMZK protocols because the committed values of NMCom need to be switched during the simulation, one potential strategy for statistical CNMZK is to construct a protocol such that the adversary's view can be simulated *without switching the committed value of* NMCom *(and of any other computationally hiding commitment)*. However, when the simulator commits to the same value in NMCom as a honest prover, it is not clear how non-malleability of NMCom can be used in the security proof. Below, we show that the CNMZK property can be shown even in this case *if we use a stronger variant of* NMCom.

A key technical tool in our technique is *CCA-secure commitment schemes* [4], which is a stronger variant of (concurrent) non-malleable commitment schemes. Roughly speaking, CCA security guarantees that the scheme is hiding even against adversaries that have access to the *committed-value oracle*, which receives concurrent commitments from the adversary and returns their committed values to the adversary. (In non-malleability, the oracle receives only parallel commitments from the adversary and returns the committed values only after the adversary finishes the interaction with the committer.) Several CCA-secure commitment schemes were constructed from one-way functions [4, 15, 12, 8]; furthermore, although CCA security itself does not provide any extractability, all of these schemes satisfy concurrent extractability as well.

Using CCA-secure commitment schemes, we construct the following protocol as a starting point.

Stage 1. (V commits to trapdoor)

1. The verifier V chooses random $r_V \in \{0,1\}^n$ and commits to r_V by using a statistically binding commitment scheme Com, which can be constructed from one-way functions [18, 11]. Let (r_V, d) be the decommitment.
2. V commits to (r_V, d) by using $\mathsf{CCA\text{-}CECom}$, where $\mathsf{CCA\text{-}CECom}$ is a CCA-secure commitment scheme that is also concurrent extractable [4, 15, 12, 8].

Stage 2. (P proves $x \in L$ or knowledge of trapdoor) The prover P proves that it knows a witness for $x \in L$ or a valid decommitment (r_V, d) of the Com commitment that V gives in Stage 1. P proves this statement by using a statistical witness-indistinguishable argument of knowledge sWIAOK, which can be constructed from

[2] Specifically, Orlandi et al. [20] used the scheme such that (i) when the string is sampled from a uniform distribution, the scheme is statistically hiding and (ii) when the string is taken from another (computationally indistinguishable) distribution, the scheme is non-malleable.

one-way functions by instantiating Blum's Hamiltonian-cycle protocol with the statistically hiding commitment scheme of [10].

In this protocol, the verifier's view can be statistically simulated by a simulator that extracts (r_V, d) from $\mathsf{CCA\text{-}CECom}$ and uses it as a witness in sWIAOK. (Note that this simulator executes Stage 1 honestly; thus, even if computationally hiding commitment schemes are used as building blocks in $\mathsf{CCA\text{-}CECom}$, the simulator commits to the same values by using them as a honest prover.) Also, intuitively this protocol seems to be CNMZK from the following reason.

- The CCA security of $\mathsf{CCA\text{-}CECom}$ guarantees that the trapdoors of the right sessions are hidden from the adversary even when the trapdoors of the left sessions are extracted and returned to the adversary.
- Then, since the simulated proofs are generated in the left sessions by extracting the trapdoors, the trapdoors in the right sessions are hidden from the adversary even when the adversary receives simulated proofs in the left sessions.
- Thus, even when the adversary receives the simulated proofs in the left sessions, the adversary cannot "cheat" in the right sessions, and therefore witnesses for the statements must be extractable from sWIAOK in the right sessions.

Of course, to formally show the statistical CNMZK property, we need to show a simulator-extractor that statistically simulates the adversary's view and also extracts witnesses for the statements in the right sessions.

As the simulator-extractor, we consider the following $\mathcal{SE}$.

1. First, $\mathcal{SE}$ simulates the view of the adversary $\mathcal{A}$ by executing the following simulator $\mathcal{S}$: Simulator $\mathcal{S}$ internally invokes $\mathcal{A}$ and interacts with it in the left and right sessions honestly except that in each left session, $\mathcal{S}$ extracts (r_V, d) by using the concurrent extractor of $\mathsf{CCA\text{-}CECom}$ and uses it as a witness in sWIAOK.
2. After simulating the view of $\mathcal{A}$ as above, $\mathcal{SE}$ extracts witnesses from the right sessions by doing the following for each right session. First, $\mathcal{SE}$ rewinds $\mathcal{S}$ until the point just before $\mathcal{S}$ sends the challenge message of sWIAOK to $\mathcal{A}$.[3] Then, $\mathcal{SE}$ repeatedly executes $\mathcal{S}$ from this point with flesh randomness until it obtains another accepted transcript of sWIAOK. After obtaining another accepted transcript, $\mathcal{SE}$ extracts a witness by using the argument-of-knowledge property of sWIAOK.

It is easy to see that $\mathcal{SE}$ statistically simulates the real view of $\mathcal{A}$. Thus, it remains to show that $\mathcal{SE}$ extracts witnesses for the statements in the right sessions.

To show the witness extractability of $\mathcal{SE}$, a natural approach is to follow the above-mentioned approach of [1, 16] and show the following.

1. When $\mathcal{A}$ receives honest proofs in the left sessions, a witness for the statement is extracted from the sWIAOK proof in every accepted right session.
2. When the honest proofs in the left sessions are switched to the simulated ones, the value extracted from sWIAOK does not change in every accepted right session.

[3] Since $\mathcal{S}$ rewinds $\mathcal{A}$ during the concurrent extraction of $\mathsf{CCA\text{-}CECom}$, $\mathcal{S}$ may send the challenge message of sWIAOK of a right session to $\mathcal{A}$ multiple times. Here, $\mathcal{SE}$ rewinds $\mathcal{S}$ until the point just before $\mathcal{S}$ sends it to $\mathcal{A}$ on the "main thread."

Note that here we argue about the extracted values instead of the committed values. At first sight, it seems that this is not a big difference and it seems that the above can be shown by using an argument similar to the one used in [1, 16].

However, this approach does not work. In particular, we cannot show the second part—that is, we cannot show that the extracted values remain to be the same when the honest proofs in the left sessions are switched to the simulated ones. To see this, observe the following. Since the witnesses used in sWIAOK are switched in the simulated proofs, we need to use the witness indistinguishability of sWIAOK of the left sessions. However, since $\mathcal{A}$ is rewound during the witness extraction of the sWIAOK proofs of the right sessions, if the left and the right sessions are scheduled so that the sWIAOK proofs of the left sessions are executed in parallel with the sWIAOK proofs of the right sessions, the sWIAOK proofs of the left sessions are also rewound, and thus we cannot use their witness indistinguishability.[4]

Thus, we instead use the following approach. Informally, the above approach does not work because the honest proofs and the simulated proofs are "too different." We thus introduce a hybrid experiment in which $\mathcal{A}$ receives *hybrid proofs* in the left sessions, where a hybrid proof is generated by extracting (r_V, d) by brute force and using it as a witness in sWIAOK. (Notice that the only difference between the hybrid proofs and the simulated proofs is how the trapdoors are extracted.) We then show that (i) witnesses for the statements are extracted in the right sessions when $\mathcal{A}$ receives hybrid proofs in the left sessions, and (ii) when hybrid proofs are switched to the simulated ones, the extracted values do not change. In particular, our analysis proceeds as follows.

- First, we show the second part, i.e., we show that the values extracted in the right sessions do not change when the proofs in the left sessions are switched from the hybrid proofs to the simulated ones. Since the only difference between the hybrid proofs and the simulated ones is how the committed values of the CCA-CECom commitments are extracted (by brute-force or by the concurrent extractability), we can show this by using the concurrent extractability of CCA-CECom. We note however that there is a subtlety since CCA-CECom in the left sessions can be rewound not only by the concurrent extractor of CCA-CECom but also by the extractor of sWIAOK. Nonetheless, by carefully using a standard technique (the "good prefix" argument), we can show that the concurrent extractor of CCA-CECom works even in this case.
- Next, we show that in the hybrid experiment, witnesses for the statements are extracted from the right sessions. Since the simulated proofs can be efficiently generated given access to the committed-value oracle of CCA-CECom, at first sight it seems that this follows directly from the CCA security of CCA-CECom and argument-of-knowledge property of sWIAOK—if a witness for the statement is not

[4] If we use the robust extraction technique [8], for each left session there exists a rewinding strategy that allows us to extract witnesses from the right sessions without rewinding sWIAOK of this left session. However, since what we want to show is that the values extracted in the right sessions *by the rewinding strategy that $\mathcal{SE}$ uses* are unchanged, the robust extraction technique cannot be used here (unless there exists a rewinding strategy that allows us to extract witnesses from the right sessions without rewinding the sWIAOK proof of *every* left session).

extracted, (r_V, d) must be extracted, and thus we can break the CCA security of $\mathsf{CCA\text{-}CECom}$. However, there are two problems.

1. Since $\mathsf{CCA\text{-}CECom}$ in the left sessions can be rewound during the witness extraction of sWIAOK of the right sessions, the hybrid experiment cannot be emulated even given access to the committed-value oracle of $\mathsf{CCA\text{-}CECom}$. Hence, the CCA-secure commitments in the right sessions may not be hiding in the hybrid experiment.
2. Since the adversary obtains hybrid proofs, which are generated in super-polynomial time, the argument-of-knowledge property of sWIAOK may not hold in the hybrid experiment. We note that although existing CCA-secure commitment schemes provides *robustness*, which guarantees that arbitrary "small"-round protocol remains secure even when adversaries have access to the committed-value oracle, we cannot use robustness here since $\mathsf{CCA\text{-}CECom}$ in the left sessions can be rewound during the witness extraction of sWIAOK of the right sessions and therefore the hybrid experiment cannot be emulated even given access to the committed-value oracle.

Because of these problems, we cannot use the security of $\mathsf{CCA\text{-}CECom}$ directly in the analysis. Thus, instead of using existing CCA-secure commitment schemes in a modular way, we directly use their building blocks in the protocol and directly use their proof technique in the analysis. (In particular, we use the robust concurrent extraction technique of [8] and a one-one CCA-secure commitment scheme of [13].) The proof techniques of existing CCA-secure commitment schemes are strong enough to solve the above problems, and thus we can show that witnesses for the statements are extracted in the hybrid experiment.

From the above two, it follows that even when $\mathcal{A}$ receives simulated proofs in the left session, valid witnesses are extracted in right sessions. This completes the overview of our technique.

3 Definitions

In this section, we sketch the definitions used in this paper. The formal definitions are given in the full version.

3.1 Statistical Concurrent Non-malleable Zero-knowledge Arguments

The definition of (statistical) concurrent non-malleable zero-knowledge [1, 20] is closely related to the definition of simulation extractability of [22]. Let $\langle P, V\rangle$ be an interactive argument for a language $L \in \mathcal{NP}$. For any man-in-the-middle adversary $\mathcal{A}$, let us consider a probabilistic experiment in which $\mathcal{A}$ participates in the following left and right interactions. In the left interaction, $\mathcal{A}$ interacts with a honest prover P of $\langle P, V\rangle$ and verifies the validity of statements $x_1, \ldots, x_m$ using identities $\mathsf{id}_1, \ldots, \mathsf{id}_m$. In the right interaction, $\mathcal{A}$ interacts with a honest verifier V of $\langle P, V\rangle$ and proves the validity of statements $\widetilde{x}_1, \ldots, \widetilde{x}_m$ using identities $\widetilde{\mathsf{id}}_1, \ldots, \widetilde{\mathsf{id}}_m$. The statements proven in the left interaction, $x_1, \ldots, x_m$, are given to P and $\mathcal{A}$ prior to the experiment. In contrast, the statements proven in the right interaction, $\widetilde{x}_1, \dots, \widetilde{x}_m$, and the identities used in the left and the right interactions, $\mathsf{id}_1, \dots, \mathsf{id}_m$ and $\widetilde{\mathsf{id}}_1, \dots, \widetilde{\mathsf{id}}_m$, are chosen by $\mathcal{A}$ during the experiment. Then, roughly speaking, $\langle P, V \rangle$ is *statistical concurrent non-malleable zero-knowledge* (statistical CNMZK) if for any adversary $\mathcal{A}$, there exists a PPT machine called the *simulator-extractor* that can statistically simulate the view of $\mathcal{A}$ in the above experiment while extracting witnesses for the statements proven by $\mathcal{A}$ in the accepted right interactions that use different identities from the left interactions.

3.2 Concurrently Extractable Commitment Schemes

Roughly speaking, a commitment scheme is *concurrently extractable* if there exists a PPT extractor such that for any adversarial committer that concurrently commits to many values by using the scheme, the extractor can extract the committed value from the adversarial committer in every valid commitment.[5]

Micciancio et al. [17] showed a $\omega(\log n)$-round concurrently extractable commitment CECom (Fig. 1), which is an abstraction of the preamble stage of the concurrent zero-knowledge protocol of [25] and can be constructed from one-way functions. The extractor of CECom performs the extraction by rewinding the adversarial committer according to the rewinding strategy of [25, 23]—the extractor internally invokes the adversarial committer C^* and interacts with C^* as honest receivers on the "main thread"; at the same time, the extractor rewinds the main thread and generates "look-ahead threads" on which the extractor interacts with C^* again as honest receivers with flesh randomness; then, at the end of each commitment on each thread, the extractor extracts the committed values by using the information collected on the other threads.

CECom can be seen as concurrent executions of the extractable commitment scheme ExtCom of [24], which consists of three stages—`commit`, `challenge`, and `reply`—and can be constructed from one-way functions.

Commit phase. The committer C and the receiver R receive common input 1^n and parameter ℓ. (In [17], $\ell = \omega(\log n)$.) To commit to $v \in \{0,1\}^n$, the committer C commits to v concurrently ℓ times by using ExtCom as follows.

1. C and R execute `commit` stage of ExtCom ℓ times in parallel.
2. For each $j \in [\ell]$ in sequence, C and R do the following.
   (a) R sends the `challenge` message of ExtCom for the j-th session.
   (b) C sends the `reply` message of ExtCom for the j-th session.

Decommit phase. C sends v to R and decommits all the ExtCom commitments.

Fig. 1. Concurrently extractable commitment CECom [17].

Robust concurrent extraction. On the concurrently extractable commitment scheme CECom of [17], Goyal et al. [8] showed a very useful lemma called the *robust concurrent extraction lemma*. Roughly speaking, this lemma states that even when the

[5] A commitment is *valid* if there exists a value to which it can be decommitted.

adversarial committer additionally participates in an external protocol, the committed values can be extracted from the adversarial committer *without rewinding the external protocol* as long as the round complexity of the external protocol is "small." In particular, the lemma guarantees that the robust concurrent extraction is possible as long as $\ell - O(k \cdot \log n) = \omega(\log n)$, where ℓ is the parameter of CECom and k is the round complexity of the external protocol. (Thus, we need to set $\ell := \omega(\log n)$ when $k = O(1)$ and set $\ell := \mathsf{poly}(n)$ when $k = \mathsf{poly}(n)$.)

In this work, we cannot use the lemma in a black-box way since in the security analysis we use a specific property of the extractor shown in [8]. In particular, in our security analysis, it is important that the extractor of [8] performs the extraction by generating the main thread and the look-ahead threads as in the rewinding strategies of [25, 23].

3.3 (One-one) CCA-secure Commitment Schemes

We recall the definition of (one-one) CCA security and κ-robustness of commitment schemes [4, 15, 13].

(One-one) CCA security. Roughly speaking, a tag-based commitment scheme $\langle C, R\rangle$ (i.e., a commitment scheme that takes an n-bit string—a *tag*—as an additional input) is *CCA-secure* if it is hiding even against adversary $\mathcal{A}$ that interacts with the following *committed-value oracle*: The committed-value oracle O interacts with $\mathcal{A}$ as an honest receiver in many concurrent sessions of the commit phase of $\langle C, R\rangle$ using tags chosen adaptively by $\mathcal{A}$; at the end of each session, if the commitment of this session is invalid or has multiple committed values, O returns $\perp$ to $\mathcal{A}$; otherwise, O returns the unique committed value to $\mathcal{A}$.

If $\langle C, R\rangle$ is CCA secure only against adversaries that interact with the *one-session committed-value oracle*, which is the same as the committed-value oracle except that it interacts with the adversary only in a single session, $\langle C, R\rangle$ is *one-one CCA secure*.

κ-robustness. Roughly speaking, a tag-based commitment scheme is *κ-robust* if for any adversary $\mathcal{A}$ and any ITM B, the joint output of a κ-round interaction between $\mathcal{A}^{O}$ and B can be simulated without O by a PPT simulator. Intuitively, κ-robustness guarantees that the security of any κ-round protocol (say, the hiding property of a κ-round commitment scheme) holds even against the adversary that interacts with O.

The scheme we use. From a result shown in [8], we can obtain a constant-round κ-robust one-one CCA-secure commitment scheme for every constant $\kappa \in \mathbb{N}$ from one-way functions. In [8], Goyal et al. constructed a $\omega(\log n)$-round CCA-secure commitment scheme from one-way functions. This scheme has $\omega(\log n)$ rounds because CECom with parameter $\ell = \omega(\log n)$ is used as a building block. The reason why ℓ is set to be $\omega(\log n)$ is that in the security analysis, the committed values of CECom need to be extracted when polynomially many CECom commitments are concurrently executed. In the setting of *one-one* CCA security, however, the security analysis works even if the committed values of CECom are extractable only when a single CECom commitment is executed; hence, we can set $\ell := O(1)$. For completeness, we give the protocol and the proof of one-one CCA security in the full version.

4 Our Statistical Concurrent Non-malleable ZK Argument

We show that a statistical concurrent non-malleable zero-knowledge argument can be constructed from any statistically hiding commitment scheme.

Theorem 2. *Assume the existence of statistically hiding commitment schemes with round complexity $R_{\mathsf{SH}}(n)$. Then, there exists an $\omega(R_{\mathsf{SH}}(n)\log n)$-round statistical concurrent non-malleable zero-knowledge argument* sCNMZK.

Since $\mathsf{poly}(n)$-round statistically hiding commitment schemes can be constructed from one-way functions [10] and constant-round ones can be constructed from a family of collision-resistant hash functions [19, 5], our main theorem (Theorem 1) follows from Theorem 2.

Proof (of Theorem 2). In sCNMZK, we use the following building blocks, all of which can be constructed from $R_{\mathsf{SH}}(n)$-round statistically hiding commitment schemes (or one-way functions, which can be obtained from statistically hiding commitment schemes).

- Two-round statistically binding commitment scheme $\mathsf{Com}_{\mathsf{SB}}$ [18, 11].
- Constant-round 4-robust one-one CCA-secure commitment scheme $\mathsf{CCACom}^{1:1}$ (see Section 3.3).
- Four-round witness-indistinguishable proof of knowledge WIPOK, which is a parallel version of Blum's Hamiltonian-cycle protocol [2].
- $(R_{\mathsf{SH}}(n)+2)$-round statistical witness-indistinguishable argument of knowledge sWIAOK, which is a parallel version of Blum's Hamiltonian-cycle protocol that is instantiated with a $R_{\mathsf{SH}}(n)$-round statistically hiding commitment scheme $\mathsf{Com}_{\mathsf{SH}}$.
- $\omega(R_{\mathsf{SH}}(n)\log n)$-round concurrently extractable commitment scheme CECom, which is the scheme of [17] with parameter $\ell = \omega(R_{\mathsf{SH}}(n)\log n)$. From the robust concurrent extraction lemma [8], we can extract the committed values from any adversarial committer even when it additionally participates in any $O(R_{\mathsf{SH}}(n))$-round external protocol.

Protocol sCNMZK is shown in Fig. 2. Roughly speaking, soundness can be proven as follows. Assume that an adversary breaks the soundness. From the witness extractability of sWIAOK, a valid decommitment (r'_V, d') of the $\mathsf{Com}_{\mathsf{SB}}$ commitment of Stage 1 can be extracted from this adversary in Stage 3. Furthermore, from the hiding property of CECom and the witness indistinguishability of WIPOK, it can be shown that (r'_V, d') can be extracted even when Stage 2 is simulated by extracting r_P in Stage II-1 and using it in Stage II-2 and II-4. Then, since Stage 2 is now simulated without using the decommitment of the $\mathsf{Com}_{\mathsf{SB}}$ commitment of Stage 1, we can derive a contradiction by breaking the hiding property of $\mathsf{Com}_{\mathsf{SB}}$ or CECom by using (r'_V, d'). The formal proof is given in the full version.

In the following, we prove the statistical CNMZK property.

Simulator-extractor $\mathcal{SE}$.

Recall that to prove the statistical CNMZK property, we need to show a simulator-extractor that simulates the view of the adversary $\mathcal{A}$ and also extracts a witness in

Input. The common input is statement $x \in L$ and identity $\mathsf{id} \in \{0,1\}^n$. The prover's private input is witness $w \in \mathbf{R}_L(x)$.

Stage I. (V commits to trapdoor)

1. V chooses random $r_V \in \{0,1\}^n$ and commits to r_V by using $\mathsf{Com}_{\mathsf{SB}}$. Let (r_V, d) be the decommitment of this commitment.
2. V commits to (r_V, d) by using CECom.

Stage II. (V proves knowledge of trapdoor)

1. P chooses random $r_P \in \{0,1\}^n$ and commits to r_P by using $\mathsf{CCACom}^{1:1}$ with tag id.
2. V commits to 0^n by using CECom.
3. P decommits the $\mathsf{CCACom}^{1:1}$ commitment of Stage II-1 to r_P.
4. V proves the following by using WIPOK:
   - the committed value of the CECom commitment of Stage I-2 is a valid decommitment of the $\mathsf{Com}_{\mathsf{SB}}$ commitment of Stage I-1, or
   - the committed value of the CECom commitment of Stage II-2 is r_P.

Stage III. (P proves $x \in L$ or knowledge of trapdoor)

1. P proves the following by using sWIAOK.
   - $x \in L$, or
   - There exists (r'_V, d') such that (r'_V, d') is a valid decommitment of the $\mathsf{Com}_{\mathsf{SB}}$ commitment of Stage I-1.

Fig. 2. Statistical concurrent non-malleable zero-knowledge argument sCNMZK.

every accepted right session. We construct our simulator-extractor step by step. First, we construct a super-polynomial-time simulator $\hat{\mathcal{S}}$ that simulates the view of $\mathcal{A}$ but does not extract witnesses in the right seasons. Next, we construct a super-polynomial-time simulator-extractor $\hat{\mathcal{SE}}$ that simulates the view of $\mathcal{A}$ by executing $\hat{\mathcal{S}}$ and then extracts the witnesses by rewinding $\hat{\mathcal{S}}$. Finally, we construct a polynomial-time simulator-extractor $\mathcal{SE}$ that emulates the execution of $\hat{\mathcal{SE}}$ in polynomial time.

Remark 1. In the following, we use the hat symbol in the names of simulators and simulator-extractors if they run in super-polynomial time (e.g., $\hat{\mathcal{S}}$ and $\hat{\mathcal{SE}}$). Also, we use the tilde symbol in the names of the messages of sCNMZK if they are the messages of the right sessions (e.g., $\widetilde{r}_V$ and $\widetilde{r}_P$); if necessary, we use subscript to denote the index of the session.

Super-polynomial-time simulator $\hat{\mathcal{S}}$. First, we show the simulator $\hat{\mathcal{S}}$, which simulates the view of $\mathcal{A}$ in super-polynomial time as follows. $\hat{\mathcal{S}}$ internally invokes $\mathcal{A}$ and interacts with $\mathcal{A}$ as provers and verifiers in the following way.

- In each left session, $\hat{\mathcal{S}}$ interacts with $\mathcal{A}$ in the same way as a honest prover except for the following. In Stage I-2, $\hat{\mathcal{S}}$ extracts the committed value (r_V, d) of the CECom commitment by brute force. (If the committed value is not uniquely determined, (r_V, d) is defined to be $(\perp, \perp)$.) In Stage III, $\hat{\mathcal{S}}$ checks whether (r_V, d) is a valid decommitment of the $\mathsf{Com}_{\mathsf{SB}}$ commitment of Stage I-1; if so, $\hat{\mathcal{S}}$ gives a sWIAOK proof by using (r_V, d) as a witness; otherwise, $\hat{\mathcal{S}}$ terminates with output fail.

- In each right session, $\hat{\mathcal{S}}$ interacts with $\mathcal{A}$ in the same way as a honest verifier.

Finally, $\hat{\mathcal{S}}$ outputs the view of internal $\mathcal{A}$. Notice that $\hat{\mathcal{S}}$ does not rewind $\mathcal{A}$.

Super-polynomial-time simulator-extractor $\hat{\mathcal{SE}}$. Next, we show the simulator-extractor $\hat{\mathcal{SE}}$, which simulates the view of $\mathcal{A}$ in super-polynomial time and also extracts witnesses in every accepted right session as follows. First, $\hat{\mathcal{SE}}$ simulates the view of $\mathcal{A}$ by executing $\hat{\mathcal{S}}$. We call this execution of $\hat{\mathcal{S}}$ the WI-*main thread*. Next, for each $i \in [m]$, if the i-th right session is accepted on the WI-main thread and uses a different identity from every left session, $\hat{\mathcal{SE}}$ extracts a witness from this session as follows.

- $\hat{\mathcal{SE}}$ rewinds the WI-main thread until the point just before the challenge message of sWIAOK of the i-th right session is sent. Then, from this point, $\hat{\mathcal{SE}}$ executes $\hat{\mathcal{S}}$ again with flesh randomness (i.e., interacts with $\mathcal{A}$ as $\hat{\mathcal{S}}$ does with flesh randomness). $\hat{\mathcal{SE}}$ repeats this rewinding until it obtains another accepting transcript of the i-th right session. We call each execution of $\hat{\mathcal{S}}$ in this step a WI-*auxiliary thread*.
- After obtaining two accepting transcripts of the i-th right session (one is on the WI-main thread and the other is on an WI-auxiliary thread), $\hat{\mathcal{SE}}$ extracts a witness from sWIAOK by using the witness extractability of sWIAOK. If $\hat{\mathcal{SE}}$ fails to extract a witness for $\widetilde{x}_i \in L$ (the statement proven in the i-th right session), $\hat{\mathcal{SE}}$ terminates with output $\mathsf{fail}_{\mathsf{WI}}$. Otherwise, let $\widetilde{w}_i$ be the extracted witness.

If the i-th right session is not accepted or uses the same identity as a left session, define $\widetilde{w}_i \stackrel{\text{def}}{=} \bot$. The output of $\hat{\mathcal{SE}}$ is $(\mathsf{view}, \{\widetilde{w}_i\}_{i\in[m]})$, where view is the view of $\mathcal{A}$ on the WI-main thread.

Polynomial-time simulator-extractor $\mathcal{SE}$. Finally, we show the simulator-extractor $\mathcal{SE}$, which emulates the execution of $\hat{\mathcal{SE}}$ in polynomial time as follows. First, $\mathcal{SE}$ emulates the WI-main thread in polynomial time as follows.

- $\mathcal{SE}$ internally invokes $\mathcal{A}$ and interacts with $\mathcal{A}$ as $\hat{\mathcal{S}}$ does except that in each left session, $\mathcal{SE}$ extracts (r_V, d) by using the concurrent extractability of CECom. Recall that a concurrent extraction of CECom involves the generation of a main thread and many look-ahead threads. We call the main thread generated during the concurrent extraction of CECom the CEC-*main thread*, and call the look-ahead threads generated during the concurrent extraction of CECom the CEC-*auxiliary threads*.[6]

Next, for each $i \in [m]$, if the i-th right session is accepted on the emulated WI-main thread and uses a different identity from every left session, $\mathcal{SE}$ emulates WI-auxiliary threads as follows.

- $\mathcal{SE}$ rewinds the emulation of the WI-main thread until the point just before the challenge message of sWIAOK of the i-th right session is sent on the CEC-main thread. Then, from this point, $\hat{\mathcal{SE}}$ emulates the WI-main thread again with flesh randomness (i.e., generates the rest of CEC-main thread and CEC-auxiliary threads with flesh randomness). $\mathcal{SE}$ repeats this rewinding until it obtains another accepted transcript of the i-th right session on an emulated WI-auxiliary thread.

Let $(\mathsf{view}, \{\widetilde{w}_i\}_{i\in[m]})$ be the output of the emulated $\hat{\mathcal{SE}}$. Then, $\mathcal{SE}$ outputs $(\mathsf{view}, \{\widetilde{w}_i\}_{i\in[m]})$.

[6] Note that the WI-main thread is also a CEC-main thread.

Analysis of poly-time simulator-extractor $\mathcal{SE}$.

To prove the statistical CNMZK property, we show that $\mathcal{SE}$ statistically simulates the view of $\mathcal{A}$ and also extracts witnesses for the statements in the right sessions.

Lemma 1. *The view of $\mathcal{A}$ simulated by $\mathcal{SE}$ is statistically indistinguishable from the view of $\mathcal{A}$ in the real experiment. Furthermore, except with negligible probability, $\mathcal{SE}$ outputs witnesses for the statements proven by $\mathcal{A}$ in the accepted right sessions that use different identities from the left sessions.*

Proof (sketch). In this proof, we use the following claim, which states that the super-polynomial-time simulator-extractor $\hat{\mathcal{SE}}$ statistically simulates the view of $\mathcal{A}$ and also extracts the witnesses from the right sessions.

Claim 1. *The view of $\mathcal{A}$ simulated by $\hat{\mathcal{SE}}$ is statistically indistinguishable from the view of $\mathcal{A}$ in the real experiment. Furthermore, except with negligible probability, $\hat{\mathcal{SE}}$ outputs witnesses for the statements proven by $\mathcal{A}$ in the accepted right sessions that use different identities from the left sessions.*

Before proving this claim, we finish the proof of Lemma 1. Given Claim 1, we can prove Lemma 1 by showing that the output of $\mathcal{SE}$ is statistically indistinguishable from that of $\hat{\mathcal{SE}}$. This indistinguishability can be shown by observing the following.

- In $\mathcal{SE}$, the emulation of $\hat{\mathcal{SE}}$ is perfect if in every left session that reaches Stage III, the value extracted by the concurrent extractability of CECom is equal to the value that would be extracted by brute force.
- In every such left session, the value extracted by the concurrent extractability of CECom is indeed equal to the value that would be extracted by brute force. This is because the CECom commitment in Stage I-2 is valid in every such left session except with negligible probability, which in turn is because of the soundness of WIPOK and the hiding property of $\mathsf{CCACom}^{1:1}$.

We note that there is a subtlety since the concurrent extraction of CECom itself is rewound in $\mathcal{SE}$ when the witnesses are extracted from the right sessions. The formal proof is given in the full version. □

Analysis of super-poly-time simulator-extractor $\hat{\mathcal{SE}}$.

It remains to prove Claim 1, which states that (i) super-polynomial-time simulator-extractor $\hat{\mathcal{SE}}$ statistically simulates the real view of $\mathcal{A}$ and (ii) $\hat{\mathcal{SE}}$ also extracts a valid witness from every accepted right session in the simulated view.

Proof (of Claim 1). First, we show that $\hat{\mathcal{SE}}$ statistically simulates the real view of $\mathcal{A}$. Since $\hat{\mathcal{SE}}$ simulates the view of $\mathcal{A}$ by executing $\hat{\mathcal{S}}$, it suffices to show that the output of $\hat{\mathcal{S}}$ is statistically indistinguishable from the real view of $\mathcal{A}$. In $\hat{\mathcal{S}}$, each left session is simulated by extracting (r_V, d) from the CECom commitment in Stage I-2 and giving a sWIAOK proof in Stage III with witness (r_V, d). Hence, the indistinguishability follows from the the statistical witness indistinguishability of sWIAOK and the following claim.

Claim 2. *In $\hat{\mathcal{S}}$, the following holds except with negligible probability: In every left session that reaches Stage III, the* CECom *commitment in Stage I-2 of this session is valid and its committed value is a valid decommitment of the* $\mathsf{Com}_{\mathsf{SB}}$ *commitment of Stage I-1.*

We do not prove Claim 2, since it is implied by the claim that we prove later (Claim 5).

Next, we show that $\hat{\mathcal{SE}}$ extracts a valid witness from every accepted right session except with negligible probability. Since $\hat{\mathcal{SE}}$ outputs $\mathsf{fail}_{\mathsf{WI}}$ when it fails to extract a witness in an accepted right session, it suffices to show that $\hat{\mathcal{SE}}$ outputs $\mathsf{fail}_{\mathsf{WI}}$ only with negligible probability. Assume for contradiction that there exists $\widetilde{i}^* \in [m]$ such that $\hat{\mathcal{SE}}$ outputs $\mathsf{fail}_{\mathsf{WI}}$ during the witness extraction of the $\widetilde{i}^*$-th right session with non-negligible probability. Then, let us consider the following hybrid simulator-extractor $\hat{\mathcal{SE}}_{\widetilde{i}^*}$.

- $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ is the same as $\hat{\mathcal{SE}}$ except that $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ tries to extract a witness only from the $\widetilde{i}^*$-th right session (and therefore rewinds the WI-main thread only from the challenge message of sWIAOK of the $\widetilde{i}^*$-th right session).

Clearly, $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ outputs $\mathsf{fail}_{\mathsf{WI}}$ with non-negligible probability. Then, we reach a contradiction roughly as follows.

Step 1. First, we show that in $\hat{\mathcal{SE}}_{\widetilde{i}^*}$, the probability that $\widetilde{r}_V$ is extracted as a witness during the witness extraction of the $\widetilde{i}^*$-th right session is non-negligible, where $\widetilde{r}_V$ is the value chosen by the verifier in Stage I-1 of the $\widetilde{i}^*$-th right session.

Step 2. Next, we define a sequence of hybrid simulator-extractors. The first hybrid is the same as $\hat{\mathcal{SE}}_{\widetilde{i}^*}$, and we gradually modify the $\widetilde{i}^*$-th right session so that it is independent of $\widetilde{r}_V$ in the last hybrid.

Step 3. Finally, we show that even in the last hybrid, the probability that $\widetilde{r}_V$ is extracted during the witness extraction of the $\widetilde{i}^*$-th right session is non-negligible. Since the $\widetilde{i}^*$-th right session is independent of $\widetilde{r}_V$ in the last hybrid, we reach a contradiction.

Details are given below.

Step 1. Prove that $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ extracts $\widetilde{r}_V$. We first prove the following claim.

Claim 3. *Let $\widetilde{r}_V$ be the value chosen by the verifier in Stage I-1 of the $\widetilde{i}^*$-th right session. If $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ outputs* $\mathsf{fail}_{\mathsf{WI}}$ *with non-negligible probability, then in $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ the probability that $\widetilde{r}_V$ is extracted during the witness extraction of the $\widetilde{i}^*$-th right session is non-negligible.*

Proof. Assume for contradiction that $\widetilde{r}_V$ is extracted during the witness extraction of the $\widetilde{i}^*$-th right session with at most negligible probability. Then, since we assume that $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ outputs $\mathsf{fail}_{\mathsf{WI}}$ with non-negligible probability, the following occurs in $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ with non-negligible probability:

- $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ obtains two accepting transcript of the $\widetilde{i}^*$-th right session (and therefore that of sWIAOK) such that the commit-messages of sWIAOK are the same,[7] but
- from these two transcript, $\hat{\mathcal{SE}}_{\widetilde{i}^*}$ fails to extract any witness from sWIAOK (either a witness for $\widetilde{x}_{\widetilde{i}^*} \in L$ or a valid decommitment of the Stage I-1 commitment).

[7] Recall that WIPOK consists of three stages: commit, challenge, and response.

We first show that when the above occurs, the two accepting sWIAOK transcripts are *admissible* except with negligible probability, where a pair of accepted transcripts of sWIAOK are admissible if their commit-messages are the same but their challenge-messages are different. Toward this end, it suffices to show that $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ chooses the same challenge-message of sWIAOK on two WI-auxiliary threads with at most negligible probability. This can be shown as follows.

- From a standard argument, we can show that the expected number of rewinding of the WI-main thread is 1 in $\hat{\mathcal{SE}}_{\widetilde{i^*}}$.[8] Thus, the probability that $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ rewinds the WI-main thread more than $2^{n/2}$ times is at most $2^{-n/2}$. Furthermore, under the condition that $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ rewinds the WI-main thread at most $2^{n/2}$ times, the probability that $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ chooses the same challenge-message on two WI-auxiliary threads is at most $2^{n/2} \cdot 2^{-n} = 2^{-n/2}$. Thus, the probability that $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ chooses the same challenge-message in two WI-auxiliary thread is at most $2^{-n/2} + 2^{-n/2} = \mathsf{negl}(n)$.

Thus, with non-negligible probability $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ obtains two admissible transcripts of sWIAOK from which no witness can be computed.

We then reach a contradiction as follows. Since sWIAOK is a parallel version of Blum's Hamiltonian-cycle protocol, if no witness is extracted from two admissible transcripts of sWIAOK, a $\mathsf{Com}_{\mathsf{SH}}$ commitment in the commit-messages is decommitted to two different values in the transcripts. Thus, we derive a contradiction by breaking the binding property of $\mathsf{Com}_{\mathsf{SH}}$ using $\hat{\mathcal{SE}}_{\widetilde{i^*}}$. A problem is that since $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ runs in super-polynomial time, the *computational* hiding property of $\mathsf{Com}_{\mathsf{SH}}$ may not hold in $\hat{\mathcal{SE}}_{\widetilde{i^*}}$. To overcome this problem, we consider hybrid simulator-extractor $\mathcal{SE}_{\widetilde{i^*}}$ that emulates the execution of $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ in polynomial time. Specifically, $\mathcal{SE}_{\widetilde{i^*}}$ emulates $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ in the same way as $\mathcal{SE}$ emulates $\hat{\mathcal{SE}}$ (i.e., by using the concurrent extractability of CECom instead of the brute-force extraction) except for the following.

- During the emulation of the WI-main thread, the value (r_V, d) is extracted in Stage I-2 of each left session by using the *robust* concurrent extractability of CECom so that the commit-message of sWIAOK of the $\widetilde{i^*}$-th right session is not rewound.

As in the proof of Lemma 1, we can show that $\mathcal{SE}_{\widetilde{i^*}}$ statistically emulates the execution of $\hat{\mathcal{SE}}_{\widetilde{i^*}}$. Thus, with non-negligible probability, $\mathcal{SE}_{\widetilde{i^*}}$ obtains two valid decommitments of a $\mathsf{Com}_{\mathsf{SH}}$ commitment (in the commit-messages of sWIAOK of the $\widetilde{i^*}$-th right session) such that decommitted values are different. Then, since $\mathcal{SE}_{\widetilde{i^*}}$ runs in polynomial time and since the commit-messages of sWIAOK (and therefore the $\mathsf{Com}_{\mathsf{SH}}$ commitment) of the $\widetilde{i^*}$-th right session is not rewound in $\mathcal{SE}_{\widetilde{i^*}}$,[9] we can break the binding property of $\mathsf{Com}_{\mathsf{SH}}$. Thus, we reach a contradiction. □

[8] For any prefix ρ of the transcript up until the challenge message of sWIAOK of the i-th right session, let p_ρ be the probability that the i-th right session is accepted when the prefix of the transcript is ρ. Then, we have $\mathrm{E}\left[T_i \mid \mathsf{prefix}_\rho\right] = p_\rho \cdot 1/p_\rho = 1$, where T_i is the random variable representing the number of rewinding of the WI-main thread and prefix_ρ is the event that the prefix of the transcript is ρ. Thus, we have $\mathrm{E}\left[T_i\right] = \sum_\rho \mathrm{E}\left[T_i \mid \mathsf{prefix}_\rho\right] \Pr\left[\mathsf{prefix}_\rho\right] = 1$.

[9] Note that the commit-messages of sWIAOK of the $\widetilde{i^*}$-th right session appear only on the WI-main thread.

Step 2. Introduce hybrid simulator-extractor. Next, we introduce hybrid simulator-extractors. To clarify the exposition, we first define a sequence of hybrid simulators by gradually modifying $\hat{\mathcal{S}}$ and then define the hybrid simulator-extractors by using them. Below, when we refer to a particular stage of sCNMZK, we always means the corresponding stage of sCNMZK in the $\widetilde{i^*}$-th right session.

Hybrid simulator $h\text{-}\hat{\mathcal{S}}_0$ is identical with $\hat{\mathcal{S}}$.

Hybrid simulator $h\text{-}\hat{\mathcal{S}}_1$ is the same as $h\text{-}\hat{\mathcal{S}}_0$ except that $\widetilde{r_P}$ is extracted by brute force in Stage II-1 and the committed value of the CECom commitment in Stage II-2 is switched from 0^n to $\widetilde{r_P}$.

Hybrid simulator $h\text{-}\hat{\mathcal{S}}_2$ is the same as $h\text{-}\hat{\mathcal{S}}_1$ except that in Stage II-4, the WIPOK proof is computed by using a witness for the fact that the committed value of the CECom commitment of Stage II-2 is $\widetilde{r_P}$.

Hybrid simulator $h\text{-}\hat{\mathcal{S}}_3$ is the same as $h\text{-}\hat{\mathcal{S}}_2$ except that in Stage I-2, the committed value of the CECom commitment is switched from $(\widetilde{r_V}, \widetilde{d})$ to $(0^{|\widetilde{r_V}|}, 0^{|\widetilde{d}|})$.

Hybrid simulator $h\text{-}\hat{\mathcal{S}}_4$ is the same as $h\text{-}\hat{\mathcal{S}}_3$ except that in Stage I-1, the committed value of the $\mathsf{Com}_{\mathsf{SB}}$ commitment is switched from $\widetilde{r_V}$ to 0^n.

Then, for each $k \in \{0, \ldots, 4\}$, hybrid simulator-extractor $h\text{-}\hat{\mathcal{SE}}_k$ is defined as follows.

Hybrid simulator-extractor $h\text{-}\hat{\mathcal{SE}}_k$ is the same as $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ except that the execution of $\hat{\mathcal{S}}$ is replaced with that of $h\text{-}\hat{\mathcal{S}}_k$. The output of $h\text{-}\hat{\mathcal{SE}}_k$ is the value extracted during the witness extraction of the $\widetilde{i^*}$-th right session.

Note that the value $\widetilde{r_V}$ is not used anywhere in $h\text{-}\hat{\mathcal{SE}}_4$.

Step 3. Prove that $\widetilde{r_V}$ is extracted in every hybrid. Finally, we show that $\widetilde{r_V}$ is extracted with non-negligible probability in each hybrid. First, we consider $h\text{-}\hat{\mathcal{SE}}_1$.

Claim 4. *Let $\widetilde{r_V}$ be the value chosen by the verifier in Stage I-1 of the $\widetilde{i^*}$-th right session. If $\hat{\mathcal{SE}}_{\widetilde{i^*}}$ outputs $\mathsf{fail}_{\mathsf{WI}}$ with non-negligible probability, then in $h\text{-}\hat{\mathcal{SE}}_1$ the probability that $\widetilde{r_V}$ is extracted during the witness extraction of the $\widetilde{i^*}$-th right session is non-negligible.*

Proof. In this proof, we use intermediate hybrid simulator-extractors in which the CECom commitment in Stage II-2 of the $\widetilde{i^*}$-th right session is gradually modified. Again, we first introduce hybrid simulators. Recall that a CECom commitment consists of $\ell = \omega(R_{\mathsf{SH}}(n) \log n)$ ExtCom commitments. Then, the intermediate hybrid simulators $h\text{-}\hat{\mathcal{S}}_{0:0}$, $\ldots$, $h\text{-}\hat{\mathcal{S}}_{0:\ell}$ are defined as follows.

Hybrid simulator $h\text{-}\hat{\mathcal{S}}_{0:0}$ is the same as $h\text{-}\hat{\mathcal{S}}_0$ except that $\widetilde{r_P}$ is extracted by brute force in Stage II-1 of the $\widetilde{i^*}$-th right session.

Hybrid simulator $h\text{-}\hat{\mathcal{S}}_{0:k}$ ($k \in [\ell]$) is the same as $h\text{-}\hat{\mathcal{S}}_{0:k-1}$ except that the committed value of the k-th ExtCom commitment in the CECom commitment of Stage II-2 is switched from 0^n to $\widetilde{r_P}$ in the $\widetilde{i^*}$-th right session.

Then, for each $k \in \{0, \ldots, \ell\}$, hybrid simulator-extractor $h\text{-}\hat{\mathcal{SE}}_{0:k}$ is defined as follows.

Hybrid simulator-extractor $h\text{-}\hat{\mathcal{SE}}_{0:k}$ is the same as $h\text{-}\hat{\mathcal{SE}}_0$ except that the execution of $h\text{-}\hat{\mathcal{S}}_0$ is replaced with that of $h\text{-}\hat{\mathcal{S}}_{0:k}$.

Note that $h\text{-}\hat{\mathcal{SE}}_{0:\ell}$ is identical with $h\text{-}\hat{\mathcal{SE}}_1$.

Below, we show that for every $k \in [\ell]$, the output of $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$ and that of $h\text{-}\hat{\mathcal{SE}}_{0:k}$ are indistinguishable. (Recall that the outputs of $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$ and $h\text{-}\hat{\mathcal{SE}}_{0:k}$ are the value extracted in the $\widetilde{i^*}$-th right session.) Since the probability that $\widetilde{r}_V$ is extracted in $h\text{-}\hat{\mathcal{SE}}_{0:0}$ is non-negligible from Claim 3, this suffices to prove Claim 4.

Roughly speaking, we show this indistinguishability as follows. Since $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$ and $h\text{-}\hat{\mathcal{SE}}_{0:k}$ differ only in the committed values of a ExtCom commitment, we use the hiding property of the ExtCom commitment to show the indistinguishability. A problem is that we cannot use it directly since $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$ and $h\text{-}\hat{\mathcal{SE}}_{0:k}$ run in super-polynomial time. To overcome this problem, we observe that the only super-polynomial computations in $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$ and $h\text{-}\hat{\mathcal{SE}}_{0:k}$ are the brute-force extraction of $\mathsf{CCACom}^{1:1}$ in the $\widetilde{i^*}$-th right session and those of CECom in the left sessions. Based on this observation, we first show that the execution of $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$ and $h\text{-}\hat{\mathcal{SE}}_{0:k}$ can be emulated in polynomial-time by using the one-session committed-value oracle $\mathcal{O}$ of $\mathsf{CCACom}^{1:1}$ and the concurrent extractability of CECom. We then combine the 4-robustness of $\mathsf{CCACom}^{1:1}$ with the hiding property of ExtCom (which has only four rounds) to argue that the output of $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$ and that of $h\text{-}\hat{\mathcal{SE}}_{0:k}$ are indistinguishable. To formally implement this idea, we need to make sure that the ExtCom commitment and the $\mathsf{CCACom}^{1:1}$ commitment are not rewound during the concurrent extraction of CECom. Details are given below.

First, we introduce hybrid simulator-extractors $h\text{-}\mathcal{SE}^{\mathcal{O}}_{0:k-1}$ and $h\text{-}\mathcal{SE}^{\mathcal{O}}_{0:k}$, where $\mathcal{O}$ is the one-session committed-value oracle of $\mathsf{CCACom}^{1:1}$. Hybrid $h\text{-}\mathcal{SE}^{\mathcal{O}}_{0:k}$ (resp., $h\text{-}\mathcal{SE}^{\mathcal{O}}_{0:k-1}$) emulates $h\text{-}\hat{\mathcal{SE}}_{0:k}$ (resp., $h\text{-}\hat{\mathcal{SE}}_{0:k-1}$) in the same way as $\mathcal{SE}$ emulates $\hat{\mathcal{SE}}$ except for the following.

- During the emulation of the WI-main thread, the value (r_V, d) is extracted in Stage I-2 of each left session by using the robust concurrent extractability so that the $\mathsf{CCACom}^{1:1}$ commitment of Stage II-1 and the k-th ExtCom commitment of the CECom commitment of Stage II-2 are not rewound in the $\widetilde{i^*}$-th right session. In addition, in the $\widetilde{i^*}$-th right session, the committed value of $\mathsf{CCACom}^{1:1}$ is extracted by forwarding the commitment to $\mathcal{O}$. Note that the $\mathsf{CCACom}^{1:1}$ commitment in the $\widetilde{i^*}$-th right session is not rewound and therefore it can be forwarded to $\mathcal{O}$.

Next, we show that for each $h \in \{k-1, k\}$, the output of $h\text{-}\hat{\mathcal{SE}}_{0:h}$ and that of $h\text{-}\mathcal{SE}^{\mathcal{O}}_{0:h}$ are indistinguishable. This can be proven in a similar way to Lemma 1. In particular, we can use the same argument if we use the following claim instead of Claim 2.

Claim 5. *In $h\text{-}\hat{\mathcal{S}}_{0:h}$ for each $h \in \{k-1, k\}$, the following holds except with negligible probability: In every left session that reaches Stage III, the CECom commitment in Stage I-2 of this session is valid and its committed value is a valid decommitment of the $\mathsf{Com}_{\mathsf{SB}}$ commitment of Stage I-1.*

Note that since $h\text{-}\hat{\mathcal{S}}_{0:0}$ is identical to $\hat{\mathcal{S}}$, Claim 5 implies Claim 2.

Proof (of Claim 5). Let us say that a left session is *bad* if it reaches Stage III and either the CECom commitment in Stage I-2 is invalid or its committed value is not a valid decommitment of the $\mathsf{Com}_{\mathsf{SB}}$ commitment in Stage I-1; a left session is *good* if it is not bad. What we want to prove is that every left session is good except with negligible probability.

Roughly speaking, the proof proceeds as follows. From the soundness of WIPOK, if a left session is bad, then in Stage II-2 of this left session, the committed value of the CECom commitment is r_P, which is the committed value of the CCACom$^{1:1}$ commitment of Stage II-1; thus, before r_P is decommitted to in Stage II-3, we can obtain r_P by extracting the committed value from CECom in Stage II-2. This itself does not contradict to the hiding property of CCACom$^{1:1}$ since h-$\hat{\mathcal{S}}_{0:h}$ runs in super-polynomial time in the brute-force extraction of CECom and CCACom$^{1:1}$. Thus, we again replace the brute-force extraction with the concurrent extraction of CECom and an oracle access to the one-session committed-value oracle $\mathcal{O}$ of CCACom$^{1:1}$, and use the one-one CCA-security of CCACom$^{1:1}$ instead of its hiding property. Here, since we want to use the one-one CCA-security of CCACom$^{1:1}$, we perform the concurrent extraction of CECom so that the CCACom$^{1:1}$ commitment in a left session and the CCACom$^{1:1}$ in the $\widetilde{i^*}$-th right session are not rewound. Details are given below.

Assume for contradiction that there exists $h \in \{k-1, k\}$ such that in h-$\hat{\mathcal{S}}_{0:h}$, a left session is bad with non-negligible probability. (Here, the indices of the left sessions are determined by the order in which Stage III begins; the reason why we define the indices in this way will become clear later.) Then, there exists $i^* \in [m]$ such that in h-$\hat{\mathcal{S}}_{0:h}$, the first $(i^* - 1)$ left sessions are good except with negligible probability but the i^*-th left session is bad with non-negligible probability. Note that from the soundness of WIPOK, when the i^*-th left session is bad, the committed value of the CECom commitment in Stage II-2 is r_P in the i^*-th left session except with negligible probability, where r_P is the value committed to in Stage II-1 of the i^*-th left session. In the following, we use BAD to denote the event that the i^*-th left session is bad, and use CHEAT to denote the event that the committed value of the CECom commitment in Stage II-2 is r_P in the i^*-th left session. Then, let us consider the following hybrids.

Hybrid simulator h-$\hat{\mathcal{S}}_{0:h:0}$ is the same as h-$\hat{\mathcal{S}}_{0:h}$. From our assumption, BAD occurs in h-$\hat{\mathcal{S}}_{0:h:0}$ with non-negligible probability. Thus, from the above argument, CHEAT occurs in h-$\hat{\mathcal{S}}_{0:h:0}$ with non-negligible probability.

Hybrid simulator h-$\hat{\mathcal{S}}_{0:h:1}$ is the same as h-$\hat{\mathcal{S}}_{0:h:0}$ except that h-$\hat{\mathcal{S}}_{0:h:1}$ terminates just before Stage III of the i^*-th left session begins. Clearly, BAD and CHEAT also occur in h-$\hat{\mathcal{S}}_{0:h:1}$ with non-negligible probability.

Hybrid simulator h-$\mathcal{S}^{\mathcal{O}}_{0:h:1}$ emulates h-$\hat{\mathcal{S}}_{0:h:1}$ in polynomial time as follows.

- At the beginning, a random left session s is chosen. (Here, we guess that session s is the i^*-th left session.)
- In every left session, in Stage I-2, the committed value (r_V, d) is extracted by the robust concurrent extractor of CECom in such a way that the CCACom$^{1:1}$ commitment of left session s and the CCACom$^{1:1}$ commitment of the $\widetilde{i^*}$-th right session are not rewound. In addition, in the $\widetilde{i^*}$-th right session, the committed value of CCACom$^{1:1}$ is extracted by forwarding the commitment to $\mathcal{O}$.
- In left session s, the committed value is also extracted in Stage II-2 by the robust concurrent extractor of CECom without rewinding the CCACom$^{1:1}$ commitment of the $\widetilde{i^*}$-th right session.

Note that when Stage III of a left session is executed, the CECom commitment in Stage I-2 of that session is valid except with negligible probability (since that session is one of the first $(i^* - 1)$ left sessions and therefore it is good except with negligible probability). Thus, the values extracted from the concurrent extractor are equal to the values that would be extracted by brute force except with negligible probability; therefore, $h\text{-}\mathcal{S}^{\mathcal{O}}_{0:h:1}$ statistically emulates $h\text{-}\hat{\mathcal{S}}_{0:h:1}$, and BAD and CHEAT occur in $h\text{-}\mathcal{S}^{\mathcal{O}}_{0:h:1}$ with non-negligible probability.

Note that session s is the i^*-th left session with non-negligible probability. Then, since CHEAT occurs in $h\text{-}\mathcal{S}^{\mathcal{O}}_{0:h:1}$ with non-negligible probability, r_P is extracted from the CECom commitment in Stage II-2 of session s with non-negligible probability, where r_P is the value committed to in Stage II-1 of session s. Then, since the $\mathsf{CCACom}^{1:1}$ commitment of session s is not rewound in $h\text{-}\mathcal{S}^{\mathcal{O}}_{0:h:1}$, we can break the one-one CCA security of $\mathsf{CCACom}^{1:1}$. Thus, we reach a contradiction. □

Thus, for each $h \in \{k-1, k\}$, the outputs of $h\text{-}\hat{\mathcal{S}}\mathcal{E}_{0:h}$ and $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:h}$ are indistinguishable.

To show that the outputs of $h\text{-}\hat{\mathcal{S}}\mathcal{E}_{0:k-1}$ and $h\text{-}\hat{\mathcal{S}}\mathcal{E}_{0:k}$ are indistinguishable, it remains to prove that the outputs of $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k-1}$ and $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k}$ are indistinguishable. This can be shown as follows. Observe that $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k-1}$ and $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k}$ differ only in the k-th ExtCom commitment of the CECom commitment of the $\widetilde{i}^*$-th right session, and this ExtCom commitment is not rewound in $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k-1}$ and $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k}$. In addition, $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k-1}$ and $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k}$ run in polynomial time given oracle access to the one-session committed-value oracle $\mathcal{O}$ of $\mathsf{CCACom}^{1:1}$. Thus, from the hiding property of ExtCom and the 4-robustness of $\mathsf{CCACom}^{1:1}$, the output of $\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k-1}$ and that of $h\text{-}\mathcal{S}\mathcal{E}^{\mathcal{O}}_{0:k}$ are indistinguishable.

Thus, we conclude that the probability that $\widetilde{r}_V$ is extracted in $h\text{-}\hat{\mathcal{S}}\mathcal{E}_1$ is non-negligible. This concludes the proof of Claim 4. □

By using essentially the same argument as in the proof of Claim 4, we can show that $\widetilde{r}_V$ is extracted with non-negligible probability also in $h\text{-}\hat{\mathcal{S}}\mathcal{E}_2$, $h\text{-}\hat{\mathcal{S}}\mathcal{E}_3$, and $h\text{-}\hat{\mathcal{S}}\mathcal{E}_4$.

Concluding the proof of Claim 1. In $h\text{-}\hat{\mathcal{S}}\mathcal{E}_4$, the $\widetilde{i}^*$-th right session is independent of $\widetilde{r}_V$, and therefore the probability that $\widetilde{r}_V$ is extracted is negligible. However, we show above that this probability is non-negligible. Thus, we reach a contradiction. □

This concludes the proof of Theorem 2. □

References

1. Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *FOCS*, pages 345–354, 2006.
2. Manuel Blum. How to prove a theorem so no one else can claim it. In *International Congress of Mathematicians*, pages 1444–1451, 1987.
3. Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires (almost) logarithmically many rounds. *SIAM J. Comput.*, 32(1):1–47, 2002.
4. Ran Canetti, Huijia Lin, and Rafael Pass. Adaptive hardness and composable security in the plain model from standard assumptions. In *FOCS*, pages 541–550, 2010.
5. Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical secrecy and multibit commitments. *IEEE Transactions on Information Theory*, 44(3):1143–1151, 1998.

6. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.
7. Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6):851–898, 2004.
8. Vipul Goyal, Huijia Lin, Omkant Pandey, Rafael Pass, and Amit Sahai. Round-efficient concurrently composable secure computation via a robust extraction lemma. In *TCC*, 2015.
9. Vipul Goyal, Ryan Moriarty, Rafail Ostrovsky, and Amit Sahai. Concurrent statistical zero-knowledge arguments for NP from one way functions. In *ASIACRYPT*, pages 444–459, 2007.
10. Iftach Haitner, Minh-Huyen Nguyen, Shien Jin Ong, Omer Reingold, and Salil P. Vadhan. Statistically hiding commitments and statistical zero-knowledge arguments from any one-way function. *SIAM J. Comput.*, 39(3):1153–1218, 2009.
11. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.
12. Susumu Kiyoshima. Round-efficient black-box construction of composable multi-party computation. In *CRYPTO*, pages 351–368, 2014.
13. Susumu Kiyoshima, Yoshifumi Manabe, and Tatsuaki Okamoto. Constant-round black-box construction of composable multi-party computation protocol. In *TCC*, pages 343–367, 2014.
14. Huijia Lin and Rafael Pass. Concurrent non-malleable zero knowledge with adaptive inputs. In *TCC*, pages 274–292, 2011.
15. Huijia Lin and Rafael Pass. Black-box constructions of composable protocols without set-up. In *CRYPTO*, pages 461–478, 2012.
16. Huijia Lin, Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable zero knowledge proofs. In *CRYPTO*, pages 429–446, 2010.
17. Daniele Micciancio, Shien Jin Ong, Amit Sahai, and Salil P. Vadhan. Concurrent zero knowledge without complexity assumptions. In *TCC*, pages 1–20, 2006.
18. Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.
19. Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43, 1989.
20. Claudio Orlandi, Rafail Ostrovsky, Vanishree Rao, Amit Sahai, and Ivan Visconti. Statistical concurrent non-malleable zero knowledge. In *TCC*, pages 167–191, 2014.
21. Rafail Ostrovsky, Omkant Pandey, and Ivan Visconti. Efficiency preserving transformations for concurrent non-malleable zero knowledge. In *TCC*, pages 535–552, 2010.
22. Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *STOC*, pages 533–542, 2005.
23. Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent zero knowledge, revisited. *J. Cryptology*, pages 1–22, 2012.
24. Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *TCC*, pages 403–418, 2009.
25. Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *FOCS*, pages 366–375, 2002.
26. Muthuramakrishnan Venkitasubramaniam. On adaptively secure protocols. In *SCN*, pages 455–475, 2014.