

Non-Malleable Statistically Hiding Commitment from Any One-Way Function

Zongyang Zhang, Zhenfu Cao, Ning Ding, and Rong Ma

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, P.R.China
{zongyangzhang,zfcao,dingning,marong}@sjtu.edu.cn

Abstract. We give a construction of non-malleable statistically hiding commitments based on the existence of one-way functions. Our construction employs statistically hiding commitment schemes recently proposed by Haitner and Reingold [1], and special-sound WI proofs. Our proof of security relies on the message scheduling technique introduced by Dolev, Dwork and Naor [2], and requires only the use of black-box techniques.

1 Introduction

A commitment scheme is an interactive protocol between two parties, the committer, who holds a value, and the receiver. It usually consists of two phases: the commit phase and the reveal phase. During the commit phase, the committer puts a value in a “locked box” and sends it to the receiver. In the reveal phase, the committer sends the “key” to the receiver, then the receiver opens the box and retrieves the value. Two basic properties of a commitment scheme are the hiding property (the receiver cannot learn the committed value before the reveal phase) and the binding property (the committer is bounded to one value after the commit phase). There are two fundamental types of commitment schemes, statistical hiding and statistical binding. In this work, we focus mainly on statistically hiding commitment schemes, where the hiding property holds against unbounded receivers while the binding property is required to hold only against polynomially bounded senders.

The concept of non-malleability was first introduced by Dolev et al. [2]. The basic properties of commitment schemes cannot prevent malleable attacks mounted by a man-in-the-middle adversary who has full control of the communication channel between the committer and the receiver. Loosely speaking, a commitment scheme is non-malleable if one cannot transform the commitment of a value into a commitment of a related value. This kind of non-malleability is called *non-malleability with respect to commitment* [3]. The notion of non-malleability used by Di Crescenzo et

al. [4] is called *non-malleability with respect to opening*, i.e., the adversary cannot construct a commitment from a given one, such that after having seen the opening of the original commitment, the adversary is able to correctly open his commitment with a related value. In the rest of this paper, when we say non-malleability, we actually mean non-malleability with respect to opening.

1.1 Related Work

Statistically hiding commitment schemes were first shown to exist based on number-theoretic assumptions [5, 6], or more generally, based on any collection of claw-free permutations [7] with an efficiently-recognizable index set [8]. Subsequent work on constructing statistically hiding commitment schemes are based on collision-resistant hash functions [9], or based on any one-way permutation [10], or based on regular one-way functions [11]. Nguyen et al. [12] and Haitner and Reingold [1] made fundamental progress by constructing statistically hiding commitment schemes based on the minimal cryptographic assumption that one-way functions exist.

Based on number-theoretic assumptions, non-malleable statistically hiding commitment schemes were designed in [13, 3] assuming the existence of a common reference string that is shared by the two players before the protocol execution. Thus, their schemes do not work in the plain model (i.e., without setup assumptions). More recently, Pass and Rosen [14] constructed a non-malleable commitment scheme that was statistically hiding based on a family of collision-resistant hash functions. Their scheme is round-efficient and needs only constant-round communication. However, the security proof relies on non-black-box techniques and is not efficient.

As one of the central goals of cryptography is to reduce complexity assumptions for various cryptographic primitives and construct them under more standard assumptions, there remain open questions as to *whether or not non-malleable statistically hiding commitment can be based solely on the existence of one-way functions, and be shown secure relying only on black-box techniques*.

1.2 Our Result

In this paper, we give affirmative answers to both of the questions posed above. We show that the existence of one-way function is a suf-

ficient condition for the existence of non-malleable statistically hiding commitment.

Theorem 1. *If one-way functions exist, then there exists a non-malleable statistically hiding commitment scheme.*

Our commitment scheme uses the commitment scheme [1] to commit to the desired value, but modify the opening process by adding a “trap-door” that can be extracted and used by the simulator to cheat in the reveal phase, and would not be known to the committer in a real execution. Although the extraction requires rewinding, we rely on the message scheduling technique of Lin et al. [15], which is a slight modification of the message scheduling technique introduced by Dolev et al. [2], to show this will suffice to prove the non-malleability. Our proof requires only standard black-box techniques. As a tradeoff, however, our protocol needs polynomial rounds of interaction.

The preliminaries and definitions are illustrated in section 2. Our non-malleable statistically hiding commitment scheme is shown in section 3.

2 Preliminaries and Definitions

For any NP languages L , note that there is a natural witness relation R_L containing pairs (x, w) where w is the witness for the membership of x in L . A function $\mu(\cdot)$, where $\mu : \mathbb{N} \rightarrow [0, 1]$ is called *negligible* if for every positive polynomial $p(\cdot)$, for all sufficiently large $n \in \mathbb{N}$, $\mu(n) < \frac{1}{p(n)}$. A *probability ensemble* is a sequence $X = \{X_i\}_{i \in I}$ of random variables, where I is a countable index set and X_i is a random variable ranging over $\{0, 1\}^{p(|i|)}$ for some polynomial $p(\cdot)$. Two probability ensembles $X = \{X_i\}_{i \in I}$ and $Y = \{Y_i\}_{i \in I}$ are *computationally indistinguishable*, if no probabilistic polynomial-time (PPT) algorithm distinguishes between them with more than negligible probability. For page limited, we assume the readers are familiar with interactive proofs.

Special-sound proofs. A 3-round public-coin interactive proof for the language $L \in \text{NP}$ with witness relation R_L is **special-sound** with respect to R_L , if for any two accepting transcripts (α, β, γ) and $(\alpha', \beta', \gamma')$ for some statement $x \in L$, such that $\alpha = \alpha'$ and $\beta \neq \beta'$, a witness w such that $(x, w) \in R_L$ can be computed by a polynomial-time deterministic procedure.

2.1 Witness Indistinguishability

The concept of witness indistinguishability was proposed by Feige and Shamir [16]. An interactive proof system is witness indistinguishable (WI) if the verifier cannot tell which of the witnesses is being used by the prover to carry out the proof, even if the verifier knows both witnesses. We focus on NP languages L with a corresponding witness relation R_L . The readers are referred to [16] for formal definition.

Special-sound WI proofs for NP languages can be based on the existence of non-interactive commitment schemes. Assuming only one-way functions, 4-round special-sound WI proofs for NP languages exist.¹ More precisely, there is a 3-round special-sound WI proof for the language of Hamiltonian Graphs [17], assuming one-way permutation families exist. If the commitment scheme used by the protocol [17] is replaced by Naor's commitment scheme [18], then it becomes a 4-round special-sound WI proof while the assumption is reduced to the existence of one-way functions. For simplicity, we use 3-round special-sound WI proofs in our protocol though our proof works also with 4-round special-sound WI proofs.

2.2 Commitment Schemes

In this work, we consider statistically hiding commitment schemes.

Definition 1 (Commitment Scheme). *A pair of PPT interactive machines $\langle C, R \rangle$ is said to be a commitment scheme if the following two properties hold:*

Statistical hiding: *For every unbounded interactive Turing machine R^* , it holds that the ensemble $\{\text{sta}_{\langle C, R \rangle}^{R^*}(v_1, z)\}_{v_1 \in \{0,1\}^n, n \in \mathbb{N}, z \in \{0,1\}^*}$ and the ensemble $\{\text{sta}_{\langle C, R \rangle}^{R^*}(v_2, z)\}_{v_2 \in \{0,1\}^n, n \in \mathbb{N}, z \in \{0,1\}^*}$ have negligible statistical difference,² where $\text{sta}_{\langle C, R \rangle}^{R^*}(v, z)$ denotes the random variable describing the output of R^* after receiving a commitment to v using $\langle C, R \rangle$.*

Computational binding: *A malicious (expected) PPT committer S^* can succeed in opening a given commitment in two different ways only with negligible probability. The reader is referred to [19, 1] for more details.*

¹ A 4-round protocol is special sound if there exists polynomial-time deterministic procedure to extract the witness from any two accepting transcripts $(\tau, \alpha, \beta, \gamma)$ and $(\tau', \alpha, \beta, \gamma)$ such that $\tau = \tau', \alpha = \alpha'$ and $\beta \neq \beta'$.

² The statistical difference between two ensembles $\{X_i\}_{i \in I}$ and $\{Y_i\}_{i \in I}$ is defined by $\frac{1}{2} \cdot \sum_{\alpha} |\Pr[X_i = \alpha] - \Pr[Y_i = \alpha]|$.

2.3 Non-Malleable Commitments

As stated in [14], we formalize the notion of non-malleability by a comparison between a *man-in-the-middle* execution and a *simulated* execution. Just as [2, 15], we consider a tag-based variant of non-malleability.

Let $\langle C, R \rangle$ be a commitment scheme. Let $n \in \mathbb{N}$ be a security parameter. Let $\mathcal{R} \in \{0, 1\}^n \times \{0, 1\}^n$ be a polynomial-time computable valid relation [13] (i.e., for all $v \in \{0, 1\}^n$, $\mathcal{R}(v, \perp) = 0$). In the man-in-the-middle execution, the adversary A is simultaneously participating in a left and right interaction. In the left interaction, the man-in-the-middle adversary A interacts with the committer C to receive a commitment to a value v using tag tag . In the right interaction, A interacts with the receiver R and tries to commit to a related value using tag of its choice $\tilde{\text{tag}}$. After commit phase execution in both interactions, A receives decommitment keys from C and then generates the corresponding decommitment key for \tilde{v} . Prior to the interaction, the value v is given to C as local input. A receives an auxiliary input z , which might contain a priori information about v . If the right commitment or decommitment fails, or $\text{tag} = \tilde{\text{tag}}$, \tilde{v} is set to \perp . Let the boolean random variable $\text{mim}_{\text{open}}^A(\mathcal{R}, v, z)$ denote whether A succeeds. Note $\text{mim}_{\text{open}}^A(\mathcal{R}, v, z) = 1$ if and only if A decommits to a value \tilde{v} such that $\mathcal{R}(v, \tilde{v}) = 1$.

In the simulated execution, a simulator S directly interacts with honest receiver R . As in the man-in-the-middle execution, the value v is chosen prior to the interaction, and S receives some a priori information about v as part of its auxiliary input z . S also receives tag tag . S first executes the commitment scheme with R . Once the commitment phase has been completed, S receives the value v and attempts to decommit to a value \tilde{v} with tag $\tilde{\text{tag}}$. If $\text{tag} = \tilde{\text{tag}}$, \tilde{v} is set to \perp . Let the boolean random variable $\text{sim}_{\text{open}}^S(\mathcal{R}, v, z)$ denote whether S succeeds. Note $\text{sim}_{\text{open}}^S(\mathcal{R}, v, z) = 1$ if and only if S decommits to a value \tilde{v} such that $\mathcal{R}(v, \tilde{v}) = 1$.

Definition 2 (Non-malleable Commitment [14]). *A commitment scheme $\langle C, R \rangle$ is said to be non-malleable with respect to opening if for every PPT man-in-the-middle adversary A , there exists an expected PPT simulator S and a negligible function $\mu : \mathbb{N} \rightarrow [0, 1]$, such that for every polynomial-time computable valid relation $\mathcal{R} \subseteq \{0, 1\}^n \times \{0, 1\}^n$, for all tags of polynomial length, for every $v \in \{0, 1\}^n$ and every $z \in \{0, 1\}^*$, the following holds:*

$$\Pr[\text{mim}_{\text{open}}^A(\mathcal{R}, v, z) = 1] < \Pr[\text{sim}_{\text{open}}^S(\mathcal{R}, v, z) = 1] + \mu(n)$$

A commitment scheme that is non-malleable according to Definition 2 is liberal non-malleable rather than strict non-malleable [2, 3]. Note we follow [14] in that non-malleability is guaranteed only if the commit phase and the reveal phase do not overlap.

3 Construction

We begin by presenting a high-level overview of our protocol. Our protocol is based on the statistically hiding commitment scheme [1] while relying on the messages scheduling technique [15] which is a slight modification of the message scheduling technique of [2]. The commit phase of our protocol is the same as that of the commitment protocol in [1]. The reveal phase, however, comes in two parts. Roughly, the reveal phase employs the two-witness technique by Feige [20] and the well known FLS-technique [21]. First, the receiver proves that it knows one of the preimages of either element s_0 or element s_1 computed by itself in the domain of a one-way function. Then, the committer sends the committed value v and proves it knows how to open the commitment or one of the preimages of either element s_0 or element s_1 . The proofs used by the prover and the verifier are all tag-based WI proofs elaborately scheduled as [15]. For simplicity of exposition, our description relies on the existence of one-way functions with efficiently recognizable range.³ We also assume the one-way function is length-preserving. Since any one-way function can be transformed into length-preserving one-way function [19].

3.1 Tag-Based Witness-Indistinguishable Proof

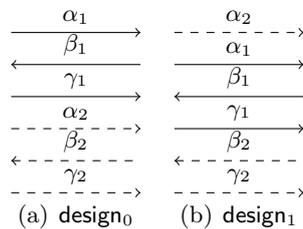


Fig. 1. Two schedules

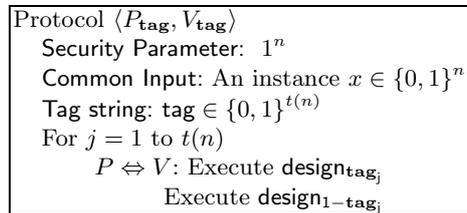


Fig. 2. Tag-based WI proof $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$

³ The protocol can be easily modified to work with arbitrary one-way function by providing a witness hiding proof that an element is in the range of the one-way function.

First, we propose a tag-based WI proof for every NP language L which is used as a basic tool in the final commitment scheme. The length of the tag is polynomial bounded to the length of the security parameter n . Denote the polynomial by $t(\cdot)$. In Fig. 1, both design_0 and design_1 contain two executions of special-sound WI proofs for L but with elaborately designed scheduling. The tag-based WI proof $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ for L is shown in Fig. 2. The protocol is composed of $4t$ -round special-sound WI proofs for language L . More precisely, there are t rounds, where in round j , the schedule $\text{design}_{\text{tag}_j}$ is followed by $\text{design}_{1-\text{tag}_j}$. The properties of $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ are easy to verify. The details are omitted.

One basic technique in proving the security of most zero-knowledge and commitment protocols is standard rewinding. However, the rewinding technique is problematic when extending to concurrent (here one-left one-right) execution environment as an adversary may adaptively schedule its messages that withstand any targeted simulator (i.e., the simulator may run super-polynomial time or is exposed to malleability attack.). Considering the non-malleability property for commitment schemes, the pivot is to design the stand-alone simulator that satisfying Definition 2. Here we also come up with the problem of how to simulate when the adversary adaptively schedules its messages.

The scheduling in Fig. 1 which is identical to [15] is vital in achieving the non-malleability. The main advantage of this scheduling is that for the proof given by a man-in-the-middle adversary, there exists a point at which the adversary cannot answer the challenge from the verifier by simply modifying the proof on the other side (provided the tag of the proof is different from that of the proof on the other side.).

Related to the above scheduling is a notion called *safe-point*, from which it is possible to perform extraction by standard rewinding until we obtain a second proof transcript, without “affecting” the other side interaction. Below is the formal definition of *safe-point*, which is mainly taken from [15] and abridged to our setting.

Definition 3 (Safe-point [15]). *A prefix ρ of a transcript τ is called a safe-point, if there exists an accepting proof $(\alpha_r, \beta_r, \gamma_r)$ in the right interaction, such that*

1. α_r occurs in ρ , but not β_r (and γ_r).
2. For any proof $(\alpha_l, \beta_l, \gamma_l)$ in the left interaction, if only α_l occurs in ρ , then β_l occurs after γ_r .

When protocol $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ is run concurrently, it is guaranteed there is a **safe-point** for right interaction that has a tag different from the left interaction following from the next lemma.

Lemma 1 (Safe-point Lemma [15]⁴). *In any one-one man-in-the-middle execution of $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$, if the right interaction has a different tag from the tag of the left interaction, there exists a **safe-point** for the right interaction.*

3.2 Non-Malleable Statistically Hiding Commitment Scheme

Let $\langle \text{SHC}, \text{SHR} \rangle$ be the statistically hiding commitment scheme [1] from any one-way function ⁵ and let $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ be a tag-based WI proof for NP. The commitment protocol is shown in Fig. 3. The length of the tag is $m(n)$. Our construction in fact compiles any statistically hiding commitment scheme with non-interactive reveal phase into a non-malleable statistically hiding one with interactive reveal phase, assuming the existence of one-way functions.

Theorem 2. *Suppose that $\langle \text{SHC}, \text{SHR} \rangle$ is a statistically hiding commitment scheme with non-interactive reveal phase and $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ is a tag-based WI proof. Then $\langle C, R \rangle$ is a non-malleable statistically hiding commitment scheme.*

Remark 1. The commitment scheme shown in Fig. 3 is tag-based non-malleable. Compared with existing tag-based commitment schemes [2, 15, 22], it seems a bit strange that our construction uses tags only in the reveal phase. In fact, this approach is inspired by the work of [14, 15]. Even tag-based non-malleable commitments can be transformed into content-based non-malleable commitments in a standard way [2], we explicitly present one in Appendix A for reference.

Remark 2. The high level approach of our commitment scheme is to combine [14] with [2, 15]. That is, to commit to v , in the commit phase, a sender commits v using the statistically hiding commitment scheme [1],

⁴ The **safe-point lemma** in [15] applies to any one-many concurrent execution environment, where the adversary participates in one left interaction and polynomial many right interactions. Here we use a simpler version of the **safe-point lemma**, where the adversary participates in one left interaction and one right interaction.

⁵ Note the commitment scheme [1] is only for a single bit. By running their scheme in parallel, we obtain a commitment scheme of any polynomial length. Hence, we also assume that the basic statistically hiding commitment scheme is for a string.

<p>Protocol $\langle C, R \rangle$ Security Parameter: 1^n Tag string: $\text{tag} \in \{0, 1\}^{m(n)}$ String to be committed: $v \in \{0, 1\}^n$ Commit Phase: $C \Leftrightarrow R$: Run the commit phase of commitment scheme $\langle \text{SHC}, \text{SHR} \rangle$, where C runs SHC and R runs SHR. R : Abort if the above commit phase fails. Let com be the transcripts of messages obtained. C records the decommitment key in dec. Reveal Phase: Stage 1: $R \rightarrow C$: Pick uniformly $r_0, r_1 \in \{0, 1\}^n$, compute $s_0 = f(r_0)$ and $s_1 = f(r_1)$ and send s_0, s_1. $R \Leftrightarrow C$: R and C engage in an execution of $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ with tag tag, where R uses r_b as witness ($b \in \{0, 1\}$) and runs P_{tag} to prove to C (running V_{tag}) knowledge of a value r s.t. $s_0 = f(r)$ or $s_1 = f(r)$. The challenge length of the verifier (i.e., C) is $2n$. C : Abort if either s_0 or s_1 is not in the range of f or the proof fails. Stage 2: $C \rightarrow R$: Send v. Stage 3: $C \Leftrightarrow R$: C and R engage in an execution of $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ with tag tag, where C runs P_{tag} to prove to R (running V_{tag}) that there exists a value dec s.t. dec is the valid decommitment key of com corresponding to v or there exists a value r s.t. $s_0 = f(r)$ or $s_1 = f(r)$. The challenge length of the verifier (i.e., R) is $2n$.</p>
--

Fig. 3. Non-malleable statistically hiding commitment scheme $\langle C, R \rangle$

and in the reveal phase, a sender sends v and proves using a “simulation-extractable” argument [2, 15] that the commit phase transcript opens to v . The simulation strategy at a high level is from [14]. For technical reasons, naively using the simulation-extractable arguments from [2, 15] does not work. We need to modify the opening process by adding a “trapdoor” that can be extracted and used by the simulator to cheat in the reveal phase. This is the reason why we add one more phase (i.e., Stage 1). Whereas in [2, 15], the trapdoor is only used in the hybrid experiment for analysis and may therefore hard-wired via a different analysis.

Proof (sketch). We need to prove the scheme satisfies the following three properties: statistical hiding, computational binding and non-malleability with respect to opening. We start by proving the hiding and non-malleability properties and then return to the proof of the binding property.

Statistical hiding. The hiding property follows directly from the hiding property of the commitment scheme $\langle \text{SHC}, \text{SHR} \rangle$. Note that $\langle \text{SHC}, \text{SHR} \rangle$ is statistically hiding, and so $\langle C, R \rangle$ is also statistically hiding.

Non-malleability. We show that for every PPT man-in-the-middle adversary A , there exists a probabilistic expected polynomial-time simulator S and a negligible function μ such that for every polynomial-time computable relation $\mathcal{R} \subseteq \{0, 1\}^n \times \{0, 1\}^n$, for every tag tag of length $m(n)$, for every $v \in \{0, 1\}^n$ and every $z \in \{0, 1\}^*$, it holds that

$$\Pr[\text{mim}_{\text{open}}^A(\mathcal{R}, v, z) = 1] < \Pr[\text{sim}_{\text{open}}^S(\mathcal{R}, v, z) = 1] + \mu(n) \quad (1)$$

Denote by A_{rev} the state of A after the the commit phase, i.e., A_{rev} contains A 's description along with its configuration at that time just before the reveal phase starts.

We proceed to describing the simulator S . S on input z and security parameter 1^n interacts with an honest receiver R and runs the adversary A internally. During the commit phase, on a high level, S internally incorporates A and emulates the commit phase of the left execution for adversary A by honestly committing to 0^n , while externally relaying messages in the right execution between A and R .

Once the commit phase is finished, S receives a value v and has to perform the reveal phase internally with A_{rev} . In **Stage 1**, S plays as an honest sender in the left reveal phase and as an honest receiver in the right reveal phase. Once the simulation of **Stage 1** completes, S applies the **safe-point lemma** to find a **safe-point** and extract a witness w to the statement proved by A_{rev} in the left reveal phase by standard rewinding.⁶ In **Stage 2**, S just sends v to A_{rev} in the left reveal phase. Then the simulation for **Stage 3** begins. S uses a fake witness (i.e. the trapdoor w) to simulate the left interaction for A_{rev} , while emulating the right interaction as an honest receiver. When the simulation for **Stage 3** completes, S again applies the **safe-point lemma** to find a **safe-point** and extract a witness \tilde{w} (i.e., the decommitment keys of A) in the right interaction. Finally, by using \tilde{w} , S can complete the reveal phase of the external execution with R .

More formally, S proceeds as follows on auxiliary input z and tag tag :

1. S internally incorporates $A(z)$.
2. During the commit phase S proceeds as follows:
 - (a) S internally emulates left interaction for A by honestly committing to 0^n .
 - (b) Messages from right execution are forwarded externally to R .
3. Once the commit phase has finished, S receives the value v . Let $\text{com}, \widetilde{\text{com}}$ denote the left and right execution transcripts respectively.

⁶ In **Stage 1**, the committer acts as a prover and the receiver acts as a verifier. The **safe-point** and **safe-point lemma** still work by interchanging right and left.

4. During the reveal phase S internally incorporates A_{rev} and proceeds as follows:

- (a) **Stage 1 Main Execution Phase:** S emulates a one-one man-in-the-middle execution by playing as honest sender with tag tag on the left and as honest receiver on the right. After completing the execution, denote by Δ the transcripts of messages obtained. Denote the right tag by $\tilde{\text{tag}}$. We emphasize here that S can emulate left interaction independent of v in Stage 1.

Stage 1 Rewinding Phase: Next, S attempts to extract the witness used by A_{rev} on the left if $\text{tag} \neq \tilde{\text{tag}}$.

- i. In Δ , find the first point ρ that is a safe-point. Let the associated proof be $(\alpha_\rho, \beta_\rho, \gamma_\rho)$.
- ii. Repeat until a second proof transcript $(\alpha_\rho, \beta'_\rho, \gamma'_\rho)$ is obtained:

Emulate the left interaction as in the Stage 1 Main Execution phase. For the right interaction:

- If A_{rev} expects to get a new proof from the right receiver, S then emulates the proof by generating design_0 himself. Forward one of the two proofs internally.
- If A_{rev} sends a challenge for a proof whose first message occurs in ρ : cancel the execution, rewind to ρ and continue.

- iii. If $\beta_\rho \neq \beta'_\rho$, extract and record the witness w from $(\alpha_\rho, \beta_\rho, \gamma_\rho)$ and $(\alpha_\rho, \beta'_\rho, \gamma'_\rho)$. Otherwise halt and output fail.

Finally, if the above (i.e. step 4a) runs for more than 2^n steps, halt and output fail.

- (b) **Stage 2:** Send v to the adversary A_{rev} .
- (c) **Stage 3 Main Execution Phase:** By using w as witness, S can easily simulate left interaction for A_{rev} . The right interaction is emulated by S adopting honest receiver strategy. After completing the execution, denote by Δ' the transcripts of messages obtained in the execution of Stage 2 and Stage 3 .

Stage 3 Rewinding Phase: S attempts to extract the decommitment key of A_{rev} on the right:

- i. In Δ' , find the first point $\tilde{\rho}$ that is a safe-point. Let the associated proof be $(\tilde{\alpha}_{\tilde{\rho}}, \tilde{\beta}_{\tilde{\rho}}, \tilde{\gamma}_{\tilde{\rho}})$.
- ii. Repeat until a second proof transcript $(\tilde{\alpha}_{\tilde{\rho}}, \tilde{\beta}'_{\tilde{\rho}}, \tilde{\gamma}'_{\tilde{\rho}})$ is obtained:

Emulate the right interaction as in the Stage 3 Main Execution Phase. For the left interaction:

- If A_{rev} expects to get a new proof from the committer, S is free to answer the request by using the witness w , except when A_{rev} sends a challenge for a proof whose first message occurs in $\tilde{\rho}$, S cancels the execution, rewinds to $\tilde{\rho}$ and continues.
 - iii. If $\tilde{\beta}_{\tilde{\rho}} \neq \tilde{\beta}'_{\tilde{\rho}}$, extract a witness \tilde{w} from $(\tilde{\alpha}_{\tilde{\rho}}, \tilde{\beta}_{\tilde{\rho}}, \tilde{\gamma}_{\tilde{\rho}})$ and $(\tilde{\alpha}_{\tilde{\rho}}, \tilde{\beta}'_{\tilde{\rho}}, \tilde{\gamma}'_{\tilde{\rho}})$. Otherwise halt and output fail.
 - iv. If \tilde{w} is a valid decommitment key for $\langle \text{SHC}, \text{SHR} \rangle$, i.e., $(\widetilde{\text{com}}, \tilde{w}, \tilde{v})$ is a legal transcript for $\langle \text{SHC}, \text{SHR} \rangle$, set $\widetilde{\text{rev}} = \tilde{w}$. Otherwise halt and output fail.
- Finally, if the above (step 4b) runs for more than 2^n steps, halt and output fail.
- (d) If the right interaction is accepting and $\text{tag} \neq \widetilde{\text{tag}}$, and $\widetilde{\text{rev}}$ contains a valid decommitment key, run the honest committer strategy on input $\widetilde{\text{com}}$ and decommitment key $\widetilde{\text{rev}}$, value \tilde{v} with tag $\widetilde{\text{tag}}$.

Running time of S . We show that the running time of S is expected PPT. Note the time spent by S in the commit phase is $\text{poly}(n)$. After S extracts the witness \tilde{w} , the time spent by S in step 4d is also $\text{poly}(n)$. Next, we show that the expected time spent by S in the reveal phase (except running time in step 4d) is also $\text{poly}(n)$. For simplicity, we assume that S does not check the fail condition and may run for more than 2^n steps (since this only increases the total running time).

Recall that in the reveal phase, S rewinds A from two safe points. We need to show the time spent in step 4a and step 4c are all expected PPT. We first analyze the time spent in step 4a during the simulation. Then using the same method, we show that the time spent in step 4c is also expected PPT.

Note the time spent by S in the Stage 1 Main Execution Phase is $\text{poly}(n)$. We then show the time spent in Stage 1 Rewinding Phase is expected PPT. The analysis hereafter is similar to that in [15] but is simpler. Let $T(i)$ be the random variable that describes the time spent in rewinding a proof after i messages have been exchanged. We show that $\mathbb{E}[T(i)] \leq \text{poly}(n)$ and then by linearity of expectation, we conclude that the expected time spent by S in the Stage 1 Rewinding Phase is $\sum_i \mathbb{E}[T(i)] \leq \sum_i \text{poly}(n) \leq \text{poly}(n)$.

Next we will bound the time $\mathbb{E}[T(i)]$. Given a partial transcript of messages ρ , let $\Pr[\rho]$ denote the probability that ρ occurs as a prefix of the execution emulated in Stage 1 Main Execution Phase. Let p_ρ denote the probability that ρ is a safe-point⁷ and is rewound. From the construction of S , we know that S keeping rewinding until it finds another accepting

transcript $(\alpha_\rho, \beta'_\rho, \gamma'_\rho)$ for ρ , canceling each rewinding for which ρ is not a **safe-point**, i.e., A_{rev} requests the second message of a proof in the right-interaction whose first message occurs in ρ . As the emulated committer and receiver act identically as real committer and real receiver in this stage, conditioned on ρ , a view occurring in a rewinding from ρ is same as occurring in the **Stage 1 Main Execution Phase**. Thus, the probability of canceling a rewinding from ρ is at most $1 - p_\rho$. Furthermore, the expected number of rewindings is at most $\frac{1}{p_\rho}$. Therefore, the expected number of rewindings from ρ is at most $p_\rho \cdot \frac{1}{p_\rho} = 1$ and each rewinding takes at most $\text{poly}(n)$ steps, i.e., $\mathbb{E}[T(i)|\rho] \leq \text{poly}(n)$. Thus,

$$\mathbb{E}[T(i)] = \sum_{\rho \text{ of length } i} \mathbb{E}[T(i)|\rho] \cdot \Pr[\rho] \leq \text{poly}(n) \cdot \sum_{\rho \text{ of length } i} \Pr[\rho] \leq \text{poly}(n)$$

The expected running time of S in step 4c is also polynomial-time using similar analysis as above. We omit the details.

Analysis of the simulator S . In order to show equation (1), we define a hybrid stand-alone simulator HYB_1 that also receives v as auxiliary input. HYB_1 proceeds exactly as S except that in the commit phase, instead of feeding A a commitment to 0^n , HYB_1 feeds A a commitment to v .

Since both the experiment S and HYB_1 are efficiently computable, the following claim follows directly from the hiding property of $\langle \text{SHC}, \text{SHR} \rangle$.

Claim 1 *There exists some negligible function μ' such that*

$$\left| \Pr[\text{sim}_{\text{open}}^S(\mathcal{R}, v, z) = 1] - \Pr[\text{sim}_{\text{open}}^{\text{HYB}_1}(\mathcal{R}, v, z) = 1] \right| < \mu'(n)$$

Next we proceed to showing the following claim.

Claim 2 *There exists some negligible function μ'' such that*

$$\left| \Pr[\text{mim}_{\text{open}}^A(\mathcal{R}, v, z) = 1] - \Pr[\text{sim}_{\text{open}}^{\text{HYB}_1}(\mathcal{R}, v, z) = 1 | \neg \text{fail}] \right| < \mu''(n)$$

Proof (sketch). Note the view of A in the commit phase in a real interaction is identical to the view of A in HYB_1 . Furthermore, HYB_1 feeds A messages according to the correct distribution in **Stage 1**, the view of A_{rev} in the simulation of **Stage 1** by experiment HYB_1 is identical to the view of A_{rev} in a real interaction. The view of A_{rev} in the simulation of **Stage 3**

⁷ Note the roles of C and R interchange in **Stage 1** where C acts as a verifier and R acts as a prover. The **safe-point lemma** will be used by interchanging the right and the left.

by HYB_1 is computationally indistinguishable following from the witness-indistinguishability of $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$. As the **safe-point lemma** shows, when the right interaction has a different tag from the left interaction, there is a safe-point. Hence, according to the actions of HYB_1 , it will either output fail or succeed in the extraction from A_{rev} . Conditioned on HYB_1 not outputting fail, by the computational-binding property of $\langle \text{SHC}, \text{SHR} \rangle$, except with negligible probability, the witness \tilde{w} and the value \tilde{v} extracted by HYB_1 are the valid decommitment key and committed value of A , respectively.

We next show $|\Pr[\text{sim}_{\text{open}}^{\text{HYB}_1}(\mathcal{R}, v, z) = 1] - \Pr[\text{sim}_{\text{open}}^{\text{HYB}_1}(\mathcal{R}, v, z) = 1 | \text{fail}]|$ is negligible by proving that the probability that event fail happens is negligible. This together with Claim 1 and Claim 2 conclude Eq. (1).

Claim 3 HYB_1 outputs fail with negligible probability.

Proof. The proof of this claim is similar to that of [15]. More precisely, HYB_1 outputs fail only in three cases: HYB_1 runs for more than 2^n steps; or the same proof transcript is obtained from some safe-point; or the witness extracted is not a valid decommitment. The arguments of the first two cases are almost the same as those in [15]. The main difference lies in the analysis of the third case.

HYB₁ runs for more than 2^n steps: We know that the expected running time of HYB_1 and S are same, i.e., $\text{poly}(n)$. Using Markov inequality, we conclude that the probability that HYB_1 runs more than 2^n steps is at most $\frac{\text{poly}(n)}{2^n}$.

The same proof transcript is obtained from some safe-point: This case occurs if HYB_1 picks some challenge β (resp. $\tilde{\beta}$) in Stage 1 (resp. Stage 3) Rewinding Phase that appeared as a challenge in the Stage 1 (resp. Stage 3) Main Execution Phase. As HYB_1 runs for at most 2^n steps, it picks at most 2^n challenges. Furthermore, the length of each challenge is $2n$. By applying the union bound, we obtain that the probability that a β (resp. $\tilde{\beta}$) is picked twice is at most $\frac{2^n}{2^{2n}}$. Since there are at most polynomial many challenges in Stage 1 (resp. Stage 3), using union bound again, we conclude that the probability that it outputs fail in this case is negligible.

The witness extracted is not a valid decommitment:⁸ Suppose, on the contrary, the witness extracted is not the decommitment key for $\langle \text{SHC}, \text{SHR} \rangle$, then by the special-sound property, it follows that it must be a value r' such that $f(r') = s_{b'}$ for some $b' \in \{0, 1\}$. Denote by r_b

($b \in \{0, 1\}$) the witness used by HYB_1 in **Stage 1** of right interaction. If $b' = 1 - b$, then we can break the one-way function f . Given A, z and v , we construct an algorithm B that inverts f . The input to B is an n -bit string $y = f(x)$ where x was chosen randomly from $\{0, 1\}^n$. B wants to output a pre-image of y under f . B proceeds as follows: B runs identically as HYB_1 with inputs z, v with the exception that when simulating the right receiver for A in **Stage 1** of reveal phase, it picks a random bit $b \in \{0, 1\}$ and a random string $r_b \in \{0, 1\}^n$, and sets $s_b = f(r_b), s_{1-b} = y$. By using r_b as witness, it can simulate the right interaction with A_{rev} easily. Finally, if B extracts a witness r' where $f(r') = y$, then we break the one-wayness of f . The probability that B inverts f is identical to the probability that HYB_1 inverts f which is non-negligible. This contradicts the one-wayness of f .

We therefore have only to deal with the case that B always outputs r' such that $f(r') = s_b$, i.e., B always outputs same preimage it knows. Then we can break the witness indistinguishability of the underlying special-sound proofs as follows: Recall that the proof $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ in **Stage 1** of right interaction contains $4m$ number of special-sound WI proofs. The above assumption is that B always extracts the same preimage used by itself in **Stage 1** of right interaction. We know that if the $4m$ number proofs use r_0 , B outputs r_0 , and if the $4m$ number proofs use r_1 , B outputs r_1 . Applying standard hybrid arguments, there exists $i \in [4m]$, by using r_0 for the first $i - 1$ proofs and r_1 for the last $4m - i$ proofs, the witness used in the i -th special-sound proof is the same as that of the witness extracted by B . We can use this session to break the witness-indistinguishability of special-sound WI proof. The probability we break the witness-indistinguishability property of the underlying special-sound proof is $\frac{1}{4m}$ times the probability that HYB_1 inverts f which is non-negligible. This contradicts the witness-indistinguishability property of the underlying special-sound proof.

Computational binding. The binding property intuitively follows from the binding property of the underlying commitment scheme $\langle \text{SHC}, \text{SHR} \rangle$ and the special-sound property (or more precisely proof of knowledge property) of the underlying proof in $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$. A formal proof proceeds along

⁸ The proof in this case heavily relies on the “simulation-extractability” property of $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ in **Stage 1**. An ordinary WI proof of knowledge is not suffice here, as the problem in this case is reduced to the security of one-way functions or witness-indistinguishability of underlying subprotocols, in the presence of an expected PPT adversary who can *rewind* the same subprotocols.

the lines of the proof of non-malleability. More precisely, suppose, there exists an adversary A that can violate the binding property of $\langle C, R \rangle$, then we design an algorithm A' that violates the binding property of $\langle \text{SHC}, \text{SHR} \rangle$. A' incorporates A and relays the commit phase messages to an external honest receiver SHR . In the reveal phase, there is no need of A' to simulate the left interaction for A . Note in the non-malleability proof, two extraction are executed. Here, we only execute one extraction by standard rewinding, and obtain the decommitment key. Using this information, A' can easily complete the reveal phase with SHR . It follows from the witness-indistinguishability property of $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ that the probability that A' breaks the binding property of $\langle \text{SHC}, \text{SHR} \rangle$ is negligible close to the probability that A breaks the binding property of $\langle C, R \rangle$.

Schedule of messages: In the non-malleability proof, the design of S is based on an unspecified assumption, i.e., in the reveal phase, **Stage 3** on both interactions will not start unless the simulations for **Stage 1** are completed. Without loss of generality, this assumption is reasonable.

Consider the scenario where the simulation for **Stage 1** of the left interaction and **Stage 3** of the right interaction overlap. The simulation goes well as the adversary runs as a prover in **Stage 3** of the right interaction, and the rewinding of **Stage 1** of the left interaction will not “rewind” the **Stage 3** of the right interaction (i.e., the adversary can only answer the left challenge by itself, without the help from the right interaction). By using the **safe-point lemma**, the simulator can still find a **safe-point** and extract the witness to the statement proved by the adversary by standard rewinding. Furthermore, the adversary also runs as a prover in **Stage 1** of the left interaction, and the rewinding of **Stage 3** of the right interaction will not “rewind” the **Stage 1** of the left interaction. Due to a more simpler but similar reason, when the simulation for **Stage 3** of the left interaction and **Stage 1** of the right interaction overlap, the simulator has no difficulty and the two extractions also performs well. We take a special note of the fact that the **safe-point lemma** depicts the existence of **safe-point** in any one-one concurrent execution environment, and considers an environment where one-side of the interaction is empty as a special case.

4 Acknowledgement.

We thank the anonymous reviewers for their valuable comments. Zongyang Zhang thank Huijia Lin for helpful discussions on the definition of **safe-point**. The work is supported by the National Nature Science Foundation

of China No.60673079 and No.60773086, and by the National 973 Program No. 2007CB311201.

References

1. Haitner, I., Reingold, O.: Statistically-hiding commitment from any one-way function. In Johnson, D.S., Feige, U., eds.: STOC, ACM (2007) 1–10
2. Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. *SIAM J. Comput.* **30** (2000) 391–437
3. Fischlin, M., Fischlin, R.: Efficient non-malleable commitment schemes. In Bellare, M., ed.: CRYPTO. Volume 1880 of Lecture Notes in Computer Science., Springer (2000) 413–431
4. Di Crescenzo, G., Ishai, Y., Ostrovsky, R.: Non-interactive and non-malleable commitment. In: STOC. (1998) 141–150
5. Boyar, J., Kurtz, S.A., Krentel, M.W.: A discrete logarithm implementation of perfect zero-knowledge blobs. *J. Cryptology* **2** (1990) 63–76
6. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.* **37** (1988) 156–189
7. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.* **17** (1988) 281–308
8. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. *J. Cryptology* **9** (1996) 167–190
9. Damgård, I., Pedersen, T.P., Pfitzmann, B.: On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *J. Cryptology* **10** (1997) 163–194
10. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for NP using any one-way permutation. *J. Cryptology* **11** (1998) 87–108
11. Haitner, I., Horvitz, O., Katz, J., Koo, C.Y., Morselli, R., Shaltiel, R.: Reducing complexity assumptions for statistically-hiding commitment. In Cramer, R., ed.: EUROCRYPT. Volume 3494 of Lecture Notes in Computer Science., Springer (2005) 58–77
12. Nguyen, M.H., Ong, S.J., Vadhan, S.P.: Statistical zero-knowledge arguments for NP from any one-way function. In: FOCS, IEEE Computer Society (2006) 3–14
13. Di Crescenzo, G., Katz, J., Ostrovsky, R., Smith, A.: Efficient and non-interactive non-malleable commitment. In Pfitzmann, B., ed.: EUROCRYPT. Volume 2045 of Lecture Notes in Computer Science., Springer (2001) 40–59
14. Pass, R., Rosen, A.: New and improved constructions of nonmalleable cryptographic protocols. *SIAM J. Comput.* **38** (2008) 702–752
15. Lin, H., Pass, R., Venkatasubramanian, M.: Concurrent non-malleable commitments from any one-way function. In Canetti, R., ed.: TCC. Volume 4948 of Lecture Notes in Computer Science., Springer (2008) 571–588
16. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: STOC, ACM (1990) 416–426
17. Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians. (1986) 1444–1451
18. Naor, M.: Bit commitment using pseudorandomness. *J. Cryptology* **4** (1991) 151–158
19. Goldreich, O.: *The Foundations of Cryptography - Volume 1*. Cambridge University Press, UK (2001)

20. Feige, U.: Alternative Models for Zero Knowledge Interactive Proofs. PhD thesis, The Weizmann Institute of Science, Rehovot, Israel (1990)
21. Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.* **29** (1999) 1–28
22. MacKenzie, P., Yang, K.: On simulation-sound trapdoor commitments. *Cryptology ePrint Archive, Report 2003/252* (2003) <http://eprint.iacr.org/>.

A A Content-Based Non-Malleable Commitment Scheme

Let $\langle \text{SHC}, \text{SHR} \rangle$ be the statistically hiding commitment scheme [1] from any one-way function and let $\langle P_{\text{tag}}, V_{\text{tag}} \rangle$ be a tag-based WI proof for all NP. Let $\text{SS} = (\text{SG}, \text{Sig}, \text{SVer})$ be a secure signature scheme. The content-based non-malleable statistically hiding commitment scheme is shown in Fig. 4. Due to page limit, the formal proof is omitted here.

Protocol $\langle C, R \rangle$
 Security Parameter: 1^n
 String to be committed: $v \in \{0, 1\}^n$
 Commit Phase:
 $C \Leftrightarrow R$: Run the commit phase of commitment scheme $\langle \text{SHC}, \text{SHR} \rangle$.
 R : Abort if the above commit phase fails.
 Denote the above transcript as com . C records the decommitment key in dec .
 Reveal Phase:
 Stage 1:
 $R \rightarrow C$: Set $(pk_0, sk_0) \leftarrow \text{SG}(1^n)$ and send pk_0 .
 $R \rightarrow C$: Pick uniformly $r_0, r_1 \in \{0, 1\}^n$, compute $s_0 = f(r_0)$ and $s_1 = f(r_1)$ and send s_0, s_1 .
 $R \Leftrightarrow C$: R and C engage in an execution of $\langle P_{pk_0}, V_{pk_0} \rangle$ with tag pk_0 , where R uses r_b as witness ($b \in \{0, 1\}$) and runs P_{pk_0} to prove to C (running V_{pk_0}) that there exists a value r s.t. $s_0 = f(r)$ or $s_1 = f(r)$. The challenge length of the verifier (i.e., C) is $2n$. C aborts if either s_0 or s_1 is not in the range of f or the proof fails.
 $R \rightarrow C$: Let tr_0 be the transcript so far. Set $\sigma_0 \leftarrow \text{Sig}(tr_0, sk_0)$ and send σ_0 .
 C : Abort if $\text{Sver}(pk_0, tr_0, \sigma_0) \neq 1$.
 Stage 2: $C \rightarrow R$: Send v .
 Stage 3:
 $C \rightarrow R$: Set $(pk_1, sk_1) \leftarrow \text{SG}(1^n)$ and send pk_1 .
 $C \Leftrightarrow R$: C and R engage in an execution of $\langle P_{pk_1}, V_{pk_1} \rangle$ with tag pk_1 , where C uses witness dec and runs P_{pk_1} to prove to R (running V_{pk_1}) that there exists a value dec s.t. dec is the decommitment key of com corresponding to v or there exists a value r s.t. $s_0 = f(r)$ or $s_1 = f(r)$. The challenge length of the verifier (i.e., R) is $2n$.
 $C \rightarrow R$: Let tr_1 be the transcript so far. Set $\sigma_1 \leftarrow \text{Sig}(tr_1, sk_1)$ and send σ_1 .
 R : Abort if $\text{Sver}(pk_1, tr_1, \sigma_1) \neq 1$.

Fig. 4. Non-malleable statistically hiding commitment scheme $\langle C, R \rangle$