Constant-Round Non-Malleable Commitments from Any One-Way Function

Huijia Lin[*] Rafael Pass[†]

September 1, 2011[‡]

Abstract

We show *unconditionally* that the existence of commitment schemes implies the existence of *constant-round* non-malleable commitments; earlier protocols required additional assumptions such as collision resistant hash functions or subexponential one-way functions.

Our protocol also satisfies the stronger notions of concurrent non-malleability and robustness. As a corollary, we establish that constant-round non-malleable zero-knowledge arguments for **NP** can be based on one-way functions and constant-round secure multi-party computation can be based on enhanced trapdoor permutations; also here, earlier protocols additionally required either collision-resistant hash functions or subexponential one-way functions.

[*] Cornell University, E-Mail: huijia@cs.cornell.edu.

[†] Cornell University, E-Mail: rafael@cs.cornell.edu. Pass is supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract GG11413-137380. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US government.

[‡] A preliminary version of this paper appeared in STOC 2011.

1 Introduction

Commitment schemes are one of the most fundamental cryptographic building blocks. Often described as the "digital" analogue of sealed envelopes, commitment schemes enable a *sender* to commit itself to a value while keeping it secret from the *receiver*. This property is called *hiding*. Furthermore, the commitment is *binding*, and thus in a later stage when the commitment is opened, it is guaranteed that the "opening" can yield only a single value determined in the committing stage. Their applications range from coin flipping [Blu83] to the secure computation of any efficiently computable function [GMW91, GMW87]. In light of their importance, commitment schemes have received a considerable amount of attention. This has resulted in a fairly comprehensive understanding of the hardness assumptions under which they can be realized; in particular, by the results of Naor [Nao91] and Håstad et al [HILL99], the existence of one-way functions implies the existence of two-round commitments.

For many applications, however, the most basic security guarantees of commitments are not sufficient. For instance, the basic definition of commitments does not rule out an attack where an adversary, upon seeing a commitment to a specific value v, is able to commit to a related value (say, $v-1$), even though it does not know the actual value of v. This kind of attack might have devastating consequences if the underlying application relies on the *independence* of committed values (e.g., consider a case in which the commitment scheme is used for securely implementing a contract bidding mechanism). Indeed, for the general task of secure multi-party computation [GMW87], such independence is cruicial. The state of affairs is even worsened by the fact that many of the known commitment schemes are actually susceptible to this kind of attack. In order to address the above concerns, Dolev, Dwork and Naor (DDN) introduced the concept of *non-malleable commitments* [DDN00]. Loosely speaking, a commitment scheme is said to be non-malleable if it is infeasible for an adversary to "maul" a commitment to a value v into a commitment to a related value $\tilde{v}$.

More precisely, we consider a *man-in-the-middle* (MIM) attacker that participates in two concurrent execution of a commitment scheme $\langle C, R\rangle$; in the "left" execution it interacts with an honest committer (running C); in the "right" execution it interacts with an honest receiver (running R). Additionally, we assume that the players have n-bit identities (where n is polynomially related to the security parameter), and that the commitment protocol depends only on the identity of the committer; we sometimes refer to this as the identity of the interaction. Intuitively, $\langle C, R\rangle$ being non-malleable means that if the identity of the right interaction is different than the identity of the left interaction (i.e., A does not use the same identity as the left committer), the value A commits to on the right does not depend on the value it receives a commitment to on the left; this is formalized by requiring that for any two values v_1, v_2, the values A commits to after receiving left commitments to v_1 or v_2 are indistinguishable.

The first non-malleable commitment protocol was constructed by Dolev, Dwork and Naor [DDN00] in 1991. The security of their protocol relies on the minimal assumption of one-way functions and requires $\Omega(\log n)$ rounds of interaction, where $n \in N$ is the length of party identities. Non-malleable commitments have since been extensively studied in the literature; the main question has been to determine the number of communication rounds needed for non-malleable commitments. Let us briefly survey some of this literature.

1.1 The State of the Art of Non-malleable Commitments

As mentioned, the original work by DDN assumes only one-way functions, and considers the "plain" model of execution; that is, there is no trusted infrastructure. DiCrenenzo, Ishai and Ostrovsky [CKOS01] and follow-up works in e.g., [CKOS01, CIO01, CF01, FF09, DG03] showed how to improve the round-complexity of the DDN construction when assuming the existence of some trusted infrastructure (e.g. a common random string); in such models *non-interactive* (i.e., single message) non-malleable commitments based on only one-way function are known [DG03]. The first improvement to the round-complexity of the DDN construction without any trusted infrastructure came more than a decade later. Following the ground-breaking work by Barak on *non-black-box simulation* [Bar01], in 2002, Barak [Bar02] presented a constant-round protocol for non-malleable commitments; the security of this protocol however relies on the existence of trapdoor permutations and hash functions that are collision-resistant against circuits of sub-exponential size. A few years later, Pass and Rosen [PR05b] (relying on a technique from [Pas04]), showed that collision resistant hash functions secure against polynomially-sized circuits are sufficient to obtain a constant-round protocol. Next, Pandey, Pass and Vaikuntanathan [PPV08] provided a construction of a non-interactive non-malleable commitment based on a new hardness assumption with a strong non-malleability flavour; in contrast to the earlier constant-round constructions, their protocol has a black-box proof of security.

Last year, the authors showed a $O(1)^{\log^* n}$-round protocol that is based on the existence of one-way functions and uses a black-box proof of security [LP09]. Subsequent work by Pass and Wee [PW10] obtained a constant-round protocol based on sub-exponetially hard one-way functions (again using a black-box proof of security). Finally, a very recent work by Wee [Wee10] improved the round-complexity of the [LP09] protocol to $O(\log^* n)$-rounds, based on one-way functions.

So, summarizing the state of the art, we have:

- Based on the minimal assumption of (polynomially-hard) one-way functions, the best protocols require $O(\log^*)$-rounds and have black-box proofs of security.
- Constant-round protocols are known based on either collision-resistant hash functions (and using a non-black-box proofs of security), or subexponetially hard one-way functions (using a black-box proof of security).

1.2 Settling the Round-complexity of Non-Malleable Commitments

In this work, we settle the round-complexity of non-malleable commitments: we present a constant-round protocol in the "plain" model that is based on the assumption of (polynomially-hard) one-way functions, and has a black-box proof of security. Since the existence of commitment schemes already implies the existence of one-way functions (cl. [IL89]), we have:

Theorem 1. *Assume the existence of a commitment scheme. Then, there exists a constant-round non-malleable commitment scheme with a black-box proof of security.*

Concurrent Non-malleability: As mentioned, the original notion of non-malleability considers an MIM attacker participating in a single execution on the left and a single execution on the right. Already the original DDN paper suggested that a stronger notion of non-malleability—*concurrent non-malleability*—where the MIM may participate in an unbounded number of executions on both the left and the right, is desirable. Pass and Rosen [PR05a] provided the first construction of a concurrently non-malleable commitment scheme; their scheme only has a constant number of rounds but relies on the existence of claw-free permutations (and non-black-box techniques). Subsequently, Lin, Pass and Venkitasubramaniam [LPV08] provided an $O(n)$-round construction based on one-way functions. Finally, the protocol in [LP09] and its variants [PW10, Wee10] yield a $O(\log^*)$-round concurrent non-malleable protocol based on one-way functions, or a constant-round concurrent non-malleable protocol based on subexponential one-way functions. As we show, our protocol is also concurrently non-malleable.

Robust Non-malleability: In [LP09], we introduced the notion of *robust non-malleable* commitments. Roughly speaking, whereas non-malleability considers a scenario where a MIM participates in the same commitment protocol on the left and the right, r-robustness considers a notion of non-malleability for commitments where the MIM attacker participates in an arbitrary r-round protocol on the left, and the commitment protocol on the right. Robustness is useful when using non-malleable commitments as subprotocols within larger protocols. As we show, for any constant r, our protocol can be made r-robust while still remaining constant-round.

Thus summarizing the above discussion, we have:

Theorem 2. *Assume the existence of a commitment scheme. Then, for any constant r, there exists a constant-round commitment scheme that is r-robust concurrently non-malleable with a black-box proof of security.*

1.3 Applications to Secure Computation

As mentioned, "independence" of inputs is crucial for secure multi-party computation protocols. Indeed, there has been a tight interplay between work on the round-complexity of multi-party computation (MPC) and work on non-malleable commitments.

Goldreich, Micali and Wigderson's [GMW87] original work on secure multi-party computation showed a $\Omega(m)$-round multi-party computation protocol based on the existence of enhanced trapdoor permutations (TDPs), where m is the number of players in the execution; implicit in their work is a $O(n)$-round non-malleable commitment for the special case of so-called "synchronizing" adversaries that have identities of length $\log n$. Subsequent works improved the round-complexity by making stronger assumptions. Katz, Ostrovsky, and Smith [KOS03], following the work by Chor and Rabin [CR87], obtained a $O(\log m)$-round MPC protocol assuming TDPs and dense-crypto systems by relying on the non-malleable commitments from [DDN00]. By additionally assuming the existence of hash-function collision-resistant against circuits of sub-exponential size (and non-black-box techniques), they also obtained a $O(1)$-round MPC protocol by instead relying on the non-malleable commitment from [Bar02]. More recently, Pass [Pas04], showed the existence of a $O(1)$-rounds MPC protocol assuming only TDPs and (standard) collision resistant hash functions (but still using non-black box techniques); this technique in turned was used in the non-malleable commitment of [PR05a].

The implicit connection between the round-complexity of non-malleable commitments and secure multi-party was formalized by Lin, Pass and Venkitasubramaniam in [LPV09]: they show that *the existence of k-round 4-robust non-malleable commitments and the existence of TDPs implies the existence of $O(k)$-round secure multi-party computation.* All the more recent constructions of non-malleable commitments [LP09, PW10, Wee10] are robust, and as a consequence, the state of the art of secure multi-party computation was the same as for non-malleable commitments.

Combining the result of [LPV09] with Theorem 2, we get that secure multi-party computations can be performed in a constant number of round based on only TDPs.

Theorem 3. *Assume the existence of enhanced trapdoor permutations. Then there exists a constant-round protocol for secure multi-party computation.*

1.4 Applications to Non-malleable ZK

Non-malleable zero-knowledge [DDN00] consider the execution of zero-knowledge protocols in the presence of a MIM attacker. Roughly speaking, a zero-knowledge protocol is non-malleable if the MIM attacker can only provide convincing right-interaction proofs of statements that it could have proved without participating in the left interaction. The recent result of [LPTV10] shows that *the* *existence of k-round 4-robust non-malleable commitments implies the existence of $O(k)$-round non-malleable zero-knowledge arguments for* **NP**. By combining their results with Theorem 2 we directly have that the existence of one-way functions implies the existence a constant-round zero-knowledge argument for **NP**.

Theorem 4. *Assume the existence of one-way functions. Then there exists a constant-round non-malleable zero-knowledge argument for* **NP** *with a black-box proof of security.*

1.5 A New Technique: "Message-scheduling in the head"

The main idea underlying all non-malleable commitment schemes is to "encode" the identity of the committer into the protocol. At the very least, this ensure that unless the attacker copies the identity of the left committer, the attacker cannot simply forward messages between the left and the right executions. But we also need to ensure that the attacker cannot in a clever way maul the messages it receives on the left so they become useful on the right. For instance, in the original DDN construction, the identity is encoded into the scheduling of messages in the protocol; on a very high-level (and oversimplifying), the idea is to ensure that at some point in the execution, the MIM must "speak" while only receiving "useless" messages. The problem with this approach is that it requires a high round-complexity.

We will revisit the DDN approach. The main idea behind our scheme is to perform the message scheduling "in the head". A bit more precisely, our protocol follows the "simulation-soundness" paradigm of Sahai [Sah99], first used in the context of CCA-secure encryption, and next used by Pass and Rosen [PR05a] and Lin and Pass [LP09] in the context of non-malleable commitments; that is, the main component of our construction is a method for enabling us to "simulate" the left interaction, while ensure that the right interaction remains "sounds". Towards this, we embedd a "trapdoor" into the protocol which depends on the identity of the interaction; proving simulation-soundness then essentially amounts to showing that there exists a way to recover the trapdoor for the left intraction, while ensuring that the adversary does not recover the trapdoor for the right interaction (as long as the right interaction has a different identity than the left interaction).

The idea is to have a protocol where the trapdoor can be recovered by "rewinding" some specific messages in the protocol—called "slots"—in *a specific order which depends on the identity of the interaction.* Furthemore, the protocol should have the property that if this specific rewinding order is not the rewinding order actually used, then a trapdoor cannot be recovered. So, if we rewind the left interaction according to the rewinding order corresponding to the identity of the left interaction, this will still not enable the adversary to recover the trapdoor corresponding to the right interaction (unless the identity of the right interaction is the same as the identity of the left interaction). In our particular instantiation of this idea, the trapdoor will be a "signature-chain" (i.e., a signature on a signature on a signature, etc.) of length n (i.e., the identity length) using different keys; the choice of the keys in the signature chain are determined by the identity of the interaction. Next, the protocol will have a "slot" for each of the keys where the receiver is willing to sign a *single* message for the committer using the key corresponding to the slot. The key point is that the simulator is able to rewind the slots in an appropriate order to recover a signature-chain corresponing to the identity of the left interaction; but the rewindings will still not enable the adversary to recover a signature-chain corresponding to any other identity.

1.6 Outline

In Section 2, we provide some preliminaries. In Section 3, we provide an overview of our protocol construction and its security proof. In Section 4 we provide some formalizations and results abouts "signature-chains". Our protocol (which relies on the notion of a signature chain) is presented in

Section 5. We provide the proof of (stand-alone) non-malleabilty in Section 6; in Section 7 and 8, we demonstrate that our protocol is also concurrent non-malleable, and can be made r-robust for any constant r.

2 Preliminaries

Let N denote the set of all positive integers. For any integer $n \in N$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$, We denote by $\{0,1\}^n$ the set of binary strings of length n, and $\{0,1,2\}^n$ the set of trinary strings of length n. Given a binary (or trinary) string ψ of length n, we denote by $[\psi]_1^i$ the prefix of ψ of length i. We denote by $\mathcal{PPT}$ probabilistic polynomial time Turing machines. We assume familiarity with interactive Turing machines, denoted ITM, interactive protocols, and computational indistinguishability; the formal definitions of interactive protocols and comutational indistinguishability are provided in Appendix A. Given a pair of ITMs, A and B, we denote by $\langle A(x), B(y)\rangle(z)$ the random variable representing the (local) output of B, on common input z and private input y, when interacting with A with private input x, when the random tape of each machine is uniformly and independently chosen.

2.1 Signature Schemes

We focus on *fixed-length signature schemes* $\Pi = (Gen, Sign, Ver)$, that is, the signing algorithm $Sign$ on input 1^n, a public key pk and a message $m \in \{0,1\}^*$, always outputs a signature of length n. We refer the reader to [Gol04] for a formal definition. Such signature schemes can be constructed relying on universal one-way hash functions [NY89], which in turn can be based on any one-way function [Rom90]. Below, a signature scheme always refers to a fixed-length signature scheme.

2.2 Commitment Schemes

Commitment schemes are used to enable a party, known as the *sender*, to commit itself to a value while keeping it secret from the *receiver* (this property is called hiding). Furthermore, the commitment is binding, and thus in a later stage when the commitment is opened, it is guaranteed that the "opening" can yield only a single value determined in the committing phase. In this work, we consider commitment schemes that are statistically-binding, namely while the hiding property only holds against computationally bounded (non-uniform) adversaries, the binding property is required to hold against unbounded adversaries. We refer the reader to [Gol01] for a formal definition.

Two-round (i.e., a single message from the receiver followed by a single message from the committer) commitment schemes are known to exist based on the minimal assumption of one-way functions [Nao91, HILL99]. In the sequel of the paper, a commitment scheme always refers to a statistically-binding commitment.

Tag-based Commitment Scheme. Following [PR05a, DDN00], we consider *tag-based commitment schemes* where, in addition to the security parameter, the committer and the receiver also receive a "tag"—a.k.a. the identity—id as common input.

2.3 Concurrent Non-Malleable Commitments

We recall the definition of concurrent non-malleability from [LPV08]. For convenience, we use a slightly different presentation (based on indistinguishability rather than simulation); equivalence follows using a standard argument (c.f. [GM84, PR05a]). Let $\langle C, R\rangle$ be a tag-based commitment scheme, and let $n \in N$ be a security parameter. Consider a man-in-the-middle adversary A (as shown in figure 1) that, on inputs n and z (where z is received as an auxiliary input), participates

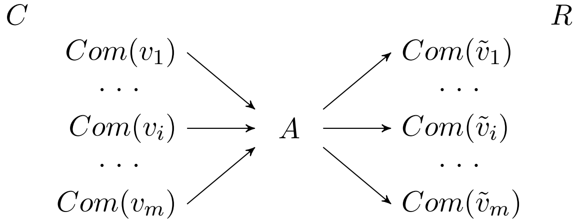
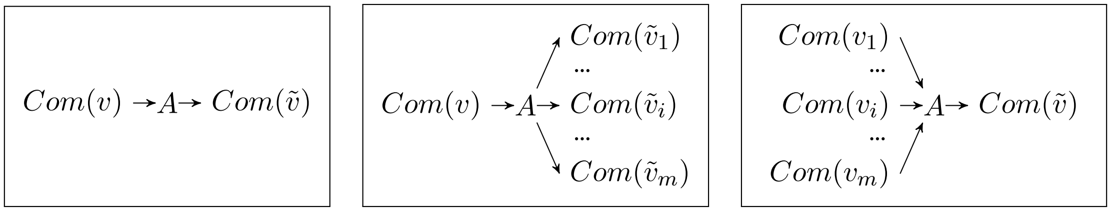


Figure 1: A concurrent man-in-the-middle adversary.


$Com(v) \to A \to Com(\tilde{v})$
$Com(v) \to A \to Com(\tilde{v}_i)$
$Com(\tilde{v}_1)$
$Com(\tilde{v}_m)$
$Com(v_1)$
$Com(v_i) \to A \to Com(\tilde{v})$
$Com(v_m)$

(i) one-one (ii) one-many (iii) many-one

Figure 2: Restricted man-in-the-middle adversaries.

in m left and right interactions simultaneously. In the left interactions the man-in-the-middle adversary A interacts with C, receiving commitments to values $v_1, \ldots, v_m$, using identities of length n, $\mathsf{id}_1, \ldots, \mathsf{id}_m \in \{0,1\}^n$, of its choice. In the right interactions A interacts with R attempting to commit to a sequence of related values $\tilde{v}_1, \ldots, \tilde{v}_m$, again using identities of length n $\tilde{\mathsf{id}}_1, \ldots, \tilde{\mathsf{id}}_m$ of its choice. If any of the right commitments are invalid, or undefined, its value is set to $\perp$. For any i such that $\tilde{\mathsf{id}}_i = \mathsf{id}_j$ for some j, set $\tilde{v}_i = \perp$—i.e., any commitment where the adversary uses the same identity as one of the left interactions is considered invalid. Let $\mathsf{mim}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)$ denote a random variable that describes the values $\tilde{v}_1, \ldots, \tilde{v}_m$ and the view of A, in the above experiment.

Definition 1. *A commitment scheme $\langle C, R\rangle$ is said to be* concurrent non-malleable (with respect to itself) *if for every polynomial $p(\cdot)$, and every $\mathcal{PPT}$ man-in-the-middle adversary A that participates in at most $m = p(n)$ concurrent executions, the following ensembles are computationally indistinguishable.*

$$\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v_1, \ldots, v_m, z)\right\}_{n\in N, v_1,\ldots,v_m \in \{0,1\}^n, v'_1,\ldots,v'_m\in\{0,1\}^n, z\in\{0,1\}^*}$$

$$\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v'_1, \ldots, v'_m, z)\right\}_{n\in N, v_1,\ldots,v_m \in \{0,1\}^n, v'_1,\ldots,v'_m\in\{0,1\}^n, z\in\{0,1\}^*}$$

We also consider relaxed notions of concurrent non-malleability: one-one, one-many, and many-one secure non-malleable commitments (See Figure 2 below.) In a one-one (a.k.a., a stand-alone secure) non-malleable commitment, we consider only adversaries A that participate in one left and one right interaction; in one-many, A participates in one left and many right, and in many-one, A participates in many left and one right.

As shown in [LPV08], any protocol that is one-many non-malleable is also concurrent non-malleable.

Proposition 1 ([LPV08])**.** *Let $\langle C, R\rangle$ be a one-many concurrent non-malleable commitment. Then, $\langle C, R\rangle$ is also a concurrent non-malleable commitment.*

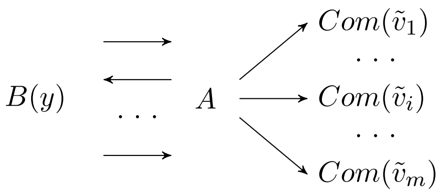


Figure 3: A concurrent man-in-the-middle adversary with respect to protocol B on input y.

2.4 Robustness: Non-Malleability w.r.t. k-round Protocols

The concept of non-malleability is traditionally only considered in a setting where a man-in-the middle adversary is participating in two (or more) executions of the *same* protocol. We here consider a notion of non-malleability with respect to arbitrary k-round protocols. Below, we recall the definition of k-robustness from [LP09].

Consider a one-many man-in-the-middle adversary A (as shown in figure 3) that participates in one left interaction—communicating with a machine B—and many right interactions—acting as a committer using the commitment scheme $\langle C, R\rangle$. As in the standard definition of non-malleability, A can adaptively choose the identities in the right interactions. We denote by $\mathsf{mim}^{B,A}_{\langle C,R\rangle}(y,z)$ the random variable consisting of the view of $A(z)$ in a man-in-the-middle execution when communicating with $B(y)$ on the left and honest receivers on the right, combined with the values $A(z)$ commits to on the right. Intuitively, we say that $\langle C, R\rangle$ is one-many non-malleable w.r.t B if $\mathsf{mim}^{B,A}_{\langle C,R\rangle}(y_1,z)$ and $\mathsf{mim}^{B,A}_{\langle C,R\rangle}(y_2,z)$ are indistinguishable, whenever interactions with $B(y_1)$ and $B(y_2)$ cannot be distinguished.

Definition 2. *Let $\langle C, R\rangle$ be a commitment scheme, and B a $\mathcal{PPT}$ ITM. We say the commitment scheme $\langle C, R\rangle$ is* one-many non-malleable w.r.t. *B, if for every two sequences $\{y^1_n\}_{n\in N}$ and $\{y^2_n\}_{n\in N}$, $y^1_n, y^2_n \in \{0,1\}^n$, such that, for all $\mathcal{PPT}$ ITM $\tilde{A}$, it holds that*

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

then it also holds that, for every $\mathcal{PPT}$ one-many man-in-the-middle adversary A,

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

We say that $\langle C, R\rangle$ is one-many k-robust if $\langle C, R\rangle$ is one-many non-malleable w.r.t. any machine B that interacts with the man-in-the-middle adversary in k rounds.

3 Proof Overview

To explain the main ideas behind our construction, we here focus on outlining the construction of a constant-round non-malleable commitment scheme that is secure for *synchronizing* and *non-aborting* adversaries; we next comment on how to deal with general adversaries. An adversary is said to be synchronizing if it "aligns" the left and the right executions; that is, whenever it receives message i on the left, it directly sends message i on the right, and vice versa. An adversary is said to be non-aborting if it never sends any invalid messages in the left interaction (where it is acting as a receiver); it might still send invalid messages on the right.

As mentioned in the introduction, the idea is to have a protocol with an "identity-based trapdoor" embedded into it. The trapdoor will be a "signature-chain" using a sequence of keys that are determined by the identity of the protocol. More precisely, we say that $(\sigma_0, \sigma_1, \ldots, \sigma_n)$ is a *plain signature chain*[1] with respect to the signature scheme Π, the verification keys vk_0, vk_1 and the pattern $\psi \in \{0,1\}^n$ if $\sigma_0 = 0$ and for all $0 \leq i < n$, σ_{i+1} is a signature on the message (i, σ_i) with respect to the key $vk_{\psi_{i+1}}$. For convenience of notation, for the remainder of this section we fix a particular signature scheme Π; all signatures we use are with respect to this this particular scheme.

The following simple claim regarding signature chains will be useful. Consider a "signature game" where an adversary A gets access to two randomly chosen verification keys vk_0, vk_1 and additionally has access to signature oracles with respect to vk_0 and vk_1; let φ denote the "access pattern" of the adversary to the signature oracle (that is, if the i'th oracle call is to the signature oracle w.r.t. vk_b, then $\varphi_i = b$). The claim now is that, with overwhelming probability, if in the signature game, A manages to output a plain signature chain with respect to vk_0, vk_1 and pattern ψ, then ψ is a substring of φ.

The protocol for committing to a string v with identity id proceeds as follows:

- **Slot 1**: The receiver R generates a key-pair (sk_0, vk_0) for the signature scheme Π, and sends vk_0 to the committer C. C next send a random message r_0 to R who signs r_0 and then returns the signature to C.

- **Slot 2**: R generates another key-pair (sk_1, vk_1) and sends vk_0 to the committer C. As in Slot 1, C next send a random message r_1 to R who signs r_1 and then returns the signature to C.

- **Commit phase**: C commits to v using a standard statistically binding commitment.

- **Proof phase**: C gives R a "special-purpose"[2] witness indistinguishable argument of knowledge of the fact that it either knows the value committed to in the commit phase, or that it knows a plain signature chain with respect to vk_0, vk_1 and id.

We now turn to argue that this protocol is non malleable with respect to non-aborting and synchronizing adversaries. For simplicity, we here focus only on one-one (i.e., stand-alone) non-malleability (but the same proof actually also works for concurrent non-malleability). Consider a man-in-the-middle adversary A that uses identity id on the left and identity $\tilde{\mathsf{id}} \neq \mathsf{id}$ on the right, and receives a commitment to the value v on the left. We will argue that no matter what the value of v is, the value it commits to on the right will be indistinguishable. Towards this goal, consider a hybrid experiment where the left interaction is simulated by acting honestly in Slot 1 and 2, next committing to 0, and finally using a "fake-witness"—namely a signature chain—in the proof phase; the simulator obtains this fake witness by simply rewinding Slot 1 and 2 (that is, to rewinding slot b, we restore the state of A after vk_b has been sent, and send a new message to be signed) in the appropriate order to obtain a signature chain with respect to id (note that since A is non-aborting, each time the simulator asks it to sign a message, it does). To show the above claim, we now argue that no matter what the value of v is, the value A commits to on the right in the real execution (when receiving a commitment to v), is indistinguishable from the value it commits to on the right when the left interaction instead is simulated.

The key-point of the proof is the claim that even in the simulation, A cannot use a fake-witness in the right interaction. This follows from the fact that since A is synchronizing, when we rewind Slot 1 and 2 on the left, the same slots are rewound on the right *in exactly the same order*. Thus, by

[1] We use the name "plain signature chain" (instead of just "signature chain"), since the actual signature chains we will use in the final construction will be a bit more complicated.

[2] We will shortly explain what makes this proof special.

the signature-game claim, if A manages to get a signature chain it must be a subset of the pattern $01\mathsf{id}$ (the reason we need to append 01 is that A gets two signatures in the honest emulation of Slot 1 and 2, already before we start the rewindings). So, if we appropriately restrict the identity set (for instance, by requiring that all identities start with 10) then the only valid identity that is a substring of $01\mathsf{id}$ is id, and thus $\tilde{\mathsf{id}} = \mathsf{id}$, which is a contradiction.

To argue that the value committed to on the right does not change when we move from the real interaction to the simulation, consider an intermediary hybrid where we only change the witness used in the proof phase (but keep the value committed to in the commit phase to v). Note that since the adversary is synchronizing, the proof phase of the left interaction appears completely after the commitment (in Stage 2) in the right interaction. Therefore, the right value does not change at all when switching the the witness used in the proof phase on the left.

Finally, we simply have to argue that the value on the right does not change once we change the value committed to in the commit phase on the left. By the hiding property of the left commitment, the view of the adversary does not change when the left committed value switches. But since the value committed to on the right cannot be efficiently recovered, this does not directly imply that the committed value also is indistinguishable. To resolve this problem, we rely on the argument of knowledge property of the proof phase: A witness on the right can be extracted efficiently from the proof phase. Since the witness used in the right interaction cannot be a fake witness (by the key-claim above), it must be the value committed to in the commit phase, so indistinguishability of the committed value follows from the hiding property of the the left commitment.

Dealing with aborting adversaries: When considering aborting adversaries, we run into two obstacles:

- The adversary might notice that the simulator is feeding it signature chains to sign (instead of random messages) and thus decide to abort the left execution. We handle this by adapting the definition of a signature chain: instead of requiring the chain to be "a signature on a signature on a signature... etc", we require a signature-chain to be a signature on "a *commitment* of a signature on a commitment of a signature... etc". And next, in the protocol, we let C send commitments to 0 instead of random strings. To be able to establish an analog of the above signature-game claim, we additionally require C to give a zero-knowledge argument of knowledge of the value it committed to before R agrees to sign it.

- Another problem is that A might abort the left execution with some probability p. This means that we might have to rewind the left execution many times (roughly $1/p$ times) before getting the signature we are looking for. As a consequence, the "access pattern" on the right will be a substring of $01\mathsf{id}_1^*\mathsf{id}_2^* \ldots \mathsf{id}_n^*$. To get around this problem, we add an additional slot (and a corresponding signature key). Next, we require that the signature-chain corresponding to the identity id to be with respect to the pattern $2\mathsf{id}_1 2\mathsf{id}_2 2\mathsf{id}_3 \ldots 2\mathsf{id}_n$.

Dealing with non-synchronizing adversaries: As is usually the case, synchronizing adversaries are the "hardest" to deal with. To prove security against non-synchronizing adversaries, we follow basically the same argument: First, if A is not synchronizing there exists some slot that is never rewound and so if the identity of the right interaction contains at least two 0's and two 1's, we can still establish the above key-claim. Next, to argue that the committed value on the right does not change, we consider again the intermediary hybrid above. However, when the adverary is not synchronizing, it may choose to interleave messages in the proof phase of the left interaction and the commitment of the right interaction, and thus the right committed value may change when the witness on the left changes. To overcome this problem, we again rely on the argument of knowledge property of the proof phase to extract a witness from the right intearction. Since the witness cannot be a signature-chain (by the key-claim), it must be the committed value; then the indistinguishablity of the committed vallue follows from the witness-indistinguishability of the left proof phase. However, one problem is that extraction on the right may rewind the left proof phase and thus break the witness indistinguishability property. One way of resolving this problem would be to (in analogy with [PR05a]) have the proof phase be statistically witness indistinguishable; but this requires additional assumptions (to keep it constant-round). Instead, we here rely on an idea from [LP09]: we let the proof phase consist of *multiple sequentially ordered* witness indistinguishable special-sound proofs. This allows us to change the witness in each of the proofs, one by one, while ensuring that the witness on the right can be extracted from some other proof, without rewinding the left proof where the witness currently is being changed.

4 Signature Chains and Games

Let $\Pi = (Gen, Sign, Ver)$ be a fixed-length signature scheme, and com a statistically-binding commitment scheme. For simplicity of notation, we keep these schemes fixed, and provide our definitions and protocols with respect to those particular schemes. Furthermore, for simplicity of exposition, we assume that com that is non-interactive; however, all of our definitions and protocols can be easily modified to work with any two-round statistically-binding commitment schemes; see Remark 1 for further details.

We now turn to formally defining the notion of a *signature-chain* and then proceed to defining *signature-games.*

Definition 3 (Signature-Chain)**.** *Let $\ell \in N$, $\psi \in \{0,1,2\}^\ell$ and $vk_0, vk_1, vk_2 \in \{0,1\}^*$ be three verification keys for the signature scheme Π. We say that a triplet $\delta = (\bar\sigma, \bar c, \bar r)$ is a* signature-chain w.r.t. keys vk_0, vk_1, vk_2 and pattern ψ*, if $\bar\sigma$, $\bar c$, and $\bar r$ are vectors of length ℓ satisfying the following properties.*

- *For all $i \in [\ell]$, $\bar\sigma_i$ is valid signature of the message $\bar c_i$ under key vk_{ψ_i}, i.e., $Ver(vk_{\psi_i}, \bar\sigma_i, \bar c_i) = 1$.*
- *For all $1 < i \leq \ell$, $\bar c_i$ is a commitment to the tuple $(i-1, \bar\sigma_{i-1})$ using* com *and randomness $\bar r_i$; and $\bar c_1$ is a commitment to 0^m using* com *and randomness $\bar r_1$, where $m = \log \ell + n$.*

We say that a signature-chain $\delta = (\bar\sigma, \bar c, \bar r)$ has length ℓ if $|\bar\sigma| = l$.

We proceed to define a signature-game $\mathrm{SG}^{A,\ell}(n,z)$, where A on input $1^n, z$ interacts with a *Challenger* in the following three stages:

Stage 1: the Challenger samples three pairs of signing and verification keys at random, $(sk_b, vk_b) \leftarrow Gen(1^n)$, where $b \in \{0,1,2\}$, and sends A the verification keys, vk_0, vk_1, and vk_2.

Stage 2: A interacts with the Challenger in a sequence of iterations for as long as it wishes. Iteration i proceeds as follows:

- A sends the Challenger a tuple (φ_i, c), where $\varphi_i \in \{0,1,2\}$, followed by a 5-round ZKAOK proof of the statement that c is a valid commitment of com.
- if the proof is convncing, the Challenger signs the commitment c using the signing key s_{φ_i} and returns the signature to A; otherwise, it aborts the iteration (without giving back a signature).

Stage 3: Finally, A outputs the tuple (δ, ψ).

We call the sequence $\varphi = \varphi_1, \varphi_2, \ldots$ of signing request, the "access pattern" of A. We say that the output of A is *well-formed* if δ is a length $l(n)$ signature-chain with respect to vk_0, vk_1, vk_2 and ψ. Finally, we say that A *wins* if its output is well-formed at ψ is not a substring of its access pattern φ (and *looses* otherwise).

Lemma 1. *For every $\mathcal{PPT}$ adversary A and every polynomial ℓ, there exists a negligible function μ, such that for every $n \in N, z \in \{0,1\}^*$, the probability that A wins in $SG^{A,\ell}(n,z)$ is at most $\mu(n)$.*

Proof. Consider any adversary A, polynomial ℓ, $n \in N$, and $z \in \{0,1\}^*$. Without loss of generality, we can assume that A always outputs tuples $(\delta = (\bar{\sigma}, \bar{c}, \bar{r}), \psi)$ such that $|\bar{\sigma}| = |\bar{c}| = |\bar{r}| = \psi| = l(n)$ (since whenever it doesn't it loses). For each $i \in [l(n)]$, define the random variable $\mathcal{I}_i$ to be the index of the *first* iteration (in Stage 2 of the game $\mathrm{SG}^{A,\ell}(n,z)$) in which A queries the Challenger for a signature of the commitment $\bar{c}_i$ under key v_{ψ_i}; if A never queries the Challenger for a signature of $\bar{c}_i$, $\mathcal{I}_i$ is set to $\perp$.

Note that if the output of A is well-formed, it contains a signature-chain $\delta = (\bar{\sigma}, \bar{c}, \bar{r})$ w.r.t. pattern ψ, such that for every i, $\bar{\sigma}_i$ is a valid signature of $\bar{c}_i$ under key v_{ψ_i}. It thus follows from the unforgibility of the signature scheme that, except with negligible probability, for each i, A must have queried $\bar{c}_i$ for a signature of v_{ψ_i} in some iteration. We thus have the following claim.

Claim 1. *For every $\mathcal{PPT}$ adversary A and polynomial ℓ, there exists a negligible function μ_1, such that for all $n \in N, z \in \{0,1\}^*$, the probability that the output of A in $SG^{A,\ell}(n,z)$ is of A is well-formed and there exists an $i \in [\ell(n)]$ such that $\mathcal{I}_i = \perp$, is smaller than $\mu_1(n)$.*

We also have the following claim.

Claim 2. *For every $\mathcal{PPT}$ adversary A and polynomial ℓ, there exists a negligible function μ_2, such that, for all $n \in N, z \in \{0,1\}^*$, the probability that the output of A in $SG^{A,\ell}(n,z)$ is well-formed and there exists an $i \in [\ell(n)-1]$ such that , $\mathcal{I}_i \neq \perp$, $\mathcal{I}_{i+1} \neq \perp$ and $\mathcal{I}_i \geq \mathcal{I}_{i+1}$, is smaller than $\mu_2(n)$.*

Before proceeding to the proof of Claim 2, we let us first prove Lemma 1 using Claim 1 and 2. It follows from the two claims that, except with negligible probability, *either* the output of A is not well-formed, *or* the output is well-formed and for all i, $\mathcal{I}_i \neq \perp$ and $\mathcal{I}_i < \mathcal{I}_{i+1}$. In the former case, the adversary loses the game. In the latter case, as $\mathcal{I}_i \neq \perp$ for all i, A must have asked for a signature using key v_{ψ_i} in the $\mathcal{I}_i{}^{\text{th}}$ iteration, which means $\varphi_{\mathcal{I}_i} = \psi_i$. Furthermore, as $\mathcal{I}_i < \mathcal{I}_{i+1}$ for all i, it follows that ψ is a substring of φ. Therefore, A loses in this case as well. Thus, except with negligible probability, A looses.

Proof of Claim 2. First notice that it follows from the (statistical) binding property of com, that except with negligible probability[3], if the output $(\delta = (\bar{\sigma}, \bar{c}, \bar{r}), \psi)$ of A is well-formed, then for all i, $\bar{c}_i \neq \bar{c}_{i+1}$, since $\bar{c}_i, \bar{c}_{i+1}$ are respectively commitments to tuples of the form $(i, \cdot)$ and $(i+1, \cdot)$. It follows that, except with negligible probability, if the output of A is well-formed, there doesn't exists some i such that $\mathcal{I}_i, \mathcal{I}_{i+1} \neq \perp$ but $\mathcal{I}_i = \mathcal{I}_{i+1}$. Thus, it suffices to bound the probability that the output of A is well formed and there exists some i such that $\mathcal{I}_i, \mathcal{I}_{i+1} \neq \perp$ and $\mathcal{I}_i > \mathcal{I}_{i+1}$.

Towards this, assume for contradiction that there exists an adversary A and a polynomial ℓ, such that there exists a function $i : N \rightarrow N$ and a polynomial p, such that for infinitely many $n \in N, z \in \{0,1\}^*$, the probability that the output of A in the game $\mathrm{SG}^{A,\ell}(n,z)$ is well-formed, $\mathcal{I}_i, \mathcal{I}_{i+1} \neq \perp$, and $\mathcal{I}_i > \mathcal{I}_{i+1}$ for $i = i(n)$, is at least $1/p(n)$. We can construct a machine B that violate the unforgibility of the signature scheme Π.

[3] Since we assume that com is non-interactive, we actually have perfect binding, but given that we want an analysis that works also for two-round commitments, we here directly consider the more general case of statistical binding.

B, on input $1^n, z$ and a randomly generated verification key vk, has access to the signing oracle corresponding to vk, and tries to forge a signature (of vk) as follows: it internally emulates an execution of the signature game $\mathrm{SG}^{A,\ell}(n, z)$ with A honestly, with the following exceptions:

- In Stage 1, it picks an index $t \in \{0, 1, 2\}$ at random and forwards the verification key vk to the adversary as the t^{th} verification key.
- In Stage 2, whenever A requests a signature of a message m under key vk, it obtains such a signature from the signing oracle and forwards it to A.

  Furthermore, it guesses that $\mathcal{I}_i = u$ and $\mathcal{I}_{i+1} = k$, for random $u > k$. Then, in the k^{th} iteration (in Stage 2 of $\mathrm{SG}^{A,\ell}(n, z)$), after receiving a request from A to sign the commitment c, it extracts out the value (j, σ^*) committed to in c from the ZKAOK that A provides following the signing request. Later, in the u^{th} iteration, when A submits a query c^* to the Challenger, it checks whether σ^* is a valid signature of c^* under key vk. If so, it halts and outputs the message-signature pair (c^*, σ^*); otherwise, it halts and outputs fail.

By construction, B emulates the view of A in the signature game $\mathrm{SG}^{A,\ell}(n, z)$ perfectly before it halts. Therefore, by our hypothesis, with probability at least $1/p(n)$, in emulation by B, A would query for the first time the commitments $\bar{c}_i$ and $\bar{c}_{i+1}$ in iterations $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ respectively, such that $\mathcal{I}_{i+1} < \mathcal{I}_i$ and $\bar{c}_{i+1}$ is a commitment to a tuple $(i + 1, \bar{\sigma}_{i+1})$, where $\bar{\sigma}_{i+1}$ is a signature of $\bar{c}_i$ under the verification key v_{ψ_i}. Let $M(n)$ be the maximum number of iterations in the game; M is polynomially bounded since the running-time of A is. With probability at least $\frac{1}{q(n)} = \frac{1}{3M(n)^2 p(n)}$, it holds that (1) the above event occurs in the emulation by B and (2) B correctly guesses the values of $\mathcal{I}_i$, $\mathcal{I}_{i+1}$ and v_{ψ_i}. In this case, except with negligible probability, the committed value σ^* that B extracts out from the ZKAOK following $c = \bar{c}_{i+1}$ contains a valid signature of $\bar{c}_i$, which is queried *for the first time* in the u^{th} iteration for a signature using key vk. Hence B will output a valid message-signature pair $(\bar{c}_i, \sigma^*)$ for vk, without querying the signing oracle $\bar{c}_i$ (since once the query $\bar{c}_i$ is submitted for the first time in iteration u, B halts immediately and outputs the pair); this violates the unforgibility of the signature scheme Π. □

□

5 The Protocol

Let $\Pi = (Gen, Sign, Ver)$ be the fixed-length signature scheme, and com the non-interactive statistically-binding commitment scheme considered in the last section. To simplify the presentation of the proof, we assume that both Π and com can be "broken"—i.e., signatures can be generated for *any* message, and the value committed to can be recovered for *any* commitment—in time $2^{n/2}$ where n is the security parameter; this is without loss of generality since we can always appropriately "scale-down" the security parameter in Π and com (and make sure that com commits to values "bit-by-bit"). To further simplify the presentation, we provide the construction of a non-malleable commitment $\langle C, R \rangle$ that works assuming player identities are ℓ-bit binary strings that contains at least two 0-bits and two 1-bits; any such scheme can trivially be turned into one that works for arbitrary identities (by simply appending two 0's and two 1's to the identity).

To commit to a value v, the Committer and the Receiver of $\langle C, R \rangle$, on common input a security parameter 1^n (in unary) and an identity $\mathsf{id} \in D^\ell$, proceed in the following three stages:

Stage 1: The receiver interacts with the Committer in three iterations, where iteration $i \in \{0, 1, 2\}$ proceeds in the following steps:

1. The Receiver generates a pair of signing and verification keys, $(s_i, v_i) \leftarrow Gen(1^n)$, of the signature scheme Π, and sends the verification key v_i.
2. The Committer commits to 0^m, where $m = \log \ell + n$, using com. Let c_i be the commitment sent to the Receiver.
3. The Committer proves that c_i is a valid com commitment using a 5-round ZKAOK protocol.
4. The Receiver signs the commitment c_i using the signing key s_i, and sends the generated signature θ_i to the Committer.

Stage 2: The Committer commits to the value v using com. Let c' be the commitment generated.

Stage 3: The Committer proves that

- *either* c' is a valid com commitment,
- *or* there exists a signature-chain δ w.r.t. v_0, v_1, v_2 and pattern $\mathsf{pattern}(\mathsf{id})$, where the function $\mathsf{pattern} : \{0,1\}^* \to \{0,1,2\}^*$ maps a (binary) identity id of length ℓ to a trinary string of length 2ℓ as follows:

$$\mathsf{pattern}(\mathsf{id}) = 2, \mathsf{id}_1, 2, \ldots, \mathsf{id}_i, 2, \ldots, \mathsf{id}_\ell$$

This statement is proved using $k+5$ sequential invocations of a 4-round $\mathcal{WI}$ special sound proof system, where k is the number of messages in Stage 1 of the protocol; we additionally require that the length of the "challenge" in each special-sound proof is n.

We refer to the last three steps of an iteration in Stage 1 as a slot, which *opens* when the Committer send the com commitment to 0^m, and *closes* when the Receiver returns a signature to the commitment. We call the slot in iteration i, the i'th slot.

It is easy to see that the protocol $\langle C, R\rangle$ consists of a constant number of messages. Furthermore, it follows using standard techniques that $\langle C, R\rangle$ is a valid commitment scheme.

Proposition 2. *$\langle C, R\rangle$ is a commitment scheme.*

Proof. We show that the $\langle C, R\rangle$ scheme satisfies the binding and hiding properties.

Binding: The binding property follows directly from the statistically binding property of com used in Stage 2.

Hiding: The hiding property essentially follows from the hiding property of com and the fact that Stage 3 of the protocol is $\mathcal{WI}$ (since $\mathcal{WI}$ proofs are closed under concurrent composition [FS90]). For completeness, we provide the proof. We show that any adversary R^* that violates the hiding property of $\langle C, R\rangle$ can be used to violate the hiding property of com. More precisely, given any adversary R^*, such that, for infinitely many $n \in N$, and $v_1, v_2 \in \{0,1\}^n$, R^* distinguishes commitments to v_1 and v_2 made using $\langle C, R\rangle$, we construct a machine R' that distinguishes commitments to v_1 and v_2 made using com. Note that the execution of a commitment of $\langle C, R\rangle$ to v_1 proceeds identically as that of a commitment to v_2 before the Stage 2 commitment of com is sent. Then by our hypothesis, there must exist a partial joint view ρ of the committer and R^* that determines the execution of the commitment before Stage 2, such that, conditioned on ρ occurring, R^* distinguishes commitments to v_1 and v_2. Let δ be a valid signature-chain corresponding to the transcript of Stage 1 in ρ. R' on auxiliary input ρ and δ proceeds as follows: it internally incorporates R^*, and feed R^* its part of view in ρ; it then forwards the external commitment made using com to R^* in Stage 2;

in Stage 3, it gives $\mathcal{WI}$ proofs using δ as a "fake witness". Finally, it outputs whatever R^* outputs. From the $\mathcal{WI}$ property of Stage 3, it follows that R' distinguishes the commitment made using com, if R^* distinguishes the commitment made using $\langle C, R\rangle$ conditioned on ρ occurring.

□

Remark 1. *Both the definition of of signature-games and our non-malleable commitment protocols makes use of a* non-interactive *statistically-binding commitment scheme* com*. Both can be easily modified to work also with any two-round statistically binding commitment schemes* $\overline{\mathsf{com}}$*. In both cases, we the first message r of a commitment of* $\overline{\mathsf{com}}$ *is sent at the beginning of the execution, and then the rest of the execution proceeds just as if* $\overline{\mathsf{com}}$ *had been non-interactive. (Additionally, in the last stage of the protocol $\langle C, R\rangle$, the sender proves that either the Stage 2 message is the second message of a valid* $\overline{\mathsf{com}}$ *commitment with first message r, or it knows a signature-chain $\delta = (\bar{\sigma}, \bar{c}, \bar{r})$, such that, δ is well-formed, except that, for all i, $\bar{c}_i$ is the second message of a* $\overline{\mathsf{com}}$ *commitment to $\bar{\sigma}_{i-1}$, generated in responding to the first message r using randomness $\bar{r}_i$). Exactly the same proof as in Section 4 and Section 6 still go through using these modified construction, since commitments of* $\overline{\mathsf{com}}$ *are hiding, no matter what the first message is, and even if the first message is reused.*

6 Proof of Non-malleability

In this section, we show that $\langle C, R\rangle$ is stand-alone non-malleable. In Sections 8 and 7, we extend the proof to show that $\langle C, R\rangle$ is also robust and concurrent non-malleable.

Theorem 5. *$\langle C, R\rangle$ is (one-one) non-malleable.*

Proof. The goal is to show that for every one-one man-in-the-middle adversary A that participates in one left and one right execution, the following ensembles are indistinguishable:

$$\left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(v_1, z)\right\}_{n\in N, v_1, v_2\in\{0,1\}^n, z\in\{0,1\}^*}$$

$$\left\{\mathsf{mim}^{A}_{\langle C,R\rangle}(v_2, z)\right\}_{n\in N, v_1, v_2\in\{0,1\}^n, z\in\{0,1\}^*}$$

Towards this, we define a series of hybrid experiments $H_0, \dots, H_{k+6}$. In each of these experiments, we show that the view of A, combined with the value that A commits to on the right, are indistinguishable. Let $\mathsf{hyb}_i(v, z)$ denote the random variable describing the view of $A(z)$, combined with the value it commits to in the right interaction in hybrid H_i (as usual, the committed value is replaced with $\perp$ if the right interaction fails or if A has copied the identity of the left interaction).

Hybrid H_0: In H_0 we first perfectly emulate a real execution of $\mathsf{mim}^{A}_{\langle C,R\rangle}(v, z)$—we call this the *Main Execution*—and next, if A successfully completed Stage 1 in the Main Execution, we try extract a "fake-witnesses" (i.e., a signature-chain) for the left interaction. More precisely, let id_l, v_0, v_1, v_2, respectively be the identity and the verification keys of the left interaction in the Main Execution, and let $\psi = \mathsf{pattern}(\mathsf{id}_l)$; the *Extraction Procedure* now proceeds in $|\psi| = 2\ell$ iterations described below.

Iteration 1: If A successfully completes Stage 1 of the left interaction in the Main Execution, it must have provided three valid signatures $\theta_0, \theta_1, \theta_2$ of commitments to 0^m, where $m = \log \ell + n$, under keys v_0, v_1, v_2 respectively. Since a signature-chain with pattern ψ starts off with a signature $\bar{\sigma}_1$ of a commitment to 0^m under key $v_{\psi_1} = v_2$, the procedure simply sets $\bar{\sigma}_1 = \theta_2$, $\bar{c}_1$ to be the transcript of the commitment to 0^m generated in iteration 2 (in Stage 1) of the left interaction, and $\bar{r}_1$ to be the randomness used in the commitment.

Iteration $i+1$: Assume that at the end of the i^{th} iteration, for $i \in [2\ell - 1]$, the procedure has obtained a signature-chain δ_i of length i w.r.t. (keys v_0, v_1, v_2 and) pattern $[\psi]_1^i$, containing signatures $\bar{\sigma}_1, \ldots, \bar{\sigma}_i$. Then, in iteration $i+1$, we obtain a signature-chain δ_{i+1} of length $i+1$, w.r.t. pattern $[\psi]_1^{i+1}$ by rewinding the appropriate slot in Stage 1 of the left interaction. More precisely, the procedure repeatedly rewinds A from where the slot ψ_{i+1} opens on the left in the Main Execution, and commits to the tuple $(i, \bar{\sigma}_i)$ (instead of 0^m) in the rewindings, until this left-slot closes successfully (i.e., A returns a valid signature on the commitment under key $v_{\psi_{i+1}}$). In each of these rewindings, the right executions are emulated using fresh randomness; in particular, this means that whenever a rewinding goes beyond the point when a verification key is sent in the right interaction, in each such rewinding a fresh verification key is picked. Then the extraction procedure simply sets $\bar{\sigma}_{i+1}$ to be this signature, and again sets $\bar{c}_{i+1}$ and $\bar{r}_{i+1}$ to be the commitment and randomness used.

If the extraction procedure takes more than $2^{n/2}$ steps, it is "cut-off"; in this case, a signature chain can be recovered in time $\text{poly}(2^{n/2})$ by our assumption on the signature scheme Π. The extraction procedure thus always terminates and always recovers a valid signature chain for the left interaction.

Since the view of A in the Main Execution in H_0 is perfectly emulated as in $\mathsf{mim}^A_{\langle C,R\rangle}(v,z)$, we trivially have that the view and value A commits to in H_0 is identically distributed to that in the real execution.

Claim 3. *For every $\mathcal{PPT}$ adversary A, it holds that:*

$$\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*} = \left\{\mathsf{hyb}_0(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*}$$

Hybrid H_1 to H_{k+5}: In hybrids H_1 to H_{k+5}, we change the witness used in the $k+5$ $\mathcal{WISSP}$ proofs in Stage 3 of the left interaction. More specifically, experiment H_i proceeds identically to H_{i-1}, except that in the first i proofs in Stage 3 of the left interaction, we prove that there exists a signature-chain w.r.t. v_0, v_1, v_2 and pattern $\mathsf{pattern}(\mathsf{id}_l)$, by using the extracted signature-chain δ as a "fake-witness". We show that the view and value committed to on the right interaction in H_{i-1} and H_i are indistinguishable.

Proposition 3. *For every $\mathcal{PPT}$ adversary A, and every function $i : N \to N$, it holds that:*

$$\left\{\mathsf{hyb}_{i(n)-1}(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{hyb}_{i(n)}(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*}$$

Towards this, we reduce the indistinguishability of $\{\mathsf{hyb}_{i(n)-1}(v,z)\}$ and $\{\mathsf{hyb}_{i(n)}(v,z)\}$ to the witness indistinguishabilty of the Stage 3. More specifically, consider some adversary A, a function i, and a polynomial p, such that (for infinitely many $n \in N$, inputs $v \in \{0,1\}^n$ and $z \in \{0,1\}^*$,) $\mathsf{hyb}^{i(n)-1}(v,z)$ and $\mathsf{hyb}^i(n)(v,z)$ are distinguishable with probability $1/p(n)$. We show that there exists a $\mathcal{PPT}$ machine B that can violate the $\mathcal{WI}$ property of the $\mathcal{WISSP}$ protocol $\langle P, V\rangle$ used in Stage 3 of the protocol.

On a high-level, the machine B, on common input 1^n and auxiliary input v, z, externally interacts with an honest prover P and receives a left-interaction Stage 3 proof, generated

Description of B

Input: B receives a security parameter 1^n and v and z as auxiliary input.

Procedure: B externally interacts with a prover P of the $\mathcal{WISSP}$ protocol $\langle P, V\rangle$, receiving a proof of a statment x using witness w_0 or w_1, where x, w_0 and w_1 are chosen by B. Internally, it proceeds in the following three phases:

Simulation Phase: B internally emulates an execution of the experiment $\mathsf{hyb}_i(v, z)$ with A, with the exception that messages in the i^{th} left-proof of the Main Execution are forwarded externally to P. More precisely, at the beginning of the i^{th} left-proof, B sends the external prover P the statement x of the i^{th} proof, together with the "real witness" $w_0 = (v, r)$ (the decommitment of the Stage 2 commitment of the left interaction) and the "fake witness" $w_1 = \delta$ (the signature-chain of the left interaction extracted from A); B next forwards the proof of x generated by P (using either w_0 or w_1) to A as the i^{th} left-proof. Let Δ be the simulated view of A in the Main Execution.

Rewinding Phase: If the right interaction is successful and has a different identity from the left interaction in Δ, B extracts the value committed to in this interaction as follow:

- Find the first $\mathcal{WISSP}$ proof $(\alpha_1, \alpha_2, \beta, \gamma)$ in Δ, such that, during its the execution, no messages belonging to Stage 1 or the i^{th} proof of the left interaction are exchanged. (Such a $\mathcal{WISSP}$ proof must exist since there are $k + 5$ $\mathcal{WISSP}$ proofs, whereas only $k + 4$ messages in Stage 1 and the i^{th} proof of the left interaction.)
- Rewinds the proof by sending new random challenges β' until a second transcript $(\alpha_1, \alpha_2, \beta', \gamma')$ is obtained.

  In the rewindings, emulate the left and right interaction for A in identically the same way as in the Main Execution, except that, whenever A expects a new message in Stage 1 or the i^{th} proof of the left interaction, cancel the execution and start a new rewinding again.
- If $\beta_\rho \neq \beta'_\rho$, extract witness w from $(\alpha_1, \alpha_2, \beta, \gamma)$ and $(\alpha_1, \alpha_2, \beta', \gamma')$. Otherwise halt and output fail_1.
- If $w = (v, r)$ is valid decommitment for the right interaction, then set $\hat{v} = v$. Otherwise halt and output fail_2.

Output Phase: If the right interaction that is not convincing or the identity of the right interaction is the same as the left interaction, set $\hat{v} = \perp$. Output $\hat{v}$ and Δ.

Figure 4: The construction of B

using either the real witness w_0—the decommitment of the Stage 2 commiment in the left interaction—or the fake witness w_1—a signature chain for the left intearction. Internally, B emulates an execution of either hyb^{i-1} or hyb^{i} with A (depending on the witness used in the external proof), except that, messages in the i^{th} proof in Stage 3 of the left interactions are fowarded externally. Furthermore, if the right interaction is successful and has a different identity from the left, B attempts to extract the value committed to on the right by repeatedly rewinding the $\mathcal{WISSP}$ proofs in Stage 3 of the right inteaction by sending new challenge messages in this proof. Since the i^{th} left-proof is forwarded externally, the rewinding has to be done in a manner that does not "affect" the i^{th} left-proof. Roughly speaking, this is possible since there are more $\mathcal{WISSP}$ proofs in Stage 3 of the right interaction, than the number of messages in the i^{th} left-proof. Therefore, in the right interaction, there exist some $\mathcal{WISSP}$ proofs that does not interleave with any messages in the i^{th} left-proof, and B can use rewindings to extract a witness *without rewinding* the left-proof. Our actual rewinding strategy also avoids rewinding Stage 1 of the left interaction, so that the fake-witness δ of the left interaction remains a valid signature chain also in the rewindings, and thus can be reused to simulate the left interaction also in the rewindings. This is again possible since there are more right-proofs than the number of messages in Stage 1 and the i^{th} proof in the left interaction. To slightly simplify the analysis, we additionally "cut-off" the rewindings if B takes more than $2^{n/2}$ steps and simply recover the value committed to in time $\text{poly}(2^{n/2})$; recall that this is possible due to our assumption on com.

If during the rewindings, B sends the same challenge message twice, it aborts outputting fail_1. Additionally, if the witness extracted from the right interaction is not a valid decommitment (it could also be a fake-witness), B aborts outputting fail_2. Otherwise, B outputs the emulated view of A, together with the value committed to in the right interaction.,

See Figure 4 for a formal description of B. Below, in Lemma 2, we show that the running-time of machine B is "bounded", in the sense that the probability that B runs for super-polynomial time is negligible.

Lemma 2. *There exists a polynomial function T, such that for every polynomial function q, every $b \in \{0,1\}$, every sufficiently large $n \in N$, and inputs $v \in \{0,1\}^n$ and $z \in \{0,1\}^*$, the probability that machine B runs for more than $q(n)T(n)$ steps in an execution of the experiment $\mathsf{STA}_b(\langle P, V\rangle, B, v, z)$ is smaller than $1/q(n)$.*

Roughly speaking, the Lemma is proven by first bounding the running-time of a "hypothetical procedure" which perform all the same rewindings, but otherwise acts honestly (i.e., always commits to 0^m in Stage 1, and always uses the honest witness in Stage 3); it follows using a simple "$p \times 1/p$" argument (similar to those in [LPV08, LP09]) that the expected running-time of this procedure is polynomial. Next we show that, with high probability, the running-time of the actual procedure is not too far off. We note that due to reasons similar to those in [GK96] we are not able to bound the expected running-time of B. Additionally, it seems unclear if the methods of [GK96] could be applicable to obtains a simulation with an expected polynomial running-time. Fortunately, in our application, since we do not actually per se care about the running time of the simulation (but only care about breaking some specific security property, namely witness indistinguishability) our weaker bound suffices.

A formal proof of Lemma 2 can be found in Section 6.1.

The following lemma is the core of our analysis.

Lemma 3. *The following holds.*

$$\{\mathsf{STA}_0(\langle P, V\rangle, B, v, z)\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{hyb}^{i-1}(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*}$$
$$\{\mathsf{STA}_1(\langle P, V\rangle, B, v, z)\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{hyb}^{i}(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*}$$

Before proceeding to the proof of 3, let us see how Lemma 3 and 2 together violate the $\mathcal{WI}$ property of the Stage 3 proofs. Recall that by our assumption, $\mathsf{hyb}^i(v,z)$ and $\mathsf{hyb}^i - 1(v,z)$ can be distinguished with probability $1/p(n)$; by Lemma 3, $\mathsf{STA}_0(\langle P, V\rangle, B, v, z)$ and $\mathsf{STA}_0(\langle P, V\rangle, B, v, z)$ can thus be distinguished with probability at least, say, $3/4p(n)$. By Lemma 2, the probability that B runs for more than, say, $4p(n)T(n)$ steps in either experiment is at most $1/4p(n)$. Therefore, by the union bound, the outputs of B (in STA_0 and STA_1) are still distinguishable with probability at least $1/4p(n)$, even if we cut-off the execution of B after $4p(n)T(n)$ steps (and output $\perp$ if B fails to complete), which is a contradiction.

Let us now turn to proving Lemma 3.

Proof. (of Lemma 3) By construction, B perfectly emulates the view of A in $\mathsf{hyb}^{i-1}(v,z)$ when receiving an external proof generated using the real witness w_0, and that in $\mathsf{hyb}^i(v,z)$ when receiving a proof generated using the fake witness w_1. Therefore, to show Lemma 3, it suffices to show that B (almost) always extracts a valid decommitment for the right interaction if it is successful and has a different identity from the left interaction (recall that by statistical binding of $\langle C, R\rangle$, the committed value is unique with overwhelming probability). In other words, showing Lemma 3 amounts to showing that the probability that B outputs fail_1 or fail_2 is negligible.

Claim 4. *There exists a negligible function μ, such that for every $b \in \{0,1\}$, every sufficiently large $n \in N$, and inputs $v \in \{0,1\}^n$ and $z \in \{0,1\}^*$, the probability that B outputs fail_1 in $\mathsf{STA}_b(\langle P, V\rangle, B, v, z)$ is smaller than $\mu(n)$.*

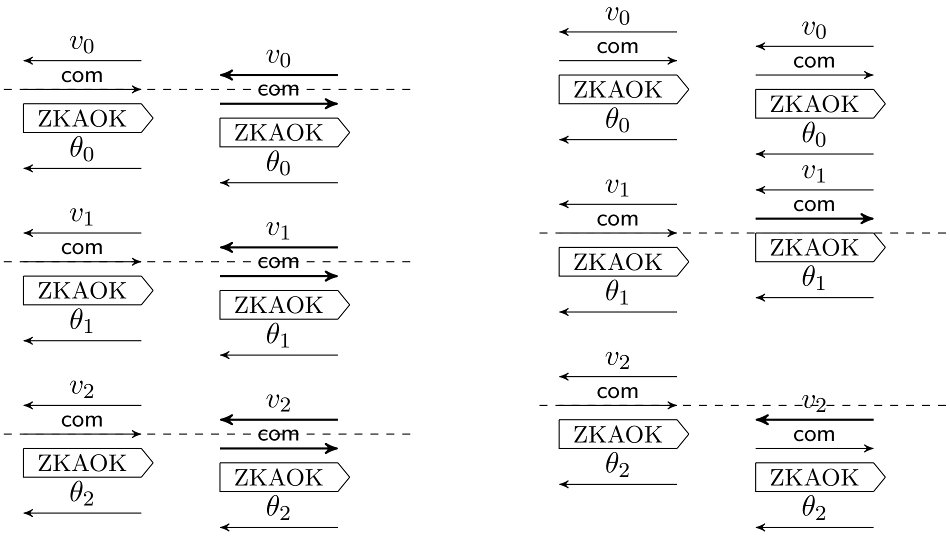
Proof. Recall that B outputs fail_1 only if in some rewinding it picks the same challenge β' as the challenge β used in the same proof in the Main Execution. Since the number of rewindings by B is bounded by $2^{n/2}$ and the length of each challenge is n, by the union bound, the probability that this happens is negligble. □

Claim 5. *There exists a negligible function μ, such that, for every $b \in \{0,1\}$, every sufficiently large $n \in N$, and inputs $v \in \{0,1\}^n$ and $z \in \{0,1\}^*$, the probability that B outputs fail_2 in $\mathsf{STA}_b(\langle P, V\rangle, B, v, z)$ is smaller than $\mu(n)$.*

Proof. Assume for contradiction that there exists a polynomial $g(n)$, such that, with probability $1/g(n)$, B extracts an invalid decommitment from the right interaction. Towards reaching a contradiction, we consider another machine B', which proceeds identically to B except that it cuts-off the execution after $g(n)T(n)$ steps (and outputs $\perp$ in this case). It follows from Lemma 2 that the probabilty that B runs for more than $g(n)T(n)$ steps is at most $1/2g(n)$. Therefore, the probability that B' extracts out an invalid decommitment from the right interaction k is at least $1/2g(n)$. Furthermore, by the special-soundness property of the right-proofs, if the witness is not a valid decommitment, it must be a signature-chain δ w.r.t. the right-interaction keys v'_0, v'_1, v'_2 and pattern $\mathsf{pattern}(\mathsf{id}_r)$. Consider the following two possible adversarial schedulings w.r.t. the left and the k^{th} right interactions in the Main Execution:

Scheduling 1: A "aligns" the slots in the left and right interactions *one by one*: a right-slot is said to be *aligned* with a left-slot if (1) its corresponding verification key is sent before the left-slot opens, and (2) its opening message (i.e., the commitment from A) is sent after the left-slot opens; see Figure 5 (i).

Scheduling 2: A does not align the slots in the left and right interactions; see Figure 5 (ii). This means that there exists some right-interaction slot that is not aligned with *any* left-interaction slot.



(i) Scheduling 1

(ii) Scheduling 2

Figure 5: The two schedulings of the messages in Stage 1 of the left and right interactions.

Since Scheduling 1 and 2 are the only two possible schedulings, by our hypothesis, at least one of the following two conditions holds.

Condition 1: The probabilty that Scheduling 1 occurs in the Main Execution and that B' extracts an invalid decommitment from the right interaction is non-negligible.

Condition 2: The probabilty that Scheduling 2 occurs in the Main Execution and that B' extracts an invalid decommitment from the right interaction is non-negligible.

We show that neither condition can hold.

Assume Condition 1 holds. We reach a contradiction by constructing a machine C that externally participates in the signature game, while internally emulating an execution of $\mathsf{STA}_b(\langle P, V\rangle, B', v, z)$ except that messages in Stage 1 of the right interaction are emulated by fowarding the appropriate messages from the signature games to A. More precisely, C forwards the three verification keys vk_0, vk_1, vk_2 in the signature game to A as the verification keys in Stage 1 of the right interaction in the Main Execution. If Schedule 1 does not occur in the Main Execution, C simply aborts. Otherwise, whenever during some rewinding, A requests another signature in one of the slots on the Main Execution (and thus using one of vk_0, vk_1, vk_2), C obtains such a signature by accessing the appropriate signature oracle in the game and forwards it to A. Recall that whenever we rewind beyond the point where a verification key is sent, a new verification key is generated by B and thus B can obtain the appropriate signatures without querying the oracle. Since the left and right slots in the Main Execution are aligned one by one, we have that whenever the t^{th} left-slot is rewound, the adversary A may only request new signatures using key v'_t on the right. It follows that the "access-pattern" of the signatures requested is a substring of

$$\varphi = 012\|(\mathsf{id}_l)_1^*, 2^*, \ldots, (\mathsf{id}_l)_i^*, 2^*, \ldots, (id_l)_\ell^*$$

So, whenever B' extracts out a signature-chain δ w.r.t. (keys v'_0, v'_1, v'_2 and pattern $\mathsf{pattern}(\mathsf{id}_r)$), C wins in the signature game since $\mathsf{pattern}(\mathsf{id}_r)$ is not a substring of φ (as $\mathsf{id}_r \neq \mathsf{id}_l$). Since the running-time of C is polynomial this contradicts Lemma 1.

Assume Condition 2 holds. We construct a machine C' just as in the previous case, except that C' abort whenever Schedule 2 does not happen in the Main Execution. When Scheduling 2 does occurs in the Main Execution, there exists a right-slot t that is not aligned with any left-slots; in other words, in all the rewindings where A gets to request a new signature in Slot t on the right, the rewinding goes beyond the point where the verification key for slot t is sent (and so new keys gets generated in each rewinding) and thus the t'th oracle is *never* used during the extraction phase. It follows that the access pattern in the signature game has a single character t, but the signature extracted is with respect to a pattern with two of each character. So, as in Condition 1, whenever B' extracts out a signature-chain δ, C' wins in the signature game. There is just one slight complication with the implementation of C': in the rewindings, B might rewind A in the middle of one of the ZKAOK in Stage 1, and since the ZKAOKs are not public-coin, we might not be able to emulate the continuation of the verifier strategy for this protocol. Note, however, that Lemma 1 still holds even if consider a slight variant of the signature game where after each ZKAOK the verifier reveals all of its random coins; this follows since this adjusted protocol would still be an ZKAOK and Lemma 1 no mayyer what ZKAOK we use in the signature game. C' can now easily be implemented as an adversary for this modified signature game.

□

□

Hybrid H_{k+6} : Hybrid H_{k+6} proceeds identically to H_{k+5} except that the Stage 2 commitment of the left execution is emulated by committing to 0^n. It follows using the same argument as in hybrids H_i, for $i \in [k+5]$, that the value committed in the right interaction can be extracted without rewinding Stage 2 of the left interaction. It then follows from the hiding property of the Stage 2 commitment that the combined view and values committed to by A in H_{k+5} are indistinguishable from that in H_{k+6}.

It follows by a hybrid argument that,

$$\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{hyb}_{k+6}(v,z)\right\}_{n\in N, v\in\{0,1\}^n, z\in\{0,1\}^*}$$

Since the above holds for every value v, we have

$$\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v_1,z)\right\}_{n\in N, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{hyb}_{k+6}(v_1,z)\right\}_{n\in N, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*}, \text{ and}$$

$$\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v_2,z)\right\}_{n\in N, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{hyb}_{k+6}(v_2,z)\right\}_{n\in N, v_1,v_2\in\{0,1\}^n, z\in\{0,1\}^*}$$

Finally, since by the definition of hyb_{k+6}, it holds that for every v_1, v_2 and z, $\mathsf{hyb}_{k+6}(v_1, z) = \mathsf{hyb}_{k+6}(v_2, z)$, we conclude that,

$$\left\{\mathsf{mim}^A_{\langle C,R\rangle}(v_1, z)\right\}_{n\in N, v_1, v_2\in\{0,1\}^n, z\in\{0,1\}^*} \approx \left\{\mathsf{mim}^A_{\langle C,R\rangle}(v_2, z)\right\}_{n\in N, v_1, v_2\in\{0,1\}^n, z\in\{0,1\}^*}$$

□

6.1 Proof of Lemma 2

Proof of Lemma 2. The running-time of B consists of three parts:

Part 1—Time spent simulating the Main Execution: Since A runs in strict polynomial time, the time $T_1(n)$ that B spends in the Main Execution polynomially bounded.

Part 2—Time spent extracting the left "fake-witness": We show that there exists a polynomial $T_2(n)$ such that for every polynomial q_2, (every $b \in \{0, 1\}$, every sufficiently large $n \in N$, and inputs $v \in \{0, 1\}^n$, $z \in \{0, 1\}^*$,) the probability that B spends more than $q_2(n)T_2(n)$ steps extracting the left "fake-witness" in $\mathsf{STA}_b(\langle P, V\rangle, B, v, z)$ is smaller than $1/q_2(n)$.

Part 3—Time spent extracting the committed value on the right: We show that there exists a polynomial $T_3(n)$ such that for every polynomial q_3, the probability that B spends more than $q_3(n)T_3(n)$ steps extracting the right committed value in $\mathsf{STA}_b(\langle P, V\rangle, B, v, z)$ is smaller than $1/q_3(n)$.

So given an arbitrary polynomial q, we get by the union bound that, the probability B spends more than $2q(n)T_2(n)$ step in part 2, or more than $2q(n)T_3(n)$ steps in part 3, is smaller than $1/q(n)$. We conclude that there exists some sufficiently big polynomial $T(n) \geq T_1(n) + 2T_2(n) + 2T_3(n)$ such that for every polynomial q, the probability that B takes more than $q(n)T(n)$ steps is smaller than $1/q(n)$.

Analysis of Part 2: Recall that in an execution of $\mathsf{STA}_b(\langle P, V\rangle, B, v, z)$, the extraction of the left "fake-witness" proceeds in 2ℓ iterations. The running-time of the first iteration is clearly polynomial; we proceed to analyze the time spent in the remainder of the iterations. Recall that in an iteration $i > 1$, B takes the signature-chain $\delta_{i-1} = ([\bar{\sigma}]_1^{i-1}, [\bar{c}]_1^{i-1}, [\bar{r}]_1^{i-1})$ of length $i-1$, w.r.t. (keys v_0, v_1, v_2 and) pattern $[\psi]_1^{i-1}$, (where $\psi = \mathsf{pattern}(\mathsf{id}_l)$,) obtained in the previous iteration, and extends it to a signature-chain $\bar{\sigma}_i$ of length i w.r.t. pattern $[\psi]_1^i$. This is done by repeatedly rewinding A from the start of the left-slot ψ_i and committing to $(i-1, \bar{\sigma}_{i-1})$ in the rewindings, until A closes this left-slot successfully. (Below we assume for simplicity that the extraction procedure is never cut-off and may run for more than $2^{n/2}$ steps, since this only increases the running time). Towards bounding the running-time of this extraction procedure, we first consider a *hypothetical procedure*, which proceeds almost the same as the actual extraction procedure, except that in the rewindings in iteration $i > 1$, instead of committing to $(i-1, \bar{\sigma}_{i-1})$, it commits to 0^m. In other words, the hypothetical procedure simulates the view of A in the rewindings using identically the same distribution as in the Main Execution. We show that the expected running-time of this hypothetical procedure is $poly(n)$; we next bound the running-time of the actual extraction procedure.

Running-time Analysis of the Hypothetical Procedure: Let $\psi = \mathsf{pattern}(\mathsf{id}_l)$ be the pattern of the "fake-witness" of the left intearction. In iteration $i > 1$, the hypothetical extraction procedure repeatedly rewinds the left-slot ψ_i; let $\mathcal{T}^i$ be the random variable that describes the time spent in rewinding the left-slot ψ_i in iteration $i > 1$. We show that $E[T^i] \leq poly(n)$ and then by linearity of expectation, we conclude that the expected running-time of the hypothetical procedure is

$$\sum_{i=2}^{2\ell} E[T^i] \leq \sum_{i=2}^{2\ell} poly(n) \leq poly(n),$$

since the number of iterations is $poly(n)$.

Let us turn to bounding $E[T^i]$. Let Γ_{ψ_i} denote the set of prefixes ρ—i.e., partial transcripts of the Main Execution—from where the left-slot ψ_i opens. Given a prefix $\rho \in \Gamma_{\psi_i}$, we introduce the following notations:

- let $\Pr[\rho]$ denote the probability that ρ occurs as a prefix in the Main Execution;
- let p_ρ denote the probability that, conditioned on the prefix ρ occurring (in the Main Execution), the left-slot ψ_i closes successfully in the Main Execution.

Take any ρ from Γ_{ψ_i}. We claim that conditioned on ρ occurring, the expected value of T^i—denoted $E[T^i|\rho]$—is $poly(n)$. This follows since, first, the hypothetical procedure starts rewinding the left-slot ψ_i in iteration i only if this slot closes successfully in the Main Execution; hence, (conditioned on ρ occurring,) the probability the left-slot ψ_i is rewound is at most p_ρ. Secondly, once it starts rewinding the left-slot ψ_i, it continues until the slot closes successfully again; since the hypothetical procedure proceeds identically in the rewindings as in the Main Execution, the probability that the left-slot ψ_i closes successfully in any rewinding is also p_ρ, and thus, (conditioned on ρ occurring,) the expected number of rewindings performed before this happens is $1/p_\rho$. Therefore, the overall expected number of rewindings from ρ is $p_\rho \times \frac{1}{p_\rho} = 1$. As each rewinding takes at most $poly(n)$ steps, we conclude that $E[T^i|\rho] \leq poly(n)$. Thus,

$$E[T^i] = \sum_{\rho \in \Gamma_{\psi_i}} E[T^i|\rho] \Pr[\rho] \ \leq \ poly(n) \times \sum_{\rho \in \Gamma_{\psi_i}} \Pr[\rho] \leq \ poly(n)$$

Running-time Analysis of the Actual Extraction Procedure: Given that the expected running time of the hypothetical procedure is bounded by a polynomial $\tilde{T}(n)$, it follows using the Markov inequality that, for every polynomial q_2, (every b, every $n \in N$, and inputs v, z,) the probability that the hypothetical procedure takes more than $q_2(n)\tilde{T}(n)/2$ steps is smaller than $2/q_2(n)$. Then we claim that the probability that actual extraction procedure takes more than $q_2(n)\tilde{T}(n)/2$ steps is smaller than $1/q_2(n)$. This follows since the only difference between the hypothetical and the actual extraction procedures is that, in the former the rewindings are simulated by committing to 0^m using com, whereas in the latter rewindings are simulated by committing to a tuple that contains a signature. Since the ZKAOK proof following the commitment is never rewound, it follows directly from the hiding property of com and the zero knowledge property of the ZKAOK proof that, the probability that the actual extraction procedure runs for more than $q_2(n)\tilde{T}(n)/2$ steps differs from that of the hypothetical procedure by at most a negligible amount. Thus, for sufficiently large n, we have that the probablity B spends more than $T_2(n) = q_2(n)\tilde{T}(n)/2$ steps is smaller than $1/q_2(n)$.

Analysis of Part 3: We show that the time that B spends in the Rewinding Phase is bounded by a polynomial $T_3(n)$ in expectation. It then follows by the Markov inequality that, for every polynomial q_3, the probability that B takes more than $q_3(n)T_3(n)$ steps is smaller than $1/q_3(n)$.

It follows from the same argument as in the above “running-time analysis of the hypothetical procedure” that to bound the expected time spent extracting the right committed value (also here, we consider the running-time without cut-offs), it suffices to bound the expected time spent in rewinding each right $\mathcal{WISSP}$ proof, since the total number of right-proofs is $poly(n)$. Then recall that a right-proof is rewound only if the proof completes successfully in the Main Execution, without interleaving with any message in Stage 1 or the i^{th} proof of the left interaction. On the other hand, once the rewinding starts, it continues until this right-proof completes successfully again, while cancelling every rewinding in which the proof interleaves with any message in Stage 1 or the i^{th} proof of the left interaction. Furthermore, as every rewinding is simulated exactly the same as in the Main Execution, it follows using the same "p times $1/p$ argument" as in the analysis of part 2 that the expected number of rewindings for every right-proof is 1, and hence the expected time spent in extracting the right committed value is bounded by a polynomial $T_3(n)$.

□

7 Proof of Concurrent Non-Malleability

As we show in the next section $\langle C, R\rangle$ is 4-robust; thus, we can already conclude the existence of a constant-round concurrent non-malleable commitment, by relying on the work of [LP09], which shows how any stand-alone non-malleable commitment that is 4-robust can be transformed into a concurrently non-malleable (and 4-robust) scheme, while incurring only a constant (additive) overhead in round complexity.

We here show that $\langle C, R\rangle$, in fact, directly is also concurrently non-malleable. Recall that ny Proposition 1, to show concurrent non-malleability, it suffices to prove that $\langle C, R\rangle$ is one-many non-malleable; that is, for every one-many man-in-the-middle adversary A, that participates in one left and many right interactions, the view of A and the values it commits to on the right are indistinguishable, no matter what value it is receiving a commitment to on the left. Towards this, we consider the same hybrid experiments H_0 to H_{k+6} as in the proof of stand-alone non-malleability. It follows from almost the same proof as before that the view of A and the values it commits to on the right are indisitnguishable in sequential hybrids, except that, in hybrids H_1 to H_{k+6}, we (or more precisely, the simulator B) now need to extract out the values that A commits to in *all* the right interactions (recall that the proof relies on the fact that the value that A commits to in the right interaction can be extracted "efficiently", to show the indistinguishability of hybrid H_i and H_{i+1} for $1 \leq i \leq k+6$). This is easy to achieve, since we can simply extract the values that A commits to in each right interaction *one by one*, after the Main Execution completes. More precisely, in the Rewinding Phase, for *every* successful right interaction that has a different identity from the left interaction in the Main Execution, B finds a $\mathcal{WISSP}$ proof in Stage 3 of this right interaction that does not interleave with any message in Stage 1 and the i^{th} proof (or Stage 2 for hybrid H_{k+6}) of that left interaction, and repeatedly rewinds the proof until a second transcript is obtain; it then computes a witness, if the two transcripts are different. Since there are only polynomial number of right interactions, it follows using almost the same proof of Lemma 2 that the running time of B is "bounded", and further using exactly the same proof of Lemma 3 that, except with negligible probability, the witnesses that B extracts out are indeed the values committed to in the right interactions. Thus by the $\mathcal{WI}$ property of the Stage 3 proofs (or the hiding property of Stage 2 resp.), the view and the values committed to by A are indistinguishable in hybrids H_i and H_{i+1} for $1 \leq i \leq k+4$ (or in H_{k+5} and H_{k+6} resp.). We thus have:

Theorem 6. *$\langle C, R\rangle$ is concurrent non-malleable.*

8 Proof of Robust Non-Malleability

In this section, we show that, for any $r \in N$, $\langle C, R\rangle$ can be easily modified into a $O(r)$-round (concurrent) non-malleable commitment scheme $\langle \tilde{C}, \tilde{R}\rangle$ that is additionally one-many r-robust.

It is shown in [LP09] that one-many r-robust commitment schemes are easy to construct: any commitment scheme that is "extractable" and has more than r "rewinding slots" is directly one-many non-malleable w.r.t. r-round protocols. Therefore, to make our constant-round non-malleable commitment scheme $\langle C, R\rangle$ one-many r-robust, we simply add more $\mathcal{WISSP}$ proofs in Stage 3 of the protocol. More precisely, the commitment scheme $\langle C_r, R_r\rangle$ proceeds identitcally to $\langle C, R\rangle$, except that in Stage 3 of the protocol, the Committer $\tilde{C}$ needs to provide $\max(r+1, l)$ $\mathcal{WISSP}$ proofs (of the statement that either the Stage 2 message is a valid commitment or that it knows a "trapdoor"), where l is the number of $\mathcal{WISSP}$ proofs in Stage 3 of the original protocol $\langle C, R\rangle$. It follows using the same proof as in [LP09] that $\langle C_r, R_r\rangle$ is one-many r-robust. Roughly speaking, the main idea of the proof is to reduce the one-many r-robustness to the indistinguishability of the interaction with machine $B(y_n^1)$ or $B(y_n^2)$, by extracting the value committed to in the right interactions from the $\mathcal{WISSP}$ proofs in Stage 3 of the protocol, *without rewinding the left interactions*. This is achievable, (similar to the proof of the indistinguishability of Hybrid H_i and H_{i+1} in Section 6,) as there are more $\mathcal{WISSP}$ proofs in Stage 3 than the nubmer of messages in the left interaction, and one can always find a $\mathcal{WISSP}$ proof that does not interleave with the left interaction and extract a witness from this proof, without rewinding the left interactions. The witness extracted must be a valid decommitment, as otherwise, by the special-soundness of the proof, it must be a valid signature-chain, which violates the soundness of the signature-game (since the adversary here is never rewound and obtains only three signatures during the straight-line execution of the right interaction). Therefore, we conclude that $\langle C_r, R_r\rangle$ is one-many r-robust. It follows using the same proof in Section 6 that $\langle C_r, R_r\rangle$ is stand-alone non-malleable; and it further follows using the same proof as in Section 7 that it is, in fact, also concurrent non-malleable.

Lemma 4. *For every $r \in N$, the protocol $\langle C_r, R_r\rangle$ has $O(r)$-round, and is concurrently non-malleable and one-many r-robust.*

Theorem 2 follows directly from Lemma 4. Furthermore, for $r < l$, the protocol $\langle C_r, R_r\rangle$ is the same as $\langle C, R\rangle$; thus,

Corollary 1. *For any $r < l$, $\langle C, R\rangle$ is concurrently non-malleable and one-many r-robust.*

9 Acknowledgements

We are very grateful to Boaz Barak, Ran Canetti, Danny Dolev, Cynthia Dwork, Johan Håstad, Oded Goldreich, Shafi Goldwasser, Silvio Micali, Moni Naor, Tal Rabin, Alon Rosen, Amit Sahai, Wei-lung Tseng, Muthuramakrishnan Venkitasubramaniam and Hoeteck Wee for many enligthening discussions about non malleability over the years. We are particularly grateful to Oded Goldreich for encouraging us to include a self-contained description of a non-malleable commitment for constant length identities into the journal version of [LP09]; the ideas in this paper came out of thinking about how to simplify the presentation of this (weak) primitive. The second author is also indebted to Alon Rosen for introducing him to the area of non malleability, and for many fruitful discussions about it.

References

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, volume 0, pages 106–115, 2001.

[Bar02] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *FOCS*, pages 345–355, Washington, DC, USA, 2002. IEEE Computer Society.

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

[Blu83] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *SIGACT News*, 15(1):23–27, 1983.

[CDSMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In *TCC*, pages 387–402, 2009.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *CRYPTO*, pages 19–40, 2001.

[CIO01] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Universal service-providers for private information retrieval. *J. Cryptology*, 14(1):37–74, 2001.

[CKOS01] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Efficient and non-interactive non-malleable commitment. In *EUROCRYPT*, pages 40–59, 2001.

[CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. Adaptive hardness and composable security in the plain model from standard assumptions. To appear in FOCS 2010, 2010.

[CR87] Benny Chor and Michael O. Rabin. Achieving independence in logarithmic number of rounds. In *PODC*, pages 260–268, 1987.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DG03] Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *STOC*, pages 426–437, 2003.

[FF09] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. *J. Cryptology*, 22(4):530–571, 2009.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC*, pages 416–426, 1990.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987.

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

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

[Gol04] Oded Goldreich. *Foundations of Cryptography — Basic Applications.* Cambridge University Press, 2004.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28:12–24, 1999.

[IL89] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235, 1989.

[KOS03] Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Round efficiency of multi-party computation with a dishonest majority. In *EUROCRYPT*, pages 578–595, 2003.

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

[LPTV10] Huijia Lin, Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable zero knowledge proofs. In *CRYPTO*, pages 429–446, 2010.

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

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

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4:151–158, 1991.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43, 1989.

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *STOC*, pages 232–241, New York, NY, USA, 2004. ACM.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In *CRYPTO*, pages 57–74, 2008.

[PR05a] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *FOCS*, pages 563–572, 2005.

[PR05b] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *STOC*, pages 533–542, 2005.

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *TCC*, pages 403–418, 2009.

[PW10] Rafael Pass and Hoeteck Wee. Constant-round non-malleable commitment from strong one-way functions. In *Eurocrypt*, 2010.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394, 1990.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.

[Wee10] Hoeteck Wee. Black-box, round-efficient secure computation via non-malleability amplification. To appear in FOCS 2010, 2010.

A General Definitions

A.1 Witness Relations

We recall the definition of a witness relation for a $\mathcal{NP}$ language [Gol01].

Definition 4 (Witness relation). *A* witness relation *for a language $L \in \mathcal{NP}$ is a binary relation R_L that is polynomially bounded, polynomial time recognizable and characterizes L by $L = \{x : \exists y \, s.t. \, (x, y) \in R_L\}$*

We say that y is a witness for the membership $x \in L$ if $(x, y) \in R_L$. We will also let $R_L(x)$ denote the set of witnesses for the membership $x \in L$, i.e., $R_L(x) = \{y : (x, y) \in L\}$. In the following, we assume a fixed witness relation R_L for each language $L \in \mathcal{NP}$.

A.2 Indistinguishability

Definition 5 (Computational Indistinguishability). *Let Y be a countable set. Two ensembles $\{A_{n,y}\}_{n\in N, y\in Y}$ and $\{B_{n,y}\}_{n\in N, y\in Y}$ are said to be* computationally indistinguishable *(denoted by $\{A_{n,y}\}_{n\in N, y\in Y} \approx \{B_{n,y}\}_{n\in N, y\in Y}$), if for every $\mathcal{PPT}$ "distinguishing" machine D, there exists a negligible function $\nu(\cdot)$ so that for every $n \in N, y \in Y$:*

$$|\Pr[a \leftarrow A_{n,y} \ : \ D(1^n, y, a) = 1] - \Pr[b \leftarrow B_{n,y} \ : \ D(1^n, y, b) = 1]| < \nu(n)$$

A.3 Interactive Proofs

We use the standard definitions of interactive proofs (and interactive Turing machines) [GMR89] and arguments (a.k.a. computationally-sound proofs) [BCC88]. Given a pair of interactive Turing machines, P and V, we denote by $\langle P(w), V\rangle(x)$ the random variable representing the (local) output of V, on common input x, when interacting with machine P with private input w, when the random input to each machine is uniformly and independently chosen.

Definition 6 (Interactive Proof System). *A pair of interactive machines $\langle P, V\rangle$ is called an interactive proof system for a language L if there is a negligible function $\nu(\cdot)$ such that the following two conditions hold :*

- Completeness: *For every $x \in L$, and every $w \in R_L(x)$,* $\Pr[\langle P(w), V\rangle(x) = 1] = 1$
- Soundness: *For every $x \in \{0,1\}^n - L$, and every interactive machine B,* $\Pr[\langle B, V\rangle(x) = 1] \leq \nu(n)$

In case that the soundness condition is required to hold only with respect to a computationally bounded prover, the pair $\langle P, V\rangle$ is called an interactive argument *system.*

A.4 Zero-Knowledge

We recall the standard definition of $\mathcal{ZK}$ proofs. Loosely speaking, an interactive proof is said to be *zero-knowledge* ($\mathcal{ZK}$) if a verifier V learns nothing beyond the validity of the assertion being proved, it could not have generated on its own. As "feasible" computation in general is defined though the notion of probabilistic polynomial-time, this notion is formalized by requiring that the output of every (possibly malicious) verifier interacting with the honest prover P can be "simulated" by a probabilistic expected polynomial-time machine S (a.k.a. the *simulator*). The idea behind this definition is that whatever V^* might have learned from interacting with P, he could have learned by himself by running the simulator S.

The notion of $\mathcal{ZK}$ was introduced and formalized by Goldwasser, Micali and Rackoff in [GMR89]. We present their definition below.

Definition 7 ($\mathcal{ZK}$)**.** *Let L be a language in* **NP***, R_L a witness relation for L, (P, V) an interactive proof (argument) system for L. We say that (P, V) is* statistical/computational $\mathcal{ZK}$*, if for every probabilistic polynomial-time interactive machine V there exists a probabilistic algorithm S whose expected running-time is polynomial in the length of its first input, such that the following ensembles are statistically close/computationally indistinguishable over L.*

- $\Big\{\langle P(y), V(z)\rangle(x)\Big\}_{n\in N, x\in\{0,1\}^n\cap L, y\in R_L(x), z\in\{0,1\}^*}$
- $\Big\{S(x, z)\Big\}_{n\in N, x\in\{0,1\}^n\cap L, y\in R_L(x), z\in\{0,1\}^*}$

where $\langle P(y), V(z)\rangle(x)$ denotes the view of V in interaction with P on common input x and private inputs y and z respectively.

A.5 Witness Indistinguishability

An interactive proof (or argument) is said to be *witness indistinguishable* ($\mathcal{WI}$) if the verifier's output is "computationally independent" of the witness used by the prover for proving the statement. In this context, we focus on languages $L \in \mathcal{NP}$ with a corresponding witness relation R_L. Namely, we consider interactions in which, on common input x, the prover is given a witness in $R_L(x)$. By saying that the output is computationally independent of the witness, we mean that for any two possible **NP**-witnesses that could be used by the prover to prove the statement $x \in L$, the corresponding outputs are computationally indistinguishable.

Definition 8 (Witness-indistinguishability)**.** *Let $\langle P, V\rangle$ be an interactive proof (or argument) system for a language $L \in \mathcal{NP}$. We say that $\langle P, V\rangle$ is* witness-indistinguishable *for R_L, if for every probabilistic polynomial-time interactive machine V^* and for every two sequences $\{w^1_{n,x}\}_{n\in N, x\in L}$ and $\{w^2_{n,x}\}_{n\in N, x\in L}$, such that $w^1_{n,x}, w^2_{n,x} \in R_L(x)$ for every $x \in L \cap \{0,1\}^n$, the following probability ensembles are computationally indistinguishable over $n \in N$.*

- $\{\langle P(w^1_{n,x}), V^*(z)\rangle(x)\}_{n\in N, x\in L\cap\{0,1\}^n, z\in\{0,1\}^*}$
- $\{\langle P(w^2_{n,x}), V^*(z)\rangle(x)\}_{n\in N, x\in L\cap\{0,1\}^n, z\in\{0,1\}^*}$

A.6 Proofs (Arguments) of Knowledge

Loosely speaking, an interactive proof is a proof of knowledge if the prover convinces the verifier that it *possesses*, or can *feasibly compute*, a witness for the statement proved. The notion of a proof of knowledge is essentially formalized as follows: an interactive proof of $x \in L$ is a proof of knowledge if there exists a probabilistic expected polynomial-time *extractor* machine E, such that for any prover P, E on input the description of P and any statement $x \in L$ readily outputs a valid witness for $x \in L$ if P succeeds in convincing the Verifier that $x \in L$. Formally,

Definition (Proof of knowledge [Gol01])**.** *Let (P, V) be an interactive proof system for the language L. We say that (P, V) is a* proof of knowledge *for the witness relation R_L for the language L it there exists an probabilistic expected polynomial-time machine E, called the extractor, and a negligible function $\nu(n)$ such that for every machine P^*, every statement $x \in \{0,1\}^n$, every random tape $r \in \{0,1\}^*$ and every auxiliary input $z \in \{0,1\}^*$,*

$$\Pr\left[\langle P'_r(z), V\rangle(x) = 1\right] \leq Pr[E^{P'_r(x,z)}(x) \in R_L(x)] + \nu(n)$$

consider $\mathcal{PPT}$ provers. An interactive argument system $\langle P, V\rangle$ is an *argument of knowledge* if the above condition holds w.r.t. probabilistic polynomial-time provers.

Special-sound $\mathcal{WI}$ proofs A 4-round public-coin interactive proof for the language $L \in \mathcal{NP}$ with witness relation R_L is special-sound with respect to R_L, if for any two transcripts $(\delta, \alpha, \beta, \gamma)$ and $(\delta', \alpha', \beta', \gamma')$ such that the initial two messages, δ, δ' and α, α', are the same but the challenges β, β' are different, there is a deterministic procedure to extract the witness from the two transcripts and runs in polynomial time. Special-sound $\mathcal{WI}$ proofs for languages in $\mathcal{NP}$ can be based on the existence of 2-round commitment schemes, which in turn can be based on one-way functions [GMW91, FS90, HILL99, Nao91].