New and Improved Constructions of Non-Malleable Cryptographic Protocols

Rafael Pass
CSAIL, MIT
Cambridge, MA, USA
pass@csail.mit.edu

Alon Rosen
CSAIL, MIT
Cambridge, MA, USA
alon@csail.mit.edu

ABSTRACT

We present a new constant round protocol for non-malleable zero-knowledge. Using this protocol as a subroutine, we obtain a new constant-round protocol for non-malleable commitments. Our constructions rely on the existence of (standard) collision resistant hash functions. Previous constructions either relied on the existence of trapdoor permutations and hash functions that are collision resistant against sub-exponential sized circuits, or required a super-constant number of rounds. Additional results are the first construction of a non-malleable commitment scheme that is statistically hiding (with respect to opening), and the first non-malleable protocols that satisfy a strict polynomial-time simulation requirement. The latter are constructed by additionally assuming the existence of trapdoor permutations.

Our approach differs from the approaches taken in previous works in that we view non-malleable zero-knowledge as a building-block rather than an end goal. This gives rise to a modular construction of non-malleable commitments and results in a somewhat simpler analysis.

The techniques that we use to construct our zero-knowledge protocol are non black-box, but are different than the non black-box techniques previously used in the context of non-malleable coin-tossing.

Categories and Subject Descriptors

F.1.2 [**Theory of Computation**]: Interactive and reactive computation

General Terms

Theory

Keywords

Cryptography, zero-knowledge, non-malleability, man-in-the-middle, round-complexity, non black-box simulation

STOC'05, May 22-24, 2005, Baltimore, Maryland, USA.

1. INTRODUCTION

We consider the execution of two-party protocols in the presence of an adversary that has full control of the communication channel between the parties. The adversary has the power to omit, insert or modify messages at its choice. It has also full control over the scheduling of the messages. The honest parties are not necessarily aware to the existence of the adversary, and are not allowed to use any kind of trusted set-up (such as a common reference string).

The above kind of attack is often referred to as a *man-in-the-middle* attack. It models a natural scenario whose investigation is well motivated. Protocols that retain their security properties in face of a man-in-the-middle are said to be *non-malleable* [11]. Due to the hostile environment in which they operate, the design and analysis of non-malleable protocols is a notoriously difficult task. The task becomes even more challenging if the honest parties are not allowed to use any kind of trusted set-up. Indeed, only a handful of such protocols have been constructed so far.

The rigorous treatment of two-party protocols in the man-in-the-middle setting has been initiated in the seminal paper by Dolev, Dwork and Naor [11]. The paper contains definitions of security for the tasks of non-malleable commitment and non-malleable zero-knowledge. It also presents protocols that meet these definitions. The protocols rely on the existence of one-way functions, and require $O(\log n)$ rounds of interaction, where $n \in N$ is a security parameter.

A more recent result by Barak presents constant-round protocols for non-malleable commitment and non-malleable zero-knowledge [2]. This is achieved by constructing a coin-tossing protocol that is secure against a man in the middle, and then using the outcome of this protocol to instantiate known constructions for non-malleable commitment and zero-knowledge in the common reference string model (see Section 1.3). The proof of security makes use of non black-box techniques and is highly complex. It relies on the existence of trapdoor permutations and hash functions that are collision-resistant against sub-exponential sized circuits.

In this paper we continue the line of research initiated by the above papers. We will be interested in the construction of new constant-round protocols for non-malleable commitment and non-malleable zero-knowledge. Similarly to the above works, we will refrain from relying on any kind of set-up assumption.

1.1 Non-Malleable Protocols

In accordance with the above discussion, consider a man-in-the-middle adversary A that is simultaneously participat-

ing in two executions of a two-party protocol. These executions are called the left and the right interaction. Besides controlling the messages that it sends in the left and right interactions, A has control over the scheduling of the messages. In particular, it may delay the transmission of a message in one interaction until it receives a message (or even multiple messages) in the other interaction.

The adversary is trying to take advantage of its participation in the left interaction in order to violate the security of the protocol executed in the right interaction, where the exact interpretation of the term "violate the security" depends on the specific task at hand.

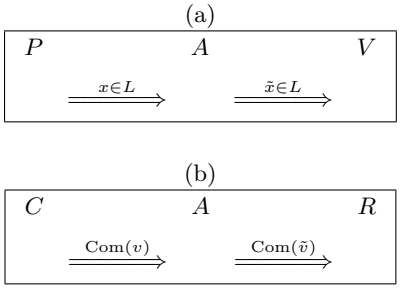

Figure 1: The man-in-the-middle adversary. (a) Interactive proofs. (b) Commitments.

A two-party protocol is said to be non-malleable if the left interaction does not "help" the adversary in violating the security of the right interaction. Following the simulation paradigm [19, 20, 17, 18], this is formalized by defining appropriate "real" and "idealized" executions.

In the real execution, called the man-in-the-middle execution, the adversary participates in both the left and the right interactions. In the idealized execution, called the stand-alone execution, the adversary is only participating in a single interaction. Security is defined by requiring that the adversary cannot succeed better in the man-in-the middle execution than he could have done in the stand-alone execution. In the specific instances of zero-knowledge and string commitment, the definition of security takes the following forms (detailed definitions can be found in Section 2).

Non-malleable zero-knowledge. [11] Let $\langle P, V\rangle$ be an interactive proof system (and see Figure 1.a). In the left interaction the adversary A is verifying the validity of a statement x by interacting with an honest prover P. In the right interaction A proves the validity of a statement $\tilde{x} \neq x$ to the honest verifier V. The objective of the adversary is to convince the verifier in the right interaction that $\tilde{x} \in L$. Non-malleability of $\langle P, V\rangle$ is defined by requiring that for any man-in-the-middle adversary A, there exists a stand-alone prover S that manages to convince the verifier with essentially the same probability as A. The interactive proof $\langle P, V\rangle$ is said to be non-malleable zero-knowledge if it is non-malleable and (stand-alone) zero-knowledge.

Non-malleable commitments. [11]. Let $\langle C, R\rangle$ be a commitment scheme (and see Figure 1.b). In the left interactions the adversary A is receiving a commitment to a value v from the committer C. In the right interaction A is sending a commitment to a value $\tilde{v}$ to the receiver R. The objective of the adversary is to succeed in committing in the right interaction to a value $\tilde{v} \neq v$ that satisfies $\mathcal{R}(v, \tilde{v}) = 1$ for some polynomial-time computable relation $\mathcal{R}$. Non-malleability of $\langle C, R\rangle$ is defined by requiring that for any man-in-the-middle adversary A, there exists a stand-alone committer S that manages to commit to the related $\tilde{v}$ with essentially the same probability as A.

Schemes that satisfy the above definition are said to be non-malleable with respect to commitment. In a different variant, called non-malleable commitment with respect to opening [14], the adversary is considered to have succeeded only if it manages to *decommit* to a related value $\tilde{v}$.

1.2 Our Results

Our main result is the construction of a new constant-round protocol for non-malleable $\mathcal{ZK}$. The proof of security relies on the existence of (ordinary) collision resistant hash functions and does not rely on any set-up assumption.

Theorem 1 (Non-malleable $\mathcal{ZK}$) *Suppose that there exists a family of collision resistant hash functions. Then, there exists a constant-round non-malleable $\mathcal{ZK}$ argument for every $L \in \mathcal{NP}$.*

Using our non-malleable $\mathcal{ZK}$ protocol as a subroutine, we construct constant round protocols for non-malleable string commitment. One of our constructions achieves statistically binding commitments that are non-malleable w.r.t. commitment, and the other achieves statistically hiding commitments that are non-malleable w.r.t. opening.

Theorem 2 (Statistically binding non-malleable commitment) *Suppose that there exists a family of collision-resistant hash functions. Then, there exists a constant-round statistically binding commitment scheme that is non malleable with respect to commitment.*

Theorem 3 (Statistically hiding non-malleable commitment) *Suppose that there exists a family of collision-resistant hash functions. Then, there exists a constant-round statistically hiding commitment scheme that is non malleable with respect to opening.*

Underlying cryptographic assumptions. The main quantitative improvement of our construction over the constant round protocols in [2] is in the underlying cryptographic assumption. Our constructions rely on the existence of ordinary collision resistant hash functions. The protocols in [2] relied on the existence of both trapdoor permutations and hash functions that are collision resistant against sub exponential sized circuits. The constructions in [11] assumed only the existence of one-way functions, but had a super-constant number of rounds.

Statistically hiding non-malleable commitments. Theorem 3 gives the first construction of a non-malleable commitment scheme that is statistically hiding and that does not rely on set-up assumptions. We mention that the existence of collision resistant hash functions is the weakest assumption currently known to imply constant round statistically hiding commitment schemes (even those that are not of the non-malleable kind) [27, 7].

Strict vs. liberal non-malleability. The notion of non malleability that has been considered so far in all works (including the current one), allows the stand alone adversary S to run in expected polynomial time. A stronger ("tighter") notion of security, named strict non-malleability [11], requires

S to run in strict polynomial time. By additionally assuming the existence of trapdoor permutations, we are able to construct the first protocols that are strictly non-malleable.

Theorem 4 (Strict non-malleability) *Suppose that there exists a family of collision resistant hash functions. Further suppose that there exist a family of trapdoor permutations. Then,*

1. *There exists a constant-round strictly non-malleable $\mathcal{ZK}$ argument for every $L \in \mathcal{NP}$.*
2. *There exists a constant round statistically hiding commitment scheme that is strictly non-malleable with respect to opening.*

Techniques. Our protocols rely on non black-box techniques used by Barak to obtain constant-round public-coin $\mathcal{ZK}$ argument for $\mathcal{NP}$ [1] (in a setting where no man in the middle is considered). They are closely related to previous works by Pass [29], and Pass and Rosen [30] that appeared in the context of bounded-concurrent two-party and multi-party computation. Our techniques are different than the ones used by Barak in the context of non-malleable coin-tossing [2].

The approach we follow in this work is fundamentally different than the approach used in [11]. Instead of viewing non-malleable commitments as a tool for constructing non-malleable $\mathcal{ZK}$ protocols, we reverse the roles and use non-malleable $\mathcal{ZK}$ protocols in order to construct non-malleable commitments. Our approach is also different from the one taken by [2], who uses a coin-tossing protocol to instantiate constructions that rely on the existence of a common reference string.

Additional contributions. Our approach gives rise to a modular and natural construction of non-malleable commitments. This construction emphasizes the role of non-malleable $\mathcal{ZK}$ as a building block for other non-malleable cryptographic primitives. In principle, our definitions of non-malleability are compatible with the ones appearing in [11]. However, the presentation is more detailed and somewhat different (see Section 2). Our definitional approach, as well as our construction of non-malleable $\mathcal{ZK}$ highlights a potential distinction between the notions of non-malleable interactive proofs and non-malleable $\mathcal{ZK}$. This distinction was not present in the definitions given in [11].

1.3 Related Work

Assuming the existence of a common random string, Di Crescenzo, Ishai and Ostrovsky [10], and Di Crescenzo, Katz, Ostrovski, and Smith [9] construct non-malleable commitment schemes. Sahai [31], and De Santis, Di Crescenzo, Ostrovski, Persiano and Sahai [8] construct a non-interactive non-malleable $\mathcal{ZK}$ protocol under the same assumption. Fischlin and Fischlin [14], and Damgård and Groth [6] construct non-malleable commitments assuming the existence of a common reference string. We note that the non-malleable commitments constructed in [10] and [14] only satisfy non-malleability with respect to opening [14]. Canetti and Fischlin [5] construct a universally composable commitment assuming a common random string. Universal composability implies non malleability. However, it is impossible to construct universally composable commitments without making set-up assumptions [5].

Goldreich and Lindell [16], and Nguyen and Vadhan [28] consider the task of session-key generation in a setting where the honest parties share a password that is taken from a relatively small dictionary. Their protocols are designed having a man-in-the-middle adversary in mind, and only requires the usage of a "mild" set-up assumption (namely the existence of a "short" password).

1.4 Future Work

Our constructions (and even more so the previous ones) are quite complex. A natural question is whether they can be simplified. A somewhat related question is whether non-black box techniques are necessary for achieving constant-round non-malleable $\mathcal{ZK}$ or commitments. Our constructions rely on the existence of collision resistant hash functions, whereas the non constant-round construction in [11] relies on the existence of one-way functions. We wonder whether the collision resistance assumption can be relaxed.

Another interesting question is whether it is possible to achieve non-malleability under concurrent executions. The techniques used in this paper do not seem to extend to the (unbounded) concurrent case and new ideas seem to be required. Advances in that direction might shed light on the issue of concurrent composition of general secure protocols.

2. DEFINITIONS

The notion of non-malleability was introduced by Dolev, Dwork and Naor [11]. In this paper we focus on non malleability of interactive proofs and string commitment. The definition of interactive proofs extends in a natural way to notions of non-malleable zero-knowledge and proofs of knowledge (by additionally requiring the zero-knowledge and proof of knowledge properties from a non-malleable interactive proof). In principle, our definitions are compatible with the ones appearing in [11]. However, the presentation is more detailed and somewhat different.

2.1 Non-Malleable Interactive Proofs

Let $\langle P, V \rangle$ be an interactive proof. Consider a scenario where a man-in-the-middle adversary A is simultaneously participating in two interactions. These interactions are called the left and the right interaction. In the left interaction the adversary A is verifying the validity of a statement x by interacting with an honest prover P. In the right interaction A proves the validity of a statement $\tilde{x}$ to the honest verifier V (see Figure 1.a). As discussed in Section 1.1, A has control over the scheduling of the messages. We consider two types of executions.

Man-in-the-middle execution. The man-in-the-middle consists of the scenario described above. The input of P is an instance-witness pair (x, w), and the input of V is the instance $\tilde{x}$. A receives both $x, \tilde{x}$ and an auxiliary input z. Let $\mathsf{mim}_V^A(x, \tilde{x}, w, z)$ be a random variable describing the the output of V in the above experiment when the random tapes of P, A and V are uniformly and independently chosen.

Stand-alone execution. In the stand-alone execution only one interaction takes place. The stand-alone adversary S directly interacts with the honest verifier V. As in the man-in-the-middle execution, V receives as input an instance $\tilde{x}$. S receives instances $x, \tilde{x}$ and auxiliary input z. Let $\mathsf{sta}_V^S(x, \tilde{x}, z)$ be a random variable describing the the output

of V in the above experiment when the random tapes of S and V are uniformly and independently chosen.

DEFINITION 2.1. *An interactive proof $\langle P, V\rangle$ for a language L is said to be* non-malleable *if for every probabilistic polynomial time man-in-the-middle adversary A, there exists a probabilistic expected polynomial time stand-alone prover S and a negligible function $\nu : N \to N$, such that for every $(x, w) \in L \times R_L(x)$, every $\tilde{x} \in \{0,1\}^{|x|}$ so that $\tilde{x} \neq x$, and every $z \in \{0,1\}^*$:*

$$\Pr\left[\mathsf{mim}_V^A(x, \tilde{x}, w, z) = 1\right] \;<\; \Pr\left[\mathsf{sta}_V^S(x, \tilde{x}, z) = 1\right] \;+\; \nu(|x|)$$

Note that the above-defined notion refers to interactive-proofs that are non-malleable *with respect to themselves*, since the definition considers a setting where the same protocol is executed in the left and the right interaction. In principle, one could consider two different protocols that are executed on the left and on the right which are be non-malleable *with respect to each other*.

Non-malleability with respect to tags. Definition 2.1 rules out the possibility that the statement proved on the right interaction is identical to the one on the left. Indeed, if the same protocol is executed on the left and on the right this kind of attack cannot be prevented, as the man-in-the-middle adversary can always copy messages between the two executions (c.f., the chess-master problem [11]).

In many situations it is, nevertheless, important to be protected against an attacker that attempts to prove even same statement. In order to deal with this problem, one might instead consider a "tag-based" variant of non-malleability.

We consider a family of interactive proofs, where each member of the family is labelled with a tag string TAG $\in \{0,1\}^m$, and $m = m(n)$ is a parameter that potentially depends on the length of the common input (security parameter) $n \in N$. As before, we consider a scenario where a man-in-the-middle adversary A is simultaneously participating in a left and a right interaction. In the left interaction, A is verifying the validity of a statement x by interacting with an honest prover P while using a protocol that is labelled with a string TAG. In the right interaction A proves the validity of a statement $\tilde{x}$ to the honest verifier V while using a protocol that is labelled with a string $\tilde{\text{TAG}}$.

Similarly to Definition 2.1, non-malleability with respect to tags is formalized by considering a man-in-the-middle and a stand-alone execution. The only difference in the definitions is that instead of requiring non-malleability whenever $x \neq \tilde{x}$, we will require non-malleability whenever TAG $\neq \tilde{\text{TAG}}$.

Tags vs. statements. A non-malleable interactive proof can be turned into a tag-based one by simply concatenating the tag to the statement being proved. On the other hand, an interactive proof that is non-malleable with respect to tags of length $m(n) = n$ can be turned into a non-malleable interactive proof by using the statement $x \in \{0,1\}^n$ as tag.

The problem of constructing a tag-based non-malleable interactive proof is already non-trivial for tags of length, say $m(n) = O(\log n)$ (and even for $m(n) = O(1)$), but is still potentially easier than for tags of length n. This opens up the possibility of reducing the construction of interactive proofs that are non-malleable w.r.t. long tags into interactive proofs that are non-malleable w.r.t. shorter tags. Even though we do not know whether such a reduction is possible in general, our work follows this path and demonstrates that in specific cases such a reduction is indeed possible.

2.2 Non-malleable Commitments

We proceed to give a definition of non-malleable commitments. Informally, a commitment scheme is non-malleable if a man-in-the-middle adversary that receives a commitment to a value v will not be able to "successfully" commit to a related value $\tilde{v}$. The literature discusses two different interpretations of the term"success":

Non-malleability with respect to commitment [11]. The adversary is said to succeed if it manages to commit to a related value, even without being able to later decommit to this value. This notion makes sense only in the case of statistically-binding commitments.

Non-malleability with respect to opening [10]. The adversary is said to succeed only if it is able to both commit and decommit to a related value. This notion makes sense both in the case of statistically-binding and statistically-hiding commitments.

As in the case of non-malleable interactive proofs, we formalize the definition by comparing a man-in-the-middle and a stand-alone execution. Let $n \in N$ be a security parameter. Let $\langle C, R\rangle$ be a commitment scheme, and let $\mathcal{R} \subseteq \{0,1\}^n \times \{0,1\}^n$ be a polynomial-time computable non-reflexive relation (i.e., $\mathcal{R}(v, v) = 0$). As before we consider man-in-the-middle adversaries that are simultaneously participating in a left and a right interaction in which a commitment scheme is taking place. The adversary is said to succeed in mauling a left commitment to a value v, if he is able to come up with a right commitment to a value $\tilde{v}$ such that $\mathcal{R}(v, \tilde{v}) = 1$. Since we cannot rule out copying, we will only be interested in relations where copying is not considered success, and we therefore require that the relation $\mathcal{R}$ is non-reflexive. The man-in-the-middle and the stand-alone executions are defined as follows.

The man-in-the-middle execution. In the man-in-the-middle execution, the adversary A is simultaneously participating in a left and a right interaction. In the left interaction the man-in-the-middle adversary A interacts with C receiving a commitment to a value v. In the right interaction A interacts with R attempting to commit to a related value $\tilde{v}$. Prior to the interaction, the value v is given to C as local input. A receives an auxiliary input z, which in particular might contain a-priori informantion about v.[1] The success of A is defined using the following two Boolean random variables:

- $\mathsf{mim}_{\mathsf{com}}^A(\mathcal{R}, v, z) = 1$ if and only if A produces a valid committment to $\tilde{v}$ such that $\mathcal{R}(v, \tilde{v}) = 1$.
- $\mathsf{mim}_{\mathsf{open}}^A(\mathcal{R}, v, z) = 1$ if and only if A decommits to a value $\tilde{v}$ such that $\mathcal{R}(v, \tilde{v}) = 1$.

The stand-alone execution. In the stand-alone execution only one interaction takes place. The stand-alone

[1]The original definition by Dwork et al. [11] accounted for such a-priori information by providing the adversary with the value $\mathsf{hist}(v)$, where the function $\mathsf{hist}(\cdot)$ be a polynomial-time computable function.

adversary S directly interacts with R. As in the man-in-the-middle execution, the value v is chosen prior to the interaction and S receives some a-priori information about v as part of its an auxiliary input z. S first executes the commitment phase with R. Once the commitment phase has been completed, S receives the value v and attempts to decommit to a value $\tilde{v}$. The success of S is defined using the following two Boolean random variables:

- $\mathsf{sta}^S_{\mathsf{com}}(\mathcal{R}, v, z) = 1$ if and only if S produces a valid committment to $\tilde{v}$ such that $\mathcal{R}(v, \tilde{v}) = 1$.
- $\mathsf{sta}^S_{\mathsf{open}}(\mathcal{R}, v, z) = 1$ if and only if A decommits to a value $\tilde{v}$ such that $\mathcal{R}(v, \tilde{v}) = 1$.

DEFINITION 2.2. *A commitment scheme $\langle C, R\rangle$ is said to be* non-malleable with respect to commitment *if for every probabilistic polynomial-time man-in-the-middle adversary A, there exists a probabilistic expected polynomial time stand-alone adversary S and a negligible function $\nu : N \to N$, such that for every non-reflexive polynomial-time computable relation $\mathcal{R} \subseteq \{0,1\}^n \times \{0,1\}^n$, every $v \in \{0,1\}^n$, and every $z \in \{0,1\}^*$, it holds that:*

$$\Pr\left[\mathsf{mim}^A_{\mathsf{com}}(\mathcal{R}, v, z) = 1\right] \;<\; \Pr\left[\mathsf{sta}^S_{\mathsf{com}}(\mathcal{R}, v, z) = 1\right] \;+\; \nu(n)$$

Non-malleability with respect to opening *is defined in the same way, while replacing the random variables $\mathsf{mim}^A_{\mathsf{com}}(\mathcal{R}, v, z)$ and $\mathsf{sta}^S_{\mathsf{com}}(\mathcal{R}, v, z)$ with $\mathsf{mim}^A_{\mathsf{open}}(\mathcal{R}, v, z)$ and $\mathsf{sta}^S_{\mathsf{open}}(\mathcal{R}, v, z)$.*

Content-based v.s. tag-based commitments. Similarly to the definition of interactive proofs non-malleable with respect to statements, the above definitions only require that the adversary should not be able to commit to a value that is related, *but different*, from the value it receives a commitment of. Technically, the above fact can be seen from the definitions by noting that the relation R, which defines the success of the adversary, is required to be non-reflexive. This means that the adversary is said to fail if it only is able to produce a commitment to the same value.[2] Indeed, if the same protocol is executed in the left and the right interaction, the adversary can always copy messages and succeed in committing to the same value on the right as it receives a commitment of, on the left. To cope with this problem, the definition can be extended to incorporate tags, in analogy with the definition of interactive proofs non-malleable with respect to tags. The extension is straight-forward and therefore omitted.

We note that any commitment scheme that satisfies Definition 2.2 can easily be transformed into a scheme which is tag-based non-malleable, by prepending the tag to the value before committing. Conversely, in analogy with non-malleable interactive proof, commitment schemes that are non-malleable with respect to tags of length $m(n) = \text{poly}(n)$ can be transformed into commitment schemes non-malleable with respect to content in a standard way (e.g., [11, 24]).

[2]Potentially, one could consider a slightly stronger definition, which also rules out the case when the adversary is able to construct a *different* commitment to the *same* value. Nevertheless, we here adhere to the standard definition of non-malleable commitments which allows the adversary to produce a different commitment to the same value.

2.3 Comparison with Previous Definitions

Our definitions of non-malleability essentially follow the original definitions by Dwork et al.[11]. However, whereas the definitions by Dwork et al. quantifies the experiments over all distributions D of inputs for the left and the right interaction (or just left interaction in the case of commitments), we instead quantify over all possible input values $x, \tilde{x}$ (or, in the case of commitments over all possible input values v for the left interaction). Our definitions can thus be seen as non-uniform versions of the definitions of [11].

Our definition of non-malleability with respect to opening is, however, different from the definition of [10] in the following ways: (1) The definition of [10] does not take into account possible a-priori information that the adversary might have about the commitment, while our (following [11]) does. (2) In our definition of the stand-alone execution the stand-alone adversary receives the value v after having completed the commitment phase and is thereafter supposed to decommit to a value related to v. The definition of [10] does not provide the simulator with this information.

In our view, the "a-priori information" requirement is essential in many situations and we therefore present a definition that satisfies it. (Consider, for example, a setting where the value v committed to is determined by a different protocol, which "leaks" some information about v.) In order to be able to satisfy this stronger requirement we relax the definition of [10] by allowing the stand-alone adversary to receive the value v before decommitting.

3. A NON-MALLEABLE $\mathcal{ZK}$ PROTOCOL

Our construction of non-malleable $\mathcal{ZK}$ protocols proceeds in two phases. In the first phase, we construct a small set of "atomic" $\mathcal{ZK}$ arguments that are non-malleable with respect to each other. These protocols can be viewed as being non-malleable for tags of size $O(\log n)$, where $n \in N$ is the size of the common input (security parameter). In the second phase, the atomic protocols are appropriately combined into a "full-fledged" non-malleable $\mathcal{ZK}$ protocol for tags of length n (which yields non-malleability w.r.t. statements).

3.1 The Atomic $\mathcal{ZK}$ Protocols

The basic building block for our construction is a set of $m = \text{poly}(n)$ atomic protocols $\{ZK_{\mathsf{tag}}\}^m_{\mathsf{tag}=1}$. The protocols rely on Barak's non black-box techniques for obtaining constant-round public-coin $\mathcal{ZK}$ for $\mathcal{NP}$ [1], and are essentially identical to the $\mathcal{ZK}$ protocols used by Pass in [29] (which in turn rely on the protocols by Pass and Rosen [30]). In this work we actually prove that the atomic protocols satisfy a "weak" non-malleability property. This property is somewhat stronger than the property proved in [29] (which is "simulation-soundness"). We start by presenting the ideas underlying Barak's protocol. We then show how to extend Barak's techniques in order to obtain the atomic protocols.

Barak's non-black-box protocol. Let $n \in N$, and let $T : N \to N$ be a "nice" function that satisfies $T(n) = n^{\omega(1)}$. Barak's protocol relies on a "special" $\mathbf{NTIME}(T(n))$ relation. It also makes use of a witness-indistinguishable universal argument ($WIUARG$) [13, 12, 22, 25, 3]. We start by describing a variant of Barak's relation, which we denote by R_{sim}. Usage of this variant will facilitate the presentation of our ideas in later stages. Let $\{\mathcal{H}_n\}_n$ be a family of hash functions where a function $h \in \mathcal{H}_n$ maps $\{0,1\}^*$ to $\{0,1\}^n$,

and let Com be a statistically binding commitment scheme for strings of length n, where for any $\alpha \in \{0,1\}^n$, the length of $\mathrm{Com}(\alpha)$ is upper bounded by $2n$. The relation R_{sim} is described in Figure 2.

Instance: A triplet $\langle h, c, r\rangle \in \mathcal{H}_n \times \{0,1\}^n \times \{0,1\}^{\mathrm{poly}(n)}$.

Witness: A program $\Pi \in \{0,1\}^*$, a string $y \in \{0,1\}^*$ and a string $s \in \{0,1\}^{\mathrm{poly}(n)}$.

Relation: $R_{\mathsf{Sim}}(\langle h, c, r\rangle, \langle \Pi, y, s\rangle) = 1$ if and only if:

1. $|y| \leq |r| - n$.
2. $c = \mathrm{Com}(h(\Pi); s)$.
3. $\Pi(y) = r$ within $T(n)$ steps.

Figure 2: R_{sim} - A variant of Barak's relation.

Let L be any language in $\mathcal{NP}$, let $n \in N$, and let $x \in \{0,1\}^n$ be the common input for the protocol. Barak's protocol is described in Figure 3.

Common Input: An instance $x \in \{0,1\}^n$

Security parameter: 1^n.

Stage 1:

$V \to P$: Send $h \stackrel{\mathrm{R}}{\leftarrow} \mathcal{H}_n$.

$P \to V$: Send $c = \mathrm{Com}(0^n)$.

$V \to P$: Send $r \stackrel{\mathrm{R}}{\leftarrow} \{0,1\}^{3n}$.

Stage 2 (Body of the proof):

$P \Leftrightarrow V$: A *WI UARG* $\langle P_{\mathsf{UA}}, V_{\mathsf{UA}}\rangle$ proving the OR of the following two statements:

1. $\exists\ w \in \{0,1\}^{\mathrm{poly}(|x|)}$ s.t. $R_L(x, w) = 1$.
2. $\exists\ \langle \Pi, y, s\rangle$ s.t. $R_{\mathsf{Sim}}(\langle h, c, r\rangle, \langle \Pi, y, s\rangle) = 1$.

Figure 3: Barak's $\mathcal{ZK}$ argument for $\mathcal{NP}$ - BZK.

As shown in [1], BZK is computationally sound. We sketch why it is also $\mathcal{ZK}$. Let V^* be the program of a potentially malicious verifier. The $\mathcal{ZK}$ property of BZK follows by letting the simulator set $\Pi = V^*$ and $y = c$, where $c = \mathrm{Com}(h(V^*); s)$. Since $|c| = 2n \leq |r| - n$ and since, by definition $V^*(c)$ always equals r, the simulator can set $c = \mathrm{Com}(h(V^*); s)$ in Stage 1, and use the triplet $\langle V^*, c, s\rangle$ as a witness for R_{sim} in the $WIUARG$.[3] This enables the simulator to produce convincing interactions, even without knowing a valid witness for $x \in L$. The $\mathcal{ZK}$ property then follows (with some work) from the hiding property of Com and the $\mathcal{WI}$ property of the $WIUARG$.

The atomic protocols. The main difference between the atomic protocols and BZK is that in the atomic protocol the prover (simulator) is given two opportunities to guess the verifier's next message. Since the probability of guessing the verifier's next message in any one of the two opportunities is negligible, the soundness of the protocol will not be harmed. The simulation task, on the other hand, will be somewhat facilitated, since the simulator will get two opportunities to succeed in predicting the verifier's next message. What will differentiate between two atomic protocols ZK_{tag} and $ZK_{\tilde{\mathsf{tag}}}$ is the fact that the length of the verifier's next messages in ZK_{tag} is a parameter that depends on tag and m (as well as on the security parameter n).

Common Input: An instance $x \in \{0,1\}^n$

Parameters: Security parameter 1^n, length parameter $\ell(n)$.

Tag String: $\mathsf{tag} \in [m]$.

Stage 0 (Set-up):

$V \to P$: Send $h \stackrel{\mathrm{R}}{\leftarrow} \mathcal{H}_n$.

Stage 1 (Slot 1):

$P \to V$: Send $c_1 = \mathrm{Com}(0^n)$.

$V \to P$: Send $r_1 \stackrel{\mathrm{R}}{\leftarrow} \{0,1\}^{\mathsf{tag}\cdot\ell(n)}$.

Stage 1 (Slot 2):

$P \to V$: Send $c_2 = \mathrm{Com}(0^n)$.

$V \to P$: Send $r_2 \stackrel{\mathrm{R}}{\leftarrow} \{0,1\}^{(m+1-\mathsf{tag})\cdot\ell(n)}$.

Stage 2 (Body of the proof):

$P \Leftrightarrow V$: A *WI UARG* $\langle P_{\mathsf{UA}}, V_{\mathsf{UA}}\rangle$ proving the OR of the following three statements:

1. $\exists\ w \in \{0,1\}^{\mathrm{poly}(|x|)}$ s.t. $R_L(x, w) = 1$.
2. $\exists\ \langle \Pi, y, s\rangle$ s.t. $R_{\mathsf{Sim}}(\langle h, c_1, r_1\rangle, \langle \Pi, y, s\rangle) = 1$.
3. $\exists\ \langle \Pi, y, s\rangle$ s.t. $R_{\mathsf{Sim}}(\langle h, c_2, r_2\rangle, \langle \Pi, y, s\rangle) = 1$.

Figure 4: An atomic protocol – ZK_{tag}.

Stand-alone analysis of ZK_{tag}. Using similar arguments to the ones used for BZK, it can be shown that ZK_{tag} is sound. The main difference to be taken into consideration is the existence of multiple slots in Stage 1.[4] The $\mathcal{ZK}$ property is proved exactly as in the case of BZK, by letting the simulator pick either $i = 1$ or $i = 2$, and use $\langle V^*, c_i, s_i\rangle$ as the witness for $\langle h, c_i, r_i\rangle \in L_{\mathsf{sim}}$ (where L_{sim} is the language that corresponds to R_{sim}). Since for every $\mathsf{tag} \in [m]$, $|r_i| - |c_i| \geq \ell(n) - 2n$, we have that as long as $\ell(n) \geq 3n$, the protocol ZK_{tag} is indeed $\mathcal{ZK}$.

Useful properties of the atomic protocols. We highlight some properties of the atomic protocols. These properties will turn out to be relevant when dealing with a man in the middle.

Freedom in the choice of the slot: The simulator described above has the freedom to choose which $i \in \{1,2\}$ it will use in order to satisfy the relation R_{sim}. In particular, for the simulation to succeed, it is sufficient that $\langle h, c_i, r_i\rangle \in L_{\mathsf{sim}}$ for *some* $i \in \{1,2\}$.

Using a longer y in the simulation: The stand-alone analysis of ZK_{tag} only requires $\ell(n) \geq 3n$. Allowing larger values of $\ell(n)$ gives us the possibility of using a longer y in the simulation. This will turn out to be useful in cases where the verifier is allowed to receive "outside" messages that do not belong to the protocol (as indeed occurs in the man-in-the-middle setting).

[3] The usage of V^*'s program as a witness for the $WIUARG$ is precisely what makes the simulation non black-box.

[4] We mention that BZK and ZK_{tag} are known to be sound only assuming that the family $\{\mathcal{H}_k\}_n$ is collision resistant against $T(n)$-sized circuits. Nevertheless, using ideas from [3], it is possible to show how by slightly modifying the relation R_{sim}, one can guarantee soundness under "standard" collision resistance.

Proof of knowledge: ZK_{tag} is a proof of knowledge. That is, for any prover P^* and for any $x \in \{0,1\}^n$, if P^* convinces the honest verifier V that $x \in L$ with non-negligible probability then one can extract a witness w that satisfies $R_L(x,w) = 1$ in (expected) polynomial time.[5]

3.2 "Weak" Non-Malleability of ZK_{tag}

We consider man-in-the-middle (MIM) adversaries that are simultaneously involved in two different executions of ZK_{tag}. For any $\mathsf{tag}, \tilde{\mathsf{tag}} \in \{1,\ldots,m\}$ we consider a left interaction in which ZK_{tag} is executed with common input $x \in \{0,1\}^n$, and a right interaction in which $ZK_{\tilde{\mathsf{tag}}}$ is executed with common input $\tilde{x} \in \{0,1\}^n$. The witness used by the prover in the left interaction is denoted by w, and the auxiliary input used by the adversary is denoted by z.

LEMMA 3.1. *Let A be a MIM adversary as above, and suppose that $\mathsf{tag} \neq \tilde{\mathsf{tag}}$. Further suppose that $\ell(n) \geq 3n$. Then, there exists a stand alone prover S for $ZK_{\tilde{\mathsf{tag}}}$ and a negligible function $\nu(\cdot)$ so that:*

$$\Pr\left[\mathsf{mim}_V^A(x,\tilde{x},w,z) = 1\right] \;<\; \Pr\left[\mathsf{sta}_V^S(x,\tilde{x},z) = 1\right] \;+\; \nu(|x|)$$

Since $m = \mathrm{poly}(n)$ and $\mathsf{tag} \in \{1,\ldots,m\}$, the tag's are in fact strings of length $\log m = O(\log n)$. Thus, Lemma 3.1 establishes some sort of "weak" non-malleability for the protocols in the family $\{ZK_{\mathsf{tag}}\}_{\mathsf{tag}=1}^m$.

Proof Sketch: The machine S will use a variant of the stand-alone $\mathcal{ZK}$ simulator in order to "internally" generate a view of a left ZK_{tag} interaction for A. The messages sent by A in the right interaction will be forwarded by S to an "external" honest verifier V for $ZK_{\tilde{\mathsf{tag}}}$ whose replies will be then fed back to A. Since the view produced by the $\mathcal{ZK}$ simulator is indistinguishable from A's actual interactions with an honest left prover, the probability that S manages to convince V will be negligibly close to the probability that A produces a convincing right interaction.

The execution of S (with one specific scheduling of messages) is depicted in Figure 5 below. In order to differentiate between the left and right interactions, messages m in the right interaction are labelled as $\tilde{m}$. Stage 2 messages in the left and right interactions are denoted u and $\tilde{u}$ respectively.

The main hurdle in implementing S is in making the simulation of the left interaction work. The problem is that the actual code of the verifier whose view we are simulating is only partially available to S. This is because the messages sent by A in the left interaction also depend on the messages A receives in the right interaction. These messages are sent by an "external" V, and V's code (randomness) is not available to S.

Technically speaking, the problem is implied by the fact that the values of the r_i's do not necessarily depend only on the corresponding c_i, but rather may also depend on the "external" right messages $\tilde{r_i}$. Thus, setting $\Pi = A$ and $y = c_i$ in the simulation will not be sufficient, since in some cases it is simply not true that $r_i = A(c_i)$.

[5]As a side remark, we mention that the weak proof of knowledge property of the $WIUARG$ is not sufficient for our purposes. To guarantee the "traditional" proof of knowledge property, we will have to make use of a "specialized" version of $WIUARG$s (further details in the full version).

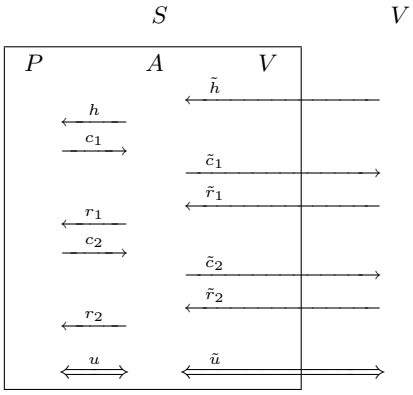

Figure 5: The "stand-alone" prover S for $ZK_{\tilde{\mathsf{tag}}}$.

Intuitively, the most difficult case to handle is the one in which $\tilde{r_1}$ is contained in Slot 1 of ZK_{tag} and $\tilde{r_2}$ is contained in Slot 2 of ZK_{tag} (as in Figure 5 above). In this case $r_i = A(c_i, \tilde{r_i})$ and so $r_i = A(c_i)$ does not hold for either $i \in \{1,2\}$. As a consequence, the simulator will not be able to produce views of convincing Stage 2 interactions with A. In order to overcome the difficulty, we will use the fact that for a given instance $\langle h, c_i, r_i\rangle$, the string c_i is short enough to be "augmented" by $\tilde{r_i}$ while still satisfying the relation R_{sim}.

Specifically, as long as $|c_i| + |\tilde{r_i}| \leq |r_i| - n$ the relation R_{sim} can be satisfied by setting $y = (c_i, \tilde{r_i})$. This guarantees that indeed $\Pi(y) = r_i$. The crux of the argument lies in the following claim (proof omitted).

CLAIM 3.2. *Suppose that $\mathsf{tag} \neq \tilde{\mathsf{tag}}$. Then, there exists $i \in \{1,2\}$ so that $|\tilde{r_i}| \leq |r_i| - \ell(n)$.*

By setting $y = (c_i, \tilde{r_i})$ for the appropriate i, the simulator is thus always able to satisfy R_{sim} for some $i \in \{1,2\}$. This is because the "auxiliary" string y used in order to enable the prediction of r_i is short enough to pass the inspection at Condition 1 of R_{sim} (i.e., $|y| = |c_i| + |\tilde{r_i}| \leq |r_i| - n$).[6]

Once R_{sim} can be satisfied, the simulator is able to produce views of convincing interactions that are computationally indistinguishable from real left interactions.[7]

The success probability of S. The success of S relies on the fact that the view of an honest $ZK_{\tilde{\mathsf{tag}}}$ verifier in right interactions with A is indistinguishable from the view of the $ZK_{\tilde{\mathsf{tag}}}$ verifier in stand-alone interactions with S. Since the acceptance bit of the verifier is a polynomial-time computable function of its view, we have that any non-negligible gap between the success probabilities of A and S directly translates into a non-negligible advantage in distinguishing between the corresponding verifier views. Thus, S convinces the honest $ZK_{\tilde{\mathsf{tag}}}$ verifier with probability that is negligibly close to the probability that A convinces the honest $ZK_{\tilde{\mathsf{tag}}}$ right verifier. ■

[6]This follows from the fact that $\ell(n) \geq 3n$ and $|c_i| = 2n$.

[7]In the above discussion we have been implicitly assuming that $\tilde{h}, \tilde{u}$ are not contained in the two slots of ZK_{tag} (where $\tilde{h}$ denotes the hash function in the right interaction and $\tilde{u}$ denotes the sequence of messages sent in the right $WIUARG$). The case in which $\tilde{h}, \tilde{u}$ are contained in the slots can be handled by setting $\ell(n) \geq 4n$, and by assuming that both $|\tilde{h}|$ and the total length of the messages sent by the verifier in the $WIUARG$ is at most n (which is indeed true [3]).

3.3 "Many-to-One" Non-Malleability of ZK_{tag}

We next consider what happens when a man-in-the-middle adversary is simultaneously involved in the verification of *many* different (parallel) executions of ZK_{tag} on the left while proving a *single* interaction on the right. As it turns out, as long as the number of left executions is bounded in advance, we can actually guarantee non-malleability even in this (more demanding) scenario.[8]

For any $\mathsf{tag}_1, \ldots, \mathsf{tag}_n, \tilde{\mathsf{tag}} \in \{1, \ldots, m\}$ we consider a left interaction in which $ZK_{\mathsf{tag}_1}, \ldots, ZK_{\mathsf{tag}_n}$ are executed in parallel with common input $x \in \{0,1\}^n$, and a right interaction in which $ZK_{\tilde{\mathsf{tag}}}$ is executed with common input $\tilde{x} \in \{0,1\}^n$. The witness used by the prover in the left interaction is denoted by w, and the auxiliary input used by the adversary is denoted by z.

LEMMA 3.3. *Let A be a MIM adversary as above, and suppose that $\mathsf{tag}_j \neq \tilde{\mathsf{tag}}$ for all $j \in [n]$. Further suppose that $\ell(n) \geq 2n^2 + n$. Then, there exists a stand alone prover S for $ZK_{\tilde{\mathsf{tag}}}$ and a negligible function $\nu(\cdot)$ so that:*

$$\Pr\left[\mathsf{mim}_V^A(x, \tilde{x}, w, z) = 1\right] \; < \; \Pr\left[\mathsf{sta}_V^S(x, \tilde{x}, z) = 1\right] \; + \; \nu(|x|)$$

Proof: We construct a stand-alone $ZK_{\tilde{\mathsf{tag}}}$ prover S for the statement "$\tilde{x} \in L$". As in the proof of Lemma 3.1, we employ a simulation procedure in order to "internally" generate a left view of $ZK_{\mathsf{tag}_1}, \ldots, ZK_{\mathsf{tag}_m}$ for A while forwarding messages from the right interaction to an "external" honest verifier. The specific way in which messages are handled is somewhat different than in the case of Lemma 3.1.

Right interaction: Messages that belong to $ZK_{\tilde{\mathsf{tag}}}$ are forwarded by S to an "external" honest verifier V for $ZK_{\tilde{\mathsf{tag}}}$. V's replies are then fed back to A.

Left interaction: The left interaction messages are generated by n "sub-simulators" $S_1, \ldots, S_n$, where each S_j is responsible for generating the messages of the sub-protocol ZK_{tag_j}. The strategy of each of the S_j's is essentially identical to the simulator used by the simulator from Lemma 3.1. The key differences between the two strategies are:

- The program A_j to which S_j commits in Slot i acts exactly like A, but instead of outputting $r_i^1, \ldots, r_i^n$ it outputs only r_i^j.
- Whenever simulator from Lemma 3.1 uses c_i^j as part of y (where c_i^j is the commitment string used in Slot i of ZK_{tag_j}), the simulator S_j will instead use $(c_i^1, \ldots, c_i^n)$.

As before, the most difficult case to handle is when $\tilde{r}_1$ is contained in Slot 1 of ZK_{tag_j} and $\tilde{r}_2$ is contained in Slot 2 of ZK_{tag_j}. We start by observing that in such a case for both $i \in \{1,2\}$ and all $j \in [n]$, if S_j sets $y = (c_i^1, \ldots, c_i^m, \tilde{r}_i)$ then

$$A_j(y) = A_j(c_i^1, \ldots, c_i^m, \tilde{r}_i) = r_i^j$$

Now, since $\mathsf{tag}_j \neq \tilde{\mathsf{tag}}$ (by Hypothesis) we can invoke Claim 3.2 and infer that there exists $i \in \{1,2\}$ so that $|\tilde{r}_i| \leq |r_i^j| - \ell(n)$. This means that for every $j \in \{1, \ldots, n\}$ the simulator S_j can choose an $i \in \{1,2\}$ so that:

$$\begin{aligned} |y| &= |c_i^1| + \ldots + |c_i^n| + |\tilde{r}_i| \\ &= 2n^2 + |\tilde{r}_i| \\ &\leq 2n^2 + |r_i^j| - \ell(n) \end{aligned}$$

Since by Hypothesis $\ell(n) \geq 2n^2 + n$ we get that $|y| \leq |r_i^j| - n$ and so S_j will always succeed in the simulation.[9] ∎

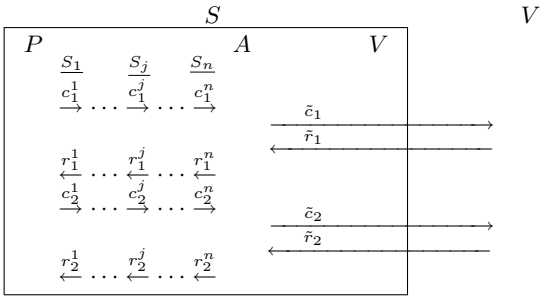

Figure 6: The "stand-alone" prover S for $ZK_{\tilde{\mathsf{tag}}}$.

3.4 "Full-Fledged" Non-Malleable $\mathcal{ZK}$

We now show how to combine the atomic protocols in order to construct a "full-fledged" non-malleable $\mathcal{ZK}$ protocol $nmZK_{\text{TAG}}$. As before, we consider adversaries that are simultaneously involved in two different executions of the protocol. This time however, the tags can be arbitrary strings in $\{0,1\}^n$. This in particular allows the usage of the statement $x \in \{0,1\}^n$ that is being proved as the tag string, yielding a construction of a $\mathcal{ZK}$ argument system that is non-malleable with respect to statements. Protocol $nmZK_{\text{TAG}}$ is described in Figure 7.

Common Input: An instance $x \in \{0,1\}^n$

Parameters: Security parameter 1^n, length parameter $\ell(n)$.

Tag String: $\text{TAG} \in \{0,1\}^n$. Let $\text{TAG} = \text{TAG}_1, \ldots, \text{TAG}_n$.

The protocol:

$P \leftrightarrow V$: For all $i \in \{1, \ldots, n\}$ (in parallel):
Run $ZK_{(i,\, \text{TAG}_i)}$ with common input x and length parameter $\ell(n)$.

V: Accept if and only if all runs are accepting.

Figure 7: A non-malleable $\mathcal{ZK}$ protocol – $nmZK_{\text{TAG}}$.

Notice that $nmZK_{\text{TAG}}$ has a constant number of rounds (since each $ZK_{(i,\, \text{TAG}_i)}$ is constant-round). Also notice that for $i \in [n]$, the length of the tag (i, TAG_i) is

$$|i| + |\text{TAG}_i| = \log n + 1 = \log 2n$$

Viewing (i, TAG_i) as elements in $[2n]$ (i.e. $m = 2n$) we infer that the length of verifier messages in $ZK_{(i,\, \text{TAG}_i)}$ is upper bounded by $m\ell(n) = 2n\ell(n)$. Hence, as long as $\ell(n) = \text{poly}(n)$ the length of verifier messages in $nmZK_{\text{TAG}}$ is upper bounded by $n \cdot m\ell(n) = 2n^2\ell(n) = \text{poly}(n)$.

[8] This is a special case of the bounded concurrency scenario considered in [23, 30, 29]. Since the current scenario is simpler than the [29] one, we are actually able to give a simpler, self-contained, proof.

[9] The messages $\tilde{h}_1, \ldots, \tilde{h}_n$ and $\tilde{u}_1, \ldots, \tilde{u}_n$ can be handled by setting $\ell(n) \geq 3n^2 + n$, and by appending them to y.

3.5 Non-Malleability of $nmZK$

Let $\text{TAG}, \tilde{\text{TAG}} \in \{0,1\}^m$, let $x, \tilde{x} \in \{0,1\}^n$, and let A be the corresponding MIM adversary. We consider a left interaction in which $nmZK_{\text{TAG}}$ is executed with common input $x \in \{0,1\}^n$, and a right interaction in which $nmZK_{\tilde{\text{TAG}}}$ is executed with common input $\tilde{x} \in \{0,1\}^n$. The witness used by the prover in the left interaction is denoted by w, and the auxiliary input used by the adversary is denoted by z.

LEMMA 3.4. *Let A be a MIM adversary as above, and suppose that $\text{TAG} \neq \tilde{\text{TAG}}$ and that $\ell(n) \geq 2n^2 + n$. Then, there exists a stand alone prover S for $nmZK_{\tilde{\text{TAG}}}$ and a negligible function $\nu(\cdot)$ so that:*

$$\Pr\left[\mathsf{mim}_V^A(x, \tilde{x}, w, z) = 1\right] \; < \; \Pr\left[\mathsf{sta}_V^S(x, \tilde{x}, z) = 1\right] \; + \; \nu(|x|)$$

Proof: The construction of the prover S proceeds in two phases. In the first phase, the adversary A is used in order to construct a stand-alone prover $\overline{S}$ for one of the sub-protocols $ZK_{(i,\, \tilde{\text{TAG}}_i)}$, where $i \in [n]$ is such that $\text{TAG}_i \neq \tilde{\text{TAG}}_i$. This $\overline{S}$ is a prover for the statement "$\tilde{x} \in L$" and will have success probability negligibly close to the success probability of A.

In the second phase, the proof of knowledge property of $ZK_{(i,\, \tilde{\text{TAG}}_i)}$ is used in order to extract an $\mathcal{NP}$-witness w for the statement "$\tilde{x} \in L$" from $\overline{S}$. Once the witness w for $\tilde{x}$ is extracted, it is possible to produce convincing (stand-alone) interactions for the full protocol $nmZK_{\tilde{\text{TAG}}}$ by playing the honest prover strategy and using w as witness for "$\tilde{x} \in L$".

Since $\overline{S}$ convinces an honest V with probability that is negligibly close to the success probability of A, and since the proof of knowledge property guarantees extraction with probability that is negligibly close to the convincing probability of $\overline{S}$, the overall success probability of S is negligibly close to the success probability of A.

Constructing the stand alone prover $\overline{S}$. Implementing the second phase is a fairly standard task. We thus focus on the first phase. Let $i \in [n]$ so that $\text{TAG}_i \neq \tilde{\text{TAG}}_i$. In order to construct a stand-alone $ZK_{(i,\, \tilde{\text{TAG}}_i)}$ prover $\overline{S}$ for the statement "$\tilde{x} \in L$", we use A in order to construct a "many-to-one" man-in-the-middle adversary A_i. This A_i verifies that $x \in L$ using $ZK_{(1,\, \text{TAG}_1)}, \ldots, ZK_{(n,\, \text{TAG}_n)}$ on the left, and proves that $\tilde{x} \in L$ using $ZK_{(i,\, \tilde{\text{TAG}}_i)}$ on the right. A_i has the same success probability as A.

Notice that for every $j \in [n]$, it holds that $(j, \text{TAG}_j) \neq (i, \tilde{\text{TAG}}_i)$ (since either $j \neq i$ or $\text{TAG}_i \neq \tilde{\text{TAG}}_i$). Thus, the tags (j, TAG_j) used in the left interaction of A_i are *all* different than the tag $(i, \tilde{\text{TAG}}_i)$ used by A_i in the right interaction. This means that, once we establish the existence of a "many-to-one" MIM A_i as above, we can invoke Lemma 3.3 to obtain a stand alone prover $\overline{S}$ that has the same success probability as A, and thus complete the proof of Lemma 3.4.

Constructing the "many-to-one" adversary A_i. The adversary A_i invokes A as a subroutine. It feeds A with messages in the following way (see Figure 8, where messages from circled sub-protocols are forwarded externally).

Right interaction: Messages that belong to $ZK_{(i,\, \tilde{\text{TAG}}_i)}$ are forwarded to an "external" honest verifier V for $ZK_{(i,\, \tilde{\text{TAG}}_i)}$. V's replies are then fed back to A. Messages that belong to $\{ZK_{(j,\, \tilde{\text{TAG}}_j)}\}_{j \neq i}$ are handled "internally". Specifically, A_i plays the role of the honest $ZK_{(j,\, \tilde{\text{TAG}}_j)}$ verifier in each of the right-interaction sub-protocols by feeding A with randomly chosen verifier messages $\tilde{r}_1^j, \tilde{r}_2^j$ and by verifying that the sub-protocol is accepting.

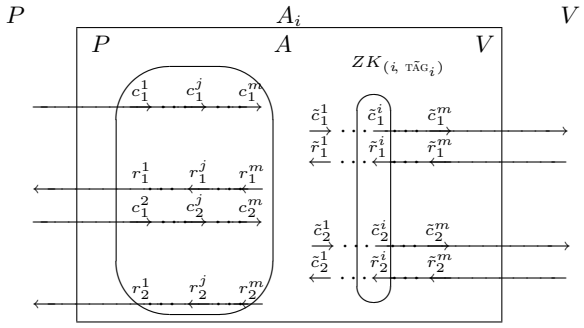

Figure 8: The "Many-to-one" MIM adversary A_i.

Left interaction: The left interaction messages are all forwarded to an "external" left prover P for the sessions $ZK_{(1,\, \text{TAG}_1)}, \ldots, ZK_{(n,\, \text{TAG}_n)}$. P's replies are fed back to A.

It can be seen that A's view of left and right hand side messages (as generated by A_i) is distributed exactly as the messages it obtains in real executions. Thus, the success probability of A_i is at least as high as that of A. ∎

4. NON-MALLEABLE COMMITMENTS

Using the construction of $nmZK_{\text{TAG}}$ as a subroutine, we present a simple construction of non-malleable commitments. We focus on (statistically-binding) commitments that are non-malleable w.r.t. commitment. At the end of this section we outline the construction of statistically-hiding commitments that are non-malleable w.r.t. opening.

Let Com be a statistically binding commitment scheme (for simplicity, assume that Com is non-interactive). Consider the following protocol for non-malleable commitments.

Security Parameter: 1^k.

String to be committed to: $v \in \{0,1\}^k$.

Commit Phase:

$C \rightarrow R$: Pick $s \in \{0,1\}^n$ and send $c = \text{Com}(v; s)$.

$C \leftrightarrow R$: Prove using $nmZK_c$ that there exist $v, s \in \{0,1\}^k$ so that $c = \text{Com}(v; s)$.

R: Verify that $nmZK_c$ is accepting.

Reveal Phase:

$C \rightarrow R$: Send v, s.

R: Verify that $c = \text{Com}(v; s)$.

Figure 9: A non-malleable commitment with respect to commitment - $nm\mathcal{C}$.

The statistical binding property of $nm\mathcal{C}$ follows from the statistical binding of Com. The computational hiding property follows from the computational hiding of Com, as well as from the (stand alone) $\mathcal{ZK}$ property of $nmZK_c$. Consider now a man-in-the-middle adversary A that, given access to a left interaction in which v is committed to, succeeds in committing to a value $\tilde{v} \neq v$ that satisfies $\mathcal{R}(v, \tilde{v}) = 1$. To prove non-malleability, one needs to construct a stand-alone committer S, that manages to commit to $\tilde{v}$ with essentially the same probability as A.

The high-level idea is to view A as a man in the middle adversary A' for the protocol $nmZK_c$, where the statement x proved in the left interaction equals $c = \text{Com}(v)$

and the statement $\tilde{x}$ proved in the right interaction equals $\tilde{c} = \mathrm{Com}(\tilde{v})$. By definition of non-malleable zero-knowledge, there exists a stand-alone prover S' that obtains c as input and produces convincing $nmZK_{\tilde{c}}$ interactions.

Notice that the stand-alone committer for $nm\mathcal{C}$, that we are supposed to construct does not have access to c. To get around this problem we let the stand-alone committer invoke S' on input $c = \mathrm{Com}(0^k)$ (rather than on input $c = \mathrm{Com}(v)$). Since the commitment Com is hiding, it will then follow that the statement $\tilde{c} = \mathrm{Com}(\tilde{v})$ that S' proves on the right will still satisfy $\mathcal{R}(v, \tilde{v}) = 1$.

Due to a subtle technical issue the above idea does not work. The reason for this is that the man in the middle adversary A' might *choose* the statement $\tilde{c}$ to be proved on the right (and thus the value committed to) adaptively. In fact, we need to make sure that the actual values committed by the stand-alone committer are indistinguishable from the values committed to by A. Note that it is not sufficient to only require that the statements proved are indistinguishable, since the statements are just commitment to these values, and commitments are always indistinguishable! We hint that we solve this problem by resorting to a statistical $\mathcal{ZK}$ version of our non-malleable $\mathcal{ZK}$ protocol. By doing this, we can instead show that the statements proved are *statistically* indistinguishable (which thus implies that the actual values committed to are indistinguishable).

Statistically Hiding Commitments. We outline the construction of statistically hiding commitments that are non-malleable with respect to opening. The protocol proceeds as follows: In the commit phase, the committer simply send a (possibly malleable) statistically hiding commitment $c = Com(v; s)$ to the receiver. In the reveal phase, the committer sends the value v and additionally gives a non-malleable $\mathcal{ZK}$ proof of knowledge of a value s' such that $c = Com(v; s')$. We note that, somewhat ironically, in the above construction it is actually sufficient to use a non-malleable proof of knowledge that is *computational* $\mathcal{ZK}$ (whereas in the construction of statistically binding commitments we rely on a statistical $\mathcal{ZK}$ protocol).

Acknowledgments. We are grateful to Johan Håstad and Moni Naor for many helpful conversations and great advice. Thanks to Boaz Barak for useful clarifications of his works. The second author would also like to thank Marc Fischlin, Rosario Gennaro, Yehuda Lindell and Tal Rabin for insightful discussions regarding non-malleable commitments. Finally, thanks to Oded Goldreich for useful comments on an earlier version of this work.

5. REFERENCES

[1] B. Barak. How to go Beyond the Black-Box Simulation Barrier. In *42nd FOCS*, pages 106–115, 2001.

[2] B. Barak. Constant-Round Coin-Tossing or Realizing the Shared Random String Model. In *43rd FOCS*, p. 345-355, 2002.

[3] B. Barak and O. Goldreich. Universal Arguments and their Applications. *17th CCC*, pages 194–203, 2002.

[4] B. Barak and Y. Lindell. Strict Polynomial-Time in Simulation and Extraction. In *34th STOC*, p. 484–493, 2002.

[5] R. Canetti and M. Fischlin. Universally Composable Commitments. In *Crypto2001*, Springer LNCS 2139, pages 19–40, 2001.

[6] I. Damgård and J. Groth. Non-interactive and Reusable Non-Malleable Commitment Schemes. In *35th STOC*, pages 426-437, 2003.

[7] I. Damgård, T. Pedersen and B. Pfitzmann. On the Existence of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures. In *Crypto93*, pages 250–265, 1993.

[8] A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano and A. Sahai. Robust Non-interactive Zero Knowledge. In *CRYPTO 2001*, pages 566-598, 2001.

[9] G. Di Crescenzo, J. Katz, R. Ostrovsky and A. Smith. Efficient and Non-interactive Non-malleable Commitment. In *EUROCRYPT 2001*, pages 40-59, 2001.

[10] G. Di Crescenzo, Y. Ishai and R. Ostrovsky. Non-Interactive and Non-Malleable Commitment. In *30th STOC*, pages 141-150, 1998

[11] D. Dolev, C. Dwork and M. Naor. Non-Malleable Cryptography. *SIAM Jour. on Computing*, Vol. 30(2), pages 391–437, 2000. Preliminary version in *23rd STOC*, pages 542-552, 1991

[12] U. Feige, D. Lapidot and A. Shamir. Multiple Noninteractive Zero Knowledge Proofs under General Assumptions. *Siam Jour. on Computing 1999*, Vol. 29(1), pages 1-28.

[13] U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. In *22nd STOC*, p. 416–426, 1990.

[14] M. Fischlin and R. Fischlin. Efficient Non-malleable Commitment Schemes. *In CRYPTO 2000*, Springer LNCS Vol. 1880, pages 413-431, 2000.

[15] O. Goldreich. *Foundation of Cryptography – Basic Tools.* Cambridge University Press, 2001.

[16] O. Goldreich and Y. Lindell. Session-Key Generation Using Human Passwords Only. In *CRYPTO 2001*, p. 408-432, 2001.

[17] O. Goldreich, S. Micali and A. Wigderson. Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *JACM*, Vol. 38(1), pages 691–729, 1991.

[18] O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. In *19th STOC*, pages 218–229, 1987.

[19] S. Goldwasser and S. Micali. Probabilistic Encryption. *JCSS*, Vol. 28(2), pages 270-299, 1984.

[20] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Jour. on Computing*, Vol. 18(1), pages 186–208, 1989.

[21] J. Håstad, R. Impagliazzo, L.A. Levin and M. Luby. Construction of Pseudorandom Generator from any One-Way Function. *SIAM Jour. on Computing*, Vol. 28 (4), pages 1364–1396, 1999.

[22] J. Kilian. A Note on Efficient Zero-Knowledge Proofs and Arguments. In *24th STOC*, pages 723–732, 1992.

[23] Y. Lindell. Bounded-Concurrent Secure Two-Party Computation Without Setup Assumptions. In *34th STOC*, pages 683–692, 2003.

[24] P. D. MacKenzie, M. K. Reiter, K. Yang: Alternatives to Non-malleability: Definitions, Constructions, and Applications. TCC 2004, pages 171-190, 2004.

[25] S. Micali. CS Proofs. *SIAM Jour. on Computing*, Vol. 30 (4), pages 1253–1298, 2000.

[26] M. Naor. Bit Commitment using Pseudorandomness. *Jour. of Cryptology*, Vol. 4, pages 151–158, 1991.

[27] M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. In *21st STOC*, pages 33–43, 1989.

[28] M. Nguyen and S. Vadhan. Simpler Session-Key Generation from Short Random Passwords. In *1st TCC*, p. 428-445, 2004.

[29] R. Pass. Bounded-Concurrent Secure Multi-Party Computation with a Dishonest Majority. In *36th STOC*, 2004, pages 232-241, 2004.

[30] R. Pass and A. Rosen. Bounded-Concurrent Secure Two-Party Computation in a Constant Number of Rounds. In *34th FOCS*, pages 404-413, 2003.

[31] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. *In 40th FOCS*, pages 543-553, 1999.