Identity-Based Zero-Knowledge

Jonathan Katz[1]* Rafail Ostrovsky[2]** Michael O. Rabin[3]

[1] Dept. of Computer Science, University of Maryland
jkatz@cs.umd.edu
[2] Dept. of Computer Science, UCLA
rafail@cs.ucla.edu
[3] Dept. of Computer Science, Harvard University
rabin@deas.harvard.edu

Abstract. We introduce and define the notion of *identity-based zero-knowledge*, concentrating on the non-interactive setting. In this setting, our notion allows any prover to widely disseminate a proof of a statement while protecting the prover from plagiarism in the following sense: although proofs are *transferable* (i.e., publicly verifiable), they are also *bound* to the identity of the prover in a way which is recognizable to any verifier. Furthermore, an adversary is unable to change this identity (i.e., to claim the proof as his own, or to otherwise change the authorship), unless he could have proved the statement on his own.

While we view the primary contribution of this work as a formal definition of the above notion, we also explore the relation of this notion to that of *non-malleable (non-interactive) zero-knowledge*. On the one hand, we show that these two notions are incomparable: that is, there are proof systems which are non-malleable but not identity-based, and vice versa. On the other hand, we show that a proof system of either type essentially implies a proof system of the other type.

* Work done while this author was at Columbia University.
** Supported in part by a gift from Teradata and Intel equipment grant

1 Introduction

One of the motivations behind the introduction of the fundamental notion of zero-knowledge (ZK) proof systems by Goldwasser, Micali, and Rackoff [9] was to allow a prover to convince a verifier about the validity of a theorem without enabling the verifier to later convince someone else [2]. When viewing ZK proofs in this way, one sees that a primary concern of such proofs is to prevent plagiarism; in other words, the prover wishes to prevent the verifier from learning some valuable information from the proof and later claiming the proof as his own (without properly referencing the original prover).

We remark that the above concerns are handled, to some extent, by ZK proofs in the *interactive* setting. Here, we have a prover P and a (possibly malicious) verifier V who will (at some *later* point) try to convince a second verifier V'. Since the transcript of the interaction between P and V can be simulated, by definition of zero-knowledge, a copy of the proof transcript will not be convincing to V'. Additionally, if V and V' interact *after* completion of the interaction between P and V, the zero-knowledge property implies that V gains no advantage in trying to convince V'.

Of course, the concern remains that V might interact with V' *while* interacting with P (i.e., act as man-in-the-middle). A related concern, in the public-key setting, was considered by Jakobsson, Sako, and Impagliazzo [10] (see also the related work by Cramer and Damgård [5]) who introduce proofs meant to convince only a single, designated verifier. Note that such a notion, if extended to the non-interactive setting, would fundamentally *limit* the widespread dissemination of proofs; on the other hand, frequently one would like to *disseminate* proofs as widely as possible (e.g., to announce results to the scientific community).

Indeed, *non-interactive* ZK (NIZK) proof systems introduced by Blum, Feldman, and Micali [3] paradoxically allow (in the presence of a common-random string available to all parties) the widespread dissemination of zero-knowledge proofs. However, although NIZK proofs "hide" the witness to the truth of the theorem, NIZK proofs do not seem to offer any guarantees against plagiarism. That is, if P gives a non-interactive proof π to V, this proof is still convincing when V transfers it to V'. Note that, here, V's interaction with V' does not need to be simultaneous with his interaction with P, since π can be copied and stored until needed. Indeed, one *advantage* of NIZK proofs is that they are *transferable* and can be passed from verifier to verifier yet still remain a convincing proof of the theorem claimed. However, NIZK proofs are not *bound* in any way to the original discoverer of the proof. That is, once a prover gives a convincing NIZK proof to the first verifier, the verifier can claim that proof as his own!

Ideally, one would like to retain the ability to disseminate proofs as widely as possible while maintaining clear (and unalterable) information about who actually created the proof. To protect the original prover P, some mechanism needs to be developed which ensures that (1) if the proof is passed from verifier to verifier it remains a convincing proof; yet (2) if the proof is simply copied, V' will recognize that P was the one who actually composed the proof. Furthermore,

(3) any adversary V' should be unable to modify the proof to make it appear as though he (V') actually composed the proof.

Toward this end, we formally define the notion of *identity-based* proof systems which satisfy the security requirements implied by the discussion above. We also show a simple and provably-secure construction of an identity-based scheme achieving the stated requirements, starting from any non-malleable zero-knowledge scheme [7]. In our construction, we do not rely on public-key infrastructure.

1.1 Related Work

The notion informally outlined above is related to the notion of non-malleability as introduced by Dolev, Dwork, and Naor [7]. Yet, these two notions are technically very different and non-malleability does not automatically imply security in the sense esdescribed above. Specifically, we note that although Dolev, et al. discuss a way to simplify the construction of non-malleable cryptosystems when identities are present, they *do not* formally define the idea of "binding" or "linking" an identity with a proof. One can also see that a non-malleable NIZK proof system does *not* achieve the security desired in our setting; in particular, the definition of non-malleability does not protect against copying (something we are explicitly concerned with here), and known non-malleable NIZK proof systems [7, 16, 6] do not consider the notion of having the prover's identity associated with the proof. Furthermore, an identity-based proof system (as defined below) is not necessarily non-malleable.

We show, however, an underlying connection between (non-interactive) non-malleable and identity-based proof systems: our construction of an identity-based proof system uses any non-malleable proof system as a building block, and we show how any identity-based system can be used to construct a non-malleable scheme without much additional complexity.

Since the original version of this manuscript was written, an improved construction of (interactive) non-malleable zero-knowledge has been proposed [1]. See also the work of [11, 15] which, *inter alia*, construct identity-based zero-knowledge proofs for identities of *logarithmic* length which are fixed *a priori* (note, however, that neither of these works formally define the notion of identity-based zero knowledge). Also related to this work is recent work of Pass [14] which is concerned with the transferability of NIZK proofs, but is not explicitly concerned with associating proofs with identities. We remark also that NIZK proof systems in the universally composable (UC) framework [4] incorporate identities to some extent (as a consequence of the definition of the UC framework), but not quite the way we do so here. For one thing, in the UC framework there is no notion of "transferability" of NIZK proofs (indeed, such proofs inherently *cannot* be transfered in the UC framework), and there is no direct requirement that identities be "extractable" from proofs. Nevertheless, known constructions of NIZK proofs in the UC framework do achieve our definition.

The complementary notion of identity-based *interactive* proof systems is also of interest. Although the notion seems not to have been considered explicitly in the early work on non-malleability [7] (and no formal definition of such a notion has previously appeared), the techniques given there may be adapted to yield identity-based proof systems in the interactive setting. Our results below, showing that identity-based proof systems can be used to construct non-malleable proof systems, extend to the interactive setting as well. In particular, the methods of Theorem 2 show that the existence of an r-round identity-based (interactive) proof system implies the existence of an r-round non-malleable proof system, indicating that the complexity of identity-based systems is not any lower than non-malleable ones.

2 Definitions

We begin with the standard definition of (adaptive) NIZK proof systems, with one additional feature: The prover algorithm $\mathcal{P}$ takes as one of its inputs a string *id* representing an identity. The verification algorithm $\mathcal{V}$, on input a proof π, outputs both a bit denoting acceptance/rejection of the proof as well as a string *id* indicating which party it believes was the one who generated the proof. The intention is that the identity information *id* is embedded in π by the prover (in some way) such that it can be extracted efficiently by the verifier $\mathcal{V}$. The following definition deals simply with the correctness of this process; however, this embedding of the *id* will be crucial when we define security for an identity-based scheme further below.

Definition 1. *$\Pi = (p, q, \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ is an* NIZK proof system with extractable identity *for language L with witness relation R if p, q are polynomial (with $q(k) = \omega(\log k)$) and $\mathcal{P}, \mathcal{V}$, and $\mathcal{S}$ are* PPT *algorithms such that:*

1. **(Completeness):** *For all $x \in L$, all w such that $(x, w) \in R$, all $\sigma \in \{0,1\}^{p(|x|)}$, and all* $\mathrm{id} \in \{0,1\}^{q(|x|)}$*, we have $\mathcal{V}(x, \mathcal{P}(x, w, \mathrm{id}, \sigma), \sigma)_1 = \mathsf{true}$ (where $\mathcal{V}(\cdot,\cdot,\cdot)_1$ represents the first component of $\mathcal{V}$'s output).*
2. **(Extractable identities):** *For all $x \in L$, all w such that $(x, w) \in R$, all $\sigma \in \{0,1\}^{p(|x|)}$, and all* $\mathrm{id} \in \{0,1\}^{q(|x|)}$*, we have $\mathcal{V}(x, \mathcal{P}(x, w, \mathrm{id}, \sigma), \sigma)_2 = \mathrm{id}$.*
3. **(Soundness):** *For all unbounded algorithms $\mathcal{P}'$, if $\sigma \in \{0,1\}^{p(|x|)}$ is chosen randomly, the probability that $\mathcal{P}'(\sigma)$ outputs (x, π) such that $\mathcal{V}(x, \pi, \sigma)_1 =$ true and $x \notin L$ is negligible.*
4. **(Zero-knowledge):** *For all $x \in L$, all w such that $(x, w) \in R$, and all* $\mathrm{id} \in \{0,1\}^{q(|x|)}$*, the following distributions are computationally indistinguishable (where $k \stackrel{\mathrm{def}}{=} p(|x|)$):*

$$\left\{\sigma \leftarrow \{0,1\}^k; \pi \leftarrow \mathcal{P}(x, w, \mathrm{id}, \sigma) : (\sigma, \pi)\right\}$$
$$\left\{(\sigma, s) \leftarrow \mathcal{S}_1(1^k); \pi \leftarrow \mathcal{S}_2(x, \mathrm{id}, s) : (\sigma, \pi)\right\}.$$

We remark that our results extend to a stronger (robust) notion of Non-Interactive Zero-knoweldge, considered in [6], where σ is identical in the real interaction and in the simulation[4].

[4] That is, the two experiemnts are as follows: First, generate $(\sigma, s) \leftarrow \mathcal{S}_1(1^k)$, where we require the distribution on σ to be uniform, and them we require that the

We further remark that the above definition says nothing about a prover who chooses to use some arbitrary identity (i.e., as opposed to their own identity) when constructing a proof. Indeed, this cannot be prevented without the additional assumption of some infrastructure who "binds" physical entities to identities.

Following [8, 6], we extend the above definition to allow for simulation of any polynomial number of proofs:

Definition 2. *$\Pi = (p, q, \mathcal{P}, \mathcal{V}, \mathcal{S})$ is an* unbounded NIZK proof system with extractable identity *for language L with witness relation R if Π is an NIZK proof system with extractable identity and for all* PPT *A, we have that:*

$$\left|\Pr[\mathsf{Expt}_{A,\Pi}(k) = 1] - \Pr[\mathsf{Expt}^{\mathcal{S}}_{A,\Pi}(k) = 1]\right|$$

is negligible; where:

$\mathsf{Expt}_{A,\Pi}(k)$:	$\mathsf{Expt}^{\mathcal{S}}_{A,\Pi}(k)$:
$\sigma \leftarrow \{0,1\}^k$	$(\sigma, s) \leftarrow \mathcal{S}_1(1^k)$
return $A^{\mathcal{P}(\cdot,\cdot,\cdot,\sigma)}(\sigma)$	return $A^{\mathcal{S}'(\cdot,\cdot,\cdot,s)}(\sigma)$

and $\mathcal{S}'(x, w, \mathrm{id}, s) \stackrel{\mathrm{def}}{=} \mathcal{S}_2(x, \mathrm{id}, s)$ (we assume, above, that if x, w, id is a query of A, then $(x, w) \in R$; note that this can be verified easily).

We now turn to the definition of security (as sketched in the Introduction) for this setting. Informally, we want to ensure that an adversary cannot take a proof π given by a prover $\mathcal{P}(x, w, id, \sigma)$ and convert it to a proof π' (for the same theorem) such that $\mathcal{V}(x, \pi', \sigma)_1 = \mathsf{true}$, yet $\mathcal{V}(x, \pi', \sigma)_2 \neq id$. In fact, our definition is even stronger as it rules out the possibility of an adversary claiming *any* proof with respect to a "new" identity unless (informally) the adversary could have proved such a statement on its own. More specifically, anything the adversary can prove with respect to a *new* identifier after seeing any (polynomial) number of proofs $\pi_1, \ldots, \pi_\ell$ given by provers with (possibly) multiple identities (adaptively chosen by the adversary), could have been proved by the adversary without seeing these proofs.[5] Our definition is based on that of [6], who present definitions in the context of non-malleable NIZK. However, we stress (as pointed out previously) that non-malleable and identity-based proof systems are incomparable, in the sense that a proof system satisfying one need not satisfy the other. We make this explicit in Lemmas 1 and 2, below.

Definition 3. *Let $\Pi = (p, q, \mathcal{P}, \mathcal{V}, \mathcal{S})$ be an unbounded NIZK proof system with extractable identity for language L with witness relation R_L. We say that Π is an* identity-based *NIZK proof system for L if there exists an extractor* Ext *such*

---

following two distributions are indistinguishable: $\{\pi \leftarrow \mathcal{P}(x, w, id, \sigma) : (\sigma, \pi)\}$ and $\{\pi \leftarrow \mathcal{S}_2(x, id, s) : (\sigma, \pi)\}$.

[5] When we say that x "could have been proved by the adversary", we mean that an actual witness w for x can be extracted from the adversary (see Definition 3).

that, for all PPT *adversaries A and all poly-time relations R, the following is negligible:*

$$\left|\Pr[\mathsf{ExptID}^{\mathcal{S}}_{A,R,\Pi}(k)] - \Pr[\mathsf{ExptID}'_{A,R,\Pi}(k)]\right|,$$

where:

$\mathsf{ExptID}^{\mathcal{S}}_{A,R,\Pi}(k)$: $(\sigma, s) \leftarrow \mathcal{S}_1(1^k)$ $(x, \pi, \mathsf{aux}) \leftarrow A^{\mathcal{S}_2(\cdot,\cdot,s)}(\sigma)$ *Let I be the list of identities queried by A* return true *iff* $\mathcal{V}(x, \pi, \sigma)_1 = \mathsf{true}$ and $\mathcal{V}(x, \pi, \sigma)_2 \notin I$ and $R(x, \mathsf{aux}) = 1$	$\mathsf{ExptID}'_{A,R,\Pi}(k)$: $(x, w, \mathsf{aux}) \leftarrow \mathsf{Ext}^A(1^k)$ return true *iff* $(x, w) \in R_L$ and $R(x, \mathsf{aux}) = 1$

(we assume, above, that if x, id is a query of A, then $x \in L$).

We remark that the above definition actually corresponds to an NIZK proof *of knowledge* (in the sense that Ext "extracts" a witness from A). It is possible to relax the definition (and our constructions) for the case of NIZK *proofs* but we omit the details here.

The next two lemmas indicate that identity-based schemes and non-malleable schemes are incomparable. For self-containment, we include in Appendix A a definition of non-malleable NIZK proof systems (adapted from [6]).

Lemma 1. *Assuming the existence of trapdoor permutations and*[6] *dense cryptosystems, there exists a proof system Π which is a non-malleable NIZK proof system yet is not an identity-based NIZK proof system.*

Proof (sketch). Consider, for example, the non-malleable schemes given in [6]. In these schemes, there is no notion of prover identities at all, and thus no connection whatsoever between a proof and the identity of the prover.

Lemma 2. *Assuming the existence of trapdoor permutations and dense cryptosystems, there exists a proof system Π which is an identity-based NIZK proof system yet is not a non-malleable NIZK proof system.*

Proof (sketch). An identity-based NIZK proof system only prevents an adversary from modifying an existing proof to yield a proof which is not associated with any of the legitimate provers, yet it may be possible for an adversary to modify an existing proof to yield a proof of a different statement (but in the name of the original prover). In particular, consider the construction Π of an identity-based proof system given in Section 3. Define proof system Π' in which a prover appends an extra bit to the end of every proof which is ignored by the verifier.

[6] The assumption of dense cryptosystems is needed only for the definitions as currently presented. By relaxing the definitions to consider *proofs* rather than *proofs of knowledge* (see the remark following Def. 3) we can, following [6, Footnote 6], base our results on the assumption of trapdoor permutations alone.

Since flipping the final bit of a valid proof yields a new valid proof, clearly the scheme is not non-malleable. Yet it is not difficult to show that Π' remains an identity-based proof system

3 An Identity-Based Proof System

We construct an identity-based NIZK proof system Π starting from any non-malleable NIZK proof system $\widetilde{\Pi} = (\tilde{p}, \widetilde{\mathcal{P}}, \widetilde{\mathcal{V}}, \widetilde{\mathcal{S}})$ for languages in $\mathcal{NP}$. We make the additional assumption that $\widetilde{\Pi}$ has *uniquely applicable proofs* (see [16]). This means that, for all $x, x'\pi, \sigma$ with $x \neq x'$, if $\widetilde{\mathcal{V}}(x, \pi, \sigma) = \mathsf{true}$ then we must have $\widetilde{\mathcal{V}}(x', \pi, \sigma) = \mathsf{false}$. Known techniques for constructing non-malleable NIZK proof systems [16, 6] give proof systems which have uniquely applicable proofs.

The intuition behind our construction[7] of proof system Π for language $L \in \mathcal{NP}$ is as follows: an identity-based proof of the theorem $x \in L$ using identity *id* will consist of a proof (under $\widetilde{\Pi}$) of the theorem that *either* $x \in L$ *or* (a portion of) the common random string specifies a commitment to *id*. A formal description follows:

- **Common random string.** Let $k \stackrel{\text{def}}{=} |x|$. Define $p(k) \stackrel{\text{def}}{=} \tilde{p}(6k^2 + 2k) + 6k^2$. The random string $\sigma \in \{0,1\}^{p(k)}$ is parsed as $\sigma_1 \circ \sigma_2$, with $|\sigma_1| = 6k^2$. String σ_1 is parsed as $r_1, c_1, \ldots, r_k, c_k$ where $|r_i| = |c_i| = 3k$, for all i. Pair (r_i, c_i) will be viewed as a bit commitment as follows [12]: let $G : \{0,1\}^k \to \{0,1\}^{3k}$ be a pseudorandom generator. If $c_i = G(y)$ for some y, then (r_i, c_i) represents a 0. If $c_i \oplus r_i = G(y)$ for some y, then (r_i, c_i) represents a 1. Note that with all but negligible probability over random choice of r_i, c_i, the pair will not represent a valid commitment to any value.
- **Prover strategy.** Any $q(k) = \mathit{poly}(k)$ is possible; for simplicity, we set $q(k) \stackrel{\text{def}}{=} k$. Define language $\widetilde{L} \in \mathcal{NP}$ as consisting of tuples (x, id), with $|x| = k$ and $|\sigma_1| = 6k^2$, such that at least one of the following is true:
  1. $x \in L$
  2. σ_1 is a commitment (see above) to the k-bit string *id*.

  (Note that $\widetilde{L}$ depends on a fixed value of σ_1. Thus, technically, we should write $\widetilde{L}_{\sigma_1}$; however, we suppress σ_1 in the notation.) Algorithm $\mathcal{P}(x, w, \mathit{id}, \sigma)$, where $\mathit{id} \in \{0,1\}^k$, is defined as follows: First, σ is parsed as $\sigma_1 \circ \sigma_2$. $\mathcal{P}$ sets $\tilde{x} := (x, \mathit{id})$ and runs $\widetilde{\mathcal{P}}(\tilde{x}, w, \sigma_2)$, where $\widetilde{\mathcal{P}}$ is the proof system for language $\widetilde{L}$. Let $\tilde{\pi}$ be the output of $\widetilde{\mathcal{P}}$. The output of $\mathcal{P}$ is then $\pi := (\mathit{id}, \tilde{\pi})$.
- Verifier strategy. $\mathcal{V}(x, (\mathit{id}, \tilde{\pi}), \sigma)$ runs as follows: First, σ is parsed as $\sigma_1 \circ \sigma_2$. The verifier sets $\tilde{x} := (x, \mathit{id})$ and outputs $(\widetilde{\mathcal{V}}(\tilde{x}, \tilde{\pi}, \sigma_2), \mathit{id})$.
- Simulation. We define $(\mathcal{S}_1, \mathcal{S}_2)$ as follows: $\mathcal{S}_1(1^k)$ chooses $\sigma_1 \in \{0,1\}^{6k^2}$ at random and then runs $\widetilde{\mathcal{S}}_1(1^k)$ to generate (σ_2, s). The output of $\mathcal{S}_1$ is (σ, s),

[7] In fact, a simpler construction is possible. Informally, to prove $x \in L$ we first construct the language $L' \stackrel{\text{def}}{=} \{(\mathit{id}, x) \mid x \in L\}$ and then give a non-malleable proof that $(\mathit{id}, x) \in L'$. We omit the details and a proof of security for this construction.

where $\sigma = \sigma_1 \circ \sigma_2$. Algorithm $\mathcal{S}_2(x, id, s)$ sets $\tilde{x} := (x, id)$, and runs $\widetilde{\mathcal{S}}_2(\tilde{x}, s)$ to obtain output $\tilde{\pi}$. Finally, $\mathcal{S}_2$ sets $\pi := (id, \tilde{\pi})$ and outputs π.

The security offered by this construction is described by the following theorem:

Theorem 1. *If $\widetilde{\Pi}$ is a non-malleable NIZK proof system (with uniquely applicable proofs) for $\tilde{L}$, then Π is an identity-based NIZK proof system for L.*

Using [6], we immediately obtain the following corollary:

Corollary 1. *Assuming the existence of trapdoor permutations and dense cryptosystems, there exists an identity-based NIZK proof system for any $L \in NP$.*

We now prove the theorem.

Proof. One-way functions are sufficient for the construction given above; furthermore, the fact that $\widetilde{\Pi}$ is an NIZK proof system for languages outside $\mathcal{BPP}$ implies that one-way functions exist (assuming $\mathcal{NP} \neq \mathcal{BPP}$) [13]. We first show that Π is an NIZK proof system with extractable identity (cf. Definition 1). Completeness and identity extraction are trivial. Soundness of Π follows from the soundness of $\widetilde{\Pi}$ and the observation that, with all but negligible probability over randomly chosen $\sigma = \sigma_1 \circ \sigma_2$, the string σ_1 cannot be interpreted as a commitment to *any* string *id*. Zero-knowledge will follow from the stronger property proved below.

To show that Π is unbounded, consider an arbitrary PPT adversary A (cf. Definition 2). Define $\widetilde{A}$ as follows: on input σ_2, adversary $\widetilde{A}$ generates $\sigma_1 \in \{0,1\}^{6k^2}$ at random and runs $A(\sigma_1 \circ \sigma_2)$. When A submits query (x, w, id), algorithm $\widetilde{A}$ sets $\tilde{x} := (x, id)$ and submits query $(\tilde{x}, w)$ to its oracle. Upon receiving $\tilde{\pi}$ in response, $\widetilde{A}$ returns to A the value $(id, \tilde{\pi})$. Finally, $\widetilde{A}$'s final output is whatever A outputs. Note that:

$$\Pr[\mathsf{Expt}_{A,\Pi}(k) = 1] = \Pr[\mathsf{Expt}_{\widetilde{A},\widetilde{\Pi}}(k) = 1]$$

and

$$\Pr[\mathsf{Expt}^{\mathcal{S}}_{A,\Pi}(k) = 1] = \Pr[\mathsf{Expt}^{\widetilde{\mathcal{S}}}_{\widetilde{A},\widetilde{\Pi}}(k) = 1].$$

Thus, if $\widetilde{\Pi}$ is an unbounded NIZK proof system (cf. Definition 5), Π is an unbounded, identifiable NIZK proof system.

We now prove that Π is an identity-based NIZK proof system. Let A be a PPT adversary, and let R be a poly-time relation. Define $\widetilde{A}$ as follows: on input σ_2, adversary $\widetilde{A}$ generates $\sigma_1 \in \{0,1\}^{6k^2}$ at random and runs $A(\sigma)$, where $\sigma = \sigma_1 \circ \sigma_2$. When A submits query x, id to its oracle for $\mathcal{S}_2$, algorithm $\widetilde{A}$ sets $\tilde{x} := (x, id)$ and submits query $\tilde{x}$ to its oracle for $\widetilde{\mathcal{S}}_2$. Upon receiving $\tilde{\pi}$ in response, $\widetilde{A}$ returns to A the response $(id, \tilde{\pi})$. When A outputs $(x_f, \pi_f = (id_f, \tilde{\pi}_f), \mathsf{aux})$, algorithm $\widetilde{A}$ checks whether id_f appears in the list of identities queried by A. If it does not, $\widetilde{A}$ outputs $(\tilde{x}_f = (x_f, id_f), \tilde{\pi}_f, \mathsf{aux})$; otherwise, $\widetilde{A}$ outputs $\perp$.

Furthermore, define relation $\widetilde{R}$ as follows: $\widetilde{R}(\tilde{x} = (x, id), \mathsf{aux}) = 1$ if and only if $R(x, \mathsf{aux}) = 1$.

We claim that:

$$\Pr[\mathsf{ExptID}^{\mathcal{S}}_{A,R,\Pi}(k)] = \Pr[\mathsf{ExptNM}^{\widetilde{\mathcal{S}}}_{\widetilde{A},\widetilde{R},\widetilde{\Pi}}(k)]. \tag{1}$$

To see this, first note that the simulation (in ExptNM) provided by $\widetilde{A}$ for A is perfect. Thus, the distribution on the values $(x_f, \pi_f = (id_f, \tilde{\pi}_f), \mathsf{aux}, \sigma)$ in the two experiments is identical. Furthermore, note that (as above, we let $\tilde{x}_f \stackrel{\text{def}}{=} (x_f, id_f)$):

$$\begin{array}{c} \mathcal{V}(x_f, \pi_f, \sigma)_1 = \mathsf{true} \\ \mathcal{V}(x_f, \pi_f, \sigma)_2 \notin I \\ R(x_f, \mathsf{aux}) = 1 \end{array} \iff \begin{array}{c} \widetilde{\mathcal{V}}(\tilde{x}_f, \tilde{\pi}_f, \sigma_1) = \mathsf{true} \\ \tilde{\pi}_f \notin Q \\ \widetilde{R}(\tilde{x}_f, \mathsf{aux}) = 1 \end{array},$$

where I is the list of identities queried by A and Q is the list of proofs which $\widetilde{A}$ received from oracle $\widetilde{\mathcal{S}}_2$ (here, we use the property that $\widetilde{\Pi}$ has uniquely applicable proofs). This completes the proof of the claim.

Let $\widetilde{\mathsf{Ext}}$ be the extractor for proof system $\widetilde{\Pi}$ guaranteed by Definition 6. We now specify extractor Ext. Algorithm $\mathsf{Ext}^A(1^k)$ first chooses $\sigma_1 \in \{0,1\}^{6k^2}$ at random and fixes it for the remainder of its execution; note that this defines $\widetilde{L}$. Next, Ext runs $\widetilde{\mathsf{Ext}}(1^k)$, responding to the oracle calls of $\widetilde{\mathsf{Ext}}$ as follows: when $\widetilde{\mathsf{Ext}}$ submits σ_2 to its oracle for $\widetilde{A}$, Ext submits $\sigma_1 \circ \sigma_2$ to its oracle for A. When A queries x, id, algorithm Ext responds by first setting $\tilde{x} := (x, id)$ and sending query $\tilde{x}$ to $\widetilde{\mathsf{Ext}}$. When $\widetilde{\mathsf{Ext}}$ responds with $\tilde{\pi}$, algorithm Ext responds to A with $\pi = (id, \tilde{\pi})$. Ultimately, when A generates its final output $(x_a, \pi_a = (id_a, \tilde{\pi}_a), \mathsf{aux}_a)$, algorithm Ext gives $(\tilde{x}_a = (x_a, id_a), \tilde{\pi}_a, \mathsf{aux}_a)$ to $\widetilde{\mathsf{Ext}}$. When $\widetilde{\mathsf{Ext}}$ outputs $(\tilde{x}_f = (x_f, id_f), \tilde{w}, \mathsf{aux}_f)$, algorithm Ext outputs $(x_f, \tilde{w}, \mathsf{aux}_f)$.

Note that, in the simulation above, Ext perfectly simulates oracle $\widetilde{A}$ for $\widetilde{\mathsf{Ext}}$ (where $\widetilde{A}$ is defined as before). Furthermore, note that if $\tilde{w}$ is a witness to $\tilde{x}_f \in \widetilde{L}$ then, with all but negligible probability, $\tilde{w}$ is also a witness to $x_f \in L$. This is so because, with all but negligible probability, string σ_1 is not a well-defined commitment to *any* string *id*. Therefore, the following is negligible:

$$\left| \Pr[\mathsf{ExptID}'_{A,R,\Pi}(k)] - \Pr[\mathsf{ExptNM}'_{\widetilde{A},\widetilde{R},\widetilde{\Pi}}(k)] \right|. \tag{2}$$

Equations (1) and (2) complete the proof that Π is identity-based.

4 From Identity-Based Schemes to Non-Malleability

In this section, we further study the relation between identity-based NIZK proof systems and non-malleable NIZK proof systems. Section 3 shows how to construct an identity-based proof system based on any non-malleable proof system. Yet, since the definition of identity-based proof systems seems weaker than the definition of non-malleable proof systems, one may wonder whether more efficient constructions of identity-based proof systems are possible. Our results indicate that, in some sense, this is not possible. More formally, we show that any identity-based NIZK proof system can be converted to a non-malleable NIZK proof system with minimal additional overhead. Below, we consider the non-interactive case; however, our results extend to the interactive setting as well. In particular, one can show (using a construction much like the one given below) that any identity-based, interactive ZK proof system can be converted to a non-malleable, interactive ZK proof system without any increase in round-complexity.

We begin with an identity-based NIZK proof system $\widetilde{\Pi} = (\tilde{p}, \tilde{q}, \widetilde{\mathcal{P}}, \widetilde{\mathcal{V}}, \widetilde{\mathcal{S}})$ in which $q(k) = \omega(\log k)$. We make the additional assumption that $\widetilde{\Pi}$ has uniquely-applicable proofs [16] (the construction given in Section 3 satisfies this assumption). In non-malleable proof system Π which we construct, a proof that $x \in L$ will consist of the following: (1) a verification key VK for a one-time signature scheme, (2) a proof $\tilde{\pi}$, in proof system $\widetilde{\Pi}$ and using $id = \mathsf{VK}$, that $x \in L$, and (3) a signature τ on $\tilde{\pi}$, using the secret key SK which corresponds to VK. A complete description of the protocol follows:

- **Common random string.** Let $|x| = k$ and define $p(k) \stackrel{\text{def}}{=} \tilde{p}(k)$. Thus, the random string σ used by Π to prove statements of length k will have the same length as that used by $\widetilde{\Pi}$.
- **Prover strategy.** We use a one-time signature scheme secure against existential forgery: algorithm $\mathsf{KeyGen}(1^k)$ generates signing/verification keys $(\mathsf{SK}, \mathsf{VK})$. We assume for simplicity that VK output by $\mathsf{KeyGen}(1^k)$ has length $\tilde{q}(k)$ (recall the definition requires $\tilde{q}(k) = \omega(\log k)$). Algorithm $\mathcal{P}(x, w, \sigma)$ first runs $\mathsf{KeyGen}(1^k)$ to generate $(\mathsf{SK}, \mathsf{VK})$. Then, $\mathcal{P}$ runs $\widetilde{\mathcal{P}}(x, w, \mathsf{VK}, \sigma)$ to give proof $\tilde{\pi}$. Finally, $\mathcal{P}$ signs $\tilde{\pi}$ (using SK) to obtain signature τ. The output is $\pi = (\mathsf{VK}, \tilde{\pi}, \tau)$.
- **Verifier strategy.** $\mathcal{V}(x, (\mathsf{VK}, \tilde{\pi}, \tau), \sigma)$ runs as follows: if τ is not a valid signature of $\tilde{\pi}$ under VK or $\widetilde{\mathcal{V}}(x, \tilde{\pi}, \sigma)_2 \neq \mathsf{VK}$, output false. Otherwise, output $\widetilde{\mathcal{V}}(x, \tilde{\pi}, \sigma)_1$.
- **Simulation.** $\mathcal{S}_1(1^k)$ simply outputs the result σ, s of running $\widetilde{\mathcal{S}}_1(1^k)$. To simulate a proof, $\mathcal{S}_2(x, s)$ runs $\mathsf{KeyGen}(1^k)$ to obtain $(\mathsf{SK}, \mathsf{VK})$, and then runs $\widetilde{\mathcal{S}}_2(x, \mathsf{VK}, s)$ to obtain $\tilde{\pi}$. Finally, $\mathcal{S}_2$ signs $\tilde{\pi}$ using SK, giving signature τ. The output is $\pi = (\mathsf{VK}, \tilde{\pi}, \tau)$.

The security of this construction is given by the following theorem:

Theorem 2. *If $\widetilde{\Pi}$ is an identity-based NIZK proof system (with $q(k) = \omega(\log k)$ and uniquely applicable proofs) for L, then Π is a non-malleable NIZK proof system for L.*

Proof. One-way functions are sufficient for the construction above; furthermore, the fact that $\widetilde{\Pi}$ is an NIZK proof system for languages outside $\mathcal{BPP}$ implies that one-way functions exist (assuming $\mathcal{NP} \neq \mathcal{BPP}$) [13]. Completeness, soundness, and (unbounded) zero-knowledge of Π follow from the fact that $\widetilde{\Pi}$ satisfies Definitions 1 and 2. Therefore, we focus on proving that Π satisfies Definition 6.

Let A be a PPT adversary and R be a poly-time relation (cf. Definition 6). Define $\widetilde{A}$ as follows: on input σ, adversary $\widetilde{A}$ simply runs $A(\sigma)$. When A submits query x to its oracle for $\mathcal{S}_2$, algorithm $\widetilde{A}$ runs algorithm $\mathsf{KeyGen}(1^k)$ to obtain $(\mathsf{SK}, \mathsf{VK})$, and submits query x, VK to its oracle for $\widetilde{\mathcal{S}}_2$. Upon receiving $\tilde{\pi}$ in response, $\widetilde{A}$ generates signature τ for $\tilde{\pi}$ using SK, and returns to A the proof $\pi = (\mathsf{VK}, \tilde{\pi}, \tau)$. When A outputs $(x_f, \pi_f = (\mathsf{VK}_f, \tilde{\pi}_f, \tau_f), \mathsf{aux})$, algorithm $\widetilde{A}$ checks that π_f is a valid proof for x and that π_f was not one of the proofs which $\widetilde{A}$ gave to A. If both conditions are satisfied, $\widetilde{A}$ outputs $(x_f, \tilde{\pi}_f, \widetilde{\mathsf{aux}} = (\mathsf{aux}, \mathsf{VK}_f, \tau_f))$; otherwise, $\widetilde{A}$ outputs $\perp$.

We claim that the following is negligible:

$$\left| \Pr[\mathsf{ExptNM}^{\mathcal{S}}_{A,R,\Pi}(k)] - \Pr[\mathsf{ExptID}^{\widetilde{\mathcal{S}}}_{\widetilde{A},R,\widetilde{\Pi}}(k)] \right| . \qquad (3)$$

To see this, note that the simulation provided by $\widetilde{A}$ for A is perfect. Thus, the distribution on $(x_f, \pi_f, \mathsf{aux})$ in the two experiments is identical. Assuming π_f is a valid proof for x_f and that π_f was not one of the proofs given to A, there are two possibilities: either VK_f is equal to one of the verification keys which $\widetilde{A}$ already used or not. The probability of the first possibility is negligible, by the security of the one-time signature scheme. On the other hand, when the second possibility occurs, we have:

$$\begin{array}{c} \mathcal{V}(x_f, \pi_f, \sigma) = \mathsf{true} \\ \pi_f \notin Q \end{array} \iff \begin{array}{c} \widetilde{\mathcal{V}}(x_f, \tilde{\pi}_f, \sigma)_1 = \mathsf{true} \\ \widetilde{\mathcal{V}}(x_f, \tilde{\pi}_f, \sigma)_2 \notin I \end{array},$$

where Q is the list of proofs received by A and I is the list of verification keys used by $\widetilde{A}$. This completes the proof of the claim.

Let $\widetilde{\mathsf{Ext}}$ be the extractor for proof system $\widetilde{\Pi}$ guaranteed by Definition 3. Define $\mathsf{Ext}(1^k)$ which runs $\widetilde{\mathsf{Ext}}(1^k)$, responding to the oracle calls of $\widetilde{\mathsf{Ext}}$ as follows: when $\widetilde{\mathsf{Ext}}$ submits σ to its oracle for $\widetilde{A}$, this query is forwarded by Ext to its oracle for A. When A queries x, algorithm Ext runs KeyGen to obtain keys $(\mathsf{SK}, \mathsf{VK})$ and submits query x, VK to $\widetilde{\mathsf{Ext}}$. When $\widetilde{\mathsf{Ext}}$ responds with $\tilde{\pi}$, algorithm Ext generates signature τ on $\tilde{\pi}$ using SK, and returns $\pi = (\mathsf{VK}, \tilde{\pi}, \tau)$ to A. When A generates its final output $(x_a, \pi_a = (\mathsf{VK}_a, \tilde{\pi}_a, \tau_a), \mathsf{aux}_a)$, algorithm Ext gives $(x_a, \tilde{\pi}_a, \mathsf{aux}_a)$ to $\widetilde{\mathsf{Ext}}$. Finally, when $\widetilde{\mathsf{Ext}}$ outputs $(x_f, w_f, \mathsf{aux}_f)$, algorithm Ext outputs the same. It is clear that:

$$\Pr[\mathsf{ExptNM}'_{A,R,\Pi}(k)] = \Pr[\mathsf{ExptID}'_{\widetilde{A},R,\widetilde{\Pi}}(k)]. \qquad (4)$$

Equations (3) and (4) complete the proof that Π is non-malleable.

Acknowledgments

We thank the referees for their helpful comments.

References

1. B. Barak. Constant-Round Coin-Tossing with a Man in the Middle or Realizing the Shared Random String Model. FOCS 2002.
2. M. Blum. How to Prove a Theorem so No One Else Can Claim It. *Proceedings of the International Congress of Mathematicians*, 1986.
3. M. Blum, P. Feldman, and S. Micali. Non-Interactive Zero-Knowledge and Its Applications. STOC '88.
4. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. FOCS 2001.
5. R. Cramer and I. Damgård. Fast and Secure Immunization Against Adaptive Man-in-the-Middle Impersonation. Eurocrypt '97.
6. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust Non-Interactive Zero Knowledge. Crypto 2001.
7. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. *SIAM J. Computing* 30(2): 391–437 (2000).
8. U. Feige, D. Lapidot, and A. Shamir. Multiple Non-Interactive Zero Knowledge Proofs Under General Assumptions. SIAM J. Comp. 29(1): 1–28 (1999).
9. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. SIAM J. Comp. 18(1): 186–208 (1989).
10. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated-Verifier Proofs and their Applications. Eurocrypt '96.
11. J. Katz, R. Ostrovsky, and A. Smith. Round Efficiency of Multi-Party Computation with a Dishonest Majority. Eurocrypt 2003.
12. M. Naor. Bit Commitment Using Pseudorandomness. J. Crypto. 4(2): 151–158 (1991).
13. R. Ostrovsky and A. Wigderson. One-Way Functions are Essential for Non-Trivial Zero-Knowledge. 2nd Israeli Symp. on Theory of Computing and Systems, 1993.
14. R. Pass. On Deniability in the Common Reference String and Random Oracle Models. Crypto 2003.
15. R. Pass. Bounded-Concurrent Multi-Party Computation with a Dishonest Majority. STOC 2004.
16. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. FOCS '99.

A Definitions for Non-Malleable NIZK

For completeness, we include relevant definitions from [6].

Definition 4. ([6, Def. 1]) *$\Pi = (p, \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ is a single-theorem NIZK proof system for a language L with witness relation R if p is polynomial and $\mathcal{P}, \mathcal{V}$, and $\mathcal{S}$ are* PPT *algorithms such that:*

1. **(Completeness):** *For all $x \in L$ and all w such that $(x, w) \in R$, for all $\sigma \in \{0,1\}^{p(|x|)}$, we have $\mathcal{V}(x, \mathcal{P}(x, w, \sigma), \sigma) = \mathsf{true}$.*
2. **(Soundness):** *For all unbounded algorithms $\mathcal{P}'$, if $\sigma \in \{0,1\}^{p(|x|)}$ is chosen randomly, the probability that $\mathcal{P}'(\sigma)$ outputs (x, π) such that $\mathcal{V}(x, \pi, \sigma) = \mathsf{true}$ and $x \notin L$ is negligible.*
3. **(Zero-knowledge):** *For all $x \in L$ and all w such that $R(x, w) = \mathsf{true}$, the following distributions are computationally indistinguishable (where $k \stackrel{\text{def}}{=} p(|x|)$):*
$$\left\{\sigma \leftarrow \{0,1\}^k; \pi \leftarrow \mathcal{P}(x, w, \sigma) : (\sigma, \pi)\right\}$$
and
$$\left\{(\sigma, s) \leftarrow \mathcal{S}_1(1^k); \pi \leftarrow \mathcal{S}_2(x, s) : (\sigma, \pi)\right\}.$$

Definition 5. ([6, Def. 2]) *$\Pi = (p, \mathcal{P}, \mathcal{V}, \mathcal{S})$ is an* unbounded *NIZK proof system for language L if Π is a single-theorem NIZK proof system for L and for all* PPT *algorithms A, we have that $|\Pr[\mathsf{Expt}_{A,\Pi}(k) = 1] - \Pr[\mathsf{Expt}^{\mathcal{S}}_{A,\Pi}(k) = 1]|$ is negligible; where:*

$\mathsf{Expt}_{A,\Pi}(k)$:	$\mathsf{Expt}^{\mathcal{S}}_{A,\Pi}(k)$:
$\sigma \leftarrow \{0,1\}^k$	$(\sigma, s) \leftarrow \mathcal{S}_1(1^k)$
return $A^{\mathcal{P}(\cdot,\cdot,\sigma)}(\sigma)$	return $A^{\mathcal{S}'(\cdot,\cdot,s)}(\sigma)$

where $\mathcal{S}'(x, w, s) \stackrel{\text{def}}{=} \mathcal{S}_2(x, s)$ (we assume, above, that if x, w is a query of A, then $(x, w) \in R$).

Definition 6. ([6, Def. 5]) *Let $\Pi = (p, \mathcal{P}, \mathcal{V}, \mathcal{S})$ be an unbounded NIZK proof system for language L with witness relation R_L. We say that Π is a* non-malleable *NIZK proof system for L if there exists an extractor Ext such that, for all* PPT *adversaries A and all poly-time relations R, the difference*

$$|\Pr[\mathsf{ExptNM}^{\mathcal{S}}_{A,R,\Pi}(k)] - \Pr[\mathsf{ExptNM}'_{A,R,\Pi}(k)]|$$

is negligible, where:

$\mathsf{ExptNM}^{\mathcal{S}}_{A,R,\Pi}(k)$:	$\mathsf{ExptNM}'_{A,R,\Pi}(k)$:
$(\sigma, s) \leftarrow \mathcal{S}_1(1^k)$	$(x, w, \mathsf{aux}) \leftarrow \mathsf{Ext}^A(1^k)$
$(x, \pi, \mathsf{aux}) \leftarrow A^{\mathcal{S}_2(\cdot,s)}(\sigma)$	return true *iff*
Let Q be the list of proofs returned by $\mathcal{S}_2$	$(x, w) \in R_L$ and
return true *iff*	$R(x, \mathsf{aux}) = 1$
$\mathcal{V}(x, \pi, \sigma) = \mathsf{true}$ and	
$\pi \notin Q$ and	
$R(x, \mathsf{aux}) = 1$	

(we assume, above, that if x is a query of A then $x \in L$).