Robust Non-interactive Zero Knowledge

Alfredo De Santis[1], Giovanni Di Crescenzo[2], Rafail Ostrovsky[2], Giuseppe Persiano[1], and Amit Sahai[3]

[1] Dipartimento di Informatica ed Applicazioni, Università di Salerno, Baronissi (SA), Italy.
ads@dia.unisa.it, giuper@dia.unisa.it

[2] Telcordia Technologies, Inc., Morristown, NJ, USA.
giovanni@research.telcordia.com, rafail@research.telcordia.com

[3] Department of Computer Science, Princeton University. Princeton, NJ 08544.
sahai@cs.princeton.edu

Abstract. Non-Interactive Zero Knowledge (NIZK), introduced by Blum, Feldman, and Micali in 1988, is a fundamental cryptographic primitive which has attracted considerable attention in the last decade and has been used throughout modern cryptography in several essential ways. For example, NIZK plays a central role in building provably secure public-key cryptosystems based on general complexity-theoretic assumptions that achieve security against chosen ciphertext attacks. In essence, in a multi-party setting, given a fixed common random string of polynomial size which is visible to all parties, NIZK allows an arbitrary polynomial number of Provers to send messages to polynomially many Verifiers, where each message constitutes an NIZK proof for an arbitrary polynomial-size NP statement.

In this paper, we take a closer look at NIZK in the multi-party setting. First, we consider *non-malleable* NIZK, and generalizing and substantially strengthening the results of Sahai, we give the first construction of NIZK which remains non-malleable after polynomially-many NIZK proofs. Second, we turn to the definition of standard NIZK itself, and propose a strengthening of it. In particular, one of the concerns in the technical definition of NIZK (as well as non-malleable NIZK) is that the so-called "simulator" of the Zero-Knowledge property is allowed to pick a *different* "common random string" from the one that Provers must actually use to prove NIZK statements in real executions. In this paper, we propose a new definition for NIZK that eliminates this shortcoming, and where Provers and the simulator use the *same* common random string. Furthermore, we show that both standard and *non-malleable* NIZK (as well as NIZK Proofs of Knowledge) can be constructed achieving this stronger definition. We call such NIZK **Robust NIZK** and show how to achieve it. Our results also yields the simplest known public-key encryption scheme based on general assumptions secure against adaptive chosen-ciphertext attack (CCA2).

1 Introduction

Interactive Zero-Knowledge. Over the last two decades, Zero-Knowledge (ZK) as defined by Goldwasser, Micali, and Rackoff [21] has become a fundamental cryptographic tool. In particular, Goldreich, Micali and Wigderson [20] showed that any NP statement can be proven in *computational* [1] ZK (see also [16]). Though ZK was originally defined for use in two-party interactions (i.e., between a single Prover and a single Verifier), ZK was shown to be useful in a host of situations where multiple parties could be involved, especially in the multi-party secure function evaluation, first considered by Goldreich, Micali and Wigderson [19]. Informally, one reason the notion of interactive ZK has been so pervasive is that in the single Prover/Verifier case, ZK essentially guarantees that any poly-time Verifier after interacting with the Prover in a ZK protocol learns absolutely nothing. Thus, informally speaking, whatever a poly-time Verifier can do after verifying a ZK protocol, it could also have done before such a ZK interaction. However, in a multiparty setting, perhaps not surprisingly, the standard two-party definition of ZK does not guarantee what we would intuitively expect from "zero knowledge': that the polynomial-time Verifier after observing such proofs can not (computationally) do anything that he was not able to do before such a proofs. Essentially, two important problems were pointed out in the literature:

One problem, formally defined by Dolev, Dwork and Naor [13] is that of *malleability*, which informally means that an adversary who takes part in some ZK interaction can also interact with other parties and can exploit fragments of ZK interactions to prove something that he was not able to prove before. Indeed, this is a real problem to which [13] propose a solution that requires polylogarithmic overhead in the number of rounds of communication. It is not known how to reduce the number of rounds further in their solution.

Another problem of ZK in the multi-party setting, pointed out by Dwork, Naor and Sahai [14], is that verifiers can "collaborate" when talking to provers, and the ZK property must be guaranteed even in concurrent executions. Indeed, unless one introduce changes in the model such as timing assumptions, in terms of the number of rounds, it was shown that a polylogarithmic number of rounds is both necessary [6] and sufficient [25] to guarantee concurrent ZK.

Non-interactive Zero-Knowledge (NIZK): A way to reduce the number of rounds in a ZK proof (to just a single message from Prover to Verifier) was

[1] Recall that several variants of ZK have been considered in the literature, in terms of the strength of the *soundness* condition and the strength of the *simulation*. In terms of the quality of the simulation, *perfect*; *statistical*; and *computational* ZK are defined [21]. In terms of *soundness* two variants were considered: ZK *proofs*, where the proof remains valid even if an infinitely-powerful Prover is involved [21,20] and ZK *arguments*, where it is required that only polynomially-bounded Provers cannot cheat (except with negligible probability), given some complexity assumption [3,26]. For ZK proofs for languages outside BPP were shown to imply the existence of one-way functions for perfect, statistical [30] (see also [34]) as well as computational [31] variants of ZK.

proposed by Blum, Feldman and Micali [2] by changing the model as follows: we assume that a *common random reference string* is available to all players. The Prover sends a single message to Verifier, which constitutes "non-interactive zero-knowledge" (NIZK) proof. In [2] it was shown that any NP statement has a NIZK proof. Extending [2], Blum, De Santis, Micali and Persiano [1] showed how a Prover can prove polynomially many proofs based on algebraic assumptions. Feige, Lapidot and Shamir further refined the definition of NIZK and constructed[2] multiple-proof NIZK based on general assumptions [15]. De Santis and Persiano extended NIZK notion to NIZK Proofs of Knowledge (NIZK-PK)[3] [8].

Again, although the notion of NIZK was defined in a two-party setting, it quickly found applications in settings with many parties, in particular where the same reference string may be used by multiple parties (see e.g. [13,28,4,22]). Because of the non-interactive nature of NIZK proofs, many multi-party issues that appear in ZK, do not arise in NIZK; for example the problem of concurrent zero-knowledge is completely gone[4]!

The definition of NIZK proposed by [2,1,15], essentially provides the following guarantee: What one can output after seeing NIZK proofs is indistinguishable from what one can output without seeing any proofs, *if you consider the reference string as part of the output.* Thus, the standard notion of NIZK says that as long as one can simulate proofs *together* with random-looking reference strings, this satisfies the notion of NIZK. This definition, however, leaves open the question of what to do about output as it relates to the *particular* reference string that is being used by a collection of parties. Since the NIZK simulator produces its own different random string, its output would make sense only relative to the reference string that it chose, different from the one used by the provers. [5] One of the contributions of this paper is to strengthen the notion of NIZK to insist that the simulator works with the *same* totally random string that all the Provers work with.

NIZK proofs are broadcastable and transferable – that is, a single proof string can be broadcasted or transferred from verifier to verifier to convince multiples parties of the validity of a statement. However, transferability causes a new problem: a user who have seen an NIZK proof (of a hard problem) can now "prove" (by simply copying) what he was not able to prove before. Indeed,

[2] Efficiency improvements to these constructions were presented in [24,9,10].

[3] In the same paper [8] defined *dense cryptosystems* and showed that dense cryptosystems and NIZK proofs of membership for NP are sufficient in order to construct NIZK-PK for all of NP. This assumption was shown to be *necessary* for NIZK-PK in [11]. (Dense cryptosystemes were also shown to be equivalent to *extractable commitment* [11].)

[4] In fact, non-malleable commitment also becomes much easier to deal with in the non-interactive setting [12]. Also, though it is not always thought of as a multi-party issue, the problem of resettable zero-knowledge [5] is also easily dealt with for NIZK as well.

[5] Indeed, it seems quite unfair to let the simulator get away with ignoring the actual reference string!

more generally the problem of *malleability* does remain for NIZK proofs: With respect to a particular (fixed) reference string, after seeing some NIZK proofs, the adversary may be able to construct new proofs that it could not have been able to otherwise. Sahai introduced *non-malleable* NIZK in [33] where he shows how to construct NIZK which remains non-malleable only as long as the number of proofs seen by any adversary is bounded. In this paper (among other contributions) we continue and extend his work, strengthening the notion and the constructions of non-malleability and removing the limitation on the number of proofs. (For further discussion on malleability issues in multi-party situations, see Appendix A.)

Our results: First, we consider the following notion of NIZK. The *sampling* algorithm produces a common random string together with auxiliary information. (We insist that the common random string comes from a uniform (or nearly uniform) distribution). Polynomially-bounded provers use this common random string to produce polynomially-many NIZK messages for some NP language. We insist that the simulator, given the *same* common random string, together with auxiliary information, can produce the proofs of theorems which are computationally indistinguishable from the proofs produced by honest provers *for the same reference string*. We call this notion *same-string* NIZK.

We show two facts regarding same-string NIZK: (1) same-string NIZK Proofs (i.e. where the prover is infinitely powerful) are impossible for any hard-on-average NP-complete languages (2) same-string NIZK Arguments (i.e. where the prover is computationally bounded) are possible given any one-way trapdoor permutation.

Next, we turn to non-malleability for NIZK, and a notion related to non-malleability called *simulation-soundness* first defined by Sahai [33]. The simulation-soundness requirement is that a polynomially-bounded prover can not prove false theorems even after seeing simulated proofs of any statements (including false statements) of its choosing. Sahai achieves non-malleability and simulation-soundness only with respect to a bounded number of proofs. In this paper, we show that assuming the existence of one-way trapdoor permutations, we can construct NIZK proof systems which remain simulation-sound even after the prover sees any polynomial number of simulated proofs[6]. Combined with [33] this also gives the simplest known construction of CCA2-secure public-key cryptosystem based on one-way trapdoor permutations.

In dealing with non-malleability, we next turn to NIZK Proofs of Knowledge (NIZK-PK), introduced by De Santis and Persiano[8]. We use NIZK-PK to propose a strengthening of the definition of non-malleability for NIZK, based

[6] We note that we can also achieve a form of non-malleability (as opposed to simulation soundness) for NIZK proofs of membership based only on trapdoor permutations. This non-malleability would also hold against any polynomial number of proofs, however the non-malleability achieved satisfies a weaker definition than the one we propose based on NIZK-PK (and in particular, the resulting NIZK proof would only be a proof of membership and not a proof of knowledge). We omit the details of this in these proceedings.

on NP-witnesses (which, in particular, implies the earlier definition [33]). We provide constructions which show that for any polynomial-time adversary, even after the adversary has seen any polynomial number of NIZK proofs for statements of its choosing, the adversary does not gain the ability to prove any new theorems it could not have produced an NP witness for prior to seeing any proofs, except for the ability to duplicate proofs it has already seen. This construction requires the assumption that trapdoor permutations exist and that public-key encryption schemes exist with an inverse polynomial density of valid public keys (called *dense cryptosystems*). Such dense cryptosystems exist under most common intractability assumptions which give rise to public-key encryption, such as the RSA assumption, Quadratic Residuosity, Diffie-Hellman [8] and factoring [11]. (In fact, in the context of NIZK-PK, we cannot avoid using such dense cryptosystems since they were shown to be *necessary* for any NIZK-PK [11].)

Finally, we call NIZK arguments that are both non-malleable and same-string NIZK **Robust NIZK**.

We highlight the contributions of our results:

- For NIZK arguments, we give the first construction where the simulator uses the same common random string as used by all the provers.
- Our Robust-NIZK proof systems are non-malleable with regard to *any* polynomial number of proofs seen by the adversary and with respect to the same proof-system. (We contrast this with the previous result of [33] which proves non-malleability against only a bounded number of proofs, and in fact the length of the reference string grew quadratically in the bound on the the number of proofs the adversary could see.) In our result, in contrast, the length of the reference string depends only on the security parameter.
- Our non-malleable NIZK definition and construction based on NIZK-PK achieves a very strong guarantee: We require that one can obtain an explicit NP witness for any statement that the adversary can prove after seeing some NIZK proofs. Thus, it intuitively matches our notion of what NIZK should mean: that the adversary cannot prove anything "new" that he was not able to prove before (except for copying proofs in their entirety).
- Finally, our construction yields the simplest known public-key encryption scheme based on general assumptions which is secure against adaptive chosen-cyphertext attacks (CCA2).

We point out some new techniques used to establish our results. All previous work on non-malleability in a non-interactive setting under general assumptions [13,12,33] used a technique called "unduplicatable set selection". Our first construction provides the first non-malleability construction based on general assumptions which *does not* use "unduplicatable set selection" at all, and rather relies on a novel use of pseudo-random functions of [18]. In our second construction, we show how to generalize the unduplicatable set selection technique to a technique we call "hidden unduplicatable set selection," and use this to build our proofs. Both techniques are novel, and may have further applications.

Organization. In Section 2, we both recall old definitions as well as give the new definitions of this paper. In Section 3, we present our first construction of Robust NIZK and non-malleable NIZK (and NIZK-PK) proofs. In Section 4, we present our second construction which uses different techniques and a yields non-malleable NIZK and NIZKPK.

2 Preliminaries and Definitions

We use standard notations and conventions for writing probabilistic algorithms and experiments. If A is a probabilistic algorithm, then $A(x_1, x_2, \ldots; r)$ is the result of running A on inputs $x_1, x_2, \ldots$ and coins r. We let $y \leftarrow A(x_1, x_2, \ldots)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \ldots; r)$. If S is a finite set then $x \leftarrow S$ is the operation of picking an element uniformly from S. $x := \alpha$ is a simple assignment statement. By a "non-uniform probabilistic polynomial-time adversary," we always mean a circuit whose size is polynomial in the security parameter. All adversaries we consider are non-uniform. (Thus, we assume our assumptions, such as the existence of one-way functions, also hold against non-uniform adversaries.)

In this section, we will formalize the notions of non-malleable, same-string and robust NIZK proofs. We will also define an extension of simulation soundness.

2.1 Basic Notions

We first recall the definition of an (efficient, adaptive) single-theorem NIZK proof systems [1,2,15,8]. Note that since we will always use the now-standard adaptive notion of NIZK, we will suppress writing "adaptive" in the future. We will also only concentrate on efficiently realizable NIZK proofs, and so we will suppress writing "efficient" as well. This first definition only guarantees that a single proof can be simulated based on the reference string. Note that our definition uses "Strong Soundness," based on Strong NIZK Proofs of Knowledge defined in [8] and a similar notion defined in [28], where soundness is required to hold even if the adversary may chose its proof after seeing the randomly selected reference string. Note that the constructions given in [15], for instance, meet this requirement. We simultaneously define the notion of an NIZK argument, in a manner completely analogous to the definition of an interactive ZK argument.

Definition 1 (NIZK [15]). *$\Pi = (\ell, \mathrm{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ is a* single-theorem NIZK proof system (resp., argument) *for the language $L \in$ NP with witness relation R if: ℓ is a polynomial, and $\mathrm{P}, \mathcal{V}, \mathcal{S}_1, \mathcal{S}_2$ are all probabilistic polynomial-time machines such that there exists a negligible function α such that for all k:*

(Completeness): *For all $x \in L$ of length k and all w such that $R(x, w) =$ `true` , for all strings σ of length $\ell(k)$, we have that $\mathcal{V}(x, \mathrm{P}(x, w, \sigma), \sigma) =$ `true` .*

(Soundness): *For all unbounded (resp., polynomial-time) adversaries A, if $\sigma \in \{0,1\}^{\ell(k)}$ is chosen randomly, then the probability that $A(\sigma)$ will output (x,p) such that $x \notin L$ but $\mathcal{V}(x,p,\sigma) = \texttt{true}$ is less than $\alpha(k)$.*

(Single-Theorem Zero Knowledge): *For all non-uniform probabilistic polynomial-time adversaries $A = (A_1, A_2)$, we have that*

$$|\Pr\left[\mathsf{Expt}_A(k) = 1\right] - \Pr\left[\mathsf{Expt}_A^S(k) = 1\right]| \leq \alpha(k),$$

where the experiments $\mathsf{Expt}_A(k)$ and $\mathsf{Expt}_A^S(k)$ are defined as follows:

$\mathsf{Expt}_A(k)$:	$\mathsf{Expt}_A^S(k)$:
$\Sigma \leftarrow \{0,1\}^{\ell(k)}$	$(\Sigma, \tau) \leftarrow \mathcal{S}_1(1^k)$
$(x,w,s) \leftarrow A_1(\Sigma)$	$(x,w,s) \leftarrow A_1(\Sigma)$
$p \leftarrow \mathrm{P}(x,w,\Sigma)$	$p \leftarrow \mathcal{S}_2(x,\Sigma,\tau)$
$\textit{return}\ A_2(p,s)$	$\textit{return}\ A_2(p,s)$

To define a notion of NIZK where any polynomial number of proofs can be simulated, we change the Zero-knowledge condition as follows:

Definition 2 (unbounded NIZK [15]). *$\Pi = (\ell, \mathrm{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ is an* unbounded NIZK proof system *for the language $L \in \mathrm{NP}$ if Π is a single-theorem NIZK proof system for L and furthermore: there exists a negligible function α such that for all k:*

(Unbounded Zero Knowledge): *For all non-uniform probabilistic polynomial-time adversaries A, we have that $|\Pr\left[\mathsf{Expt}_A(k) = 1\right] - \left[\mathsf{Expt}_A^S(k) = 1\right]| \leq \alpha(k)$, where the experiments $\mathsf{Expt}_A(k)$ and $\mathsf{Expt}_A^S(k)$ are defined as follows:*

$\mathsf{Expt}_A(k)$:	$\mathsf{Expt}_A^S(k)$:
$\Sigma \leftarrow \{0,1\}^{\ell(k)}$	$(\Sigma, \tau) \leftarrow \mathcal{S}_1(1^k)$
$\textit{return}\ A^{P(\cdot,\cdot,\Sigma)}(\Sigma)$	$\textit{return}\ A^{S'(\cdot,\cdot,\Sigma,\tau)}(\Sigma)$

where $S'(x,w,\Sigma,\tau) \overset{\text{def}}{=} \mathcal{S}_2(x,\Sigma,\tau)$.

Definition 3. *We say that an NIZK argument system is* same-string *NIZK if the (unbounded) zero knowledge requirement above is replaced with the following requirement: there exists a negligible function α such that for all k:*

(Same-String Zero Knowledge): *For all non-uniform probabilistic polynomial-time adversaries A, we have that $|\Pr\left[X = 1\right] - \Pr\left[Y = 1\right]| \leq \alpha(k)$, where X and Y are as defined in (and all probabilities are taken over) the experiment $\mathsf{Expt}(k)$ below:*

```
Expt(k) :
    $(\Sigma, \tau) \leftarrow \mathcal{S}_1(1^k)$
    $X \leftarrow A^{P(\cdot,\cdot,\Sigma)}(\Sigma)$
    $Y \leftarrow A^{S'(\cdot,\cdot,\Sigma,\tau)}(\Sigma)$
```

where $S'(x,w,\Sigma,\tau) \overset{\text{def}}{=} \mathcal{S}_2(x,\Sigma,\tau)$.

(Same-String Zero Knowledge, cont.): *The distribution on Σ produced by $\mathcal{S}_1(1^k)$ is the uniform distribution over* $\{0,1\}^{\ell(k)}$.

Remark 1. We make two observations regarding the definition of same-string NIZK:

– As done in [15], the definition could equivalently be one that states that with all but negligible probability over the choices of common random reference strings, the simulation is computationally indistinguishable from real proofs supplied by the prover. We omit the details for lack of space.
– On the other hand, the definition above differs from the standard definition on unbounded zero knowledge only in the new requirement that the simulator produce truly uniform reference strings. It is easy to verify that all other changes are cosmetic.
– In the next theorem, we show why we must speak only of same-string NIZK *arguments*, and not NIZK Proofs.

Theorem 1. *If one-way functions exist, then there cannot exist same-string (adaptive) NIZK Proof systems for any NP-complete language L, even for single-theorem NIZK. In fact, this result extends to any language that is hard-on-average with respect to an efficiently samplable distribution.*

Proof. **(Sketch)** We only sketch the proof of this impossibility result. Assume that one-way functions exist, and that a same-string (adaptive) single-theorem NIZK Proof system exists for an NP-complete language L. We will show a contradiction to the soundness of the NIZK Proof System. First we note that the existence of one-way functions and Cook's theorem implies that there is a probabilistic polynomial-time algorithm M such that for all non-uniform polynomial-time machines A, if $x \leftarrow M(1^k)$, the probability that A correctly decides whether $x \in L$ is only negligibly more than 1/2. It is implicit in the previous statement that with probability close to 1/2, if $x \leftarrow M(1^k)$, then $x \notin L$.

This hardness condition also implies that, in particular, the simulator must output proofs that are accepted with all but negligible probability when given as input $x \leftarrow M(1^k)$. At the same time, because the NIZK system is both same-string (adaptive) NIZK, it must be that the reference strings output by $S_1(1^k)$ come from a uniform distribution.

Now, consider a cheating (unbounded) prover which, for any given random string, guesses the auxiliary information τ which maximizes the probability that the simulator outputs an accepting proof on inputs chosen according to $x \leftarrow M(1^k)$. Since the reference string that the prover encounters is also uniform, it follows that the cheating prover will have at least as high a probability of convincing a verifier to accept on input $x \leftarrow M(1^k)$. But we know that the simulator causes the verifier to accept with probability negligibly close to 1. This contradicts the (unconditional) soundness of the NIZK proof system, completing the proof.

We also define the notion of an NIZK proof of knowledge [8] for an NP language L with witness relation R. Informally, the idea is that in an NIZK proof of knowledge, one should be able to extract the NP witness directly from the proof if given some special information about the reference string. We capture this notion by defining an *extractor* which produces a reference string together with some auxiliary information. The distribution on reference strings is statistically close to the uniform distribution. Given the auxiliary information and an NIZK proof, one can efficiently extract the witness. [8] show how to turn any NIZK proof system into a proof of knowledge under the assumption that public-key encryption schemes exist with sufficiently high density of valid public keys (called dense cryptosystems). We now recall the formal definition:

Definition 4 (NIZK proof of knowledge [8]). $\Pi = (\ell, \mathrm{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2), E = (E_1, E_2))$ *is a* NIZK proof (or argument) of knowledge *for the language* $L \in \mathrm{NP}$ *with witness relation* R *if:* Π *is an NIZK proof (or argument) system (of any type) for* L *and furthermore* E_1 *and* E_2 *are probabilistic polynomial-time machines such that there exists a negligible function* α *such that for all* k*:*

(Reference-String Uniformity): *The distribution on reference strings produced by*
$E_1(1^k)$ *has statistical distance at most* $\alpha(k)$ *from the uniform distribution on* $\{0,1\}^{\ell(k)}$.

(Witness Extractability): *For all adversaries* A*, we have that* $\left[\mathsf{Expt}^E_A(k)\right]| \geq \Pr\left[\mathsf{Expt}_A(k)\right] - \alpha(k)$*, where the experiments* $\mathsf{Expt}_A(k)$ *and* $\mathsf{Expt}^S_A(k)$ *are defined as follows:*

$\mathsf{Expt}_A(k)$:	$\mathsf{Expt}^E_A(k)$:
$\Sigma \leftarrow \{0,1\}^{\ell(k)}$	$(\Sigma, \tau) \leftarrow E_1(1^k)$
$(x, p) \leftarrow A(\Sigma)$	$(x, p) \leftarrow A(\Sigma)$
return $V(x, p, \Sigma)$	$w \leftarrow E_2(\Sigma, \tau, x, p)$
	return true *if* $(x, w) \in R$

2.2 Non-malleable NIZK

We now proceed to define non-malleable NIZK. The intuition that our definition will seek to capture is to achive the strongest possible notion of non-malleability: "whatever an adversary can prove after seeing polynomially many NIZK proofs for statements of its choosing, it could have proven without seeing them, except for the ability to duplicate proofs."[7] Extending the notion of NIZK-PK of De Santis and Persiano [8] we define non-malleable NIZK-PK. We will make the definition with regard to simulated proofs, but note that one can make a similar definition with regard to actual proofs; we omit it due to lack of space.

[7] When interpreting the line "it could have proven without seeing them," we insist that an actual NP witness for the statement should be extractable from the adversary, which is a very strong NIZK-PK property.

Definition 5. [Non-Malleable NIZK] Let $\Pi = (\ell, \mathcal{P}, \mathcal{V}, \mathcal{S})$ be an unbounded NIZK proof system for the NP language L with witness relation W. We say that Π is a *non-malleable NIZK proof system (or argument)*[8] for L if there exists a probabilistic polynomial-time oracle machine M such that:

For all non-uniform probabilistic polynomial-time adversaries A and for all non-uniform polynomial-time relations R, there exists a negligible function $\alpha(k)$ such that

$$\Pr\left[\mathsf{Expt}^{S}_{A,R}(k)\right] \leq \Pr\left[\mathsf{Expt}'_{A}(k)\right] + \alpha(k)$$

where $\mathsf{Expt}^{S}_{A,R}(k)$ and $\mathsf{Expt}'_{A,R}(k)$ are the following experiments:

$\mathsf{Expt}^{S}_{A,R}(k)$:	$\mathsf{Expt}'_{A,R}(k)$:
$(\Sigma, \tau) \leftarrow \mathcal{S}_1(1^k)$	
$(x, p, \text{aux}) \leftarrow A^{S_2(\cdot, \Sigma, \tau)}(\Sigma)$	
Let Q be list of proofs given by S_2 above	$(x, w, \text{aux}) \leftarrow M^A(1^k)$
`return true` iff	`return true` iff
$(\ p \notin Q\ )$ `and`	
$(\ \mathcal{V}(x, p, \Sigma) =$ `true` $)$ `and`	$(\ (x, w) \in W)$ `and`
$(\ R(x, \text{aux}) =$ `true` $)$	$(\ R(x, \text{aux}) =$ `true` $)$

We also consider (and strengthen) another notion for NIZK called simulation soundness [33] which is related to non-malleability, but also can be useful in applications – in particular, it suffices for building public-key encryption schemes secure against the strongest form of chosen-ciphertext attack (CCA2). The ordinary soundness property of proof systems states that with overwhelming probability, the prover should be incapable of convincing the verifier of a false statement. In this definition, we will ask that this remains the case even after a polynomially bounded party has seen any number of simulated proofs of his choosing. Note that simulation soundness is implied by our definition of non-malleability above.

Definition 6. [Unbounded Simulation-Sound NIZK] Let $\Pi = (\ell, \mathcal{P}, \mathcal{V}, \mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2))$ be an unbounded NIZK proof system (or argument) for the language L. We say that Π is *simulation-sound* if for all non-uniform probabilistic polynomial-time adversaries A, we have that

$$\Pr\left[\mathsf{Expt}_{A,\Pi}(k)\right] \text{ is negligible in } k,$$

where $\mathsf{Expt}_{A,\Pi}(k)$ is the following experiment:

[8] To stress the main novelty of this definition, we will sometimes write "non-malleable in the explicit witness sense," to indicate that an explicit NP-witness can be extracted from any prover. We remark that our definition clearly implies the definition of [33].

```
$\mathsf{Expt}_{A,\Pi}(k)$ :
    $(\Sigma, \tau) \leftarrow \mathcal{S}_1(1^k)$
    $(x, p) \leftarrow A^{\mathcal{S}_2(\cdot,\Sigma,\tau)}(\Sigma)$
    Let $Q$ be list of proofs given by $\mathcal{S}_2$ above
    return true iff ( $p \notin Q$ and $x \notin L$ and $\mathcal{V}(x, p, \Sigma) =$ true )
```

Definition 7. *We will call an NIZK argument that is non-malleable and has unbiased simulations a* robust NIZK *argument.*

3 First Construction

In this section, we exhibit our construction of NIZK proof systems that enjoy unbounded simulation-soundness. This construction is then readily modified using NIZK Proofs of Knowledge to construct proof systems with unbounded non-malleability (in the explicit witness sense), and robust NIZK arguments.

Assumptions needed. In order to construct our simulation-sound proof systems for some NP language L, we will require the existence of efficient single-theorem (adaptive) NIZK proof systems for a related language L', described in detail below. Such proof systems exist under the assumption that trapdoor permutations exist [15]. Further, we will require the existence of one-way functions. To construct the proof systems with full non-malleability, we will require efficient single-theorem (adaptive) NIZK proofs of knowledge for the language L'. Such proof systems exist under the assumption that dense cryptosystems exist and trapdoor permutations exist [8].

3.1 Ingredients

Let k be the security parameter. We first specify the ingredients used in our construction:
Commitment. We recall two elegant methods for constructing commitments. One, based on one-way permutations, will allow us to construct non-malleable NIZK arguments with unbiased simulations (*i.e.* robust NIZK). The other, which can be based merely on one-way functions, suffices to construct non-malleable NIZK proof systems.

The theorem of Goldreich and Levin [17] immediately yields the following bit commitment scheme from any one-way permutation f on k bits:

$$C(b, s) = (r, f(s)) \text{ where } r \in_R \{0,1\}^k \text{ such that } r \cdot s = b$$

Here, it should be that $s \in_R \{0,1\}^k$. Note that if $s = 0^k$ and $b = 1$, then no choice of r will allow for $r \cdot s = b$. In this case, r is chosen at random, but the commitment is invalid. Since invalid commitments can only occur with probability at most 2^{-k}, we can safely ignore this. To reveal the bit, the sender simply reveals s. Observe that the distribution $C(b, s)$ where both b and s are chosen uniformly has is precisely the uniform distribution over $\{0,1\}^{2k}$. We will sometimes write just $C(b)$ to mean $C(b, s)$ where $s \in_R \{0,1\}^k$. Note that in this commitment scheme, every string of length $2k$ corresponds to a commitment to *some unique* string.

On the other hand, we recall the bit commitment protocol of Naor [27] based on pseudorandom generators (which can be built from any one-way function [23]). Let G be a pseudorandom generator stretching k bits to $3k$ bits. The Naor commitment procedure commits to a bit b as follows:

$$C(b, s) = \begin{cases} (r, G(s)) & \text{if } b = 0 \\ (r, G(s) \oplus r) & \text{if } b = 1 \end{cases}$$

Here, $r \in_R \{0,1\}^{3k}$, and as above the string s should be selected uniformly at random among strings of length k. Again, we will sometimes write just $C(b)$ to mean $C(b, s)$ where $s \in_R \{0,1\}^k$. It is shown in [27] that if U and U' are both independent uniform distributions among strings of length $3k$, then the distributions (U, U'), $C(0)$, and $C(1)$ are all computationally indistinguishable (taken as ensembles of distributions indexed by k). Furthermore, it is clear that unless r is of the form $G(s_1) \oplus G(s_2)$ for some s_1 and s_2, there are no commitment strings that can arise as both commitments to 0 and commitments to 1. The probability of this being possible is thus less than 2^{-k} over the choices of r. Furthermore, the probability that a random sample from (U, U') could be interpreted as a commitment to any bit is at most 2^{-k} – in contrast to the one-way permutation based scheme above.

Pseudo-Random Functions. We also let $\{f_s\}_{s \in \{0,1\}^k}$ be a family of pseudorandom functions [18] mapping $\{0,1\}^*$ to $\{0,1\}^k$.

One-Time Signatures. Finally, let $(Gen, Sign, Ver)$ be a strong one-time signature scheme (see [29,33]), which can be constructed easily from universal one-way hash functions. Note that these objects can be constructed from one-way functions.

3.2 The Construction

Intuition. The NIZK system intuitively works as follows: First, a verification-key/signing-key pair (VK, SK) is chosen for the one-time signature scheme. Then the prover provides a NIZK proof that *either* x is in the language, *or* that the reference string actually specifies a hidden pseudo-random function and that some specified value is the output of this pseudo-random function applied to the verification key VK. Finally, this proof is itself signed using the signing key SK.

We now describe the proof system Π for L precisely. Note that a third possibility for the NIZK proof is added below; this is a technical addition which simplifies our proof of correctness.

– **Common random reference string.** The reference string consists of three parts Σ_1, Σ_2, and Σ_3.

1. Σ_1 is a string that we break up into k pairs $(r_1, c_1), \ldots, (r_k, c_k)$. If we use the one-way permutation-based commitments, each r_i and c_i are of length k; if we use the Naor commitment scheme, r_i and c_i are of length $3k$.
2. Σ_2 is a string of length $3k$.
3. Σ_3 is a string of length polynomial in k. The exact length of Σ_3 depends on an NIZK proof system described below.

– **Prover Algorithm.** We define the language L' to be the set of tuples $(x, u, v, \Sigma_1, \Sigma_2)$ such that at least one of the following three conditions hold:
  - $x \in L$
  - Σ_1 consists of commitments to the bits of the k bit string s, and $u = f_s(v)$: Formally, there exists $s = s_1 \ldots s_k$ with $s_i \in \{0,1\}$ for each i, and there exist $a_1, a_2, \ldots, a_k \in \{0,1\}^k$ such that $u = f_s(v)$ and such that for each i, (r_i, c_i) is a commitment under C to the bit s_i.
  - There exists $s \in \{0,1\}^k$ such that $\Sigma_2 = G(s)$

  We assume we have a single-theorem NIZK proof system for L' (which we denote Π'). Note that the length of the reference string Σ_3 should be $\ell_{\Pi'}(k)$. We now define the prover for L. On input x, a witness w, and the reference string $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$, the prover does the following:
  1. Use $Gen(1^k)$ to obtain a verification key / signing key pair (VK, SK) for the one-time signature scheme.
  2. Let u be uniformly selected from $\{0,1\}^k$.
  3. Using Σ_3 as the reference string and w as the witness, generate a single-theorem NIZK proof under Π' that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$. Denote this proof by π'.
  4. Output $(VK, x, u, \pi', Sign_{SK}(x, u, \pi'))$.

  As a sanity check, we observe that if $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$ is chosen uniformly, then the probability that Σ_1 can be interpreted as the commitment to any bits and the probability that Σ_2 is in the range of G are both exponentially small in k. Thus, with all but exponentially small probability over the choice of Σ_1 and Σ_2, a proof that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$ really does imply that $x \in L$.

– **Verifier Algorithm.** The verification procedure, on input the instance x and proof $(VK, x', u, \pi', \sigma)$, with respect to reference string $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$, proceeds as follows:
  1. Confirm that $x = x'$, and confirm the validity of the one-time signature — *i.e.* that $Ver_{VK}((x, u, \pi'), \sigma) = 1$.
  2. Verify that π' is a valid proof that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$.

– **Simulator Algorithm.** We now describe the two phases of the simulator algorithm. S_1 is the initial phase, which outputs a reference string Σ along with some auxiliary information τ. S_2 takes as input this auxiliary information, the reference string, and an instance x, and outputs a simulated proof for x. The intuition for the simulator is that it sets up the reference string to be such that a hidden pseudo-random function really is specified, and instead of proving that x is in the language, the simulator simply proves that it can evaluate this hidden pseudo-random function on the verification key of the signature scheme.

```
$\mathcal{S}_1(1^k)$ :
    $s, \Sigma_2 \leftarrow \{0,1\}^{3k}$; $\Sigma_3 \leftarrow \{0,1\}^{\ell_{\Pi'}(k)}$
    $a_i \leftarrow \{0,1\}^k$ for $i = 1, \ldots, k$
    $g_i \leftarrow C(s_i, a_i)$ for $i = 1. \ldots, k$
    $\Sigma_1 = (g_1, g_2, \ldots, g_k)$
    return $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$ and
        $\tau = (s, a_1, \ldots, a_k)$
```

```
$\mathcal{S}_2(\tau = (s, a_1, \ldots, a_k), \Sigma = (\Sigma_1, \Sigma_2, \Sigma_3), x)$ :
    $(VK, SK) \leftarrow Gen(1^k)$
    $u = f_s(VK)$
    Use $\Sigma_3$ as ref string and $\tau$ as witness to construct
        proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
    $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
    return $(VK, x, u, \pi', \sigma)$
```

Theorem 2. *If Π' is a single-theorem NIZK proof system for L', the proof system Π described above is either:*

- *an unbounded simulation-sound NIZK proof system for L if C is the Naor commitment scheme and one-way functions exist.*
- *an unbounded simulation-sound same-string NIZK argument for L with if C is the commitment scheme based on one-way permutations and one-way permutations exist.*

Proof. As they are standard, we only sketch the proofs for completeness, soundness, and zero-knowledge. We provide the proof of unbounded simulation soundness in full.

Completeness follows by inspection. For the case of NIZK proofs, soundness follows by the fact that if Σ is chosen uniformly at random, then the probability that Σ_1 can be interpreted as a commitment to any string is exponentially small, and likewise the probability that Σ_2 is in the image of the pseudorandom generator G is exponentially small. For the case of NIZK arguments, we will in fact establish not only soundness but the stronger simulation soundness property below.

In the case where C is based on a one-way permutation, we note that the simulator's distribution on Σ is exactly uniform, thus satisfying this property required by same-string NIZK.

The proof of unbounded zero-knowledge follows almost exactly techniques of [15]. First we note that if we modify the real prover experiment by replacing the uniform Σ_1 with the distribution from the simulation (which in the case where C is based on one-way permutations is no change at all), but keep the prover as is, then by the security of the commitment scheme, the views of the adversary are computationally indistinguishable. Now, [15] show that single-theorem NIZK implies unbounded witness-indistinguishability. Thus, since the simulator for Π uses only a different witness to prove the same statement, the view of the adversary in the simulator experiment is computationally indistinguishable from the view of the adversary in the modified prover experiment. Thus, unbounded zero-knowledge follows.

Unbounded simulation soundness – Overview. The proof of simulation soundness uses novel techniques based in part on a new application of pseudorandom functions to non-malleability. We also use a combination of techniques from [13, 33], [15], and [4]. As we do not use set selection at all, the proof is quite different from that techniques from [12,33]. The intuition is as follows: Through the use of the signature scheme, we know that any proof of a false theorem that the adversary might output which is different from the proofs provided by the simulator must use a verification key VK that is new. Otherwise, providing a valid signature would contradict the security of the signature scheme. Once we know that the verification key VK must be different, we observe that the only way to prove a false theorem with regard to the simulated reference string is to provide a value $u = f_s(VK)$. By considering several hybrid distributions, we show that this is impossible by the security of pseudorandom functions and the witness-indistinguishability of the NIZK proof system Π' for L'.

Unbounded simulation soundness – Full Proof. We recall from the definition of unbounded simulation soundness the adversary experiment, and substitute from our construction, to build experiment Expt_0.

```
$\mathsf{Expt}_0(1^k)$ (Actual Adversary Experiment):
    Make Reference String $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$:
        $\Sigma_2 \leftarrow \{0,1\}^{3k}$; $\Sigma_3 \leftarrow \{0,1\}^{\ell_{\Pi'}(k)}$
        $s \leftarrow \{0,1\}^k$
        $\Sigma_1 \leftarrow$ commitments to bits of $s$ using randomness $a_1, \ldots, a_k$.
    Run adversary $A$. When asked for proof for $x$, do:
        $(VK, SK) \leftarrow Gen(1^k)$
        $u = f_s(VK)$
        Use $\Sigma_3$ as ref string and $(s, a_1, \ldots, a_k)$ as witness
        to construct proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
        $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
        return $(VK, x, u, \pi', \sigma)$
    Let $(x, \pi)$ be output of adversary.
    Let $Q$ be list of proofs provided by simulator above.
    return true iff ( $\pi \notin Q$ and $x \notin L$ and $\mathcal{V}(x, \pi, \Sigma) =$ true )
```

Let $\Pr[\mathsf{Expt}_0(1^k)] = p(k)$. We must show that $p(k)$ is negligible.

We denote the components of the proof π output by the adversary as (VK, x, u, π', σ). Let T be the list of verification keys output by the simulator. (Note that with all but exponentially small probability, these verification keys will all be distinct.) We first consider the probability $\Pr[\mathsf{Expt}_0(1^k)$ and $VK \in T]$.

In the case where this is true, we know that $\pi \notin Q$, and therefore this implies that the adversary was able to produce a message/signature pair for VK different than the one given by the simulator. Thus, if $\Pr[\mathsf{Expt}_0(1^k)\ \texttt{and}\ VK \in T]$ is non-negligible, we can use it to forge signatures and break the (strong) one-time signature scheme. Thus, $\Pr[\mathsf{Expt}_0(1^k)\ \texttt{and}\ VK \in T]$ is negligible. Since $p(k) = \Pr[\mathsf{Expt}_0(1^k)\ \texttt{and}\ VK \in T] + \Pr[\mathsf{Expt}_0(1^k)\ \texttt{and}\ VK \notin T]$, we now need only focus on the second probability. Let $p_0(k) = \Pr[\mathsf{Expt}_0(1^k)\ \texttt{and}\ VK \notin T]$.

We now consider a second experiment, where we change the acceptance condition of the experiment:

```
$\mathsf{Expt}_1(1^k)$ (Accept only if $u = f_s(VK)$):
    Make Reference String $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$:
        $\Sigma_2 \leftarrow \{0,1\}^{3k}$; $\Sigma_3 \leftarrow \{0,1\}^{\ell_{\Pi'}(k)}$
        $s \leftarrow \{0,1\}^k$
        $\Sigma_1 \leftarrow$ commitments to bits of $s$ using randomness $a_1, \ldots, a_k$.
    Run adversary $A$. When asked for proof for $x$, do:
        $(VK, SK) \leftarrow Gen(1^k)$
        $u = f_s(VK)$
        Use $\Sigma_3$ as ref string and $(s, a_1, \ldots, a_k)$ as witness
            to construct proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
        $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
        return $(VK, x, u, \pi', \sigma)$
    Let $(x, \pi = (VK, x, u, \pi', \sigma))$ be output of adversary.
    Let $Q$ be list of proofs output by simulator above.
    Let $T$ be list of verification keys output by simulator above.
    return true iff
        ( $\pi \notin Q$ and $\mathcal{V}(x, \pi, \Sigma) =$ true and $VK \notin T$ and $u = f_s(VK)$)
```

Now, let $p_1(k) = \Pr[\mathsf{Expt}_1(1^k)]$. In Expt_1, we insist that $VK \notin T$ and replace the condition that $x \notin L$ with $f_s(VK) = u$. Note that with these changes, the experiment can be implemented in polynomial-time. Now, by the fact that Π' is a proof system for L', we know that if $x \notin L$, then with overwhelming probability the only way the adversary's proof can be accepted is if $f_s(VK) = u$. (Recall that in all cases, Π' is an NIZK proof system, not an argument.) Thus, we have that $p_0(k) \le p_1(k) + \alpha(k)$, where α is some negligible function.

We now consider a third experiment, where we change part of the reference string Σ_2 to make it pseudorandom:

Let $p_2(k) = \Pr[\mathsf{Expt}_2(1^k)]$. In Expt_2, the only change we made was to make Σ_2 be pseudorandom rather than truly random. Thus, we must have that $|p_2(k) - p_1(k)| \le \alpha(k)$, where α is some negligible function. Otherwise, this would yield a distinguisher for the generator G.

We now consider a fourth experiment, where instead of providing proofs based on proving $u = f_s(VK)$, we provide proofs based on the pseudorandom seed for Σ_2:

Let $p_3(k) = \Pr[\mathsf{Expt}_3(1^k)]$. In Expt_3, the only change we made was to have the simulator use the seed for Σ_2 as the witness to generate its NIZK

```
$\mathsf{Expt}_2(1^k)$ (Change $\Sigma_2$ to be pseudorandom):
    Make Reference String $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$:
        $d \leftarrow \{0,1\}^k$; Let $\Sigma_2 = G(d)$.
        $\Sigma_3 \leftarrow \{0,1\}^{\ell_{\Pi'}(k)}$
        $s \leftarrow \{0,1\}^k$
        $\Sigma_1 \leftarrow$ commitments to bits of $s$ using randomness $a_1, \ldots, a_k$.
    Run adversary $A$. When asked for proof for $x$, do:
        $(VK, SK) \leftarrow Gen(1^k)$
        $u = f_s(VK)$
        Use $\Sigma_3$ as ref string and $(s, a_1, \ldots, a_k)$ as witness
            to construct proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
        $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
        return $(VK, x, u, \pi', \sigma)$
    Let $(x, \pi = (VK, x, u, \pi', \sigma))$ be output of adversary.
    Let $Q$ be list of proofs output by simulator above.
    Let $T$ be list of verification keys output by simulator above.
    return true iff
        ( $\pi \notin Q$ and $\mathcal{V}(x, \pi, \Sigma) =$ true and $VK \notin T$ and $u = f_s(VK)$)
```

```
$\mathsf{Expt}_3(1^k)$ (Use seed for $\Sigma_2$ to generate NIZK proofs):
    Make Reference String $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$:
        $d \leftarrow \{0,1\}^k$; Let $\Sigma_2 = G(d)$.
        $\Sigma_3 \leftarrow \{0,1\}^{\ell_{\Pi'}(k)}$
        $s \leftarrow \{0,1\}^k$
        $\Sigma_1 \leftarrow$ commitments to bits of $s$ using randomness $a_1, \ldots, a_k$.
    Run adversary $A$. When asked for proof for $x$, do:
        $(VK, SK) \leftarrow Gen(1^k)$
        $u = f_s(VK)$
        Use $\Sigma_3$ as ref string and $d$ as witness
            to construct proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
        $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
        return $(VK, x, u, \pi', \sigma)$
    Let $(x, \pi = (VK, x, u, \pi', \sigma))$ be output of adversary.
    Let $Q$ be list of proofs output by simulator above.
    Let $T$ be list of verification keys output by simulator above.
    return true iff
        ( $\pi \notin Q$ and $\mathcal{V}(x, \pi, \Sigma) =$ true and $VK \notin T$ and $u = f_s(VK)$)
```

proof that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$. Note that this means that s and the randomness $a_1, \ldots, a_k$ are not used anywhere except to generate Σ_1. Now, [15] prove that any adaptive single-theorem NIZK proof system is also adaptive unbounded witness-indistinguishable (see [15] for the definition of witness-indistinguishable non-interactive proofs). The definition of adaptive unbounded witness-indistinguishability directly implies that $|p_3(k) - p_2(k)| \leq \alpha(k)$, where α is some negligible function.

We now consider a fifth experiment, where finally we eliminate all dependence on s by chosing Σ_1 independently of s:

```
$\mathsf{Expt}_4(1^k)$ (Make $\Sigma_1$ independent of $s$):
    Make Reference String $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$:
        $d \leftarrow \{0,1\}^k$; Let $\Sigma_2 = G(d)$.
        $\Sigma_3 \leftarrow \{0,1\}^{\ell_{\Pi'}(k)}$
        $s, s' \leftarrow \{0,1\}^k$
        $\Sigma_1 \leftarrow$ commitments to bits of $s'$ using randomness $a_1, \ldots, a_k$.
    Run adversary $A$. When asked for proof for $x$, do:
        $(VK, SK) \leftarrow Gen(1^k)$
        $u = f_s(VK)$
        Use $\Sigma_3$ as ref string and $d$ as witness
            to construct proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
        $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
        return $(VK, x, u, \pi', \sigma)$
    Let $(x, \pi = (VK, x, u, \pi', \sigma))$ be output of adversary.
    Let $Q$ be list of proofs output by simulator above.
    Let $T$ be list of verification keys output by simulator above.
    return true iff
        ( $\pi \notin Q$ and $\mathcal{V}(x, \pi, \Sigma) =$ true and $VK \notin T$ and $u = f_s(VK)$)
```

Let $p_4(k) = \Pr[\mathsf{Expt}_4(1^k)]$. In Expt_4, we choose two independent uniformly random strings s, s' and make Σ_1 into a commitment to s' rather than s. This has the effect of making Σ_1 completely independent of the string s.

Suppose $s^0, s^1 \leftarrow \{0,1\}^k$; $b \leftarrow \{0,1\}$, and $\Sigma_1 \leftarrow$ commitments to bits of s^b. By the security of the commitment scheme (either by Naor[27] or Goldreich-Levin [17], depending on which scheme we use), we know that for every polynomial-time algorithm B, we have that $\Pr[B(s^0, s^1, \Sigma_1) = b] \leq \frac{1}{2} + \alpha(k)$, where α is some negligible function.

Consider the following algorithm B: On input s^0, s^1, Σ_1, execute Expt_4 (or equivalently Expt_3), except with $s = s^0$ and $s' = s^1$, and using the value of Σ_1 specified as input to B. Return 1 if the experiment succeeds.

Then:

$$\begin{aligned}
\Pr[B = b] &= \frac{1}{2}\Pr[B = 1 | b = 1] + \frac{1}{2}\Pr[B = 0 | b = 0] \\
&= \frac{1}{2}(1 - p_4(k)) + \frac{1}{2}p_3(k) \\
&= \frac{1}{2} + \frac{1}{2}(p_3(k) - p_4(k))
\end{aligned}$$

Thus, we have that $p_3(k) - p_4(k) \leq \alpha(k)$ for some negligible function α.

Finally, we consider the last experiment, where we replace the pseudorandom function f with a truly random function:

```
$\mathsf{Expt}_5(1^k)$ (Replace $f$ with truly random function):
    Make Reference String $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$:
        $d \leftarrow \{0,1\}^k$; Let $\Sigma_2 = G(d)$.
        $\Sigma_3 \leftarrow \{0,1\}^{\ell_{\Pi'}(k)}$
        $s, s' \leftarrow \{0,1\}^k$
        $\Sigma_1 \leftarrow$ commitments to bits of $s'$ using randomness $a_1, \ldots, a_k$.
    Run $A$ with oracle to simulator. When asked for proof of $x$, do:
        $(VK, SK) \leftarrow Gen(1^k)$
        $u \leftarrow \{0,1\}^k$
        Use $\Sigma_3$ as ref string and $d$ as witness
            to construct proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
        $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
        return $(VK, x, u, \pi', \sigma)$
    Let $(x, \pi = (VK, x, u, \pi', \sigma))$ be output of adversary.
    Let $Q$ be list of proofs output by simulator above.
    Let $T$ be list of verification keys output by simulator above.
    Let $u' \leftarrow \{0,1\}^k$
    return true iff
        ( $\pi \notin Q$ and $\mathcal{V}(x, \pi, \Sigma) =$ true and $VK \notin T$ and $u = u'$)
```

Let $p_5(k) = \Pr[\mathsf{Expt}_5(1^k)]$. In Expt_5, we replace the pseudorandom function f_s with a truly random function F, which simply returns a truly random value at each query point. Note that since we only consider the case where $VK \notin T$, this means that $F(VK)$ will be a uniformly selected value (which we denote u') that is totally independent of everything the adversary sees. Thus, it follows that $p_5(k) \leq 2^{-k}$ since the probability that any value output by the adversary equals u' is at most 2^{-k}.

On the other hand, we will argue that $p_4(k)$ and $p_5(k)$ can only be negligibly apart by the pseudorandomness of $\{f_s\}$. Consider the following machine M which is given an oracle O to a function from $\{0,1\}^k$ to $\{0,1\}^k$: Execute experiment $\mathsf{Expt}_4(k)$ except replace any call to f_s with a call to the oracle. Note that s is not used in any other way in $\mathsf{Expt}_4(k)$. Return 1 iff the experiment succeeds.

Now, if the oracle provided to M is an oracle for f_s with $s \leftarrow \{0,1\}^k$, then $\Pr[M^O = 1] = p_4(k)$. If M is provided with an oracle for a truly random function F, then $\Pr[M^O = 1] = p_5(k)$. By the pseudorandomness of $\{f_s\}$, it follows that $|p_4(k) - p_5(k)| \leq \alpha(k)$ for some negligible function α.

In conclusion, we have that $p_5(k) \leq 2^{-k}$, and that $p_i(k) \leq p_{i+1}(k) + \alpha(k)$ for some negligible function α for each $i = 0, 1, 2, 3, 4$. Thus, $p_0(k) \leq \beta(k)$ for some negligible function β, which finally implies that $p(k)$ is negligible, completing the proof.

Theorem 3. *If the NIZK proof system Π' in the construction above is replaced by a single-theorem NIZK proof of knowledge for L', and assuming one-way* *functions exist, then Π is an unbounded non-malleable (in the explicit witness sense) NIZK proof system (or argument) for L. In particular if Π was also same-string NIZK, then Π is a Robust NIZK argument.*

Proof. (**Sketch**) This follows from essentially the same argument as was used above to prove that Π is unbounded simulation-sound. We sketch the details here.

To prove unbounded non-malleability in the explicit witness sense, we must exhibit a machine M that with oracle access to the adversary A produces an instance x, together with a witness w for membership of $x \in L$, satisfying some relation. Recall that since Π' is a proof of knowledge, there are extractor machines E_1 and E_2. We describe our machine M explicitly below:

```
$M^A(1^k)$ (Non-Malleability Machine):
    Make Reference String $\Sigma = (\Sigma_1, \Sigma_2, \Sigma_3)$:
        $(\Sigma_3, \tau) \leftarrow E_1(1^k)$
        $\Sigma_2 \leftarrow \{0,1\}^{3k}$
        $s \leftarrow \{0,1\}^k$
        $\Sigma_1 \leftarrow$ commitments to bits of $s$ using randomness $a_1, \ldots, a_k$.
    Interact with $A(\Sigma)$. When asked for proof of $x$, do:
        $(VK, SK) \leftarrow Gen(1^k)$
        $u = f_s(VK)$
        Use $\Sigma_3$ as ref string and $(s, a_1, \ldots, a_k)$ as witness
            to construct proof $\pi'$ that $(x, u, VK, \Sigma_1, \Sigma_2) \in L'$
        $\sigma \leftarrow Sign_{SK}(x, u, \pi')$
        return $(VK, x, u, \pi', \sigma)$
    Let $(x, \pi = (VK, x, u, \pi', \sigma), \text{aux})$ be output of adversary.
    Let $w' \leftarrow E_2(\Sigma, \tau, (x, u, VK, \Sigma_1, \Sigma_2), \pi')$
    If $w'$ is a witness for $x \in L$, return $(x, w', \text{aux})$, else abort
```

M essentially executes $\mathsf{Expt}^S_{A,R}(k)$ from the definition of non-malleability, except using E_1 to generate Σ_3, (recall that this output of E_1 is distributed negligibly close to uniformly) and using E_2 to extract a witness from the NIZK proof for L'. We immediately see therefore that M will fail to meet the conditions of non-malleability only if there is a non-negligible probability that the witness w' returned by E_2 is not a witness for $x \in L$ and yet the proof π' is valid. By construction, with all but negligible probability over Σ_2 and Σ_3, this can only happen if w' is a witness for $u = f_s(VK)$. But the proof of simulation-soundness of Π implies that the adversary can output such a u with a valid proof π with only negligible probability. This shows that the probability of M's success is only negligibly different than the probability of success in the experiment $\mathsf{Expt}^S_{A,R}(k)$.

4 Second Construction

In this section, we exhibit our second construction of NIZK proof systems with unbounded adaptive non-malleability (in the explicit NP-witness sense). Our construction uses several tools, that can all be based on any NIZK proof of knowledge. In particular, this construction is based on a novel generalization of unduplicatable set selection [13,12,33] which we call *hidden undiplicatable set selection* which can be used to achieve unbounded non-malleability, and might be useful elsewhere. interest.

An informal description. As a starting point, we still would like to use the paradigm of [15] in order to be able to simulate arbitrarily many proofs, when requested by the adversary. In other words, we want to create a proof system where the simulator can use some "fake" witness to prove arbitrarily many theorems adaptively requested by an adversary but the adversary must use a "real" witness when giving a new proof.

One important step toward this goal is to use a new variation on the "unduplicatable set selection" technique (previously used in [13,12,33]). While in previous uses of unduplicatable set selection, the selected set was sent in the clear (for instance, being determined by the binary expansion of a commitment key or a signature public key), in our construction such a set is hidden.

Specifically, on input x, the prover picks a subset S of bits of the random string and proves that $x \in L$ or the subset S enjoys property P (to ensure soundness P is such that with overwhelming probability a subset of random bits does not enjoy P). The subset S is specified by a string s that is kept hidden from the verifier through a secure commitment. The same string s is used to specify a pseudo-random function f_s and the value of f_s on a random u is then used as source of randomness for the key generation of a signature scheme. To prevent that the adversary does not follow these instructions in generating the public key, our protocol requires that a non-interactive zero-knowledge proof for the correctness of this computation is provided. Thus, the prover actually produces two zero-knowledge proofs: the "real one" (in which he proves that $x \in L$ or the set S enjoys property P) and the "auxiliary proof" (in which he proves correctness of the construction). Finally, the two proofs are signed with the public key generated.

This way, the generation of the public key for the signature scheme is tied to the selected set S in the following sense: if an adversary tries to select the same set and the same input for the pseudo-random function as in some other proof he will be forced to use the same public key for the signature scheme (for which, however, she does not have a secret key).

Let us intuitively see why this protocol should satisfy unbounded non-malleable zero-knowledge. A crucial point to notice is that the simulator, when computing the multiple proofs requested by the adversary, will select a set of strings, set them to be pseudo-random and the remaining ones to be random, and always use this single selected set of strings, rather than a possibly different set for each proof, as done by a real prover; note however that the difference between these two cases is indistinguishable. As a consequence, the adversary, even after seeing many proofs, will not be able to generate a new proof without knowing its witness as we observe in the following three possible cases.

First, if the adversary tries to select a different set S' (from the one used in the simulation), then she is forced to use a random string. Therefore S' does not enjoy P and therefore she can produce a convincing real proof only if she has a witness for $x \in L$.

Second, if the adversary tries to select the same set of strings as the one used in the simulation and the same input for the pseudo-random function as at least in one of the proofs she has seen, then she is forced to use the same signature public key and therefore will have to forge a signaturewhich violates the security of the signature scheme used.

Third, if the adversary tries to select the same set of strings as the one used in the simulation and an input for the pseudo-random function different from all the proofs she has seen, she will either break the secrecy of the commitment scheme or the pseudorandomness of the pseudo-random function used.

Tools. We use the following tools:

1. A pseudo-random generator $g = \{g_n\}_{n \in \mathsf{N}}$ where $g_n : \{0,1\}^n \to \{0,1\}^{2n}$;
2. A pseudo-random family of functions $f = \{f_s\}_{s \in \mathsf{N}}$, where $f_s : \{0,1\}^{|s|} \to \{0,1\}^{|s|}$.
3. A commitment scheme (Commit,VerCommit).
   On input a n-bit string s and a n^a-bit random *reference* string σ, for a constant a, algorithm Commit returns a commitment key *com* and a decommitment key *dec* of length n^a. On input σ, s, com, dec, algorithm VerCommit returns 1 if *dec* is a valid decommitment key of *com* as s and $\perp$ otherwise.
4. A one-time strong signature scheme (KG,SM,VS).
   On input a random string r of length n^a for a constant a, algorithm KG returns a public key pk and a secret key sk of length n. On input pk, sk, a message m, algorithm SM returns a signature sig. On input pk, m, sig, algorithm VS returns 1 if sig is a valid signature of m or 0 otherwise.

In the description of our proof system we will use the following polynomial-time relations.

1. Let g be a pseudorandom generator that stretches random strings of length n into pseudorandom string of length $2n$. The domain of relation R_1 consists of a reference string σ, n pairs of $2n$-bit strings $(\tau_{i,0}, \tau_{i,1})_{i=1}^{n}$, and a commitment com such that com is the commitment of an n-bit string $s = s_1 \circ \cdots \circ s_n$ computed with reference string σ and for each $i = 1, \cdots, n$ there exists $\mathsf{seed}_i \in \{0,1\}^n$ such that $\tau_{i,s_i} = g_n(\mathsf{seed}_i)$. A witness for membership in the domain of R_1 consists of the decommitment key dec, the string s and the seeds $\mathsf{seed}_1, \cdots, \mathsf{seed}_n$.
2. Let KG be the key-generator algorithm of a secure signature scheme, $\{f_s\}$ a pseudorandom family of functions and g a pseudorandom generator that stretches random strings of length n into pseudorandom strings of length $2n$. The domain of relation R_2 consists of a public key pk, two reference strings σ_0 and σ_1, a commitment com, and an n-bit string u such that at least one of the following holds:

   a) String com is the commitment of an n-bit string s computed using σ_1 as reference string and pk is the output of KG on input $f_s(u)$.
   b) There exists an n-bit string r_0 such that $\sigma_0 = g(r_0)$.

   Witnesses of membership into R_2 are of two forms: either consist of decommitment dec and string s or of string r_0 such that $\sigma_0 = g(r_0)$. We denote by (A_2, B_2) a NIZK proof system of knowledge for relation R_2. We denote by E_{02}, E_{12}, S_2 the simulator and extractor associated with (A_3, B_3).
3. Relation R_3 is the or of relation R_1 and relation R. We denote by (A_3, B_3) a NIZK proof system of knowledge for relation R_3. We denote by E_{03}, E_{13}, S_3 the simulator and extractor associated with (A_3, B_3).

The Construction. Let R be a polynomial-time relation.

– **Common input.** $x \in \{0,1\}^n$.
– **Common random reference string.** The reference string consists of five parts:
  $\Sigma_0, \Sigma_1, \Sigma_2, \Sigma_3$,and Σ_4,where $\Sigma_4 = (\Sigma_{4,1,0} \circ \Sigma_{4,1,1}) \circ \cdots \circ (\Sigma_{4,n,0} \circ \Sigma_{4,n,1})$.
– **Prover Algorithm.** On input a witness w such that $R(x, w) = 1$, do the following:
  1. Uniformly choose $s \in \{0,1\}^n$ and $u \in \{0,1\}^n$;
  2. let $(\mathsf{com}, \mathsf{dec}) = Commit(\Sigma_1, s)$;
  3. let $r = f_s(u)$ and $(pk, sk) = Gen(1^k, r)$;
  4. using reference string Σ_2, input $I_2 = (pk, \Sigma_0, \Sigma_1, \mathsf{com}, u)$ and and witness $W_2 = (\mathsf{dec}, s)$, generate an NIZK proof of knowledge π_2 of W_2 such that $R_2(I_2, W_2) = 1$;
  5. using reference string Σ_3, input $I_3 = (\Sigma_4, \mathsf{com}, x)$ and $W_3 = w$ as witness generate an NIZK proof of knowledge π_3 of W_3 that $R_3(I_3, W_3) = 1$;
  6. let $mes = (\mathsf{com}, u, \pi_2, \pi_3)$;
  7. compute signature $sig = Sign(pk, sk, mes)$ and output (mes, pk, sig).
– **Verifier Algorithm.** On input $(\mathsf{com}, u, \pi_2, \pi_3, sig)$ do the following:
  1. verify that sig is a valid signature of $(\mathsf{com}, u, \pi_2, \pi_3)$;
  2. verify that π_2 and π_3 are correct;
  3. if all these verification are satisfied then output: ACCEPT and halt, else output: REJECT and halt.

The above protocol, as written, can be used to show the following

Theorem 4. *If there exists an efficient NIZK proof of knowledge for an NP-complete language, then there exists (constructively) an unbounded non-malleable (in the explicit witness sense) NIZK proof system for any language in NP.*

Consider the above protocol, where NIZK proofs of knowledge are replaced by NIZK proofs of membership. The resulting protocol can be used to show the following

Theorem 5. *If there exists an efficient NIZK proof of membership for an NP-complete language, and there exist one-way functions, then there exists (constructively) an simulation-sound NIZK proof system for any language in NP.*

In Appendix B we present a proof of Theorem 4 (note that, as done for our first construction, we can use part of this proof to prove Theorem 5).

Acknowledgments. Part of this work done while the third author was visiting Universitá di Salerno and part was done while the fourth author was visiting Telcordia Technologies and DIMACS. We thank Shafi Goldwasser and Oded Goldreich for valuable discussions.

References

1. M. Blum, A. De Santis, S. Micali and G. Persiano, Non-Interactive Zero-Knowledge Proofs. *SIAM Journal on Computing*, vol. 6, December 1991, pp. 1084–1118.
2. M. Blum, P. Feldman and S. Micali, Non-interactive zero-knowledge and its applications. *Proceedings of the* 20th *Annual Symposium on Theory of Computing*, ACM, 1988.
3. G. Brassard, D. Chaum and C. Crépeau, *Minimum Disclosure Proofs of Knowledge.* JCSS, v. 37, pp 156-189.
4. M. Bellare, S.Goldwasser, New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. *Advances in Cryptology – Crypto* 89 *Proceedings*, Lecture Notes in Computer Science Vol. 435, G. Brassard ed., Springer-Verlag, 1989.
5. R. Canetti, O. Goldreich, S. Goldwasser, and S. Micali. Resettable Zero-Knowledge. ECCC Report TR99-042, revised June 2000. Available from `http://www.eccc.uni-trier.de/eccc/`. Preliminary version appeared in ACM STOC 2000.
6. R. Canetti, J. Kilian, E. Petrank, and A. Rosen Black-Box Concurrent Zero-Knowledge Requires $\tilde{\Omega}(\log n)$ Rounds. *Proceedings of the* -67th *Annual Symposium on Theory of Computing*, ACM, 1901.
7. R. Cramer and V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. *Advances in Cryptology – Crypto* 98 *Proceedings*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
8. A. De Santis and G. Persiano, Zero-knowledge proofs of knowledge without interaction. *Proceedings of the* 33rd *Symposium on Foundations of Computer Science*, IEEE, 1992.
9. A. De Santis, G. Di Crescenzo and G. Persiano, Randomness-efficient Non-Interactive Zero-Knowledge. Proceedings of 1997 *International Colloquium on Automata, Languagues and Applications* (ICALP 1997).
10. A. De Santis, G. Di Crescenzo and G. Persiano, Non-Interactive Zero-Knowledge: A Low-Randomness Characterization of NP. Proceedings of 1999 *International Colloquium on Automata, Languagues and Applications* (ICALP 1999).
11. A. De Santis, G. Di Crescenzo and G. Persiano, Necessary and Sufficient Assumptions for Non-Interactive Zero-Knowledge Proofs of Knowledge for all NP Relations. Proceedings of 2000 *International Colloquium on Automata, Languagues and Applications* (ICALP 2000).
12. G. Di Crescenzo, Y. Ishai, and R. Ostrovsky, Non-Interactive and Non-Malleable Commitment. *Proceedings of the* 30th *Annual Symposium on Theory of Computing*, ACM, 1998.
13. D. Dolev, C. Dwork, and M. Naor, Non-Malleable Cryptography. *Proceedings of the* -45th *Annual Symposium on Theory of Computing*, ACM, 1923 and SIAM Journal on Computing, 2000.

14. C. Dwork, M. Naor, and A. Sahai, Concurrent Zero-Knowledge. *Proceedings of the* 30th *Annual Symposium on Theory of Computing*, ACM, 1998.
15. U. Feige, D. Lapidot, and A. Shamir, Multiple non-interactive zero knowledge proofs based on a single random string. In *31st Annual Symposium on Foundations of Computer Science*, volume I, pages 308–317, St. Louis, Missouri, 22–24 October 1990. IEEE.
16. O. Goldreich, *Secure Multi-Party Computation*, 1998. First draft available at `http://theory.lcs.mit.edu/~oded`
17. O. Goldreich and L. Levin, *A Hard Predicate for All One-way Functions* . *Proceedings of the* 21st *Annual Symposium on Theory of Computing*, ACM, 1989.
18. O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions. *Journal of the ACM,* Vol. 33, No. 4, 1986, pp. 210–217.
19. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. *Proceedings of the* 19th *Annual Symposium on Theory of Computing*, ACM, 1987.
20. O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but their Validity or All Languages in NP have Zero-Knowledge Proof Systems. Journal of ACM 38(3): 691–729 (1991).
21. S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, February 1989.
22. S. Goldwasser, R. Ostrovsky Invariant Signatures and Non-Interactive Zero-Knowledge Proofs are Equivalent. *Advances in Cryptology – Crypto* 92 *Proceedings*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992.
23. J. Håstad, R. Impagliazzo, L. Levin, and M. Luby, Construction of pseudorandom generator from any one-way function. SIAM Journal on Computing. Preliminary versions by Impagliazzo et. al. in *21st STOC* (1989) and Håstad in *22nd STOC* (1990).
24. J. Kilian, E. Petrank An Efficient Non-Interactive Zero-Knowledge Proof System for NP with General Assumptions, Journal of Cryptology, vol. 11, n. 1, 1998.
25. J. Kilian, E. Petrank Concurrent and Resettable Zero-Knowledge in Poly-logarithmic Rounds. *Proceedings of the* -67th *Annual Symposium on Theory of Computing*, ACM, 1901
26. M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect zero-knowledge arguments for NP can be based on general complexity assumptions. *Advances in Cryptology – Crypto* 92 *Proceedings*, Lecture Notes in Computer Science Vol. 740, E. Brickell ed., Springer-Verlag, 1992 and *J. Cryptology*, 11(2):87–108, 1998.
27. M. Naor, Bit Commitment Using Pseudo-Randomness, *Journal of Cryptology*, vol 4, 1991, pp. 151–158.
28. M. Naor and M. Yung, Public-key cryptosystems provably secure against chosen ciphertext attacks. *Proceedings of the* 22nd *Annual Symposium on Theory of Computing*, ACM, 1990.
29. M. Naor and M. Yung, "Universal One-Way Hash Functions and their Cryptographic Applications", *Proceedings of the* 21st *Annual Symposium on Theory of Computing*, ACM, 1989.
30. R. Ostrovsky One-way Functions, Hard on Average Problems and Statistical Zero-knowledge Proofs. In Proceedings of 6th Annual Structure in Complexity Theory Conference (STRUCTURES-91) June 30 – July 3, 1991, Chicago. pp. 51-59

31. R. Ostrovsky, and A. Wigderson One-Way Functions are Essential for Non-Trivial Zero-Knowledge. Appeared In Proceedings of the second Israel Symposium on Theory of Computing and Systems (ISTCS-93) Netanya, Israel, June 7th-9th, 1993.
32. C. Rackoff and D. Simon, Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. *Advances in Cryptology – Crypto* 91 *Proceedings*, Lecture Notes in Computer Science Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.
33. A. Sahai Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. *Proceedings of the* 40th *Symposium on Foundations of Computer Science*, IEEE, 1999
34. A. Sahai and S. Vadhan A Complete Problem for Statistical Zero Knowledge. Preliminary version appeared in *Proceedings of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997. Newer version may be obtained from authors' homepages.

A Discussion of Usefulness of ZK in Multiparty Settings

Goldreich, Micali, and Wigderson [19] introduced a powerful paradigm for using zero-knowledge proofs in multiparty protocols. The idea is to use zero-knowledge proofs to force parties to behave according to a specified protocol in a manner that protects the secrets of each party. In a general sense, the idea is to include with each step in a protocol a zero-knowledge proof that the party has acted correctly. Intuitively, because each participant is providing a *proof*, they can only successfully give such a proof if they have, in truth, acted correctly. On the other hand, because their proof is *zero knowledge*, honest participants need not fear losing any secrets in the process of proving that they have acted correctly.

To turn this intuition into a proof that no secrets are lost, the general technique is to simulate the actions of certain parties without access to their secrets. The definition of zero knowledge (in both interactive and non-interactive settings) is based on the existence of a simulator which can produce simulated proofs of arbitrary statements. This often makes it easy to simulate the actions of parties (which we call the high-level simulation) as needed to prove that no secrets are lost.

The problem of malleability, however, can arise here in a subtle way. One feature of simulators for zero-knowledge proofs is that they can simulate proofs of false statements. In fact, this is often crucial in the high-level simulation of parties, because without knowing their secrets it is often not possible to actually follow the protocol they way they are supposed to. However, on the other hand, it may also be crucial in the high-level simulation that the proofs received by a simulated party be correct! As an example which arises in the context of chosen-ciphertext security for public-key encryption [28], consider the following: Suppose in a protocol, one party is supposed to send encryptions of a single message m under two different public keys K_1 and K_2. According to our paradigm, this party should also provide a zero-knowledge proof that indeed these two encryptions are encryptions of the same message. Now, suppose the receiver is supposed to know both decryption keys k_1 and k_2. But suppose that because we are simulating the receiver, we only know one key k_1. Suppose further that the simulator needs to decypher the message m in order to be able to continue the protocol. Now, if we could always trust proofs to be correct, knowing just one key would be enough, since we would know for sure that the two encryptions are encrypting the same message, and therefore the decryption of any one of them would provide us with m.

Here is where the malleability problem arises: Perhaps a simulated party occasionally provides simulated proofs of false statements. If the proof system is malleable, another party could turn around and provide the receiver above with two inconsistent encryptions and a false proof that they are consistent. Now, in this case, the behavior of the simulated party would be different from the behavior of the real party, because the simulator would not notice this inconsistency. Indeed, this very problem arises in the context of chosen-ciphertext security, and illustrates how malleable proofs can make it difficult to construct simulators. If we look more closely, we see that more specifically, the problem is the possibility that an adversary can use simulated proofs to construct proofs for false statements. Sahai [33] considered this problem by introducing the notion of a *simulation-sound* proof system, although he is not able to construct simulation-sound NIZK proof systems immune to any polynomial number of false proofs. (Note that our notion of non-malleability implies simulation soundness.) In this work, we show how to achieve simulation-sound NIZK proof systems immune to any polynomial number of false proofs. Our construction of such NIZK systems requires the assumption of one-way trapdoor permutations – a possibly weaker computational assumption then dense cryptosystems.

B Proof for Our Second Construction

First of all we need to show that the proposed protocol is an efficient NIZK proof system for the language equal to the domain of relation R; namely, that it satisfies the completeness and soundness requirements, and that the prover runs in polynomial-time, when given the appropriate witness. It is immediate to check that the properties of completeness and soundness are verified by the described protocol. In particular, for the completeness and the efficiency of the prover, note that since the honest prover has a witness for relation R, she can compute the proof π_3 in step 5 and make the verifier accept; for the soundness, note that if the input x is not in the domain of relation R then since the reference string is uniformly distributed, input I_3 is not in the domain of relation R_3 and therefore, from the soundness of (A_3, B_3), the verifier can be convinced with probability at most exponentially small.

In the rest of the proof, we prove the non-malleability property of our proof system. We start by presenting a construction for the adaptive simulator algorithm and the non-malleability machine, and then prove that, together with the above proof system, they satisfy the non-malleability property of Definition 5

The adaptive simulator algorithm. We now describe the simulator S algorithm for the proof system presented. S consists of two distinct machines: S_1, which constructs a reference string Σ along with some auxiliary information aux, and S_2 which takes as input Σ, aux and an instance x ad outputs a simulated proof π for x.

```
ALGORITHM $S_1(1^n)$.
1. Randomly choose $\Sigma_0 \in \{0,1\}^{2n}$, $\Sigma_1 \in \{0,1\}^{n^a}$ and $\Sigma_2$ and $\Sigma_3$;
2. randomly choose $s \in \{0,1\}^n$;
3. for $i = 1$ to $n$ do
       randomly pick $\mathsf{seed}_i$from $\{0,1\}^n$;
       set $\Sigma_{4,i,s_i} = g(\mathsf{seed}_i)$;
       randomly pick $\Sigma_{4,i,1-s_i}$from $\{0,1\}^{2n}$;
4. set $\Sigma = \Sigma_0 \circ \Sigma_1 \circ \Sigma_2 \circ \Sigma_3 \circ \Sigma_4$;
5. set aux $= (s, \mathsf{seed}_1, \cdots \mathsf{seed}_n)$;
6. output $(\Sigma, \mathrm{aux})$.
```

```
ALGORITHM $S_2(\Sigma, \mathrm{aux}, x)$.
1. Write aux as aux $= (s, \mathsf{seed}_1, \cdots \mathsf{seed}_n)$;
2. compute $(\mathsf{com}, \mathsf{dec})$from $Commit(\Sigma_1, s)$;
3. randomly pick $u$from $\{0,1\}^n$ and compute $r = f_s(u)$;
4. compute $(pk, sk) = KG(r)$;
5. using reference string $\Sigma_2$, input $I_2 = (pk, \Sigma_0, \Sigma_1, \mathsf{com}, u)$ and witness $W_2 = (\mathsf{dec}, s)$, generate an NIZK proof of knowledge $\pi_2$ of $W_2$ such that $R_2(I_2, W_2) = 1$;
6. using reference string $\Sigma_3$, input $I_3 = (\Sigma_4, \mathsf{com}, x)$ and witness $W_3 = (\mathsf{dec}, s, \mathsf{seed}_1, \cdots, \mathsf{seed}_n)$ generate an NIZK proof of knowledge $\pi_3$ of $W_3$ such that $R_3(I_3, W_3) = 1$;
7. set $mes = (\mathsf{com}, u, \pi_2, \pi_3)$;
8. compute signature $sig = Sign(pk, sk, mes)$ and output $(mes, pk, sig)$.
```

Note that the from the point of view of the adversary, the transcript output by the simulator S is indistinguishable from a real conversation with a prover, or otherwise either the secrecy of the commitment scheme or the security of the pseudorandom generator or the witness indstinguishability of the proof system used are violated. The proof of this is standard and is based on arguments from [15].

The non malleability machine M. The computation of the non-malleability machine M can be divided into three phases. During the first phase, M creates a reference string along with some auxiliary information to be used later; in the second phase M receives strings $x^1, \ldots, x^l$ from Adv and produces proofs $\pi^1, \ldots, \pi^l$; finally, in the third phase it receives a proof π^* for input x^* and extracts a witness w^* from π^*.

Input to M: security parameters 1^n.

Phase 1: Preprocessing.

0. Randomly choose $\Sigma_0 \in \{0,1\}^{2n}$;
1. randomly choose $\Sigma_1 \in \{0,1\}^{n^a}$;
2. run E_{20} on input 1^n to obtain Σ_2 along with auxiliary information aux_2;
3. run E_{30} on input 1^n to obtain Σ_3 along with auxiliary information aux_3;
4. randomly choose $s \in \{0,1\}^n$;
5. compute $(\mathsf{com}, \mathsf{dec}) = Commit(\Sigma_1, s)$;
6. `for` $i = 1$ to n `do`
   randomly pick $\mathsf{seed}_i$from $\{0,1\}^n$;
   set $\Sigma_{4,i,s_i} = g(\mathsf{seed}_i)$;
   randomly pick $\Sigma_{4,i,1-s_i}$from $\{0,1\}^{2n}$.

Phase 2: Interact with adversary Adv. When asked for proof of x^i, do:

1. compute $(\mathsf{com}^i, \mathsf{dec}^i)$from $Commit(\Sigma_1, s)$;
2. randomly pick $u^i$from $\{0,1\}^n$ and compute $r^i = f_s(u^i)$;
3. compute $(pk^i, sk^i) = KG(r^i)$;
4. using reference string Σ_2, input $I_2^i = (pk^i, \Sigma_0, \Sigma_1, \mathsf{com}^i, u^i)$ and witness $W_2^i = (\mathsf{dec}^i, s)$, generate an NIZK proof of knowledge π_2^i of W_2^i such that $R_2(I_2^i, W_2^i) = 1$;
5. using reference string Σ_3, input $I_3^i = (\Sigma_4, \mathsf{com}^i, x^i)$ and witness $W_3 = (\mathsf{dec}^i, s, \mathsf{seed}_1, \cdots, \mathsf{seed}_n)$ generate an NIZK proof of knowledge π_3^i of W_3^i such that $R_3(I_3^i, W_3^i) = 1$;
6. compute $mes^i = (\mathsf{com}^i, u^i, \pi_2^i, \pi_3^i)$;
7. compute signature $sig^i = Sign(pk^i, sk^i, mes^i)$ and output (mes^i, pk^i, sig^i).

Phase 3: Output. Receive (x^*, π^*) from the adversary and do:

1. let $W_3^* = E_{31}(\Sigma_3, \mathrm{aux}_3, x^*, \pi^*)$;
2. if W_3^* is a witness for $x \in L$ then return W_3^* else return $\perp$.

Next we prove the non-malleability property. Note that if the adversary is successful in producing a convincing new proof π^* then she is also producing a convincing proof of knowledge π_3^* that some input I_3 belongs to the domain of relation R_3. Using this proof, M can extract a witness W_3 such that $R_3(I_3, W_3) = 1$. By the construction of R_3, this witness is either a witness for R (in which case M is successful) or a witness for R_1. Therefore the non-malleability property of our proof system is proved by the following

Lemma 1. *The probability that, at Phase 3, M extracts from proof π^* a witness for relation R_1 is negligible.*

Proof. First of all we assume that the proof returned by the adversary is accepting (namely, both proofs π_2^*, π_3^* in π^* for relations R_2, R_3, respectively, are accepting), otherwise there is nothing to prove. We then consider the following cases and for each of them we show that the probability is negligible for otherwise we would reach a contradiction by showing that Adv can be used to contradict one of our original assumptions about the cryptographic tools used.
Case (a): The adversary has used a string s^* different from s.
Case (b): The adversary has used the same string s and a value u^* equal to u^j for some j.
Case (c): The adversary has used the same string s and a value u^* different from all u^i's.

Proof for Case (a). Suppose $s^* \neq s$ and let i be such that $s_i^* \neq s_i$. Then with very high probability there exists no seed_i^* such that $g(\mathsf{seed}_i^*) = \Sigma_{4,i,s_i^*}$. Therefore, there exists no witness W_3^* for I_3^* and relation R_1 and thus by the soundness of the proof system used the verifier will reject with very high probability.

Proof for Case (b). We denote by l the number of queries performed by Adv and by $u^1, \cdots, u^l$ the values used by M in answering the l queries of Adv and by u^* the value used by Adv in its proof π.

Assume that there exists $j \in \{1, \ldots, l\}$ such that $u^* = u^j$. Then, given that Adv has used the same pseudorandom functions, and that we are assuming that the proof π_2^* returned by Adv is accepting, it must be the case that Adv has used the same public key pk^j as M.

Therefore, if the proof π^* generated by Adv is different from the proofs produced by M during Phase 2, it can be for one of the following two reasons (a) π contains a tuple $(\mathsf{com}^*, u^*, \pi_2^*, \pi_3^*)$ different from the corresponding tuple $(\mathsf{com}^j, u^j, \pi_2^j, \pi_3^j)$ used by M to answer the j-th query or (b) exhibit a different signature.

In case (a), Adv can be used to violate the unforgeability of the signature scheme used as it manages to produce a message and to sign it without having access to the secret key for the signature scheme.

Case (b) is ruled out by the property of the signature scheme employed saying that, given message m and its signature sig, it is hard to provide a new signature of m that is different from sig.

Proof for Case (c). In this section we show that the probability that M obtains in Phase 3 a witness W for relation R_1 and that the proof produced by the adversary has used the same values s as M and a different u is negligible.

We consider a series of 4 polynomial-time experiments $\mathsf{Expt}_0, \ldots, \mathsf{Expt}_3$ with the event that $\mathsf{Expt}_0(1^n)$ gives 1 in output being exactly the experiment of M interacting with Adv we are interested in.

Thus, denoting by $p_i(n)$ the probability $\Pr[\,\mathsf{Expt}_i(1^n)\,] = 1$, we need to show that $p_0(n)$ is negligible. We do so, 1) by showing that the output of the experiments $\mathsf{Expt}_i(1^n)$ and $\mathsf{Expt}_{i+1}(1^n)$ are indistinguishable and thus $|p_i(n) - p_{i+1}(n)|$ is negligible for $i = 0, 1, 2$; 2) by showing that $p_3(n)$ is negligible.

1. $\mathsf{Expt}_0(1^n)$.
   $\mathsf{Expt}_0(1^n)$ is exactly experiment $\mathsf{Expt}'_{A,R}$, the experiment of the adversary interacting with algorithm M. We only modify Phase 3.

Phase 3: Output. Receive (x^*, π^*) from Adv.

```
1. Write $\pi^*$ as $\pi^* = (\mathsf{com}^*, u^*, \pi_2^*, \pi_3^*, pk^*, sig^*)$.
2. Let $W_2^* = E_{12}(\Sigma_2, aux_2, x, \pi_2)$.
3. Write $W_2^*$ as $W_2^* = (\mathsf{dec}, s)$.
4. Let $W_3^* = E_{13}(\Sigma_3, aux_3, x, \pi_3)$.
5. If $W_3^*$ is a witness for $x \in L$ then output 0.
6. Write $W_3^*$ as $W_3^* = (\mathsf{dec}^*, s^*, \mathsf{seed}_1^*, \cdots, \mathsf{seed}_n^*)$.
7. Output 1 iff $s^* = s$ and $u^* \neq u^j$, for $j = 1, \cdots, l$.
```

2. $\mathsf{Expt}_1(1^n)$.
   In $\mathsf{Expt}_1(1^n)$ random string Σ_0 is the output of generator g_n on input a random n-bit string r_0 and the proofs at steps 4 and 5 of Phase 2 of M are produced using r_0 as witness.

Phase 1: Pre-Processing. Similar to Phase 1 of M with step 0 replaced with the following.

```
0. Randomly choose $r_0 \in \{0,1\}^n$ and set $\Sigma_0 = g_n(r_0)$.
```

Phase 2: Interacting with adversary. Receive x^i from Adv.
Receive x^i from Adv.
Modify steps 4 and 5 of Phase 2 of M in the following way:

```
4. using reference string $\Sigma_2$, input $I_2^i = (pk^i, \Sigma_0, \Sigma_1, \mathsf{com}^i, u^i)$ and witness $W_2^i = (r_0)$, generate an NIZK proof of knowledge $\pi_2^i$ of $W_2^i$ such that $R_2(I_2^i, W_2^i) = 1$;
5. using reference string $\Sigma_3$, input $I_3^i = (\Sigma_4, \mathsf{com}^i, x^i)$ and witness $W_3^i = (s, \mathsf{seed}_1, \cdots \mathsf{seed}_n)$ generate an NIZK proof of knowledge $\pi_3^i$ of $W_3^i$ such that $R_3(I_3^i, W_3^i) = 1$;
```

Phase 3: Output. Same as Expt_0.

The output of Expt_0 and Expt_1 are indistinguishable for otherwise we would violate either the pseudorandomness of the generator g or the witness indistinguishability of the proof system. This can be viewed by consider an intermediate experiment in which Σ_0 is output of g but the proof do not use it as witness.

3. $\mathsf{Expt}_2(1^n)$.
   Expt_2 differs from Expt_1 in the fact that pk is computed by KG on input a random value.

> *Phase 1: Pre-Processing.* Same as Expt_1.
> *Phase 2: Interact with the adversary.* Receive x^i from Adv.
> Modify step 3. of Phase 2 of M in the following way.
> 2. Randomly select r^i from $\{0,1\}^n$ and compute $(pk^i, sk^i) = KG(r^i)$.
>
> *Phase 3: Output.* Same as Expt_1.

To prove that the distribution of the output of Expt_1 and Expt_2 are indistinguishable we define experiments $\mathsf{Expt}_{2.j}$, for $j = 0, \cdots, l$. In the first j executions of Phase 2 of $\mathsf{Expt}_{2.j}$, the public file is computed as in Expt_1 and in the subsequent executions as in Expt_2. Thus distinguishing between the output of Expt_2 and Expt_1 implies the ability to distinguish between $\mathsf{Expt}_{2.\hat{j}}$ and $\mathsf{Expt}_{2.(\hat{j}+1)}$, for some $0 \le \hat{j} \le l-1$, which contradicts either the security of the commitment scheme or the pseudorandomness of f.
To substantiate this last claim, we consider the following three experiments. For sake of compactness, we look only at the relevant components of the proof, that is, the commitment com, the value u and the public key pk; we do not consider the remaining components since they stay the same in each experiment and their construction can be efficiently simulated.

> $\mathsf{Expt}_a(1^n)$
>
> 1. Pick s, r at random from $\{0,1\}^n$.
> 2. Compute commitment com of s.
> 3. Pick u at random from $\{0,1\}^n$.
> 4. Compute $pk = KG(f_s(u))$.
> 5. Output (com, u, pk).

> $\mathsf{Expt}_b(1^n)$
>
> 1. Pick s, r at random from $\{0,1\}^n$.
> 2. Compute commitment com of s.
> 3. Pick u at random from $\{0,1\}^n$.
> 4. Compute $pk = KG(f_r(u))$.
> 5. Output (com, u, pk).

> $\mathsf{Expt}_c(1^n)$
>
> a) Pick s, r at random from $\{0,1\}^n$.
> b) Compute commitment com of s.
> c) Pick u at random from $\{0,1\}^n$.
> d) Compute $pk = KG(r)$.
> e) Output (com, u, pk).

Now we have the following two observations:

Obs. 1 Expt_a and Expt_b are indistinguishable.
Suppose they are not and consider the following adversary A that contradicts the security of the commitment scheme. A receives two random n-bit strings s and r and a commitment *com* of either s or r and performs the following two steps. First A picks u at random from $\{0,1\}^n$ and then computes *pk* as $pk = KG(f_s(u))$.
Now notice that if *com* is a commitment of s then the triplet (com, u, pk) is distributed as in the output of $\mathsf{Expt}_a(1^n)$. On the other hand if *com* is a commitment of r, then (com, u, pk) is distributed as in the output of $\mathsf{Expt}_b(1^n)$.

Obs. 2 Expt_b and Expt_c are indistinguishable.
Suppose they are not and consider the following adversary A that contradicts the pseudorandomness of f. A has access to a black box that computes a function F that is either a completely random function f or a pseudorandom function f_r for some random n-bit string r. A performes the following steps to construct a triplet (com, u, pk). A picks s at random, computes a commitment *com* of s, picks u at random, feeds the black box u obtaining $t = F(u)$ and computes *pk* as $pk = KG(t)$.
Now notice that if F is a random function then then (com, u, pk) is distributed as in the output of $\mathsf{Expt}_c(1^n)$. On the other hand if F is a pseudorandom function f_r for some random r then (com, u, pk) is distributed as in the output of $\mathsf{Expt}_b(1^n)$.

By the above observations Expt_a (the simplified version of $\mathsf{Expt}_{2.\hat{j}}$) and Expt_c (the simplified version of $\mathsf{Expt}_{2.\hat{j}+1}$) are indistinguishable.

4. $\mathsf{Expt}_3(1^n)$.
Expt_3 differs from Expt_2 in the fact that a random string s' is committed to instead of string s.

Phase 1: Pre-Processing. Same as Expt_2 with the following exception: step 4 is modified as follows:
4. randomly pick $s, s,' \in \{0,1\}^n$;

Phase 2: Interact with the adversary. Receive x^i from Adv.
Modify step 1 of M in the following way:

1. Compute $(com^i, dec^i) = Commit(\Sigma_1, s')$ uniformly choose $u^i \in \{0,1\}^n$.

Output. Same as Expt_0.

The distributions of the output of Expt_3 and Expt_2 are indistinguishable for otherwise we could distinguish commitment.
Finally, observe that in $\mathsf{Expt}_3(1^n)$, what is seen by Adv is independent from s. Thus the probability that Adv guesses s is negligible. Therefore, $p_3(n)$ is negligible.