# On Simulation-Sound Trapdoor Commitments

Philip MacKenzie[1] and Ke Yang[2]

[1] Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07040,
philmac@lucent.com

[2] Computer Science Department, Carnegie Mellon University,
Pittsburgh, PA 15213,
yangke@cs.cmu.edu

**Abstract.** We study the recently introduced notion of a *simulation-sound trapdoor commitment (SSTC)* scheme. In this paper, we present a new, simpler definition for an SSTC scheme that admits more efficient constructions and can be used in a larger set of applications. Specifically, we show how to construct SSTC schemes from any one-way functions, and how to construct very efficient SSTC schemes based on specific number-theoretic assumptions. We also show how to construct simulation-sound, non-malleable, and universally-composable zero-knowledge protocols using SSTC schemes, yielding, for instance, the most efficient universally-composable zero-knowledge protocols known. Finally, we explore the relation between SSTC schemes and non-malleable commitment schemes by presenting a sequence of implication and separation results, which in particular imply that SSTC schemes are non-malleable.

## 1 Introduction

The notion of a *commitment* is one of the most important and useful notions in cryptography. Intuitively, a commitment is the digital equivalent of a "locked combination safe." A party Alice would commit to a value by placing it into the safe, closing the safe, and spinning the lock, so that the value may later be revealed by Alice divulging the combination of the safe. Obviously, the value cannot be viewed by any other party prior to this opening (this is known as the "secrecy" or "hiding" property), and cannot be altered (this is known as the "binding" property). Commitments have been useful in a wide range of applications, from zero-knowledge protocols (e.g., [4,12,26]) to electronic commerce (e.g., remote electronic bidding), and have been studied extensively (e.g., [3,32,33]). In many cases, however, one needs commitment schemes with *additional properties* besides hiding and binding, such as those described below.

A *trapdoor commitment (TC) scheme* is a commitment scheme with an additional "equivocability" property. Roughly speaking, for such a commitment scheme there is some *trapdoor* information whose knowledge would allow one to open a commitment in more than one way (and thus "equivocate"). Naturally, without the trapdoor, equivocation would remain computationally infeasible [4,20,2].

C. Cachin and J. Camenisch (Eds.): EUROCRYPT 2004, LNCS 3027, pp. 382–400, 2004.

A *non-malleable commitment (NMC) scheme* is a commitment scheme with the property that (informally) not only is the value $v$ placed inside a commitment secret, but seeing this commitment does not give another party any advantage in generating a new commitment that, once $v$ is revealed, can then be opened to a value related to $v$ [18,16,23,17,14].[3]

Finally, a *universally composable commitment (UCC) scheme* is a commitment scheme with a very strong property that intuitively means that the security of a commitment is guaranteed even when commitment protocols are concurrently composed with arbitrary protocols [5,6,15]. To achieve universal composability, a commitment scheme seems to require equivocability, non-malleability, and furthermore, *extractability*. Roughly speaking, an extractable commitment scheme has a modified secrecy definition, which states that there is a *secret key* whose knowledge would allow one to extract the value placed in a commitment. Naturally, without this knowledge, the value would remain hidden. We note that the notion of a UCC scheme appears to be strictly stronger than the other notions of commitment schemes. In particular, Damgård and Groth [14] show that a UCC scheme implies secure key exchange, while both TC schemes and NMC schemes can be constructed from one-way functions.

### 1.1 Simulation-Sound Trapdoor Commitments

In this paper, we focus our attention on another extension of commitment schemes, namely simulation-sound trapdoor commitment (SSTC) schemes. An SSTC scheme is a TC scheme with a strengthened binding property, called simulation-sound binding. Roughly speaking, in an SSTC scheme, an adversary cannot equivocate on a commitment with a certain tag, even after seeing the equivocation of an unbounded number of commitments with different tags (i.e., the adversary may request an equivocation oracle to generate an unbounded number of commitments with different tags, and then to open them to arbitrary values). Here, a tag for a commitment is simply a binary string associated with the commitment. We will discuss tags in more detail below.

The term “simulation soundness” was first used to describe a property of zero-knowledge proofs by Sahai [37], and intuitively meant that even though an adversary could see simulated proofs of incorrect statements, it could not itself produce a new proof of any incorrect statement. Garay *et al.* [24] first applied this term to trapdoor commitments. They gave a slightly stronger, although more complicated, simulation-sound binding property and an efficient construction based on DSA signatures [29]. Their definition was specifically tailored to the goal of developing a universally-composable zero-knowledge (UCZK) proof that was secure in the presence of adversaries that could adaptively corrupt parties.[4]

[3] The original definition of [18] states (informally) that another party does not even have any advantage in creating a new commitment to a value related to $v$, regardless of the ability to open the new commitment. However, we will use the definition based on opening.

[4] They use the term *identifier* in place of the term *tag*, and intuitively, in their definition [24], a commitment made by the adversary using identifier *id* is binding, even if

### 1.2 Summary of Results

*Simpler Definition* We provide a simpler definition of SSTC schemes than the one by Garay *et. al.* [24]. Though the binding property in our definition is weaker, it is still sufficient in many applications (e.g., to construct UCZK protocols that are secure in the presence of adversaries that can adaptively corrupt parties).

We also discuss various design issues in the definition, and most notably, the choice between definitions based on the tag of the commitment and on the body of the commitment. Informally, a tag-based definition requires that an adversary cannot equivocate a commitment com with a certain tag so long as it does not see the equivocation of any commitment with the same tag. On the other hand, a body-based definition requires that the adversary cannot equivocate a commitment com so long as the commitment com itself has not been equivocated. (Note that we use the term "body" to refer to the bit-string that is the commitment.)

In our paper, we choose to focus on tag-based schemes since they admit simpler constructions and seem to be the most appropriate for our applications. For example, in constructing secure zero-knowledge protocols in the UC framework, where the communication is normally assumed to be authenticated, it is natural to use a tag-based scheme, setting the tag to be the pair of the identities of the prover and the verifier.

*Efficient Constructions* We present various constructions of SSTC schemes. The first construction is a generic one based on the (minimal) assumption that one-way functions exist. Our construction is similar to that of a UCC commitment scheme in Canetti *et. al.* [7]. However, because SSTC schemes do not require the extractability property, we are able to simplify the construction, and have it rely on a weaker assumption. The second construction is based on the DSA assumption, and is very efficient, involving only a small constant number of modular exponentiations. It is similar to the construction from Garay *et. al.* [24], but is about twice as efficient. The third assumption is based on Cramer-Shoup signatures [11], and relies on the strong RSA assumption [1]. It is also very efficient, again requiring only a small constant number of modular exponentiations.

We remark here that our most efficient SSTC schemes are more efficient than all known UCC schemes. For instance, the UCC constructions of [6,7] are for bit commitments, and thus have an expansion factor of at least the security parameter. The UCC construction of [15] has constant expansion factor, but requires a CRS of length proportional to the number of parties times the security parameter. Recently and independent from this work, Damgård and Groth [14] presented a UCC scheme with a constant expansion factor with a CRS whose length is independent of the number of parties. However, their scheme is still quite complicated, since it requires interaction, and uses two different types of

the adversary has seen any commitment using identifier $id$ opened (using an oracle that knows a trapdoor) once to any arbitrary value, and moreover, any commitment using identifier $id' \neq id$ opened (again using the oracle) an unbounded number of times to any arbitrary values.

commitments, one a non-malleable commitment scheme, and the other a special "mixed commitment scheme."

*Applications* We show constructions of unbounded simulation-sound, unbounded non-malleable, and universally composable zero-knowledge (ZK) protocols using SSTC schemes in the common reference string (CRS) model. In particular, we show how to (1) convert a $\Sigma$-protocol [10] (which is a special three-round, honest-verifier protocol where the verifier only sends random bits) into an unbounded simulation-sound ZK protocol; and (2) convert an $\Omega$-protocol [24] (which is a $\Sigma$-protocol with a straight-line extractor) into an unbounded non-malleable ZK protocol, and further into a universally-composable ZK protocol. The constructions are conceptually very simple. In fact, they all share the same structure, and all use a technique from Damgård [13] and Jarecki and Lysyanskaya [28]. The same technique was also used in Garay *et. al.* [24] in constructing a universally-composable ZK protocol that is secure against adaptive corruptions.

Our constructions are very efficient, and in particular our construction of a universally-composable ZK protocol is more efficient than previous constructions, at least when starting with a $\Sigma$-protocol. Compared to UCZK protocols based on universally-composable commitment schemes [6,7,14,15], our efficiency gain comes mainly from the fact that we avoid the Cook-Levin theorem [8,30],[5] but also from the fact that some of our SSTC schemes are more efficient than any UCC schemes, as discussed above. Compared to the UCZK protocol in Garay *et. al.* [24], our savings are twofold: the simpler SSTC construction (with a weaker definition) cuts the overhead of the SSTC commitments by half, and the direct use of the identities as tags eliminates the need for one-time signatures on the protocol transcripts.

In recent and independent work, Gennaro [25] presented an SSZK protocol[6] that is similar to our construction in Section 4. It uses a new type of commitment scheme called *multi-trapdoor commitments*, and an efficient implementation of this scheme based on the strong RSA assumption and a special hash property. A multi-trapdoor commitment scheme is similar to an SSTC scheme, except that it requires the existence of a different trapdoor (i.e., secret key) corresponding to each tag, and its security property corresponding to simulation-sound binding requires tags to be pre-chosen by the adversary.[7]

*Relation to Non-malleable Commitments* We discuss the relation between SSTC schemes and NMC schemes [18,16,17,14].[8] At first glance, binding and non-

[5] In previous constructions, they build a UCZK protocol $\Pi^L$ for an NP-complete language $L$ (e.g. Hamiltonian Cycle or Satisfiability), and then the UCZK protocols for any NP language is reduced to $\Pi^L$ via the Cook-Levin theorem, which is not very efficient.

[6] It is also concurrent non-malleable ZK, if rewinding is allowed in witness extraction

[7] We have recently defined a *static SSTC* scheme as a commitment scheme with only the second requirement, and note that it is also sufficient in our SSZK and NMZK constructions.

[8] Technically, when we refer to an NMC scheme, we will always mean an $\epsilon$-non-malleable commitment scheme, following the notation proposed in [17].

malleability (or analogously, equivocation and malleability) seem like very different notions: while the former concerns the adversary's ability to open a commitment to multiple values, the latter concerns the adversary's ability to produce and open a commitment to a single value related to a previously committed value. However, they are actually closely related, and we shall show that simulation-sound binding implies non-malleability (when both are appropriately defined). In fact, a similar observation was used implicitly in [16,17,14] to construct NMC schemes. In particular, these NMC schemes are all based on trapdoor commitment schemes that satisfy a weak notion of simulation-sound binding. (Note that these results all use body-based definitions instead of tag-based definitions.) However, the *exact* relationship between the notions of simulation-sound binding and non-malleability was not known, e.g., if simulation-sound binding is strictly stronger than non-malleability, or if they are equivalent.

We study the exact relationship between these two notions in this paper. To do this, we need to resolve some technical issues. First, just as SSTC schemes can be tag-based or body-based, NMC schemes can also be tag-based or body-based, where a tag-based NMC scheme is informally defined as one in which seeing a commitment (to some value $v$) with a certain tag does not give an adversary any advantage in generating a new commitment with a different tag that can later be opened to a value related to $v$. Since we focus on tag-based SSTC schemes, we will focus on their relation to tag-based NMC schemes.[9] (Analogous results could be obtained for the relationship between body-based SSTC schemes and body-based NMC schemes.) Second, an SSTC scheme is a TC scheme, so to make a useful comparison, we consider non-malleable trapdoor commitment (NMTC) schemes. Third, since an adversary for an SSTC scheme is allowed to query an equivocation oracle, we will also consider NMTC schemes in which an adversary is allowed to query an equivocation oracle.

Finally, we refine our definitions of SSTC schemes and NMTC schemes by specifying the number of equivocation oracle queries an adversary is allowed to make. An equivocation oracle, on a $\mathsf{commit}$ query, produces a commitment $\widetilde{\mathsf{com}}$ and on an $\mathsf{decommit}$ query, opens $\widetilde{\mathsf{com}}$ to an arbitrary value. We say a $\mathsf{TC}$ scheme is $\mathsf{SSTC}(\ell)$, if it remains secure if the adversary is allowed to make at most $\ell$ $\mathsf{commit}$ queries to the oracle (with no restriction on the number of $\mathsf{decommit}$ queries). We define $\mathsf{NMTC}(\ell)$ schemes similarly. We use $\mathsf{SSTC}(\infty)$ and $\mathsf{NMTC}(\infty)$ to denote the schemes where the adversary can make an unlimited number of $\mathsf{commit}$ queries. With the refined definitions (except for those related to the definition in [14], discussed below), we shall then prove that, for any constant $\ell$, $\mathsf{SSTC}(\ell+1)$ is strictly stronger than $\mathsf{NMTC}(\ell)$ and $\mathsf{NMTC}(\ell)$ is strictly stronger than $\mathsf{SSTC}(\ell)$. (In particular, note that even an $\mathsf{SSTC}(1)$ scheme is strictly stronger than an $\mathsf{NMC}$ scheme, since an $\mathsf{NMTC}(0)$ scheme is at least as strong as an $\mathsf{NMC}$ scheme.) Furthermore, $\mathsf{SSTC}(\infty)$ is equivalent to $\mathsf{NMTC}(\infty)$.

[9] Tag-based NMC schemes are also related to UCC schemes. In particular, it can be shown that a UCC scheme is also a tag-based NM commitment scheme in which the tag is the identity of the committing party.

See Figure 1. This makes it clear that the two notions, simulation-sound binding and non-malleability, are very closely related.

$$\begin{tikzcd}
\mathsf{SSTC}(0) & \mathsf{SSTC}(1) \arrow[dl] & \cdots \arrow[dl] & \mathsf{SSTC}(\ell-1) & \mathsf{SSTC}(\ell) \arrow[dl] & \cdots \arrow[dl] & \mathsf{SSTC}(\infty) \arrow[d, leftrightarrow] \\
\mathsf{NMTC}(0) \arrow[u] & \mathsf{NMTC}(1) \arrow[u] & \cdots & \mathsf{NMTC}(\ell-1) \arrow[u] & \mathsf{NMTC}(\ell) \arrow[u] & \cdots & \mathsf{NMTC}(\infty)
\end{tikzcd}$$

**Fig. 1.** The relation between $\mathsf{SSTC}$ and $\mathsf{NMTC}$ schemes, with one-sided arrows denoting strict implication and two-sided arrows denoting equivalence

The definition of non-malleable commitments in Damgård and Groth [14] (which they call *reusable* non-malleable commitments) does not quite fit into the equivalence and separation results above. Their definition states that seeing one *or more* commitments does not give another party any advantage in generating one *or more* commitments that can later be opened to values related to the values in the original commitments. However it can be shown that $\mathsf{SSTC}(\infty)$ implies a reusable NMC scheme. As mentioned above, one can characterize their construction of a reusable NMC scheme as constructing a trapdoor commitment schemes that satisfies a slightly weaker notion of simulation-sound binding, and showing that this implies a reusable NMC scheme.

Due to space limitations, some proofs to our theorems are omitted, and can be found in the full version [31].

## 2 Preliminaries and Definitions

We will use signature schemes that are existentially unforgeable against adaptive chosen-message attacks [27]. However, some of these may only be used for a single signature, and for these, more efficient *one-time* signature scheme constructions may be used [19].

A commitment scheme is a two-phase protocol[10] between a sender and a receiver, both probabilistic polynomial-time Turing machines, that runs as follows. In the commitment phase, the sender commits to a value $v$ by computing a pair $(\mathsf{com}, \mathsf{dec})$ and sending $\mathsf{com}$ to the receiver, and in the decommitment phase, the sender reveals $(v, \mathsf{dec})$ to the receiver, who checks whether the pair is valid.

Informally, a commitment scheme satisfies the hiding property, meaning that for any $v_1 \neq v_2$ of the same length, a commitment to $v_1$ is indistinguishable from a commitment to $v_2$, and the binding property, meaning that once the receiver receives $\mathsf{com}$, the sender cannot open $\mathsf{com}$ to two different values, except with negligible probability.

[10] We define a standard non-interactive commitment scheme. We do not consider relaxations to interactive commitment schemes.

We will always assume that commitments are labeled with a tag. While this is not a factor in the security of basic commitment schemes, it will be useful in defining certain enhanced commitment schemes, as will be obvious below. We also assume that there is a commitment generator function that generates a set of parameters for the commitment scheme. In other papers this is often referred to as a *trusted third party* or as the *common reference string* generation,[11] and it is especially important when we define trapdoor commitment schemes below. (We include it in the basic definition to more conveniently define trapdoor commitment schemes.)

Formally, we define a commitment scheme as follows.

**Definition 1. [Commitment Scheme]** $\mathsf{CS} = (\mathsf{Cgen}, \mathsf{Ccom}, \mathsf{Cver})$ *is a* commitment scheme *if* $\mathsf{Cgen}$*,* $\mathsf{Ccom}$*, and* $\mathsf{Cver}$ *are probabilistic polynomial-time algorithms such that*

- **Completeness** *For all $v$ and tag,*

$$\Pr[pk \leftarrow \mathsf{Cgen}(1^k); (\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{Ccom}(pk, v, tag) : \\ \mathsf{Cver}(pk, \mathsf{com}, v, tag, \mathsf{dec}) = 1] = 1.$$

- **Binding** *There is a negligible function $\alpha(k)$ such that for all non-uniform probabilistic polynomial-time adversaries $\mathcal{A}$,*

$$\Pr[pk \leftarrow \mathsf{Cgen}(1^k); (\mathsf{com}, tag, v_1, v_2, \mathsf{dec}_1, \mathsf{dec}_2) \leftarrow \mathcal{A}(pk) : \\ (\mathsf{Cver}(pk, \mathsf{com}, v_1, tag, \mathsf{dec}_1) = \mathsf{Cver}(pk, \mathsf{com}, v_2, tag, \mathsf{dec}_2) = 1) \\ \wedge (v_1 \neq v_2)] \leq_{\mathrm{ev}} \alpha(k).$$

- **Hiding** *For all $pk$ generated with non-zero probability by $\mathsf{Cgen}(1^k)$, for all $v_1, v_2$ of equal length, and for all tag, the following probability distributions are computationally indistinguishable:*

$$\{(\mathsf{com}_1, \mathsf{dec}_1) \leftarrow \mathsf{Ccom}(pk, v_1, tag) : \mathsf{com}_1\} \text{ and} \\ \{(\mathsf{com}_2, \mathsf{dec}_2) \leftarrow \mathsf{Ccom}(pk, v_2, tag) : \mathsf{com}_2\}.$$

Next, we define trapdoor commitment schemes. (We borrow some notation from Reyzin [35].) Informally a trapdoor commitment scheme has the property that there exists a trapdoor that would allow one to generate a "fake" commitment along with information that would later allow to decommit to any subsequently given value $v$, and that this commitment/decommitment pair is indistinguishable from an actual commitment to $v$ and a subsequent decommitment

[11] We do not use the term "common reference string" in our definition, since these parameters may be generated in a number of ways, and in particular, they may be generated by the receiver. In protocols where this value actually comes from a common reference string, we will make this clear.

**Definition 2.** **[Trapdoor Commitment Scheme]**
$\mathsf{TC} = (\mathsf{TCgen}, \mathsf{TCcom}, \mathsf{TCver}, \mathsf{TCfakeCom}, \mathsf{TCfakeDecom})$ *is a* trapdoor commitment scheme *if* $\mathsf{TCgen}(1^k)$ *outputs a public/secret key pair* $(pk, sk)$*,* $\mathsf{TCgen}_{pk}$ *is a function that restricts the output of* $\mathsf{TCgen}$ *to the public key,* $(\mathsf{TCgen}_{pk}, \mathsf{TCcom}, \mathsf{TCver})$ *is a commitment scheme and* $\mathsf{TCfakeCom}$ *and* $\mathsf{TCfakeDecom}$ *are probabilistic polynomial-time algorithms such that*

- **Trapdoor Property** *For all identifiers tag and values v, the following probability distributions are computationally indistinguishable:*

$$\{(pk, sk) \leftarrow \mathsf{TCgen}(1^k); (\widetilde{\mathsf{com}}, \xi) \leftarrow \mathsf{TCfakeCom}(pk, sk, tag); \widetilde{\mathsf{dec}} \leftarrow \mathsf{TCfakeDecom}(\xi, v) : (pk, tag, v, \widetilde{\mathsf{com}}, \widetilde{\mathsf{dec}})\}$$

*and*

$$\{(pk, sk) \leftarrow \mathsf{TCgen}(1^k); (\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{TCcom}(pk, v, tag) : (pk, tag, v, \mathsf{com}, \mathsf{dec})\}.$$

## 3 Simulation-Sound Trapdoor Commitments

In [24], simulation-sound trapdoor commitment (SSTC) schemes were introduced, in order to construct a universally-composable zero-knowledge (UCZK) protocol secure against adaptive corruptions. Intuitively, they defined an SSTC scheme as a trapdoor commitment scheme with a *simulation-sound binding* property that guarantees that a commitment made by the adversary using *tag* is binding, even if the adversary has seen any commitment using *tag* opened (using a simulator that knows a trapdoor) once to any arbitrary value, and moreover, any commitment using $tag' \neq tag$ opened (again using the simulator) an unbounded number of times to any arbitrary values.

Here we introduce a new definition for an SSTC scheme where the simulation-sound binding property only guarantees that a commitment made by the adversary using *tag* is binding, if the adversary has *never* seen the simulator open a commitment using *tag* (i.e., not even once, as is allowed in the previous definition).[12] Obviously this is a weaker property. However, we will show that it also suffices for the desired application in [24], namely, for constructing UCZK protocols secure against adaptive adversaries.

**Definition 3.** [SSTC **Scheme**]
$\mathsf{TC} = (\mathsf{TCgen}, \mathsf{TCcom}, \mathsf{TCver}, \mathsf{TCfakeCom}, \mathsf{TCfakeDecom})$ *is an* SSTC scheme *if* $\mathsf{TC}$ *is a trapdoor commitment scheme such that*

- **Simulation-Sound Binding** *There is a negligible function* $\alpha(k)$ *such that for all non-uniform probabilistic polynomial-time adversaries* $\mathcal{A}$*,*

[12] Note that in addition to the simulation-sound binding property being modified, our definition of the underlying trapdoor commitment scheme is slightly different than the one given in [24].

$$\begin{aligned}\Pr[&(pk, sk) \leftarrow \mathsf{TCgen}(1^k); (\mathsf{com}, tag, v_1, v_2, \mathsf{dec}_1, \mathsf{dec}_2) \leftarrow \mathcal{A}^{\mathcal{O}_{pk,sk}}(pk) :\\ &\quad (\mathsf{TCver}(pk, \mathsf{com}, v_1, tag, \mathsf{dec}_1) = \mathsf{TCver}(pk, \mathsf{com}, v_2, tag, \mathsf{dec}_2) = 1)\\ &\quad \wedge (v_1 \neq v_2) \wedge tag \notin Q]\\ \leq_{\text{ev}}\ &\alpha(k),\end{aligned}$$

*where $\mathcal{O}_{pk,sk}$ operates as follows, with $Q$ initially set to $\emptyset$:*

- *On input* $(\mathsf{commit}, tag)$*:*
  *compute* $(\widetilde{\mathsf{com}}, \xi) \leftarrow \mathsf{TCfakeCom}(pk, sk, tag)$*, store* $(\widetilde{\mathsf{com}}, tag, \xi)$*, and add* $tag$ *to* $Q$*. Return* $\widetilde{\mathsf{com}}$*.*
- *On input* $(\mathsf{decommit}, \widetilde{\mathsf{com}}, v)$*:*
  *if for some tag and some* $\xi$*, a tuple* $(\widetilde{\mathsf{com}}, tag, \xi)$ *is stored, compute* $\widetilde{\mathsf{dec}} \leftarrow \mathsf{TCfakeDecom}(\xi, v)$*. Return* $\widetilde{\mathsf{dec}}$*.*

For the remainder of the paper, SSTC will refer to this new definition, and SSTC(GMY) will refer to the old definition of [24].

Now we construct SSTC schemes based on specific cryptographic assumptions, and sketch the proofs showing that they achieve simulation-sound binding.

*SSTC scheme based on any one-way function* Here we present an efficient SSTC scheme $\mathsf{TC}$ based on a signature scheme, which in turn may be based on any one-way function [36]. $\mathsf{TC}$ is the $\mathsf{aHC}$ scheme from Canetti *et al.* [7] with the following changes:

1. The underlying commitment scheme based on one-way permutations is replaced by the commitment scheme of Naor [32] based on pseudorandom generators (which can be built from any one-way function).

2. An extra parameter $tag$ is included, and the one-way function $f$ and corresponding NP language $\{y | \exists\, x \text{ s.t. } y = f(x)\}$ used in the underlying non-interactive Feige-Shamir trapdoor commitment [21] is replaced by the signature verification relation $\{((\mathsf{sig\_vk}, tag), \sigma) | 1 = \mathsf{sig\_verify}(\mathsf{sig\_vk}, tag, \sigma)\}$.

We omit the detailed description and proof of the the simulation-soundness of the scheme in this extended abstract.

*SSTC scheme based on DSA* Here we present an efficient SSTC scheme $\mathsf{TC}$ based on DSA. It is a simplified version of the DSA-based SSTC(GMY) scheme from [24]. $\mathsf{TCgen}(1^k)$ generates a DSA public/private key pair $(pk, sk)$, where $pk = (g, p, q, y)$ and $sk = (g, p, q, x)$. For a message $m \in \mathbb{Z}_q$, $\mathsf{TCcom}((g, p, q, y), m, tag)$ first computes $\alpha \xleftarrow{R} \mathbb{Z}_q$, $g' \leftarrow g^\alpha \bmod p$, and $h \leftarrow g^{H(tag)} y^{g'} \bmod p$. (Note that if $s$ is the discrete log of $h$ over $g'$, then $(g' \bmod q, s)$ is a DSA signature for $tag$.) Then it generates a Pedersen commitment [34] to $m$ over bases $(g', h)$, i.e., it generates $\beta \xleftarrow{R} \mathbb{Z}_q$ and computes the commitment/decommitment pair $((g', c), \beta)$, where $c \leftarrow (g')^\beta h^m$. $\mathsf{TCver}((g, p, q, y), (g', c), m, tag, \beta)$ verifies that $c \equiv (g')^\beta h^m$, where $h \equiv g^{H(tag)} y^{g'} \bmod p$. $\mathsf{TCfakeCom}((g, p, q, y), (g, p, q, x), tag')$ computes a DSA signature $(g'', s)$ on $tag'$ using the secret key (g,p,q,x), computes the values $g' \leftarrow (g^{H(tag')} y^{g''})^{s^{-1}} \bmod p$ and $h \leftarrow (g')^s \bmod p$, generates $\beta' \xleftarrow{R} \mathbb{Z}_q$, and sets $c \leftarrow h^{\beta'} \bmod p$. It outputs commitment $(g', c)$ and auxiliary information $(q, \beta', s)$. Then $\mathsf{TCfakeDecom}((q, \beta', s), m)$ outputs $(m, (\beta' - m)s \bmod q)$, which is a decommitment to $m$.

To show the simulation-sound binding property, we show that if an adversary can break this property, we can break DSA as follows. (We assume that DSA is existentially unforgeable against an adaptive chosen-message attack.) Take a DSA key $vk_0$ and its corresponding DSA signature oracle (from the definition of existential unforgeability against an adaptive chosen-message attack). It is easy to see that the equivocation oracle, and in particular the $\mathsf{commit}$ queries to that oracle, may be implemented using the DSA signature oracle on the requested $tag$'s.

Now say the adversary gives a double opening with $tag$, for which no commitment was requested, and thus no call to the DSA signature oracle was made. In particular, say it gives openings $(m, \beta)$ and $(m', \beta')$ of $(g', c)$. Then $(g' \bmod q, (\beta' - \beta)/(m - m') \bmod q)$ is a signature on $tag$, breaking DSA.

*SSTC scheme based on Cramer-Shoup signatures* Here we present an efficient SSTC scheme $\mathsf{TC}$ based on Cramer-Shoup signatures [11] and as secure as strong RSA. (We note that the more efficient version of the Cramer-Shoup signature scheme in Fischlin [22] could be used here as well to obtain an even more efficient SSTC scheme.) $\mathsf{TCgen}(1^k)$ generates a public/private key pair $(pk, sk)$ for Cramer-Shoup signatures, where $pk = (N, h, x, e', H)$ and $sk = (p, q)$. For a message $m \in \{0,1\}^k$, $\mathsf{TCcom}((N, h, x, e', H), m, tag)$ first computes $(y', x', e)$ as in the Cramer-Shoup signature protocol for $tag$, and sets $x'' \leftarrow xh^{H(x')} \bmod N$. (Note that if $y$ is $e$th root of $x''$ modulo $N$, then $\langle e, y, y' \rangle$ is a Cramer-Shoup signature for $tag$.) Then it uses the unconditionally-hiding commitment scheme from [9] based on $e$-one-way homomorphisms (specifically, based on the RSA encryption function with public key $(e, N)$, i.e., $f(a) : a^e \bmod N$) over base $x''$ to commit to $m$. That is, it chooses $\beta \overset{R}{\leftarrow} \mathbb{Z}_N^*$ and computes the commitment/decommitment pair $((y', e, c), \beta)$, where $c \leftarrow (x'')^m \beta^e \bmod N$. $\mathsf{TCver}((N, h, x, e', H), (y', e, c), m, tag, \beta)$ verifies that $e$ is an odd $k+1$-bit integer different from $e'$, $c \equiv (x'')^m \beta^e \bmod N$, and $x'' \equiv xh^{H(x')} \bmod N$, where $x'$ is computed from $y'$ and $e$ as in the Cramer-Shoup signature protocol.

$\mathsf{TCfakeCom}((N, h, x, e', H), (p, q), tag')$ first computes a signature $\langle e, y, y' \rangle$ on $tag'$ using the secret key. Then it computes $x' \leftarrow (y')^{e'} h^{-H(tag')} \bmod N$ and $x'' \leftarrow xh^{H(x')} \bmod N$, generates $\beta' \overset{R}{\leftarrow} \mathbb{Z}_N^*$, and sets $c \leftarrow (\beta')^e \bmod N$. It outputs commitment $(y', e, c)$ and auxiliary information $(N, \beta', y)$. Finally, the function $\mathsf{TCfakeDecom}((N, \beta', y), m)$ outputs $(m, \beta' y^{-m} \bmod N)$, which is a decommitment to $m$.

To show the simulation-sound binding property, we show that if an adversary can break this property, we can break the Strong RSA assumption. The proof basically follows the proof of security (i.e., existential unforgeability against adaptive chosen-message attack) of the Cramer-Shoup signature scheme from [11], which for brevity we will call the *CSSig proof.* As in the CSSig proof, we divide adversaries into Types I, II, and III. For each type, we respond to $\mathsf{commit}$

queries to the equivocation oracle using signatures as computed in the responses to the corresponding signature queries in the CSSig proof. Finally, instead of the adversary producing a forged signature, the adversary gives a double opening of a commitment with some *tag* for which no commit query was made (and thus for which no corresponding signature query was necessary). In particular, say the adversary gives openings $(m, \beta)$ and $(m', \beta')$ of $(y', e, c)$ with $m > m'$. Then $(x'')^{m-m'} \equiv (\beta' \beta^{-1})^e \bmod N$. In the case of Type I and Type II adversaries, i.e., when $e$ is produced in response to a commit query, $e$ is prime and $e > m - m'$. Therefore the value $y$ such that $y^e \equiv x \bmod N$ may be computed (e.g., using the Extended Euclidean Algorithm) and $\langle e, y, y' \rangle$ is a signature on *tag*. Then as in the corresponding cases in the CSSig proof, this can be shown to break the standard RSA assumption. In the case of a Type III adversary, $e$ is not necessarily prime, so we may not necessarily obtain a signature on *tag*. However, the CSSig proof simply uses the fact that $x'' \equiv (\beta' \beta^{-1})^e \bmod N$ to show that Strong RSA can be broken, and the equation $(x'')^{m-m'} \equiv (\beta' \beta^{-1})^e \bmod N$ that we obtain can be used in a similar way to show that Strong RSA can be broken. We omit the details.

## 4 Application to ZK Proofs

We show how an SSTC scheme can be used to construct unbounded simulation-sound ZK protocols, unbounded non-malleable ZK protocols, and universally composable ZK protocols. Our constructions are conceptually simpler than those given by Garay *et al.* [24].

All our results will be in the *common reference string* (CRS) model, which assumes that there is a string uniformly generated from some distribution and is available to all parties at the start of a protocol. Note that this is a generalization of the *public random string* model, where a uniform distribution over fixed-length bit strings is assumed.

### 4.1 Unbounded Simulation Sound and Non-malleable ZK

Intuitively, a ZK protocol is unbounded simulation sound if an adversary cannot convince the verifier of a false statement with non-negligible probability, even after interacting with an arbitrary number of (simulated) provers. We refer the readers to [24] for a formal definition.

Our construction starts with a class of three-round, public-coin, honest-verifier zero-knowledge protocols, also known as $\Sigma$-protocols [10].

Consider a binary relation $R(x, w)$ that is computable in polynomial time. A $\Sigma$-protocol $\Pi$ for the relation $R$ proves membership of $x$ in the language $L_R = \{x | \exists w, s.t. \ R(x, w) = 1\}$. For a given $x$, let $(a, c, z)$ denote the conversation between the prover and the verifier. To compute the first and the final messages, the prover invokes efficient algorithms $a_\Pi(x, w, r)$ and $z_\Pi(x, w, r, c)$, respectively, where $w$ is the witness, $r$ is the random bits, and $c$ is the challenge from the verifier (as the second message). Using an efficient predicate $\phi(x, a, c, z)$, the verifier decides whether the conversation is accepting with respect to $x$. The relation $R$, and the algorithms $a(\cdot)$, $z(\cdot)$ and $\phi(\cdot)$, are public.

We assume the protocol $\Pi$ has a simulator $\mathcal{S}_\Pi$ that, taking the challenge as input, generates an accepting conversation. More precisely, $(a, c, z) \leftarrow \mathcal{S}_\Pi(c)$, where that the distribution of $(a, c, z)$ is computationally indistinguishable from the real conversation.

The protocol $\mathsf{USS}^R_{[pk]}(x)$ is shown in Figure 2, and uses an SSTC scheme $\mathsf{TC}$. Say $\Pi$ is a $\Sigma$-protocol for relation $R$. The prover generates a pair $(\mathsf{sig\_vk}, \mathsf{sig\_sk})$ for a strong one-time signature scheme and sends $\mathsf{sig\_vk}$ to the verifier. Then the prover generates the first message $a$ of $\Pi$ and sends its commitment $\mathsf{com}$ to the verifier, using the signature verification key $\mathsf{sig\_vk}$ as the commitment *tag*. After receiving the challenge $c$, the prover generates and sends the third message $z$ of $\Pi$, opens the commitment $\mathsf{com}$, signs the entire transcript using the signing key $\mathsf{sig\_sk}$, and sends the signature on the transcript to the verifier. (To be specific, the *transcript* consists of all values sent or received by the prover in the protocol, except the final signature.)

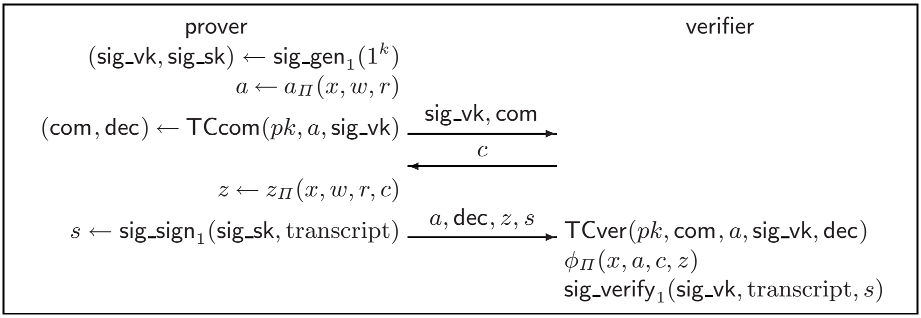
prover
verifier
$(\mathsf{sig\_vk}, \mathsf{sig\_sk}) \leftarrow \mathsf{sig\_gen}_1(1^k)$
$a \leftarrow a_\Pi(x, w, r)$
$(\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{TCcom}(pk, a, \mathsf{sig\_vk})$ $\xrightarrow{\mathsf{sig\_vk}, \mathsf{com}}$
$\xleftarrow{c}$
$z \leftarrow z_\Pi(x, w, r, c)$
$s \leftarrow \mathsf{sig\_sign}_1(\mathsf{sig\_sk}, \text{transcript})$ $\xrightarrow{a, \mathsf{dec}, z, s}$ $\mathsf{TCver}(pk, \mathsf{com}, a, \mathsf{sig\_vk}, \mathsf{dec})$
$\phi_\Pi(x, a, c, z)$
$\mathsf{sig\_verify}_1(\mathsf{sig\_vk}, \text{transcript}, s)$

**Fig. 2.** $\mathsf{USS}^R_{[pk]}(x)$: An unbounded simulation-sound ZK protocol for relationship $R$ with common input $x$ and common reference string $pk$, where $pk$ is drawn from the distribution $\mathsf{TCgen}(1^k)$. The prover also knows the witness $w$ such that $R(x, w) = 1$.

**Theorem 1.** *The protocol* $\mathsf{USS}^R_{[pk]}(x)$ *is a USSZK argument.*

Intuitively, a ZK protocol is unbounded non-malleable if an efficient witness extractor successfully extracts a witness from any adversary that causes the verifier to accept, even when the adversary is also allowed to interact with any number of (simulated) provers. Again, we refer the readers to [24] for a formal definition.

Our construction of the NMZK protocol is very similar to that of the USSZK protocol presented above, where the only difference is that the $\Sigma$-protocol is replaced by an $\Omega$-protocol. Recall that an $\Omega$-protocol [24] is like a $\Sigma$-protocol with the additional property that it admits a polynomial-time, straight-line extractor (an $\Omega$-protocol works in the CRS model).

The protocol $\mathsf{NM}^{R}_{[pk,\sigma]}(x)$ is very similar to the protocol in Figure 2, but note that here we assume that $\Pi$ is an $\Omega$-protocol with $\sigma$ being the CRS.

**Theorem 2.** *The protocol* $\mathsf{NM}^{R}_{[pk,\sigma]}(x)$ *is an NMZK argument of knowledge for the relation* $R$*.*

### 4.2 Universally Composable ZK

The universal composability paradigm was proposed by Canetti [5] for defining the security and composition of protocols. To define security one first specifies an *ideal functionality* using a trusted party that describes the desired behavior of the protocol. Then one proves that a particular protocol operating in a real-life model securely realizes this ideal functionality, as defined below. Here we briefly summarize the framework.

A (real-life) protocol $\pi$ is defined as a set of $n$ interactive Turing Machines $P_1, \ldots, P_n$, designating the $n$ parties in the protocol. It operates in the presence of an environment $\mathcal{Z}$ and an adversary $\mathcal{A}$, both of which are also modeled as interactive Turing Machines. The environment $\mathcal{Z}$ provides inputs and receives outputs from honest parties, and may communicate with $\mathcal{A}$. $\mathcal{A}$ controls (and may view) all communication between the parties. (Note that this models asynchronous communication on open point-to-point channels.) We will assume that messages are authenticated, and thus $\mathcal{A}$ may not insert or modify messages between honest parties.[13] $\mathcal{A}$ also may corrupt parties, in which case it obtains the internal state of the party. (In the non-erasing model, the internal state would encompass the complete internal history of the party.)

The ideal process with respect to a functionality $\mathcal{F}$, is defined for $n$ parties $P_1, \ldots, P_n$, an environment $\mathcal{Z}$, and an (ideal-process) adversary $\mathcal{S}$. However, $P_1, \ldots, P_n$ are now dummy parties that simply forward (over secure channels) inputs received from $\mathcal{Z}$ to $\mathcal{F}$, and forward (again over secure channels) outputs received from $\mathcal{F}$ to $\mathcal{Z}$. Thus the ideal process is a trivially secure protocol with the input-output behavior of $\mathcal{F}$.

*The zero-knowledge functionality.* The (multi-session) ZK functionality as defined by Canetti [5] is given in Figure 3. In the functionality, parameterized by a relation $R$, the prover sends to the functionality the input $x$ together with a witness $w$. If $R(x, w)$ holds, then the functionality forwards $x$ to the verifier. As pointed out in [5], this is actually a proof of knowledge in that the verifier is assured that the prover actually knows $w$.

Garay *et al.* [24] proved that any "augmentable" NMZK protocol can be easily converted to a UCZK protocol in the $\mathcal{F}^{\mathcal{D}}_{\mathrm{CRS}}$-hybrid model, assuming static corruptions. Intuitively, an NMZK protocol is augmentable if the first message sent by the prover contains the common input $x$ and a special field $\mathsf{aux}$ in which the prover can fill with an arbitrary string without compromising security. (In

[13] This feature could be added to an unauthenticated model using a message authentication functionality as described in [5].

$\hat{\mathcal{F}}^R_{\mathrm{ZK}}$ proceeds as follows, running parties $P_1, \ldots, P_n$, and an adversary $\mathcal{S}$:

- Upon receiving $(\mathsf{zk\text{-}prover}, sid, ssid, P_i, P_j, x, w)$ from $P_i$: If $R(x, w)$ then send $(\mathsf{ZK\text{-}PROOF}, sid, ssid, P_i, P_j, x)$ to $P_j$ and $\mathcal{S}$. Otherwise, ignore.

**Fig. 3.** The (multi-session) zero-knowledge functionality (for relation $R$)

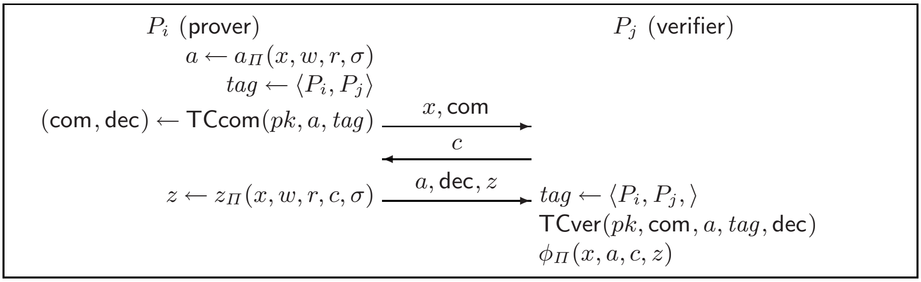
$P_i$ (prover)
$P_j$ (verifier)
$a \leftarrow a_\Pi(x, w, r, \sigma)$
$tag \leftarrow \langle P_i, P_j \rangle$
$(\mathsf{com}, \mathsf{dec}) \leftarrow \mathsf{TCcom}(pk, a, tag)$
$x, \mathsf{com}$
$c$
$z \leftarrow z_\Pi(x, w, r, c, \sigma)$
$a, \mathsf{dec}, z$
$tag \leftarrow \langle P_i, P_j, \rangle$
$\mathsf{TCver}(pk, \mathsf{com}, a, tag, \mathsf{dec})$
$\phi_\Pi(x, a, c, z)$

**Fig. 4.** $\mathsf{MYZK}^R_{[pk,\sigma]}(x)$: A UCZK protocol for relationship $R$ with common reference string $(pk, \sigma)$ where $pk$ is drawn from the distribution $\mathsf{TCgen}(1^k)$ and $\sigma$ is drawn from the distribution of the CRS for protocol $\Pi$.

the conversion to UCZK in [24], the auxiliary string contains the *sid*, the *ssid*, and the identities of the prover and verifier.)

It can be readily verified that the protocol $\mathsf{NM}^R_{[pk,\sigma]}(x)$ can be easily made augmentable by adding $x$ and $\mathsf{aux}$ in the first message. We denote the slightly modified protocol where the $\mathsf{aux}$ field is set to $(sid, ssid, P_i, P_j)$ by $\mathsf{ANM}^R_{[pk,\sigma]}(x)$. Then it follows that $\mathsf{ANM}^R_{[pk,\sigma]}(x)$ is a UCZK protocol for relation $R$, assuming static corruptions.

However, one can simplify this protocol by removing the one-time signature scheme, only including the identities of the prover and verifier in the auxiliary string, and using this auxiliary string as the tag of the commitment scheme. This simplified scheme, $\mathsf{MYZK}^R_{[pk,\sigma]}(x)$, is shown in Figure 4. (Note that since we are assuming authenticated communication in the UC framework, the identities $P_i$ and $P_j$ will be known to both parties, and thus do not need to be explicitly sent in our protocol.) Furthermore, this protocol can be easily modified into one that remains secure against adaptive corruption in the erasing model. In fact, all that is needed is to have the prover erase the randomness used in the $\Omega$-protocol before sending the final message.

**Theorem 3.** *The protocol* $\mathsf{MYZK}^R_{[pk,\sigma]}(x)$ *is a UCZK protocol for relation R, assuming static corruptions. By erasing the randomness (r) used in the Ω-protocol before the final message, it is a UCZK protocol for relation R, assuming adaptive corruption (in the erasing model).*

## 5 Comparison to Non-malleable Commitments

We explore the exact relation between SSTC schemes and NMC schemes.

Our definition for non-malleable (NM) commitments is based on the definition in [17], which, technically speaking, defines the notion of $\epsilon$-non-malleability, instead of strict non-malleability. For the clarity of presentation, we shall use the term "non-malleability" to mean $\epsilon$-non-malleability, and will note any places where our results have application to strict non-malleability.

Informally, similar to the definition in [17], we say a commitment scheme is non-malleable if when an adversary sees a commitment $\mathsf{com}_1$, generates its own commitment $\mathsf{com}_2$, and sees $\mathsf{com}_1$ opened, it cannot then open $\mathsf{com}_2$ to a value related to $\mathsf{com}_1$ with any greater probability than a simulator that never saw $\mathsf{com}_1$ in the first place.[14] Note that this is also called *non-malleability with respect to opening* [16] and differs from the original definition of [18] that was discussed in the introduction, and which is also called *non-malleability with respect to commitment.* Our definition differs from the definition in [17] as follows.

- We only define NM *trapdoor* commitment (NMTC) schemes, since that is what will be of most interest in comparisons to SSTC schemes. Non-trapdoor versions of these definitions are straightforward.
- We use tag-based definitions instead of body-based definitions. Again this is what will be of most interest in comparisons to SSTC schemes. Body-based definitions are straightforward. In fact, most of our results relating SSTC schemes and $\mathsf{NMTC}$ schemes also hold when these schemes are defined using body-based definitions. We will discuss this later.

Due to space limitations, we omit the formal definition of an NMTC scheme. It may be obtained in a straightforward manner from the formal definition in [17] and the changes described above.

As mentioned in the introduction, the recent work of Damgård and Groth [14] generalizes and strengthens the definition of non-malleable commitments to be reusable, i.e., to have the property that seeing one *or more* commitments does not give another party any advantage in generating one *or more* commitments that can later be opened to values related to the values in the original commitments. Their definition also stipulates that the distribution of committed messages is dependent on the public key. However, we will continue to use the simpler definition, since it exemplifies the relation between SSTC schemes and NMTC schemes. Later we will discuss how to obtain similar relations to reusable NMTC schemes.

Note that we can generalize the definition of $\mathsf{NMTC}$ to $\mathsf{NMTC}(\ell)$ schemes, which are $\mathsf{NMTC}$ schemes in which the adversary is allowed to query an oracle $\mathcal{O}_{pk,sk}$ as defined in the SSTC definition, but with at most $\ell$ $\mathsf{commit}$ queries allowed, and with the restriction that the commitment produced by the adversary has a tag that is not used in any of the $\mathsf{commit}$ queries. Note that an $\mathsf{NMTC}$

[14] Slightly more formally, we say that it is $\epsilon$-non-malleable if for all $\epsilon$ it cannot do this with probability non-negligibly greater than $\epsilon$.

scheme is an $\mathsf{NMTC}(0)$ scheme. We use $\ell = \infty$ to denote an oracle which accepts an unbounded number of $\mathsf{commit}$ queries.

We similarly generalize the definition of SSTC schemes and consider $\mathsf{SSTC}(\ell)$ schemes. Then an $\mathsf{SSTC}(0)$ scheme is just a TC scheme, and an $\mathsf{SSTC}(\infty)$ scheme is what we have called an SSTC scheme.

As mentioned above, we have defined NMTC schemes as tag-based, as opposed to body-based, as usually seen in literature [18,16,23,17,14]. However, this is not a significant distinction since there exists fairly generic reductions from one to the other. Our next theorem shows such a reduction from body-based NMTC schemes to tag-based ones.

Here, we assume the commitment scheme allows commitments to strings of arbitrary length. A similar theorem could be shown for commitment schemes which allow only fixed length commitments, say of length equal to the security parameter.

**Theorem 4.** *Let* $\mathsf{TC}$ *be a body-based* $\mathsf{NMTC}$ *scheme. Let* $\mathsf{TC}'$ *be* $\mathsf{TC}$*, but with the tag added to the message being committed. That is,* $\mathsf{TCgen}'(1^k)$ *returns the result of* $\mathsf{TCgen}(1^k)$*,* $\mathsf{TCcom}'(pk, v, tag)$ *returns the result of* $\mathsf{TCcom}(pk, \langle v, tag\rangle, tag)$*, and* $\mathsf{TCver}'(pk, \mathsf{com}, v, tag, \mathsf{dec})$ *returns the result of* $\mathsf{TCver}(pk, \mathsf{com}, \langle v, tag\rangle, tag, \mathsf{dec})$*. Then* $\mathsf{TC}'$ *is a tag-based* $\mathsf{NMTC}$ *scheme.*

Considering the problem of converting tag-based $\mathsf{SSTC}$ or $\mathsf{NMTC}$ schemes to body-based $\mathsf{SSTC}$ or $\mathsf{NMTC}$ schemes, it seems that a simple construction like the one in Theorem 4 does not suffice. Instead, one could construct a body-based scheme by generating a verification/signing key pair for a strong one-time signature scheme, using the verification key as the tag in the tag-based commitment, signing the tag-based commitment using the signing key, and giving the pair (the tag-based commitment and the associated signature) as the full commitment. As this is a fairly standard technique, used in, e.g. [24], we omit the analysis here.

### 5.1 Relations between SSTC and NMTC

First we show that for all $\ell \geq 0$, an $\mathsf{SSTC}(\ell + 1)$ scheme is also an $\mathsf{NMTC}(\ell)$ scheme, and an $\mathsf{NMTC}(\ell)$ scheme is also an $\mathsf{SSTC}(\ell + 1)$ scheme.

**Theorem 5.** *Let* $\mathsf{TC}$ *be an* $\mathsf{SSTC}(\ell + 1)$ *scheme. Then* $\mathsf{TC}$ *is an* $\mathsf{NMTC}(\ell)$ *scheme.*

**Theorem 6.** *Let* $\mathsf{TC}$ *be an* $\mathsf{NMTC}(\ell)$ *scheme. Then* $\mathsf{TC}$ *is an* $\mathsf{SSTC}(\ell)$ *scheme.*

To relate our results to reusable non-malleable commitment schemes as defined in [14], we need to consider adversaries that input a vector of commitments (and later decommitments), and output a vector of commitments (and later decommitments). To be specific, let $(t, u)$-$\mathsf{NMTC}(\ell)$ denote a reusable NMTC commitment scheme with an input vector of size $t$ and an output vector of size $u$.

Then using a proof similar to above, but with some additional ideas from [14], we can prove the following theorem.[15]

**Theorem 7.** *Let* $\mathsf{TC}$ *be an* $\mathsf{SSTC}(\ell + t)$ *scheme. Then* $\mathsf{TC}$ *is a* $(t, u)$-$\mathsf{NMTC}(\ell)$ *scheme.*

Finally, we show the following separation results.

**Theorem 8.** *Assuming the hardness of the discrete logarithm problem, there exists an* $\mathsf{SSTC}(\ell)$ *scheme that is not* $\mathsf{NMTC}(\ell)$*, for every* $\ell \geq 0$.

**Theorem 9.** *If there exists an* $\mathsf{NMTC}(\ell)$ *scheme, then there exists an* $\mathsf{NMTC}(\ell)$ *scheme that is not* $\mathsf{SSTC}(\ell + 1)$.

## References

1. N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology – EUROCRYPT '97* (LNCS 1233), 480–494, 1997.
2. D. Beaver. Adaptive zero-knowledge and computational equivocation. In *28th ACM Symp. on Theory of Computing*, 629–638, 1996.
3. M. Blum. Coin flipping by telephone. In *IEEE Spring COMPCOM*, pp. 133–137, 1982.
4. G. Brassard, D. Chaum, and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *JCSS*, 37(2):156–189, 1988.
5. R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd IEEE Symp. on Foundations of Computer Sci.*, 136–145, 2001.
6. R. Canetti and M. Fischlin. Universally composable commitments. In *Advances in Cryptology – CRYPTO 2001* (LNCS 2139), 19–40, 2001.
7. R. Canetti, Y. Lindell, R. Ostrovsky and A. Sahai. Universally composable two-party computation. In 34th ACM Symp. on Theory of Computing, 494–503, 2002. Full version in *ePrint archive*, Report 2002/140. `http://eprint.iacr.org/`, 2002.
8. S. A. Cook. The complexity of theorem-proving procedures. In *3rd IEEE Symp. on Foundations of Computer Sci.*, 151–158, 1971.
9. R. Cramer and I. Damgård. Zero-Knowledge Proofs for Finite Field Arithmetic, or: Can Zero-Knowledge Be for Free? In *Advances in Cryptology – CRYPTO '98* (LNCS 1462), pages 424–441, 1998.
10. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology – CRYPTO '94* (LNCS 839), pages 174–187, 1994.
11. R. Cramer and V. Shoup. Signature scheme based on the strong RSA assumption. In *ACM Trans. on Information and System Security* 3(3):161-185, 2000.
12. I. Damgård. On the existence of bit commitment schemes and zero-knowledge proofs. In *Advances in Cryptology – CRYPTO '89* (LNCS 435), 17–29, 1989.

[15] As in [14], we change the definition of a valid relation (over vectors of messages) to one in which all messages including $\perp$ are allowed, but where the probability of the relation being true cannot be increased by changing a message in the second (adversarially-chosen) vector to $\perp$.

13. I. Damgård. Efficient Concurrent Zero-Knowledge in the Auxiliary String Model. In *Advances in Cryptology – EUROCRYPT 2000* (LNCS 1807), 418–430, 2000.
14. I. Damgård and J. Groth. Non-interactive and reusable non-malleable commitment schemes. In *35th ACM Symp. on Theory of Computing*, 426–437, 2003.
15. I. Damgård and J. Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In *Advances in Cryptology – CRYPTO 2002* (LNCS 2442), 581–596, 2002. Full version in *ePrint Archive*, report 2001/091. `http://eprint.iacr.org/`, 2001.
16. G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Non-interactive and non-malleable commitment. In *30th ACM Symp. on Theory of Computing*, 141–150, 1998.
17. G. Di Crescenzo, J. Katz, R. Ostrovsky, and A. Smith. Efficient and Non-Interactive Non-Malleable Commitment. In *Advances in Cryptology – EUROCRYPT 2001* (LNCS 2045), 40–59, 2001.
18. D. Dolev, C. Dwork and M. Naor. Non-malleable cryptography. *SIAM J. on Comput.*, 30(2):391–437, 2000. Also in *23rd ACM Symp. on Theory of Computing*, 542–552, 1991.
19. S. Even, O. Goldreich, and S. Micali. On-line/Off-line digital signatures. *J. Cryptology* 9(1):35-67 (1996).
20. U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *22nd ACM Symp. on Theory of Computing*, 416–426, 1990.
21. U. Feige and A. Shamir. Zero-Knowledge Proofs of Knowledge in Two Rounds. In *Advances in Cryptology – CRYPTO '89* (LNCS 435), 526–544, 1989.
22. M. Fischlin. The Cramer-Shoup strong-RSA signature scheme revisited. In *Public Key Cryptography – PKC 2003* (LNCS 2567), 116–129, 2003.
23. M. Fischlin and R. Fischlin. Efficient non-malleable commitment schemes. In *Advances in Cryptology – CRYPTO 2000* (LNCS 1880), 413–431, 2000.
24. J. A. Garay, P. MacKenzie, and K. Yang. Strengthening Zero-Knowledge Protocols using Signatures. In *Advances in Cryptology – EUROCRYPT 2003* (LNCS 2656), 177–194, 2003.
25. R. Gennaro. Improved Proofs of Knowledge Secure under Concurrent Man-in-the-middle Attacks and their Applications. In *ePrint Archive*, report 2003/214. `http://eprint.iacr.org/`, 2003.
26. O. Goldreich, S. Micali and A. Wigderson. Proofs that yield nothing but their validity or All languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.
27. S. Goldwasser, S. Micali and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17:281–308, 1988.
28. S. Jarecki and A. Lysyanskaya. Adaptively Secure Threshold Cryptography: Introducing Concurrency, Removing Erasures. In *Advances in Cryptology – EUROCRYPT 2000* (LNCS 1807), 221–242, 2000.
29. D. W. Kravitz. Digital signature algorithm. U.S. Patent 5,231,668, 27 July 1993.
30. L. A. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9:115–116, 1973. In Russian. Engl. trans.: *Problems of Information Transmission* 9:265–266.
31. P. MacKenzie and K. Yang. On simulation-sound trapdoor commitments (full version). Available on the Cryptology ePrint Archive: `http://eprint.iacr.org/2003/252`.
32. M. Naor. Bit commitment Using Pseudo-Randomness. *J. Cryptology* 4(2):151–158 (1991).
33. M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect zero-knowledge arguments for NP can be based on general complexity assumptions. In *Advances in Cryptology – CRYPTO '92* (LNCS 740), 196–214, 1992.

34. T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology – CRYPTO '91* (LNCS 576), 129–140, 1991.
35. L. Reyzin. Zero-knowledge with public keys. Ph.D. Thesis, MIT, 2001.
36. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd ACM Symp. on Theory of Computing*, 387–394, 1990.
37. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th IEEE Symp. on Foundations of Computer Sci.*, 543–553, 1999.