Computational Soundness of Non-Malleable Commitments

David Galindo[1⋆], Flavio D. Garcia[2], and Peter van Rossum[2]

[1] Departament of Computer Science, University of Malaga, Spain
dgalindo@lcc.uma.es
[2] Institute for Computing and Information Sciences,
Radboud University Nijmegen, The Netherlands.
{flaviog,petervr}@cs.ru.nl

Abstract. This paper aims to find a proper security notion for commitment schemes to give a sound computational interpretation of symbolic commitments. We introduce an indistinguishability based security definition of commitment schemes that is equivalent to non-malleability with respect to commitment. Then, we give a construction using tag-based encryption and one-time signatures that is provably secure assuming the existence of trapdoor permutations. Finally, we apply this new machinery to give a sound interpretation of symbolic commitments in the Dolev-Yao model while considering active adversaries.

1 Introduction

Over the last few decades, two main stream approaches have been developed for the analysis of security protocols. On the one hand, the cryptographic approach considers an arbitrary computationally-bound adversary that interacts with honest participants and tries to break a security goal. This model is satisfactory as it deals with every efficient attacker. On the other hand, the symbolic or Dolev-Yao approach idealizes the security properties of the cryptographic primitives, which are axiomatized in a logic. Moreover, the capabilities of the adversary are also specified by a set of inference rules. This approach is appealing because there are automated techniques for the verification of some security properties.

Abadi and Rogaway in [AR02] pioneered the idea of relating these two models and showed that, under appropriate assumptions on the underlying cryptographic primitives, a simple language of encrypted expressions is sound with respect to the computational model in the case of passive adversaries.

Such a relation maps symbolic messages m to distributions over bitstrings $[\![m]\!]$. This map then should relate messages that are observationally equivalent in the symbolic world to indistinguishable distributions over bitstrings. Such a map allows one to use formal methods, possibly even automated, to reason about security properties of protocols and have those reasonings be valid also in the standard computational model.

⋆ partially funded by the Spanish Ministry of Science and Education through the projects ARES (CSD2007-00004) and CRISIS (TIN2006-09242)

Several extensions to the original Abadi-Rogaway logic [AR02] have been proposed in the literature. These extensions deal with public key encryption [MW04, Her05], key cycles [ABHS05], partial information leakage [ABS05], active instead of passive adversaries [MW04, JLM05], and more realistic security notions [AW05]. Other extensions add new primitives to the logic such as bilinear pairings [Maz07], modular exponentiation [BLMW07] and hash functions [CKKW06, GvR06]. There are also frameworks dealing with generic equational theories [BCK05, ABW06, KM07]. So far there is no work in the literature, that we are aware of, that relates these two approaches for commitment schemes.

Commitment schemes are fundamental cryptographic primitives and are used in protocols like zero-knowledge proofs [GMW91], contract signing [EGL85], and can be used for bidding protocols. A commitment consists of two phases: the commitment phase where the principals commit to a message without revealing any information; and the opening phase where the principals reveal the message and it is possible to verify that this message corresponds to the value committed to during the commitment phase. After the commitment phase it should be infeasible to open the commitment to a different value than the one committed. This property is called binding. In the context of bidding protocols, non-malleability is also a desirable property. This means that an adversary cannot modify an intercepted commitment, say into a commitment to a slightly higher bid.

Our contribution. The first objective of this paper is to find sufficient security assumptions to give a sound computational interpretation of commitments schemes in the Dolev-Yao model, under active adversaries. Pursuing that objective we propose a new indistinguishability-based security definition for commitment schemes in the presence of adaptive adversaries. Then we give a novel generic construction for a non-malleable commitment scheme based on one-way trapdoor permutations. This construction is secure with respect to our new definition and has some additional properties such as being non-interactive, perfectly binding and reusable, which makes it of independent interest. This new definition allows us to prove soundness of the Dolev-Yao model extended with commitments, following the directions of Micciancio and Warinschi [MW04].

Overview. Section 3 introduces basic notation and definitions from the literature. Section 4 elaborates on different definitions of non-malleability for commitment schemes and discusses the relations among them. In Section 5 we propose a new commitment scheme and we give a security proof. Section 2 describes symbolic protocol executions, its computational counterparts and the map between them and also states the soundness result. Finally in Section 7 there are some concluding remarks.

2 Symbolic Protocols

We are going to apply this theory to give sound computational interpretation to symbolic commitments. Recall from the introduction that the symbolic approach to protocol verification deals with symbolic or algebraic messages and idealized cryptographic primitives. In this setting the adversary is unbounded in running

time and has full control over the communication media but is completely incapable of breaking the underlying cryptographic primitives.

We now describe the message space and the closure operator. These messages are used to formally describe cryptographic protocols. The closure represents the knowledge that can be extracted from a message, and it is used to define what valid algebraic protocol runs are. Intuitively a protocol run is valid if every message sent by a principal can be deduced from its knowledge except maybe for some fresh randomness. Much of this is standard (see, e.g., [AR02, MW04, MP05, GvR06]), except that we model commitments and decommitments as well as encryption.

Definition 2.1. Let **Nonce** be an infinite set of *nonce symbols*, **Const** a finite set of *constant symbols*, **Key** an infinite set of *key symbols*, and **Random** an infinite set of *randomness labels*. Nonces are denoted by $n, n', \ldots$, constants by $c, c', \ldots$, keys by $k, k', \ldots$, and randomness labels by $r, r', \ldots$. Using these building blocks, *messages* are constructed using symbolic encryption, commitments, decommitments, and pairing operations:

$$\mathbf{Msg} \ni m := c \mid n \mid \{\!|m|\!\}_k^r \mid \mathsf{com}^r(m) \mid \mathsf{dec}^r(m) \mid \langle m, m \rangle.$$

A message of the form $\{\!|m|\!\}_k^r$ is called an *encryption* and the set of all such messages is denoted by **Enc**. Similarly, messages of the form $\mathsf{com}^r(m)$ are called *commitments* and the set of all these messages is denoted by **Com**. The messages of the form $\mathsf{dec}^r(m)$ are called *decommitments* and the set of all these messages is denoted by **Dec**. In a protocol run $\mathsf{dec}^r(m)$ is a valid decommitment of $\mathsf{com}^{r'}(m')$ only if $m = m'$ and $r = r'$. We say that elements in $\mathbf{Const} \cup \mathbf{Nonce} \cup \mathbf{Key}$ are primitive and we denote this set by **Prim**. For a public key k we denote its associated private key as k^{-1}.

The *closure* of a set U of messages is the set of all messages that can be constructed from U using tupling, detupling, commitment, decommitment, and encryption and decryption. It represents the information an adversary could deduce knowing U. Note that, due to secrecy of the commitment scheme, knowing $\mathsf{com}^r(m)$ does not provide an adversary with any information about m.

Definition 2.2 (Closure)**.** Let U be a set of messages. The *closure* of U, denoted by $\overline{U}$, is the smallest set of messages satisfying: 1. $\mathbf{Const} \subseteq \overline{U}$; 2. $U \subseteq \overline{U}$; 3. $m, m' \in \overline{U} \implies \langle m, m' \rangle \in \overline{U}$; 4. $m \in \overline{U} \wedge k \in \overline{U} \implies \{\!|m|\!\}_k^r \in \overline{U}$; 5. $\{\!|m|\!\}_k^r \in \overline{U} \wedge k^{-1} \in \overline{U} \implies m \in \overline{U}$; 6. $m \in \overline{U} \implies \mathsf{com}^r(m), \mathsf{dec}^r(m) \in \overline{U}$; 7. $\mathsf{dec}^r(m) \in \overline{U} \implies m \in \overline{U}$; 8. $\langle m, m' \rangle \in \overline{U} \implies m, m' \in \overline{U}$.

Next we need to find the right security notions to give sound computational interpretation to symbolic encryption and commitments.

3 Computational Setup

This section introduces syntaxis and security definitions for different cryptographic primitives. Much of this is standard, we refer the reader to [GM84,RS92] and [NY90] for a thorough explanation. Some of this primitives will be used to interpret algebraic operations and some of them are used as building blocks for our construction of Section 5.

3.1 Commitment Schemes

Definition 3.1. A *commitment scheme* is a triple $\Omega = (\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ of probabilistic polynomial-time algorithms. TTP, the *trusted third party*, takes as input the security parameter 1^η and produces a common reference string σ. We require that $|\sigma| \geq p(\eta)$ for some non-constant polynomial p. Snd, the *sender*, takes as input σ and a message m and produces a commitment *com* to this message and a corresponding decommitment *dec*. Rcv, the *receiver*, takes as input σ, *com*, and *dec* and produces a message or $\perp$.

$\mathbf{Meaningfulness}_\Omega(A)$:
$\sigma \leftarrow \mathsf{TTP}(1^\eta)$
$m \leftarrow A(\sigma)$
$(com, dec) \leftarrow \mathsf{Snd}(\sigma, m)$
$m_1 \leftarrow \mathsf{Rcv}(\sigma, com, dec)$
return $m \neq m_1$

$\mathbf{Secrecy}_{\mathsf{TTP},\mathsf{Snd}}(A_1, A_2)$:
$\sigma \leftarrow \mathsf{TTP}(1^\eta)$
$m_0, m_1, s \leftarrow A_1(\sigma)$
$b \leftarrow \{0, 1\}$
$(com, dec) \leftarrow \mathsf{Snd}(\sigma, m_b)$
$b' \leftarrow A_2(s, com)$
return $b = b'$

$\mathbf{Binding}_{\mathsf{TTP},\mathsf{Rcv}}(A)$:
$\sigma \leftarrow \mathsf{TTP}(1^\eta)$
$(com, dec_1, dec_2) \leftarrow A(\sigma)$
$m_1 \leftarrow \mathsf{Rcv}(\sigma, com, dec_1)$
$m_2 \leftarrow \mathsf{Rcv}(\sigma, com, dec_2)$
return $m_1 \neq \perp \neq m_2$
$\wedge\ m_1 \neq m_2$

The following three conditions must hold.

1. For all probabilistic polynomial-time algorithms A, the probability $\mathbb{P}[\mathbf{Meaningfullness}_\Omega(A)]$ is a negligible function of η.
2. For all probabilistic polynomial-time algorithms (A_1, A_2), the advantage $|\mathbb{P}[\mathbf{Secrecy}_{\mathsf{TTP},\mathsf{Snd}}(A_1, A_2)] - 1/2|$ is a negligible function of η.
3. For all probabilistic polynomial-time algorithms A, the probability $\mathbb{P}[\mathbf{Binding}_{\mathsf{TTP},\mathsf{Rcv}}(A)]$ is a negligible function of η.

Definition 3.2. A commitment scheme is said to be *perfectly binding* if for all unbounded algorithms A, the probability $\mathbb{P}[\mathbf{Binding}_{\mathsf{TTP},\mathsf{Rcv}}(A)]$ is zero.

Definition 3.3. A commitment scheme is said to be *perfectly hiding* if for all unbounded algorithms (A_0, A_1), $|\mathbb{P}[\mathbf{Secrecy}_{\mathsf{TTP},\mathsf{Snd}}(A_1, A_2)] - 1/2|$ is zero.

3.2 Encryption Schemes

Definition 3.4. An *encryption scheme* is a triple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ of probabilistic polynomial-time algorithms. $\mathcal{K}$ takes as input the security parameter 1^η and produces a key pair $(\mathrm{pk}, \mathrm{sk})$ where pk is the public encryption key and sk is the private decryption key. $\mathcal{E}$ takes as input a public key pk and a plaintext m and outputs a ciphertext. $\mathcal{D}$ takes as input a private key sk and a ciphertext and outputs a plaintext or $\perp$. It is required that $\mathbb{P}[(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{K}(1^\eta); c \leftarrow \mathcal{E}(\mathrm{pk}, m); m' \leftarrow \mathcal{D}(\mathrm{sk}, c) : m = m'] = 1$.

$\mathbf{IND\text{-}CCA}_\Pi(A_0, A_1)$:
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathcal{K}(1^\eta)$
$m_0, m_1, s \leftarrow A_0^{\mathcal{D}}(\mathrm{pk})$
$b \leftarrow \{0, 1\}$
$c \leftarrow \mathcal{E}(\mathrm{pk}, m_b)$
$b' \leftarrow A_1^{\mathcal{D}}(s, c)$
return $b = b'$

Definition 3.5. An encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is said to be *IND-CCA secure* if for all probabilistic polynomial-time adversaries $A = (A_0, A_1)$ the advantage of A, defined as $|\mathbb{P}[\mathbf{IND\text{-}CCA}_{\Pi}(A_0, A_1)] - 1/2|$, is a negligible function of η. This adversary has access to a decryption oracle $\mathcal{D}$ that on input c' outputs $\mathcal{D}(\mathrm{sk}, c')$ with the only restriction that $c \neq c'$.

3.3 One-time Signatures

Definition 3.6. A *signature scheme* is a triple $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ of probabilistic polynomial-time algorithms. Gen takes as input the security parameter 1^η and produces a key pair $(\mathrm{vk}, \mathrm{sk})$ where vk is the signature verification key and sk is the secret signing key. Sign takes as input sk and a message m and produces a signature s of m. Vrfy takes as input vk, a message m and a signature s and outputs whether or not s is a valid signature of m.

$\mathbf{OTS}_{\Sigma}(A_0, A_1)$:
$(\mathrm{vk}, \mathrm{sk}) \leftarrow \mathsf{Gen}(1^\eta)$
$m, s \leftarrow A_0(\mathrm{vk}, 1^\eta)$
$\sigma \leftarrow \mathsf{Sign}(\mathrm{sk}, m)$
$m', \sigma' \leftarrow A_1(s, \sigma)$
return $\sigma \neq \sigma' \wedge \mathsf{Vrfy}(\mathrm{vk}, (m', \sigma'))$

Definition 3.7. A signature scheme $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ is a *strong, one-time signature scheme* if the success probability of any probabilistic polynomial-time adversary (A_0, A_1) in the game $\mathbf{OTS}_{\Sigma}(A_0, A_1)$ is negligible in the security parameter η.

3.4 Tag-based Encryption

Definition 3.8. A *tag-based encryption scheme (TBE)* handling tags of length ℓ (where ℓ is a polynomially-bounded function) is a triple of probabilistic polynomial-time algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. KeyGen takes a security parameter 1^η and returns a public key pk and secret key sk. The public key pk includes the security parameter 1^η and $\ell(\eta)$; as well as the description of sets $\mathcal{M}, \mathcal{R}, \mathcal{C}$, which denote the set of messages, randomness and ciphertexts respectively. These descriptions might depend on the public key pk. Enc takes as inputs pk, a tag $t \in \{0,1\}^\ell$ and $m \in \mathcal{M}$. It returns a ciphertext $c \in \mathcal{C}$. Dec takes as inputs the secret key sk, a tag t and $c \in \mathcal{C}$, and returns $m \in \mathcal{M}$ or $\perp$ when c is not a legitimate ciphertext. For the sake of consistency, these algorithms must satisfy $\mathsf{Dec}(\mathrm{sk}, t, c) = m$ for all $t \in \{0,1\}^\ell$, $m \in \mathcal{M}$, where $c = \mathsf{Enc}(\mathrm{pk}, t, m)$.

Definition 3.9. Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a TBE scheme. We say $\mathcal{E}$ is *IND-TBE-CCA secure* if for any 3-tuple of PPT oracle algorithms (A_0, A_1, A_2) and any polynomially-bounded function ℓ the advantage in the following game is negligible in the security parameter 1^η:

$A_0(1^\eta, \ell(\eta))$ outputs a target tag t. $\mathsf{KeyGen}(1^\eta)$ outputs $(\mathrm{pk}, \mathrm{sk})$ and the adversary is given pk. Then the adversary A_1 may ask polynomially-many queries to a decryption oracle $\mathcal{D}(t', c') = \mathsf{Dec}(\mathrm{sk}, t', c')$ for pairs tag-ciphertext (t', c') of its choice, with the restriction $t \neq t'$. At some point, A_1 outputs two equal length

messages m_0, m_1. A bit $b \leftarrow \{0,1\}$ is chosen at random and the adversary is given a challenge ciphertext $c \leftarrow \mathsf{Enc}(\mathrm{pk}, t, m_b)$. A_2 may continue asking the decryption oracle for pairs tag-ciphertext (t', c') of its choice, with the restriction $t \neq t'$. Finally, A_2 outputs a guess b'.

IND-TBE-CCA $_{\mathcal{E}}(A_0, A_1, A_2)$:
$t, s_1 \leftarrow A_0(1^\eta, \ell(\eta))$
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathsf{KeyGen}(1^\eta)$
$m_0, m_1, s_2 \leftarrow A_1^{\mathcal{D}}(s_1, \mathrm{pk})$
$b \leftarrow \{0,1\}$
$c \leftarrow \mathsf{Enc}(\mathrm{pk}, t, m_b)$
$b' \leftarrow A_2^{\mathcal{D}}(s_2, c)$
return $b = b'$

We define the advantage of A as $|\mathbb{P}[\mathbf{IND\text{-}TBE\text{-}CCA}(A)] - 1/2|$.

3.5 Interpretation

Suppose we have an encryption scheme Π, a commitment scheme Ω and a function that maps symbolic constants to constant bitstrings. Then we can define a mapping $[\![\cdot]\!]$ from algebraic messages $m \in \mathbf{Msg}$ to distributions over bitstrings $[\![m]\!] \in \mathbf{Str}$. This interpretation maps nonces to random bitstrings of length η; encryptions are interpreted by running the encryption algorithm $\mathcal{E}$ and for interpreting commitments and decommitments we use the commit algorithm Snd.

In order to achieve sound interpretation we will explore the security requirements on these cryptographic primitives. For the case of encryption it is satisfactory to use any IND-CCA encryption scheme as shown in [MW04]. For the case of commitments, using standard security definitions is not straightforward as they are not strong enough nor indistinguishability based. To achieve sound interpretation of the idealized Dolev-Yao model, throughout the next section we elaborate on a convenient security definition for commitment schemes.

4 Definitions of Non-malleability

As noticed by Fischlin and Fischlin [FF00], there are two different versions of non-malleability for commitment schemes, namely: NM with respect to opening (NMO) and NM with respect to commitment (NMC). NMC was the version originally proposed by Dolev, Dwork and Naor in [DDN91]. It means that given a commitment to a message m, the adversary is unable to build a different commitment to m', with m related to m'. This version of non-malleability is appropriate while considering perfectly binding commitments and only makes sense for schemes that are not perfectly hiding.

The other version NMO, seemingly weaker, means that an adversary that is first given a commitment to m and on a second stage its decommitment, is unable to find a different commitment-decommitment pair that decommits to a message m' related to m. This notion was studied by Di Crescenzo, Ishai and Ostrovsky [CIO98] and later by Di Crescenzo, Katz, Ostrovsky and Smith [CKOS01]. Intuitively a commitment scheme is non-malleable if the adversary can do no better than a

simulator which has no information at all about the message that was committed to. Next we recall their definition.

$\mathbf{NMO}_{\Omega}(A_1, A_2, D, R)$:
$\sigma \leftarrow \mathsf{TTP}(1^{\eta})$
$m_1 \leftarrow D$
$com_1, dec_1 \leftarrow \mathsf{Snd}(\sigma, m_1)$
$com_2 \leftarrow A_1(\sigma, com_1)$
$dec_2 \leftarrow A_2(\sigma, com_1, com_2, dec_1)$
$m_2 \leftarrow \mathsf{Rcv}(\sigma, com_2, dec_2)$
return $com_1 \neq com_2 \wedge R(m_1, m_2)$

$\mathbf{SIM}(S, D, R)$:
$m_1 \leftarrow D$
$m_2 \leftarrow S(1^{\eta}, D)$
return $R(m_1, m_2)$

Definition 4.1 (Non-malleability [CIO98,CKOS01])**.** Let $\Omega = (\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ be a commitment scheme. Ω is called *non-malleable* if for all PPT adversaries (A_1, A_2) there is a PPT simulator S such that for all distributions D and all relations R,

$$\mathbb{P}[\mathbf{NMO}_{\Omega}(A_1, A_2, D, R)] - \mathbb{P}[\mathbf{SIM}(S, D, R)]$$

is a negligible function of η.

Remark 4.2. To prevent that the adversary trivially wins, by refusing to decommit, the following restriction over the relation R is imposed: for all messages m, we have $R(m, \perp) = 0$.

4.1 NMC-CCA: Non-malleability Against Chosen Commitment Attacks

The previous definition deals with non-malleability with respect to opening. For the relation between symbolic and computational cryptography we need the stronger notion of non-malleability with respect to commitment. Intuitively, this is because in the algebraic setting $\mathsf{com}^r(m')$ cannot be deduced from $\mathsf{com}^r(m)$, with m' somehow related to m. Therefore we adapt the NMO definition to non-malleability with respect to commitment and we strengthen it by incorporating active adaptive security, allowing the adversary to mount *chosen commitment attacks* (CCA in short). Specifically, we empower the adversary with access to a decommitment oracle $\mathcal{D}$. To do so, from now on, we restrict our attention to non-interactive, perfectly binding trapdoor commitment schemes. The oracle $\mathcal{D}$ has access to the trapdoor information. It takes as argument a commitment c with the restriction that c is not equal to the challenge commitment com_1. Then if the commitment c has been correctly generated, the oracle returns a decommitment d which opens c, and otherwise it outputs $\perp$.

$\mathbf{NMC\text{-}CCA}_{\Omega}(A_0, A_1, R)$:
$\sigma \leftarrow \mathsf{TTP}(1^{\eta})$
$D, s_1 \leftarrow A_0^{\mathcal{D}}(\sigma)$
$m_1 \leftarrow D(\sigma)$
$com_1, dec_1 \leftarrow \mathsf{Snd}(\sigma, m_1)$
$com_2, s_r \leftarrow A_1^{\mathcal{D}}(s_1, com_1)$
$dec_2 \leftarrow \mathcal{D}(com_2)$
$m_2 \leftarrow \mathsf{Rcv}(\sigma, com_2, dec_2)$
return $com_1 \neq com_2 \wedge R(s_r, m_1, m_2)$

$\mathbf{SIM\text{-}CCA}_{\mathsf{TTP}}(S_0, S_1, R)$:
$\sigma \leftarrow \mathsf{TTP}(1^{\eta})$
$D, s_1 \leftarrow S_0(\sigma)$
$m_1 \leftarrow D(\sigma)$

$com_2, s_r \leftarrow S_1(s_1)$
$dec_2 \leftarrow \mathcal{D}(com_2)$
$m_2 \leftarrow \mathsf{Rcv}(\sigma, com_2, dec_2)$
return $R(s_r, m_1, m_2)$

Definition 4.3 (NMC-CCA)**.** Let $\Omega = (\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ be a commitment scheme. Ω is called *NMC-CCA secure* if for all PPT adversaries (A_0, A_1) there is a PPT simulator (S_0, S_1) such that for all relations R (with the same restriction as in 4.2),

$$\mathbb{P}[\mathbf{NMC\text{-}CCA}_{\Omega}(A_0, A_1, R)] - \mathbb{P}[\mathbf{SIM\text{-}CCA}_{\mathsf{TTP}}(S_0, S_1, R)]$$

is a negligible function of η.

4.2 An Indistinguishability Based Definition

Next we introduce an equivalent formulation of NMC-CCA that is more convenient to prove soundness of the Dolev-Yao model with respect to commitment schemes.

IND-COM-CCA$_b(A_0, A_1)$:
$\sigma \leftarrow \mathsf{TTP}(1^\eta)$
$m_0, m_1, s_1 \leftarrow A_0^{\mathcal{D}}(\sigma)$
$com_1, dec_1 \leftarrow Snd(\sigma, m_b)$
$b' \leftarrow A_1^{\mathcal{D}}(s_1, com_1)$
return b'

Definition 4.4 (IND-COM-CCA)**.** Let $\Omega = (\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ be a commitment scheme. Ω is said to be *IND-COM-CCA secure* if for all PPT adversaries (A_0, A_1)

$$\mathbb{P}[\,\mathbf{IND\text{-}COM\text{-}CCA}_1(A_0^{\mathcal{D}}, A_1^{\mathcal{D}}) = 1] - \mathbb{P}[\,\mathbf{IND\text{-}COM\text{-}CCA}_0(A_0^{\mathcal{D}}, A_1^{\mathcal{D}}) = 1]$$

is a negligible function of η.

Next we show that NMC-CCA and IND-COM-CCA are equivalent. We discuss it briefly as it is basically the proof that NM-CCA and IND-CCA are equivalent, adapted to commitment schemes.

Theorem 4.5. *Let $\Omega = (\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ be a commitment scheme. Then Ω is IND-COM-CCA secure if and only if Ω is NMC-CCA secure.*

Proof. (IND-COM-CCA $\Leftarrow$ NMC-CCA) Let (B_0, B_1) be an adversary for IND-COM-CCA. Then we build the following adversary (A_0, A_1) against NMC-CCA.

Algorithm $A_0^{\mathcal{D}}(\sigma)$:
$m_0, m_1, s_1 \leftarrow B_0^{\mathcal{D}}(\sigma)$
$D \leftarrow U(\{m_0, m_1\})$
return $D, (\sigma, m_0, m_1, s_1)$

Algorithm $A_1^{\mathcal{D}}((\sigma, m_0, m_1, s_1), c_1)$:
$b \leftarrow B_1^{\mathcal{D}}(s_1, c_1)$
$c_2 \leftarrow \mathsf{Snd}(\sigma, m_b)$
return c_2, ϵ

where U is the uniform distribution. Now take the relation $R(s_r, m_1, m_2)$ as m_1 equal to m_2. It should be clear, after unfolding (A_0, A_1) in the **NMC-CCA** game, that this adversary has the same advantage that (B_0, B_1) has against IND-COM-CCA.

(IND-COM-CCA $\Rightarrow$ NMC-CCA) Let (A_0, A_1) be an adversary for NMC-CCA. Then we build the following adversary (B_0, B_1) against IND-COM-CCA.

Algorithm $B_0^{\mathcal{D}}(\sigma)$:
$D, s_1 \leftarrow A_0^{\mathcal{D}}(\sigma)$
$m_0, m_1 \leftarrow D$
return $m_0, m_1, (\sigma, m_0, m_1, s_1)$

Algorithm $B_1^{\mathcal{D}}((\sigma, m_0, m_1, s_1), c_1)$:
$c_2 \leftarrow A_1^{\mathcal{D}}(s_1, c_1)$
$m \leftarrow \mathcal{D}(c_2)$
if $m = m_1$ **then return** 1
else return 0

Again, just by unfolding these adversaries in the IND-COM-CCA game, it is easy to verify that they have the same advantage that (A_0, A_1) has against NMC-CCA. □

It remains to show that such a security notion for a commitment scheme is achievable. In the next section we give a practical construction that achieves IND-COM-CCA security.

5 The Construction

We now propose a new construction for IND-COM-CCA that is computationally hiding, perfectly binding, reusable, non-interactive, non-malleable under adaptive adversaries, and provably secure under the assumption that trapdoor permutations exist.

Next we outline the idea of our construction. As pointed out by Di Crescenzo, Katz, Ostrovsky and Smith [CKOS01], an IND-CCA secure public key encryption scheme can be converted into a perfectly binding non-malleable commitment scheme. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an indistinguishable against adaptive chosen-ciphertext attacks secure public key encryption scheme. The idea is to commit to a message m by encrypting it using random coins r; commitment is set to be the ciphertext $c = \mathsf{Enc}(\mathrm{pk}, m; r)$; de-commitment is set to be the pair (m, r); finally the opening algorithm takes (c, m, r) and checks whether $c = \mathsf{Enc}(\mathrm{pk}, m; r)$. When trying to directly use this construction to instantiate an IND-COM-CCA commitment scheme one might not be able to simulate the de-commitment oracle. The reason is that given a ciphertext/commitment c, one recovers the purported embedded message m by using the decryption algorithm, but not necessarily the randomness r. One way to break through this situation is to include in the commitment a second ciphertext $c' = \mathsf{Enc}(\mathrm{pk}', r; r')$ encrypting the randomness r used in the first ciphertext $c = \mathsf{Enc}(\mathrm{pk}, m; r)$. This is the key idea of our construction. We additionally use one-time signatures and this together with tag-based encryption schemes ensure the de-commitment oracle does not leak vital information.

Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a tag based encryption scheme and let $\Sigma = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme. Define $(\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ as follows:

- TTP runs $\mathsf{KeyGen}(1^\eta)$ twice to obtain $(\mathrm{pk}_1, \mathrm{sk}_1)$ and $(\mathrm{pk}_2, \mathrm{sk}_2)$. The common reference string includes $\mathrm{pk}_1, \mathrm{pk}_2$.
- To commit to a message m, the sender Snd computes and outputs the commitment $C = (\mathrm{vk}, c_1, c_2, s)$ where $c_1 = \mathsf{Enc}(\mathrm{pk}_1, \mathrm{vk}, m; r_1)$, $c_2 = \mathsf{Enc}(\mathrm{pk}_2, \mathrm{vk}, r_1; r_2)$, with $r_1, r_2 \leftarrow \mathcal{R}$, $(\mathrm{vk}, \mathrm{sk}) \leftarrow \mathsf{Gen}(1^\eta)$ and $s \leftarrow \mathsf{Sign}(\mathrm{sk}, (c_1, c_2))$. The decommitment is set to be (m, r_1).
- To de-commit ciphertext $C = (\mathrm{vk}, c_1, c_2, s)$ using (m, r_1), the receiver Rcv first checks if the signature on (c_1, c_2) is correct, and afterwards whether or not $c_1 = \mathsf{Enc}(\mathrm{pk}_1, \mathrm{vk}, m; r_1)$.

We assume $\mathcal{R} = \mathcal{M}$.

Theorem 5.1. *Assume that* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is an IND-TBE-CCA secure tag based encryption scheme and that* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is a one-time strongly unforgeable signature scheme. Then* $(\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ *is an IND-COM-CCA secure commitment scheme.*

Proof. We transform an adversary A against the IND-COM-CCA security of the commitment scheme into adversaries against the TBE and the OTS. Next we will describe a sequence of games following the methodology advocated in [Sho04, BR06]. Let X_i be the event that A learns the challenge bit b in the i-th game.

Game 0. This is the unmodified IND-COM-CCA game. Trivially, $|\mathbb{P}[X_0] - 1/2|$ equals the advantage of A against IND-COM-CCA.

Game 1. In this game we disallow decryption queries $C = (\mathrm{vk}, c_1, c_2, s)$ s.t. $\mathrm{vk} = \mathrm{vk}^\star$ where $(\mathrm{vk}^\star, c_1^\star, c_2^\star, s^\star)$ is the challenge commitment. Then, we get that $|\mathbb{P}[X_1] - \mathbb{P}[X_0]|$ is less or equal than the advantage any PPT algorithm has in breaking the one-time strong unforgeability security of the OTS.

Game 2. Still decryption queries with $\mathrm{vk} = \mathrm{vk}^\star$ are forbidden. In this game we use the IND-CCA security of the second instance of the TBE scheme. The components $c_1^\star$ and $c_2^\star$ of the challenge ciphertext are changed to $c_1^\star = \mathsf{Enc}(\mathrm{pk}_1^\star, \mathrm{vk}^\star, m_b^\star; r_1)$, and $c_2^\star = \mathsf{Enc}(\mathrm{pk}_2^\star, \mathrm{vk}^\star, r')$ where $r', r_1 \leftarrow \mathcal{R}$. Now, we have $|\mathbb{P}[X_2] - \mathbb{P}[X_1]|$ is less or equal than the advantage any PPT algorithm has in breaking the selective IND-CCA security of the TBE.

Finally it is shown that $|\mathbb{P}[X_2] - 1/2]|$ is bounded by the advantage any PPT algorithm has in breaking the selective IND-CCA security of the first instance of the TBE.

Putting everything together, we get that $|\mathbb{P}[X_0] - 1/2|$ is bounded by the advantages in breaking the OTS scheme plus twice the advantage in breaking the selective IND-CCA of the TBE scheme. Next we describe the concrete adversaries,

Game 0 $\approx$ Game 1. Assume that there is an adversary (A_0, A_1) that is able to distinguish the environments of Game 0 and 1. Then we build an adversary (B_0, B_1) against the one-time strong unforgeability of the signature scheme.

Algorithm $B_0(1^\eta, \mathrm{vk})$:
$\mathrm{pk}_1, \mathrm{sk}_1 \leftarrow \mathsf{KeyGen}(1^\eta)$
$\mathrm{pk}_2, \mathrm{sk}_2 \leftarrow \mathsf{KeyGen}(1^\eta)$
$m_0, m_1, s_1 \leftarrow A_0^{\mathcal{D}}(\mathrm{pk}_1, \mathrm{pk}_2)$
$b \leftarrow \{0, 1\}$
$r_1 \leftarrow \mathcal{R}$
$c_1 \leftarrow \mathsf{Enc}(\mathrm{pk}_1, \mathrm{vk}, m_b; r_1)$
$c_2 \leftarrow \mathsf{Enc}(\mathrm{pk}_2, \mathrm{vk}, r_1)$
return $(c_1, c_2), (s_1||\mathrm{vk}||c_1||c_2||\mathrm{sk}_1||\mathrm{sk}_2)$

and $B_1((s_1||\mathrm{vk}||c_1||c_2||\mathrm{sk}_1||\mathrm{sk}_2), s) = [b' \leftarrow A_1^{\mathcal{D}}(s_1, (\mathrm{vk}, c_1, c_2, s))]$. Calls to the decommitment oracle $\mathcal{D}(\mathrm{vk}', c_1', c_2', s')$ are simulated by firstly verifying the signature $\mathsf{Vrfy}(\mathrm{vk}', (c_1', c_2'), s')$. If the verification succeeds then the oracle returns the pair $(\mathsf{Dec}(\mathrm{sk}_1, \mathrm{vk}', c_1'), \mathsf{Dec}(\mathrm{sk}_2, \mathrm{vk}', c_2'))$ and otherwise it outputs $\perp$. If the adversary eventually performs a query $\mathcal{D}(\mathrm{vk}', c_1', c_2', s')$ with $\mathrm{vk}' = \mathrm{vk}$ then the execution of the adversary is aborted and B outputs $((c_1', c_2'), s')$, thus breaking the one-time strong unforgeability of the signature scheme.

Game 1 $\approx$ Game 2. Assume that there is an adversary (A_0, A_1) that is able to distinguish the environments of Game 1 and 2. Then we build an adversary (B_0, B_1, B_2) against the IND-CCA security of the second TBE. Take $B_0(1^\eta, \ell(\eta)) = [(\mathrm{vk}, \mathrm{sk}) \leftarrow \mathsf{Gen}(1^\eta); \mathbf{return}\ \mathrm{vk}, (\mathrm{vk}||\mathrm{sk})]$ and $B_1(s_1, \mathrm{pk}_2) = [r', r_1 \leftarrow \mathcal{R}; \mathbf{return}\ r', r_1, (s_1||r'||r_1||\mathrm{pk}_2)]$ and

$$\begin{array}{l}\hline
\mathbf{Algorithm}\ B_2^{\mathcal{O}_{\mathrm{sk}_2}}((\mathrm{vk}||\mathrm{sk}||r'||r_1||\mathrm{pk}_2), c_2): \\
\mathrm{pk}_1, \mathrm{sk}_1 \leftarrow \mathsf{KeyGen}(1^\eta) \\
m_0, m_1, s_1 \leftarrow A_0^{\mathcal{D}}(\mathrm{pk}_1, \mathrm{pk}_2) \\
b \leftarrow \{0,1\} \\
c_1 \leftarrow \mathsf{Enc}(\mathrm{pk}_1, \mathrm{vk}, m_b; r_1) \\
s \leftarrow \mathsf{Sign}(\mathrm{sk}, (c_1, c_2)) \\
b' \leftarrow A_1^{\mathcal{D}}(s_1, (\mathrm{vk}, c_1, c_2, s)) \\
\mathbf{if}\ b = b'\ \mathbf{then\ return}\ 1 \\
\mathbf{else\ return}\ 0 \\
\hline
\end{array}$$

Calls to the decommitment oracle $\mathcal{D}(\mathrm{vk}, c_1, c_2, s)$ are simulated by firstly verifying the signature $\mathsf{Vrfy}(\mathrm{vk}, (c_1, c_2), s)$. If the verification succeeds then the oracle returns $(\mathsf{Dec}(\mathrm{sk}_1, \mathrm{vk}, c_1), \mathcal{O}_{\mathrm{sk}_2}(c_2))$ and otherwise it outputs $\perp$.

Finally we show that $|\mathbb{P}[X_2] - 1/2]|$ is bounded by the advantage any PPT algorithm has in breaking the selective IND-CCA security of the first instance of the TBE. Assume that there is an adversary (A_0, A_1) for Game 2. Then we build an adversary (B_0, B_1, B_2) against the IND-CCA security of the first TBE. Take $B_0(1^\eta, \ell(\eta)) = [(\mathrm{vk}, \mathrm{sk}) \leftarrow \mathsf{Gen}(1^\eta); \mathbf{return}\ \mathrm{vk}, (\mathrm{vk}||\mathrm{sk})]$ and

$$\begin{array}{l}\hline
\mathbf{Algorithm}\ B_1^{\mathcal{O}_{\mathrm{sk}_1}}((\mathrm{vk}||\mathrm{sk}), \mathrm{pk}_1): \\
\mathrm{pk}_2, \mathrm{sk}_2 \leftarrow \mathsf{KeyGen}(1^\eta) \\
m_0, m_1, s_1 \leftarrow A_0^{\mathcal{D}}(\mathrm{pk}_1, \mathrm{pk}_2) \\
\mathbf{return}\ (m_0, m_1), (\mathrm{vk}||\mathrm{sk}||\mathrm{pk}_1||s_1||\mathrm{pk}_2||\mathrm{sk}_2) \\
\hline
\end{array}$$

$$\begin{array}{l}\hline
\mathbf{Algorithm}\ B_2^{\mathcal{O}_{\mathrm{sk}_1}}((\mathrm{vk}||\mathrm{sk}||\mathrm{pk}_1||s_1||\mathrm{pk}_2||\mathrm{sk}_2), c_1): \\
r' \leftarrow \mathcal{R} \\
c_2 \leftarrow \mathsf{Enc}(\mathrm{pk}_2, \mathrm{vk}, r') \\
s \leftarrow \mathsf{Sign}(\mathrm{sk}, (c_1, c_2)) \\
b' \leftarrow A_1^{\mathcal{D}}(s_1, (\mathrm{vk}, c_1, c_2, s)) \\
\mathbf{return}\ b' \\
\hline
\end{array}$$

Calls to the decommitment oracle $\mathcal{D}(\mathrm{vk}, c_1, c_2, s)$ are simulated by firstly verifying the signature $\mathsf{Vrfy}(\mathrm{vk}, (c_1, c_2), s)$. If the verification succeeds then the oracle returns $(\mathcal{O}_{\mathrm{sk}_1}(c_1), \mathsf{Dec}(\mathrm{sk}_2, \mathrm{vk}, c_2))$ and otherwise it outputs $\perp$. □

6 Protocol Execution and State Traces

We now prove that it is possible to port proofs in the symbolic framework to the computational one. First, for the sake of self-containment we describe the adversarial model and the execution environment following the directions of Micciancio and Warinschi [MW04]. We refer the reader to this paper for a thorough explanation.

The message space and the closure operator were defined in Section 2. Messages are used to formally describe cryptographic protocols. The closure represents the knowledge that can be extracted from a message, and is used to define what valid algebraic protocol runs are. Intuitively a protocol run is valid if every message sent by a principal can be deduced from its knowledge except maybe for some fresh randomness. In this setting an adversary is in control of the communication media and is able to interact with honest participants. Consider then an adversary that has access to an oracle that will play the role of the honest participants. This adversary can start new sessions of the protocol and send messages to a principal of a given session and get the respective answer back. Formally, the adversary A can perform one of the following queries to the execution oracle $\mathcal{O}$.

1. $\mathsf{newsession}([I_1 \dots I_n])$ that takes a list of user identities I_i and returns a new session identifier s.
2. $\mathsf{send}(s, I, m)$ that delivers the message m to the principal I of session s. Then $\mathcal{O}$ updates I's state and returns the answer to the adversary.

In case that the adversary performs a query that is not according to the protocol, for the specific state of the receiver, the oracle aborts the execution of this session.

In a formal protocol, the messages exchanged are algebraic expressions from the message algebra. A formal adversary A^f will interact with the formal oracle $\mathcal{O}^f$ in a symbolic protocol run.

On the other hand, a computational adversary A^c is a probabilistic polynomial-time Turing machine that operates on bitstrings. For a fixed value of the security parameter there is a set of primitive bitstrings for constants and nonces denoted by $\mathbf{Prim}_\eta$. The set of bitstrings $\mathbf{Msg}_\eta$ is build from $\mathbf{Prim}_\eta$ by tupling, encryptions, commitments and decommitments. There is a set **Sid** of *session identifiers*; a set **Uid** of *user identities* and a set **Vars** of *variables* in the abstract protocol description.

Let $F : \mathbf{Sid} \times \mathbf{Uid} \to (\mathbf{Vars} \to \mathbf{Msg}, \mathbb{N})$ be the state maintained by the formal oracle $\mathcal{O}^f$. On input (s, I) it returns the state of principal I in session s together with his *instruction pointer*. The instruction pointer indicates on which step of the abstract protocol this principal is. Similarly, $C : \mathbf{Sid} \times \mathbf{Uid} \to (\mathbf{Vars} \to \mathbf{Msg}_\eta, \mathbb{N})$ is the state maintained by the computational oracle $\mathcal{O}^c$. Assume without loss of generality that all the sessions are created at the beginning. Then, a formal adversary A_f is just a sequence of $\mathsf{send}(s, I, m)$ queries. We say that a formal adversary A_f is a valid Dolev-Yao adversary ($A_f \in \mathbf{DY}$) if each message he sends to the oracle is in the closure of his initial knowledge plus the answers he gets from the oracle $\mathcal{O}^f$. A protocol execution, thus, is the sequence of states $F_0, F_1, \dots$ of the formal oracle O^f and is denoted by $\mathsf{trace}(A_f, \mathcal{O}^f)$. After fixing the randomness of the adversary and that of the oracle environment to τ_A and $\tau_\mathcal{O}$, we can similarly define a computational execution trace $\mathsf{trace}(A_c(\tau_A), \mathcal{O}^c(\tau_\mathcal{O}))$ as the sequence of states $C_0, C_1, \dots$ of the computational oracle $\mathcal{O}^c$.

Definition 6.1. We say that $[\![\cdot]\!] : \mathbf{Prim} \to \mathbf{Prim}_\eta$ is an *interpretation function* if it is injective and structure preserving (i.e., maps formal nonces to nonce bitstrings, formal commitments to commitments and so on).

Definition 6.2. Let $F = F_0, F_1, \ldots$ be a formal execution trace and let $C = C_0, C_1, \ldots$ be a concrete execution trace. We say that $F \preceq C$ if there exists an interpretation function $[\![\cdot]\!]$ such that $[\![F_0]\!] = C_0, [\![F_1]\!] = C_1, \ldots$.

The following theorem shows that a computational adversary has no more power than an algebraic adversary.

Theorem 6.3. *Let* $(\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ *be an IND-COM-CCA secure commitment scheme and let* $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be an IND-CCA secure encryption scheme. For any computational adversary* A_c*, the probability*

$$\mathbb{P}[\,\exists A_f \in \boldsymbol{DY} : \mathsf{trace}(A_f, \mathcal{O}^f) \preceq \mathsf{trace}(A_c(\tau_A), \mathcal{O}^c(\tau_{\mathcal{O}}))]$$

is overwhelming. Here the probability is taken over the random choices τ_A *of the adversary and* $\tau_{\mathcal{O}}$ *of the oracle.*

Proof. First fix the randomness τ_A and $\tau_{\mathcal{O}}$. Running the computational adversary A_c, it produces a sequence of queries/answers to/from the computational oracle. Because we know all the trapdoor information that the oracle generates and because the adversary has to send properly typed messages, we can de-construct any message sent into primitive terms. Choosing new algebraic terms for each distinct primitive bitstring encountered we build a sequence of algebraic queries which constitutes an algebraic adversary A_f. Note that for different random choices of τ_A and $\tau_{\mathcal{O}}$ we get the same A_f (up to renaming) with overwhelming probability.

It remains to show that the adversary we just built is Dolev-Yao. Suppose that it is not. Then A_f must, at some point, send a query that contains a non-adversarial nonce $n^\star$ that is not in the closure of the messages he received before. If this nonce occurs inside an encryption (with an unknown key) then one can build an adversary breaking the IND-CCA security of the encryption scheme [MW04]. Assume then that it occurs inside a commitment. We now build an adversary that breaks the IND-COM-CCA security of the commitment scheme.

This adversary simulates the environment to A_c using the de-commit oracle when necessary except for the query that contains $n^\star$. There it generates two interpretations (n_0, n_1) for $n^\star$ and gives them as challenge plaintext for the IND-COM-CCA game. The challenger gives back a commitment to n_b where b is the challenge bit. This commitment to n_b is used to answer the oracle queries. At the moment A_C outputs the interpretation of $n^\star$ we can check whether it is n_0 or n_1. □

A formal security notion is a predicate $\mathsf{P_f}$ on formal traces. A protocol $\Pi \models_f \mathsf{P_f}$ if for all adversaries $A_f \in \mathbf{DY}$ holds that $\mathsf{trace}(A_f, \mathcal{O}^f) \in \mathsf{P_f}$. Similarly, a computational security notion is a predicate $\mathsf{P_c}$ on computational traces. A protocol $\Pi \models_c \mathsf{P_c}$ if for all probabilistic polynomial-time adversaries A_c holds that $\mathsf{trace}(A_f, \mathcal{O}^f) \in \mathsf{P_c}$ with overwhelming probability (taken over the random choices of the adversary and the ones of the oracle environment). The proof of the following theorem follows as in [MW04].

Theorem 6.4. *Let* $(\mathsf{TTP}, \mathsf{Snd}, \mathsf{Rcv})$ *be a IND-COM-CCA secure commitment scheme and let* $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be an IND-CCA secure encryption scheme. Let* $\mathsf{P_f}$ *and* $\mathsf{P_c}$ *be respectively formal and computational security notions such that for all formal traces*

ft and all computational traces ct it holds that $(ft \in \mathsf{P_f} \wedge ft \preceq ct) \Longrightarrow ct \in \mathsf{P_c}$. *Then*

$$\Pi \models_f \mathsf{P_f} \implies \Pi \models_c \mathsf{P_c} \,.$$

□

7 Conclusions

We presented two equivalent security notions for commitment schemes: a simulation based definition and a indistinguishability based one. We then gave a concrete scheme satisfying this security notion. This construction is of interest on itself as it is generic and has some interesting features like being reusable, perfectly binding and secure against adaptive chosen-commitment attacks. We then applied this new machinery to give sound interpretation of symbolic commitments while considering active adversaries.

References

ABHS05. Pedro Adão, Gergei Bana, Jonathan Herzog, and Andre Scedrov. Soundness of formal encryption in the presence of key-cycles. In *ESORICS'05*, volume 3679 of *LNCS*, pages 374–396. Springer, 2005.

ABS05. Pedro Adão, Gergei Bana, and Andre Scedrov. Computational and information-theoretic soundness and completeness of formal encryption. In *CSFW'05*, pages 170–184. IEEE, 2005.

ABW06. Martín Abadi, Mathieu Baudet, and Bogdan Warinschi. Guessing attacks and the computational soundness of static equivalence. In *FOSSACS'06*, volume 3921 of *LNCS*, pages 398–412. Springer, 2006.

AR02. Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

AW05. Martín Abadi and Bogdan Warinschi. Security analysis of cryptographically controlled access to XML documents. In *Proceedings of the 24th ACM Symposium on Principles of Database Systems*, pages 108–117. ACM Press, 2005.

BCK05. Mathieu Baudet, Véronique Cortier, and Steve Kremer. Computationally sound implementations of equational theories against passive adversaries. In *ICALP'05*, volume 3580 of *LNCS*, pages 652–663. Springer, 2005.

BLMW07. Emmanuel Bresson, Yassine Lakhnech, Laurent Mazaré, and Bogdan Warinschi. A generalization of DDH with applications to protocol analysis and computational soundness. In *CRYPTO'07*, LNCS. Springer, 2007.

BR06. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *EUROCRYPT'06*, volume 4004 of *LNCS*, pages 409–426. Springer, 2006.

CIO98. Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *STOC'98*, pages 141–150. ACM Press, 1998.

CKKW06. Véronique Cortier, Steve Kremer, Ralf Küsters, and Bogdan Warinschi. Computationally sound symbolic secrecy in the presence of hash functions. In *FSTTCS'06*, volume 4337 of *LNCS*, pages 176–187. Springer, 2006.

CKOS01. Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Efficient and non-interactive non-malleable commitment. In *EUROCRYPT'01*, volume 2045, pages 40–59. Springer, 2001.

DDN91. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *STOC'91*, pages 542–552. ACM Press, 1991.

EGL85. S. Even, O. Goldreich, and A. Lempel. A randomizing protocol for signing contracts. *Comm. ACM*, 28(6):637–647, 1985.

FF00. Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. In *CRYPTO'03*, volume 1880 of *LNCS*, pages 413–431. Springer, 2000.

GM84. S. Goldwasser and S. Micali. Probabilistic encryption. *J. Computer and System Sciences*, 28(2):270–299, 1984.

GMW91. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. *J. ACM*, 38(1):691–729, 1991.

GvR06. Flavio D. Garcia and Peter van Rossum. Sound computational interpretation of symbolic hashes in the standard model. In *IWSEC'06*, volume 4266 of *LNCS*, pages 33–47. Springer, 2006.

Her05. Jonathan Herzog. A computational interpretation of Dolev-Yao adversaries. *Theoretical Computer Science*, 340(1):57–81, 2005.

JLM05. Romain Janvier, Yassine Lakhnech, and Laurent Mazar. Completing the picture: Soundness of formal encryption in the presence of active adversaries. In *ESOP'05*, volume 3444 of *LNCS*, pages 172–185. Springer, 2005.

KM07. Steve Kremer and Laurent Mazaré. Adaptive soundness of static equivalence. In *ESORICS'07*, LNCS. Springer, 2007. To appear.

Maz07. Laurent Mazaré. Computationally sound analysis of protocols using bilinear pairings. In *WITS'07*, pages 6–21, 2007.

MP05. Daniele Micciancio and Saurabh Panjwani. Adaptive security of symbolic encryption. In *TCC'05*, volume 3378 of *LNCS*, pages 169–187. Springer, 2005.

MW04. Daniele Micciancio and Bogdan Warinschi. Soundness of formal encryption in the presence of active adversaries. In *TCC'04*, volume 2951 of *LNCS*, pages 133–151. Springer, 2004.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attack. In *STOC'90*, pages 427–437. ACM, 1990.

RS92. C. Rackoff and D.R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444, 1992.

Sho04. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. http://eprint.iacr.org/.