Non-Malleability vs. CCA-Security: The Case of Commitments

Brandon Broadnax, Valerie Fetzer, Jörn Müller-Quade, and Andy Rupp

Karlsruhe Institute of Technology, Karlsruhe, Germany
{brandon.broadnax, valerie.fetzer, joern.mueller-quade,
andy.rupp}@kit.edu

Abstract. In this work, we settle the relations among a variety of security notions related to non-malleability and CCA-security that have been proposed for commitment schemes in the literature. Interestingly, all our separations follow from two generic transformations. Given two appropriate security notions X and Y from the class of security notions we compare, these transformations take a commitment scheme that fulfills notion X and output a commitment scheme that still fulfills notion X but not notion Y.

Using these transformations, we are able to show that some of the known relations for public-key encryption do not carry over to commitments. In particular, we show that, surprisingly, parallel non-malleability and parallel CCA-security are not equivalent for commitment schemes. This stands in contrast to the situation for public-key encryption where these two notions are equivalent as shown by Bellare et al. at CRYPTO '99.

1 Introduction

A commitment scheme is a two-party protocol that enables one party, called the sender, to commit himself to a value, while keeping it hidden from others and to later reveal that value to the other party, called the receiver. Commitment schemes belong to the most important building blocks of cryptography and have many applications including coin flipping protocols, signature schemes and zero-knowledge proofs.

Non-malleability (first introduced in [15]) is an important security notion for commitment schemes that is, like its counterpart for encryption schemes, concerned with defending against man-in-the-middle attacks. Informally, a commitment scheme is called (stand-alone) *non-malleable* if it is impossible for a man-in-the-middle adversary that receives a commitment to a value v to "successfully" commit to a related value $\tilde{v}$.

Several variants of non-malleability have been defined in the literature. For *parallel non-malleability* [16] the adversary receives multiple commitments in parallel and commits to multiple values in parallel. For *concurrent non-malleability* [26] the adversary receives and sends multiple commitments in an arbitrary schedule determined by the adversary.

There are many works on non-malleable commitment schemes in the literature, e.g., [11, 13, 16, 18, 19, 22, 27, 31]. Non-malleable commitment schemes have numerous applications in the field of multi-party computation. For instance, parallel non-malleable commitment schemes have been used for constructing round-efficient (six round) MPC protocols [16], concurrently non-malleable commitment schemes have been used as a building block for black-box MPC protocols [31] and (stand-alone) non-malleable commitment schemes have been used for concurrently composable protocols [27].

Another security notion related to non-malleability is *CCA-security* [25]. A commitment scheme is called *CCA-secure* if it remains hiding even if the adversary has access to an oracle that "breaks" polynomially many commitments. There exist several relaxed variants of CCA-security. For *parallel CCA-security* [24] the adversary can ask the oracle a single query that consists of polynomially many commitments sent to the oracle in parallel. For *one-one CCA-security* [23] the adversary can ask the oracle a single query that consists of exactly one commitment.

CCA-secure commitment schemes are a central building block for concurrently secure multi-party computation in the *plain model*, i.e., without trusted setup apart from authenticated channels. CCA-secure commitment schemes were introduced by [9] in the context of "Angel-based security". Angel-based security, first proposed by [30], relaxes the security notion of the universal composability framework (UC) [6] in order to circumvent the broad impossibility results of the latter. In the Angel-based security framework, concurrently secure multi-party computation in the plain model can be achieved for (almost) every cryptographic task [9, 10, 23–25]. This stands in contrast to the UC framework where many important functionalities such as commitments or zero-knowledge cannot be realized in the plain model (see, e.g., [7, 8]). Moreover, parallel CCA-secure commitment schemes [5, 23] and one-one CCA-secure commitment schemes [23, 24] were used as building blocks for several recent round-efficient concurrently secure general multi-party computation protocols in the plain model.

Considering this great variety of useful security notions, it is a natural question to ask how these notions are related. Surprisingly, only a few relations have been analyzed so far (cf. Fig. 1). Most works focus either on security notions related to CCA-security *or* on security notions related to non-malleability. In this work we focus on the relations *between* the two concepts and prove a more complete relation diagram. Motivated by public-key encryption, we also define and analyze the hierarchy of *q-bounded CCA-security* [14], where the adversary can adaptively ask the oracle at most q queries for a fixed natural number q.

Related Work. This work is in the vein of a series of papers establishing relations between different variants of security definitions for public-key encryption and commitments such as [1–4, 12, 14, 29]. For instance, Bellare et al. [1] prove relations among non-malleability-based and indistinguishability-based notions of security for public-key encryption. In particular, they show that IND-CCA2-security and NM-CCA2-security are equivalent. Bellare and Sahai [3] show that the indistinguishability-based definition of non-malleable encryption is equivalent to the simulation-based definition. Moreover, they show that non-malleability is equivalent to indistinguishability for public key encryption under a "parallel chosen ciphertext attack". Bellare et al. [2] show that standard security for commitment schemes does not imply selective opening security. Böhl, Hofheinz and Kraschewski [4] analyze the relations between indistinguishability-based and simulation-based definitions of selective opening security for public key encryption.

For the set of security notions for commitment schemes that are considered in this work, only a few relations are resolved, however. Pandey, Pass and Vaikuntanathan [28] show that CCA-security implies concurrent non-malleability. In [13] Ciampi et al. show that the non-malleable commitment scheme from a preliminary version of [20] is not concurrently non-malleable. Lin, Pass and Venkitasubramaniam [26] construct a commitment scheme that separates non-malleability and parallel non-malleability. The remaining relations are, to the best of our knowledge, unsettled.

Our Contribution. We settle the relations among a variety of security notions related to non-malleability and CCA-security that have been proposed for commitment schemes in the literature (see Fig. 1).[1]

Our results show, in particular, that some of the known results from previous works that dealt with public-key encryption do not carry over to the case of commitment schemes. In particular, the result of Bellare and Sahai [3], who showed that parallel non-malleability and parallel CCA-security are equivalent for public-key encryption schemes, does not hold for commitment schemes, in general. These two notions are only equivalent for non-interactive commitment schemes (see Appendix A).

Interestingly, we are able to obtain all of our separation results using two generic transformations. Given two appropriate security notions X and Y from the class of security notions we compare in this work, these transformations take a commitment scheme that fulfills notion X and output a commitment scheme that still fulfills notion X but not notion Y. Both transformations are fully black-box and require no additional computational assumptions.

The first transformation is used for separations where Y is a CCA-related security notion. The key idea of this transformation is to expand a commitment scheme that fulfills a security notion X by a "puzzle phase" where the sender sends a specific computationally hard puzzle to the receiver. If the receiver answers with a correct solution then the sender "gives up" and sends his input to the receiver who can then trivially win in the security game in this case. If the puzzle is tailored appropriately then the expanded commitment scheme still fulfills notion X but fails to fulfill notion Y. Intuitively, this separation holds be-

[1] Note that we always use *statistically binding* commitment schemes in this work, since we want the committed values in the experiments for CCA-security and non-malleability (as well as their variants) to be uniquely defined (with overwhelming probability). We note that using *strong computationally binding* commitment schemes would also work.

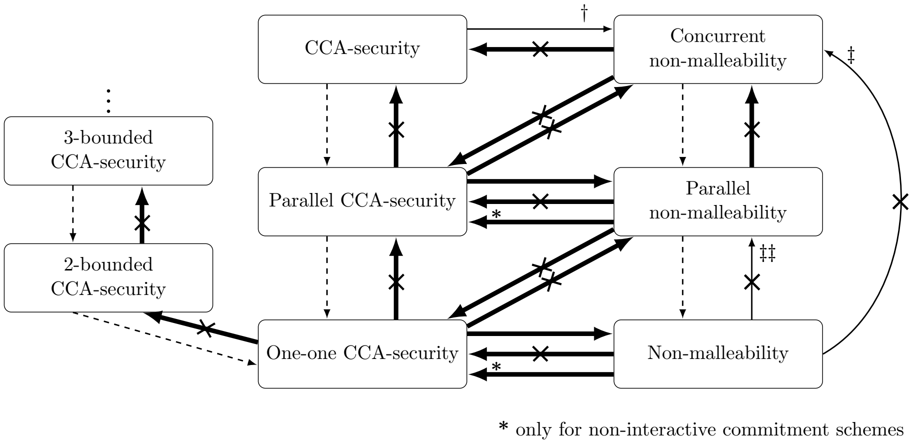

Fig. 1. The relations between several security notions for commitment schemes. The dotted arrows indicate trivial implications. The solid arrows indicate relations proved in the literature (see [28] for † and [13] for ‡) or separating commitment schemes from the literature (such as the scheme $\langle \tilde{C}, \tilde{R} \rangle$ from [26] for ‡‡). The thick arrows indicate our results.

cause an adversary in the Y-security game has access to an oracle that "breaks" the puzzle but an adversary in the X-security game does not.

The second transformation is used for separations where Y is a variant of non-malleability. This transformation expands a given commitment scheme by adding a "share phase" in which the sender commits to two random shares of his input in a specific order. This is done in such a way that a man-in-the-middle adversary is able to forward these commitments to the receiver in his experiment. After the commit phase is over, these shares will be opened by the implicit oracle in the experiment and given to the distinguisher, who can then reconstruct the committed value.

On Black-Box Separations. We note that the separations proven in this work differ from *black-box separations.* Separating a security notion X from a security notion Y by a black-box separation means that one cannot construct a scheme satisfying X from a scheme satisfying Y in a *black-box manner.*

Black-box separations are stronger than our separations. However, we note that one cannot achieve black-box separations between the security notions described in this work. This is because, given a (statistically binding) commitment scheme satisfying any of the security notions considered in this work, one can construct a commitment scheme satisfying any other security notion in this work in a black-box way. This can be shown as follows: first, each of the notions described in this work implies the standard hiding property for commitment schemes. Furthermore, given a commitment scheme that is binding and hiding, one can construct a one-way function in a black-box way [21]. Moreover, [23] showed how to construct a CCA-secure commitment scheme from any one-way function in a black-box way. Since CCA-security implies any other notion described in this work, the statement follows. This transformation is, of course, highly redundant and inefficient and therefore only of theoretical interest.

2 Preliminaries and Definitions

For any $x \in \{0,1\}^*$, we let $|x|$ denote the size of x. If S is a set then $s \overset{\$}{\leftarrow} S$ denotes the operation of picking an element s of S uniformly at random. We use the term PPT as abbreviation for probabilistic polynomial time (in the security parameter) in the context of algorithms or machines. We write $\mathcal{A}(x)$ to indicate that $\mathcal{A}$ is an algorithm with input x, we write $\mathcal{A}^{\mathcal{O}}(x)$ to indicate that $\mathcal{A}$ is an algorithm with input x and black-box access to the oracle $\mathcal{O}$ and we write $y \leftarrow \mathcal{A}(x)$ to denote the output y of $\mathcal{A}$ with input x.

The term *negligible* is used for denoting functions that are (asymptotically) smaller than one over any polynomial. More precisely, a function $\mathsf{negl}(\cdot)$ from non-negative integers to reals is called *negligible* if for every constant $c > 0$ and all sufficiently large k, it holds that $\mathsf{negl}(k) < k^{-c}$.

Commitment Schemes. A Commitment schemes is a two-phase two-party protocol in which one party, the sender, commits himself in the first phase (the commit phase) to a value while keeping it secret from the other party, the receiver. In the second phase (the reveal phase) the sender reveals the value he committed to. At the end of this phase the receiver outputs this value. In addition to the requirement that both sender and receiver run in polynomial time, we require that a commitment scheme fulfills the following two properties:

- *Hiding*: The commit phase yields no knowledge of the value to the receiver. This also applies to cheating receivers.
- *Binding*: Given the transcript of the interaction in the first phase, there exists at most one value that the receiver can accept as the correct opening in the reveal phase. This also applies to cheating senders.

For a formal definition see [17]. In this work we focus on statistically binding and computationally hiding (string) commitment schemes, i.e., the binding property holds against unbounded adversaries, while the hiding property only holds against computationally bounded (non-uniform) adversaries. This is because committed values are then uniquely defined with overwhelming probability.

In a tag-based commitment scheme both parties get a bit string called tag as additional input. We will denote by $\mathsf{Com}_{tag}(v)$ a (possibly interactive) commitment to the value $v \in \{0,1\}^k$ under the tag $tag \in \{0,1\}^k$ using the commitment scheme Com.[2] In the following, we only consider tag-based commitment schemes

[2] Note that if we later use a formulation like "the sender sends $\mathsf{Com}_{tag}(v)$ to the receiver", we do not necessarily assume that the commitment scheme is non-interactive and hence consists of only one message. We rather use this formulation as an abbreviation for "the sender commits to v under the tag tag to the receiver using the commitment scheme Com".

because the definitions of security notions considered here require tag-based commitment schemes.

CCA-Secure Commitment Schemes. Roughly speaking, a tag-based commitment scheme Com is said to be CCA-secure [25], if the value committed to using a tag *tag* remains hidden even if the receiver has access to a oracle that "breaks" polynomially many commitments using a different tag $tag' \neq tag$ for him. In this work we consider committed value oracles (oracles that return the committed value) only, but not decommitment oracles (oracles that return the full decommitment information).

The CCA-oracle $\mathcal{O}_{\mathsf{cca}}$ for Com acts as follows in an interaction with an adversary $\mathcal{A}$: It participates with $\mathcal{A}$ in polynomially many sessions of the commit phase of Com as an honest receiver. At the end of each session, if the session is valid, the oracle returns the unique value v committed to in the interaction; otherwise, it returns $\perp$. Note that if a session has multiple valid committed values, the CCA-oracle also returns $\perp$. The statistical binding property guarantees that this happens with only negligible probability.

Let $\mathsf{Exp}^{\mathsf{cca}}_{\mathsf{Com},\mathcal{A}}(k)$ denote the output of the following probabilistic experiment: Let $\mathcal{O}_{\mathsf{cca}}$ be the CCA-oracle for Com. The adversary has during the entire course of the experiment access to $\mathcal{O}_{\mathsf{cca}}$. On input $1^k, z$, the adversary $\mathcal{A}^{\mathcal{O}_{\mathsf{cca}}}$ picks a tag *tag* and two strings v_0 and v_1 with $|v_0| = |v_1|$ and sends this triple to the experiment. The experiment randomly selects a bit $b \overset{\$}{\leftarrow} \{0,1\}$ and then commits to v_b using the tag *tag* to $\mathcal{A}^{\mathcal{O}_{\mathsf{cca}}}$. Finally, $\mathcal{A}^{\mathcal{O}_{\mathsf{cca}}}$ sends a bit b' to the experiment, which outputs 1 if $b = b'$ and 0 otherwise. The output of the experiment is replaced by $\perp$ if during the execution the adversary queries the oracle on a commitment that uses the challenge tag *tag*.

Definition 1 (CCA-secure commitment scheme). *Let* Com *be a tag-based commitment scheme and* $\mathcal{O}_{\mathsf{cca}}$ *be the CCA-oracle for* Com*. We say that* Com *is* CCA-secure*, if for every* PPT*-adversary* $\mathcal{A}$ *and all* $z \in \{0,1\}^*$ *the following holds:*

$$\mathsf{Adv}^{\mathsf{cca}}_{\mathsf{Com},\mathcal{A}(z)}(k) := \mathsf{Pr}[\,\mathsf{Exp}^{\mathsf{cca}}_{\mathsf{Com},\mathcal{A}(z)}(k) = 1\,] - \frac{1}{2} \leq \mathsf{negl}(k)$$

for a negligible function negl.

Parallel CCA-Secure Commitment Schemes. Parallel CCA-secure commitment schemes are for example defined by Kiyoshima [23]. The parallel CCA-oracle $\mathcal{O}_{\mathsf{pcca}}$ is defined like the CCA-oracle, except that the adversary is restricted to a parallel query, i.e., the adversary can only send a single query that may contain multiple commitments sent in parallel, to the oracle. Let $\mathsf{Exp}^{\mathsf{pcca}}_{\mathsf{Com},\mathcal{A}}(k)$ define the output of the security game for parallel CCA-security (PCCA). The formal definition is then analogous to the definition of CCA-security.

One-One CCA-Secure Commitment Schemes. One-one CCA-secure commitment schemes are for example defined by Kiyoshima [23]. The one-one CCA-oracle $\mathcal{O}_{\mathsf{1cca}}$ is defined like the CCA-oracle, except that the adversary is restricted to a single query consisting of exactly one commitment. Let $\mathsf{Exp}^{\mathsf{1cca}}_{\mathsf{Com},\mathcal{A}}(k)$ define the output of the security game for one-one CCA-security (1CCA). The formal definition is then analogous to the definition of CCA-security.

q-bounded CCA-Secure Commitment Schemes. The q-bounded CCA-oracle $\mathcal{O}_{q\mathsf{cca}}$ is defined like the CCA-oracle, except that the adversary is restricted to $q \in \mathbb{N}$ queries where each query consists of exactly one commitment. Let $\mathsf{Exp}^{q\mathsf{cca}}_{\mathsf{Com},\mathcal{A}}(k)$ define the output of the security game for q-bounded CCA-security (qCCA). The formal definition is then analogous to the definition of CCA-security. Note that by definition 1-bounded CCA-security equals one-one CCA-security.

Non-Malleable Commitment Schemes. We now specify a definition of non-malleable commitment schemes that is essentially a game-based variant of the definition by Goyal, Pandey and Richelson [20]. It is easy to see that the two definitions are equivalent. Using a game-based variant of [20] makes it easier to compare this notion with CCA-security..

Let $\mathsf{Exp}^{\mathsf{nm}}_{\mathsf{Com},\mathcal{A},\mathcal{D}}(k)$ denote the output of the following probabilistic experiment: On input $1^k, z$, the adversary $\mathcal{A}$ picks a tag tag and two strings v_0 and v_1 with $|v_0| = |v_1|$, sends this triple to the sender and gets back the challenge commitment $\mathsf{Com}_{tag}(v_b)$, where b is a random bit chosen by the sender. The adversary then sends a commitment $\mathsf{Com}_{\widetilde{tag}}(\tilde{v}_b)$ to the receiver. If $\widetilde{tag} = tag$, $\tilde{v}_b$ is set to $\perp$. At the end of this interaction the adversary outputs his view $view_{\mathcal{A}}$ and the receiver outputs the value $\tilde{v}_b$. Note that the experiment plays the role of the sender and the receiver in the interaction. Also note that the receiver R has implicit access to a super-polynomial-time oracle $\mathcal{O}$ that breaks the received commitment for him and that the adversary's view contains the randomness of the adversary and a transcript of all messages sent and received by the adversary. After the interaction has finished, the distinguisher $\mathcal{D}$ gets z, the view $view_{\mathcal{A}}$ of the adversary and the value $\tilde{v}_b$ as input and outputs a bit b'. The experiment outputs 1 if $b = b'$ and 0 otherwise.

Definition 2 (Non-malleable commitment scheme). *A commitment scheme* Com *is* non-malleable *if for every* PPT *man-in-the-middle adversary* $\mathcal{A}$*, for every* PPT *distinguisher* $\mathcal{D}$ *and all* $z \in \{0,1\}^*$ *the following holds:*

$$\mathsf{Adv}^{\mathsf{nm}}_{\mathsf{Com},\mathcal{A}(z),\mathcal{D}}(k) := \mathsf{Pr}[\, \mathsf{Exp}^{\mathsf{nm}}_{\mathsf{Com},\mathcal{A}(z),\mathcal{D}}(k) = 1\,] - \frac{1}{2} \leq \mathsf{negl}(k)$$

for a negligible function negl.

Concurrent Non-Malleable Commitment Schemes. Tag-based concurrently non-malleable commitment schemes are examined by Lin, Pass and Venkitasubramaniam [26]. Here, man-in-the-middle adversaries are participating in left and right interactions in which $m = poly(k)$ commitments take place (where $k \in \mathbb{N}$ is the security parameter).

In the concurrent setting, the adversary $\mathcal{A}$ is simultaneously participating in m left and right interactions. He sends a triple of sequences $(\boldsymbol{tag}, \boldsymbol{v}^0, \boldsymbol{v}^1)$ with $\boldsymbol{tag} = (tag_1, \ldots, tag_m)$, $\boldsymbol{v}^0 = (v^0_1, \ldots, v^0_m)$ and $\boldsymbol{v}^1 = (v^1_1, \ldots, v^1_m)$ to the sender and receives commitments to values $v^b_1, \ldots, v^b_m$ with tags $tag_1, \ldots, tag_m$

from the sender S and commits to values $\tilde{v}_1^b, \ldots, \tilde{v}_m^b$ with tags $\widetilde{tag}_1, \ldots, \widetilde{tag}_m$ to the receiver R. For any i such that $\widetilde{tag}_i = tag_j$ for some j, set $\tilde{v}_i^b = \bot$. Let $\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com},\mathcal{A},\mathcal{D}}(k)$ define the output of the security game for concurrent non-malleability (CNM). The formal definition is then analogous to the definition of non-malleability.

Parallel Non-Malleable Commitment Schemes. A relaxed notion of concurrent non-malleability is parallel non-malleability [16]. Here, like for concurrent non-malleability, the adversary receives m commitments from the sender and sends m commitments to the receiver. However, for parallel non-malleability the commitments are always sent in parallel. Again, any commitment in the right interaction that uses a tag that is also present in the left interaction, is considered invalid. Let $\mathsf{Exp}^{\mathsf{pnm}}_{\mathsf{Com},\mathcal{A},\mathcal{D}}(k)$ define the output of the security game for parallel non-malleability (PNM). The formal definition is then analogous to the definition of non-malleability.

$\mathcal{O}$-One-Way Commitment Schemes. Informally speaking, a tag-based commitment scheme Com with message space $\{0,1\}^k$ and tag space $\{0,1\}^k$ is said to be $\mathcal{O}$-one-way, if no PPT-adversary can break a commitment to a random value, even with access to the oracle $\mathcal{O}$. The property can be formally defined with a security game. Let $\mathsf{Exp}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{A},\mathcal{O}}(k)$ denote the output of the following probabilistic experiment: The experiment generates a random value v and a random tag tag, i.e., $v \overset{\$}{\leftarrow} \{0,1\}^k$, $tag \overset{\$}{\leftarrow} \{0,1\}^k$. It then sends the commitment $\mathsf{Com}_{tag}(v)$ as challenge to the PPT-adversary $\mathcal{A}^{\mathcal{O}}$. On input 1^k, z, the adversary now tries to break the commitment and sends at some time his solution v' back to the experiment which outputs 1 if $v = v'$ and 0 otherwise. Note that during the entire course of the game the adversary has access to the oracle $\mathcal{O}$. The output of the experiment is replaced by $\bot$ if during the execution the adversary queries the oracle on a commitment that uses the challenge tag tag.

Definition 3 ($\mathcal{O}$-one-way commitment scheme). *Let* Com *be a tag-based commitment scheme and* $\mathcal{O}$ *be a specific oracle for it. We say that* Com *is* $\mathcal{O}$-one-way*, if for every* PPT*-adversary* $\mathcal{A}$ *and all* $z \in \{0,1\}^*$ *the following holds:*

$$\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{A}(z),\mathcal{O}}(k) := \mathsf{Pr}[\,\mathsf{Exp}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{A}(z),\mathcal{O}}(k) = 1\,] \leq \mathsf{negl}(k)$$

for a negligible function negl.

This definition can be instantiated with various oracles. For example, $\mathcal{O}_{\mathsf{cca}}$-one-wayness describes a security notion where the one-way adversary has access to the CCA-oracle for the commitment scheme in question. Note that CCA-security implies $\mathcal{O}_{\mathsf{cca}}$-one-wayness. Similarly, parallel CCA-security implies $\mathcal{O}_{\mathsf{pcca}}$-one-wayness, one-one CCA-security implies $\mathcal{O}_{\mathsf{1cca}}$-one-wayness and q-bounded CCA-security implies $\mathcal{O}_{q\mathsf{cca}}$-one-wayness. Also note that non-malleability (and its stronger variants) implies ε-one-wayness for the empty oracle ε. Note that the empty oracle just returns $\bot$ for each query.

Extractable Commitment Schemes. Finally, we define extractable commitment schemes:

Definition 4 (Extractable Commitment Scheme). *Let* Com *be a statistically binding commitment scheme. Then,* Com *is* extractable *if there exists a* PPT *oracle machine E (the "extractor") such that for any* PPT *committer* $\mathcal{C}^*$, $E^{\mathcal{C}^*}$ *outputs a pair* (τ, σ) *such that*

- *τ is identically distributed to the view of $\mathcal{C}^*$ at the end of interacting with an honest receiver R in the commit phase.*
- *the probability that τ is accepting and $\sigma \neq \bot$ is negligible.*
- *if $\sigma \neq \bot$, then it is statistically impossible to decommit τ to any value other than σ.*

3 The First Transformation: Puzzle-Solution Approach

In this section, we describe the first transformation in this work. We call this approach the *puzzle-solution approach* because the general idea is to expand a commitment scheme by a puzzle phase that is executed at the beginning. Let X and Y be security notions for commitment schemes for which one wants to show that X does not imply Y. For the first transformation, Y will always be a CCA-related security notion. Let $\mathcal{O}_X$ be the oracle an adversary can use in the security game for the notion X. Let analogously $\mathcal{O}_Y$ be the oracle an adversary can use in the security game for the notion Y. (note that these oracles can be the "empty oracle"). Let Com be a (possibly interactive) commitment scheme that fulfills X. We will sometimes call Com the base commitment scheme.

3.1 The Construction

Using Com, one can then define the separating commitment scheme, which we will denote by Com'. We define Com' as output of a transformation $\mathsf{PComGen}$ that gets a base commitment scheme, a number $l \in \mathbb{N}$ and a string $\mathsf{sch} \in \{seq, par\}$ as input, i.e., $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, l, \mathsf{sch})$.

In the commitment scheme Com' the sender S, who wants to commit to a value v given a tag tag, first sends a puzzle to the receiver R and, depending on whether R solves the puzzle or not, sends v either as plaintext or commits to v using the base commitment scheme Com. The puzzle consists of l commitments to random messages (using Com) that are either sent in parallel (if $\mathsf{sch} = par$) or sequentially (if $\mathsf{sch} = seq$) to R. More specifically, the sender randomly generates l tags of length k and l values also of length k, i.e., $(tag_p^1, \ldots, tag_p^l) \overset{\$}{\leftarrow} \left(\{0,1\}^k\right)^l$, $(w_1, \ldots, w_l) \overset{\$}{\leftarrow} \left(\{0,1\}^k\right)^l$.

If $\mathsf{sch} = par$, the sender commits in *parallel* to $(w_1, \ldots, w_l)$ under the tags $(tag_p^1, \ldots, tag_p^l)$ to the receiver. The receiver then answers with a possible solution to the puzzle by simply guessing, i.e., sending random $(w'_1, \ldots, w'_l)$. The sender then checks if for all $i \in \{1, \ldots, l\}$ it holds that $w_i = w'_i$. If this is the case, S sends v as plaintext to the receiver. If it does not hold, S commits to v using the tag tag and the commitment scheme Com to R.

If $\mathsf{sch} = seq$, the sender *sequentially* commits to $(w_1, \ldots, w_l)$ under the tags $(tag_p^1, \ldots, tag_p^l)$ to the receiver. More specifically, he first commits to w_1 using the tag tag_p^1 and the commitment scheme Com and waits for the possible solution. The receiver R then sends a random value w_1' to S. If the solution is incorrect, then S commits to v using the tag tag and the base commitment scheme Com to R. Otherwise, he continues the puzzle phase by sending the second puzzle commitment, i.e., $\mathsf{Com}_{tag_p^2}(w_2)$, to R and again waits for the possible solution. The receiver R then sends another random value w_2' to S. If the solution is incorrect, then S commits to v using the tag tag and the commitment scheme Com. Otherwise, he continues by sending the third puzzle commitment and so forth. If R has correctly solved all l puzzle commitments, S sends v as plaintext to the receiver.

Remark 1. When designing the separating commitment scheme, l and sch should be carefully picked. The puzzle should be selected in such a way that it can be solved with $\mathcal{O}_Y$ but not with $\mathcal{O}_X$.

3.2 The Proof Strategy

To prove that X does not imply Y, one shows that the constructed commitment scheme Com' still fulfills X if the base commitment scheme Com fulfills X, but not Y.

Show that Com' is not Y-secure. For that purpose, one constructs an adversary $\mathcal{A}$, who breaks the Y-security of Com'. The strategy for $\mathcal{A}$ is to let $\mathcal{O}_Y$ solve the puzzle for him. He then gets the challenge value as plaintext and can thus trivially win in the security game for Y.

The probability that $\mathcal{A}$ wins the game is overwhelming, because the only possibilities how $\mathcal{A}$ can lose are: 1) the oracle solves the puzzle it gets before the query, 2) a session with the oracle has multiple valid committed values and $\mathcal{O}_Y$ thus returns $\perp$, 3) during the execution the adversary queries the oracle on a commitment that uses the challenge tag (which happens if a puzzle commitment uses the challenge tag). Since one can show that each possibility occurs only with negligible probability, the overall winning probability of $\mathcal{A}$ is overwhelming.

Show that Com' is X-secure (under the assumption that Com is X-secure). Let $\mathcal{A}$ be an adversary on Com' in the security game for X, who wins the game with non-negligible advantage. Depending whether or not $\mathcal{A}$ solves at least one puzzle[3] in the security game for X, one has to distinguish two cases. For each case one builds an adversary who breaks the X-security of the commitment scheme Com.

Case 1: $\mathcal{A}$ solves at least one puzzle. In this case, one constructs an adversary $\mathcal{B}_1$ on the $\mathcal{O}_X$-one-wayness of Com. Recall that X-security implies $\mathcal{O}_X$-one-wayness for our cases. We denote by n the number of challenge commitments $\mathcal{A}$ awaits. Since each of the n corresponding puzzles contains l commitments,

[3] Note that for example in the concurrent non-malleability security game multiple puzzles (with $l = 1$ for each puzzle) are sent (one for each session).

$\mathcal{A}$ expects in total $m = l \cdot n$ puzzle commitments. The strategy of $\mathcal{B}_1$ is then first to randomly generate $m - 1$ puzzle values and tags and to randomly select $j \in \{1, \ldots, m\}$. After $\mathcal{B}_1$ hast received the challenge $\mathsf{Com}_{tag}(v)$ from the experiment, he starts to send $\mathcal{A}$ the puzzle(s). For all puzzle commitments except the j^{th} he uses the honestly generated values and tags. As j^{th} puzzle commitment, he uses the challenge. After $\mathcal{A}$ has sent the solution to the j^{th} puzzle commitment (aka the challenge), $\mathcal{B}_1$ terminates the simulation of $\mathcal{A}$ and sends $\mathcal{A}$'s solution to the j^{th} puzzle commitment as his own solution to the experiment.

If $\mathcal{A}$ asks his oracle $\mathcal{O}_X$ during the game, $\mathcal{B}_1$ sends random answers in the puzzle phase (to simulate the oracle) and forwards the actual oracle query to his own $\mathcal{O}_X$. There is a chance that $\mathcal{B}_1$'s experiment returns $\perp$ at the end of the experiment. This happens if one of $\mathcal{A}$'s oracle queries contains a tag that equals $\mathcal{B}_1$'s challenge tag. This case may occur with non-negligible probability because the challenge tags of $\mathcal{A}$ and $\mathcal{B}_1$ are not necessarily identical. Fortunately, the opposite event also occurs with non-negligible probability.

The adversary $\mathcal{B}_1$ thus wins his game if $\mathcal{A}$ solves the puzzle commitment that is the challenge and $\mathcal{A}$'s oracle queries do not involve the challenge tag.

Case 2: $\mathcal{A}$ solves none of the puzzles. In this case one builds an adversary $\mathcal{B}_2$ on the X-security of Com. The strategy of $\mathcal{B}_2$ is to send random puzzle(s) to $\mathcal{A}$, who fails to solve them (by assumption). After the puzzle phase, $\mathcal{B}_2$ forwards his own challenge to $\mathcal{A}$. The adversary $\mathcal{B}_2$ also forwards $\mathcal{A}$'s solution as his own solution to the experiment.

If $\mathcal{A}$ asks his oracle $\mathcal{O}_X$ during the game, $\mathcal{B}_2$ sends random answers in the puzzle phase (to simulate the oracle) and forwards the actual oracle query to his own $\mathcal{O}_X$. Here, the challenge tags of $\mathcal{A}$ and $\mathcal{B}_2$ are always identical (because $\mathcal{B}_2$ forwards it to his experiment), so the possibility of $\mathcal{B}_2$'s experiment outputting $\perp$ is not a problem in this case.

The adversary $\mathcal{B}_2$ thus wins his game if $\mathcal{A}$ wins his own game and solves no puzzle.

4 A Concrete Example of the Puzzle Solution Approach: Concurrent Non-Malleability Does Not Imply CCA-Security

In this section, we apply the puzzle-solution approach to separate the notion of CCA-security from the notion of concurrent non-malleability.[4] To this end, we define Com' as $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 1, seq)$ where Com is a statistically binding, concurrently non-malleable commitment scheme. The puzzle hence consists of just one commitment (thus the scheduling does not matter in this case). We follow the proof strategy described in Sec. 3.

[4] While the separation of CCA-security from concurrent non-malleability is not very surprising, we have nonetheless chosen to give a full proof for this separation. This is because this proof is one of the easier applications of our puzzle-solution approach and therefore (hopefully) a good example for the reader.

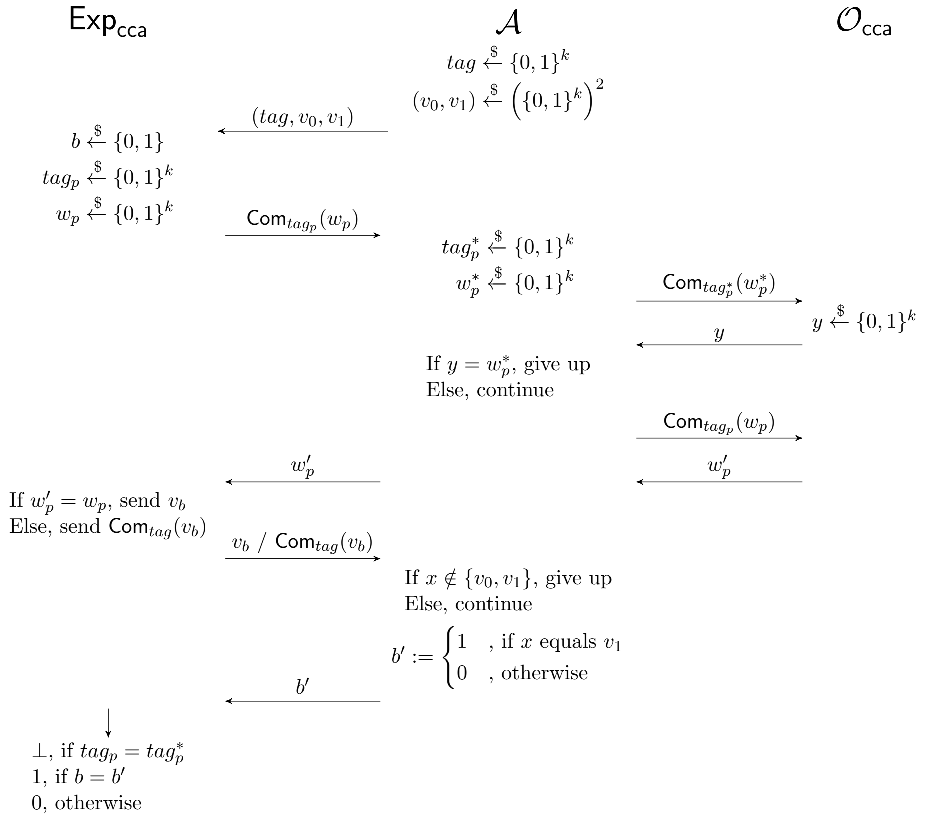
$\mathsf{Exp}_{\mathsf{cca}}$
$\mathcal{A}$
$\mathcal{O}_{\mathsf{cca}}$
$tag \xleftarrow{\$} \{0,1\}^k$
$(v_0, v_1) \xleftarrow{\$} \left(\{0,1\}^k\right)^2$
(tag, v_0, v_1)
$b \xleftarrow{\$} \{0,1\}$
$tag_p \xleftarrow{\$} \{0,1\}^k$
$w_p \xleftarrow{\$} \{0,1\}^k$
$\mathsf{Com}_{tag_p}(w_p)$
$tag_p^* \xleftarrow{\$} \{0,1\}^k$
$w_p^* \xleftarrow{\$} \{0,1\}^k$
$\mathsf{Com}_{tag_p^*}(w_p^*)$
$y \xleftarrow{\$} \{0,1\}^k$
y
If $y = w_p^*$, give up
Else, continue
$\mathsf{Com}_{tag_p}(w_p)$
w'_p
w'_p
If $w'_p = w_p$, send v_b
Else, send $\mathsf{Com}_{tag}(v_b)$
v_b / $\mathsf{Com}_{tag}(v_b)$
If $x \notin \{v_0, v_1\}$, give up
Else, continue
$b' := \begin{cases} 1 & \text{, if } x \text{ equals } v_1 \\ 0 & \text{, otherwise} \end{cases}$
b'
$\bot$, if $tag_p = tag_p^*$
1, if $b = b'$
0, otherwise

Fig. 2. Graphical depiction of the behavior of the adversary $\mathcal{A}$ in the CCA-security game for the commitment scheme Com'. Note that $w'_p \in \{w_p, \bot\}$ is either the unique committed value w_p or, if the commitment has more than one valid opening, $\bot$.

Theorem 1 (CNM $\not\Rightarrow$ CCA). *If* Com *is a statistically binding, concurrently non-malleable commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 1, seq)$ *is also statistically binding and concurrently non-malleable but not CCA-secure.*

Proof. The statistical binding property of Com' follows readily from the statistical binding property of the underlying commitment scheme Com. In the following, we prove that Com' is concurrently non-malleable but not CCA-secure.[5]

Claim 1: Com' is not CCA-secure. We show that we can build a CCA-adversary $\mathcal{A}$, such that $\mathcal{A}$ wins the CCA-security game for the commitment scheme Com' with non-negligible advantage.

[5] For ease of notation, we omit the (non-uniform) input z of the adversary and distinguisher. The proof can be easily adapted to include this input.

The CCA-adversary $\mathcal{A}$ acts as depicted in Fig. 2. His strategy is to let the oracle solve the puzzle he got from the experiment and to hence get the challenge as plaintext. There are three possibilities how $\mathcal{A}$ can lose the game:

- the oracle solves the puzzle, i.e., $y = w_p^*$
- the puzzle tag equals the challenge tag, i.e., $tag = tag_p$ (in that case the experiment returns $\perp$ as result instead of a bit)
- the query sent to the oracle has more than one valid opening (in that case the oracle returns $w'_p = \perp$)

The first possibility occurs with probability $1/2^k$ because the oracle uniformly selects a solution. The second possibility also occurs with probability $1/2^k$ because the puzzle tag is uniformly selected. The third possibility occurs with negligible probability, which we denote by $\mathsf{negl}_1(k)$, because Com is by assumption statistically binding. Thus, $\mathcal{A}$'s advantage is non-negligible:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cca}}_{\mathsf{Com}',\mathcal{A}}(k) &= \Pr[\mathsf{Exp}^{\mathsf{cca}}_{\mathsf{Com}',\mathcal{A}}(k) = 1] - \frac{1}{2} \\ &\geq 1 - \frac{1}{2^k} - \frac{1}{2^k} - \mathsf{negl}_1(k) - \frac{1}{2} \\ &= \frac{1}{2} - \frac{1}{2^{k-1}} - \mathsf{negl}_1(k)\end{aligned}$$

Claim 2: Com' is concurrently non-malleable. Let us assume Com' is not concurrently non-malleable. Then we show that Com is also not concurrently non-malleable. Consider an adversary $\mathcal{A}$ and distinguisher $\mathcal{D}_\mathcal{A}$ such that $\mathcal{A}$ wins in the concurrent non-malleability security game for the commitment scheme Com' with advantage $\mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_\mathcal{A}}(k)$. Let $m = poly(k)$, where k is the security parameter, be the number of concurrent commitment sessions initiated by the sender in the concurrent non-malleability security game for Com'. Then we can split-up $\mathcal{A}$'s advantage into

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_\mathcal{A}}(k) &= \Pr[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_\mathcal{A}}(k) = 1 \wedge \exists i : \mathcal{A} \text{ solves puzzle } i] \\ &\quad + \Pr[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_\mathcal{A}}(k) = 1 \wedge \nexists i : \mathcal{A} \text{ solves puzzle } i] - \frac{1}{2}\end{aligned} \tag{1}$$

Hence, in the following we can restrict to consider that $\mathcal{A}$ wins and

- Case 1: $\mathcal{A}$ solves at least one of the m puzzles.
- Case 2: $\mathcal{A}$ solves none of the m puzzles.

Case 1: $\mathcal{A}$ solves at least one of the m puzzles. Using $\mathcal{A}$ we construct an adversary $\mathcal{B}_1$ against the ε-one-wayness (for the empty oracle ε) of the commitment scheme Com. The adversary $\mathcal{B}_1$ acts as depicted in Fig. 3 in the ε-one-way security game for the commitment scheme Com. His strategy is to mimic the experiment for $\mathcal{A}$ in the concurrent non-malleability security game and to replace a random puzzle commitment with the challenge he got from his own experiment. Note that depending on the behavior of $\mathcal{A}$, it may at some time happen that $\mathcal{A}$ sends

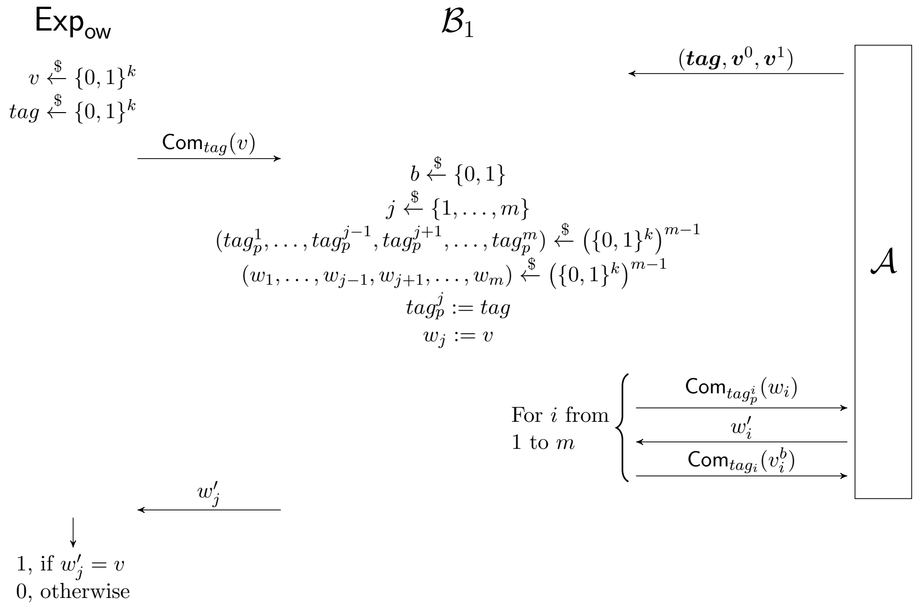
$\mathsf{Exp}_{\mathsf{ow}}$
$\mathcal{B}_1$
$(\boldsymbol{tag}, \boldsymbol{v}^0, \boldsymbol{v}^1)$
$v \xleftarrow{\$} \{0,1\}^k$
$tag \xleftarrow{\$} \{0,1\}^k$
$\mathsf{Com}_{tag}(v)$
$b \xleftarrow{\$} \{0,1\}$
$j \xleftarrow{\$} \{1,\ldots,m\}$
$(tag_p^1,\ldots,tag_p^{j-1},tag_p^{j+1},\ldots,tag_p^m) \xleftarrow{\$} \left(\{0,1\}^k\right)^{m-1}$
$(w_1,\ldots,w_{j-1},w_{j+1},\ldots,w_m) \xleftarrow{\$} \left(\{0,1\}^k\right)^{m-1}$
$tag_p^j := tag$
$w_j := v$
$\mathcal{A}$
For i from 1 to m
$\mathsf{Com}_{tag_p^i}(w_i)$
w_i'
$\mathsf{Com}_{tag_i}(v_i^b)$
w_j'
1, if $w_j' = v$
0, otherwise

Fig. 3. Graphical depiction of the behavior of the adversary $\mathcal{B}_1$ in the ε-one-way security game for the commitment scheme Com. Note that $\boldsymbol{tag} = (tag_1, \ldots, tag_m)$, $\boldsymbol{v}^0 = (v_1^0, \ldots, v_m^0)$ and $\boldsymbol{v}^1 = (v_1^1, \ldots, v_m^1)$.

a puzzle to who he believes is the receiver, but is actually $\mathcal{B}_1$. If $\mathcal{B}_1$ receives such a puzzle $\mathsf{Com}_{\widetilde{tag}_p^i}(\tilde{w}_i)$ from $\mathcal{A}$, he acts as a honest receiver and sends a random solution $\tilde{w}_i'$ back. The time of $\mathcal{A}$'s interaction with the "receiver" or the contents of the puzzle do not matter in this case, therefore this interaction is omitted in Fig. 3.

By construction, $\mathcal{B}_1$ wins the game if v equals w_j', which happens if $\mathcal{A}$ correctly solves the j^{th} puzzle. Thus, the advantage of $\mathcal{B}_1$ is as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{B}_1,\varepsilon}(k) &= \mathsf{Pr}[\mathsf{Exp}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{B}_1,\varepsilon}(k) = 1] \\
&\geq \mathsf{Pr}[\mathsf{Exp}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{B}_1,\varepsilon}(k) = 1 \mid \exists i : \mathcal{A} \text{ solves puzzle } i] \\
&\quad \cdot \mathsf{Pr}[\exists i : \mathcal{A} \text{ solves puzzle } i] \\
&\geq \frac{1}{m} \cdot \mathsf{Pr}[\exists i : \mathcal{A} \text{ solves puzzle } i] \\
&\geq \frac{1}{m} \cdot \mathsf{Pr}[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_{\mathcal{A}}}(k) = 1 \wedge \exists i : \mathcal{A} \text{ solves puzzle } i]
\end{aligned} \tag{2}$$

Case 2: $\mathcal{A}$ solves none of the m puzzles. Using $\mathcal{A}$, we construct an adversary $\mathcal{B}_2$ against the concurrent non-malleability property of the commitment scheme Com. For each $i \in \{1, \ldots, m\}$, $\mathcal{B}_2$ sends an honestly generated puzzle to $\mathcal{A}$ (thereby simulating the sender), who fails to solve it, and then forwards the i^{th} commitment he gets from the sender to $\mathcal{A}$. When $\mathcal{A}$ interacts with his receiver,

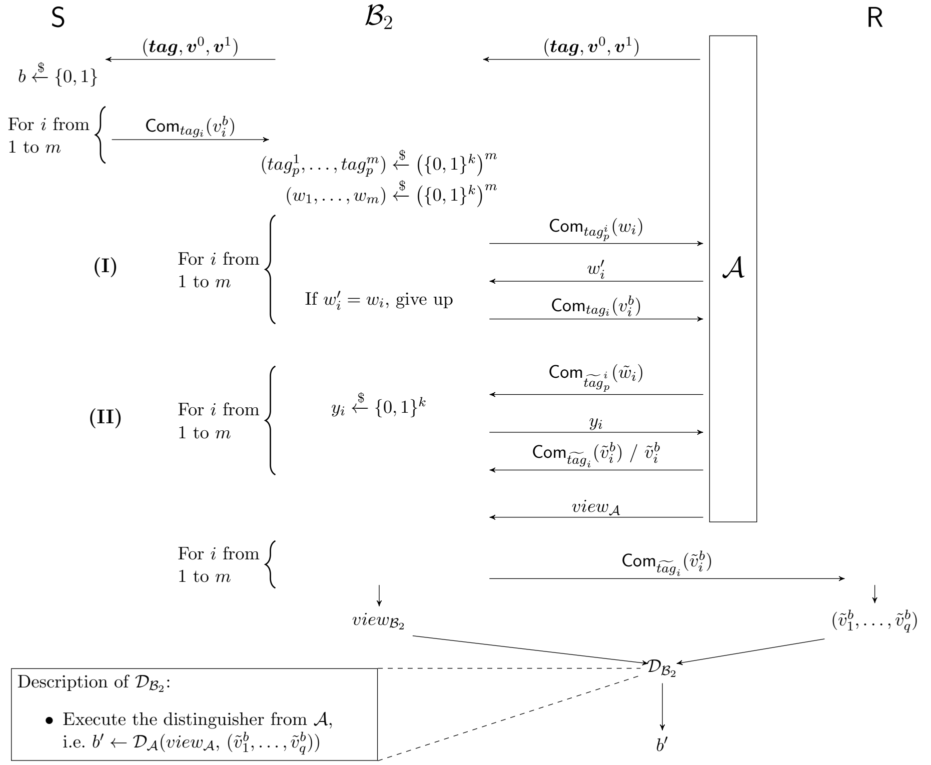
S
$\mathcal{B}_2$
R
$(\boldsymbol{tag}, \boldsymbol{v}^0, \boldsymbol{v}^1)$
$(\boldsymbol{tag}, \boldsymbol{v}^0, \boldsymbol{v}^1)$
$b \xleftarrow{\$} \{0,1\}$
For i from 1 to m
$\mathsf{Com}_{tag_i}(v_i^b)$
$(tag_p^1, \dots, tag_p^m) \xleftarrow{\$} (\{0,1\}^k)^m$
$(w_1, \dots, w_m) \xleftarrow{\$} (\{0,1\}^k)^m$
(I)
For i from 1 to m
$\mathsf{Com}_{tag_p^i}(w_i)$
w_i'
If $w_i' = w_i$, give up
$\mathsf{Com}_{tag_i}(v_i^b)$
$\mathcal{A}$
(II)
For i from 1 to m
$\mathsf{Com}_{\widetilde{tag}_p^i}(\tilde{w}_i)$
$y_i \xleftarrow{\$} \{0,1\}^k$
y_i
$\mathsf{Com}_{\widetilde{tag}_i}(\tilde{v}_i^b)$ / $\tilde{v}_i^b$
$view_{\mathcal{A}}$
For i from 1 to m
$\mathsf{Com}_{\widetilde{tag}_i}(\tilde{v}_i^b)$
$view_{\mathcal{B}_2}$
$(\tilde{v}_1^b, \dots, \tilde{v}_q^b)$
$\mathcal{D}_{\mathcal{B}_2}$
Description of $\mathcal{D}_{\mathcal{B}_2}$:
• Execute the distinguisher from $\mathcal{A}$, i.e. $b' \leftarrow \mathcal{D}_{\mathcal{A}}(view_{\mathcal{A}}, (\tilde{v}_1^b, \dots, \tilde{v}_q^b))$
b'

Fig. 4. Graphical depiction of the behavior of the adversary $\mathcal{B}_2$ in the concurrent non-malleability security game for the commitment scheme Com. At **(I)** $\mathcal{A}$'s interaction with the "sender" is depicted and at **(II)** $\mathcal{A}$'s interaction with the "receiver". Note that $\boldsymbol{tag} = (tag_1, \dots, tag_m)$, $\boldsymbol{v}^0 = (v_1^0, \dots, v_m^0)$ and $\boldsymbol{v}^1 = (v_1^1, \dots, v_m^1)$. Note that $\mathsf{Com}_{\widetilde{tag}_i}(\tilde{v}_i^b)$ / $\tilde{v}_i^b$ denotes that, depending on whether $\mathcal{B}_2$ correctly guessed the solution y_i or not, the i^{th} result value is sent as a commitment or as a plaintext value. In the (negligible) case that $\mathcal{B}_2$ correctly solves a puzzle and gets a value $\tilde{v}_i$ as plaintext, he himself commits to this value before sending the commitment to the receiver. Also note that $view_{\mathcal{B}_2}$ contains $view_{\mathcal{A}}$.

who is simulated by $\mathcal{B}_2$, $\mathcal{B}_2$ answers randomly in the puzzle phases (to simulate an honest receiver) and forwards the commitments from $\mathcal{A}$ to his own receiver (cf. Fig. 4).

The advantage of $\mathcal{B}_2$ in this case is as follows:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com},\mathcal{B}_2,\mathcal{D}_{\mathcal{B}_2}}(k) &= \mathsf{Pr}[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com},\mathcal{B}_2,\mathcal{D}_{\mathcal{B}_2}}(k) = 1] - \frac{1}{2} \\
&\geq \mathsf{Pr}[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com},\mathcal{B}_2,\mathcal{D}_{\mathcal{B}_2}}(k) = 1 \mid \nexists i : \mathcal{A} \text{ solves puzzle } i] \\
&\quad \cdot \mathsf{Pr}[\nexists i : \mathcal{A} \text{ solves puzzle } i] - \frac{1}{2} \\
&= \mathsf{Pr}[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_{\mathcal{A}}}(k) = 1 \mid \nexists i : \mathcal{A} \text{ solves puzzle } i] \\
&\quad \cdot \mathsf{Pr}[\nexists i : \mathcal{A} \text{ solves puzzle } i] - \frac{1}{2} \\
&= \mathsf{Pr}[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_{\mathcal{A}}}(k) = 1 \wedge \nexists i : \mathcal{A} \text{ solves puzzle } i] - \frac{1}{2}
\end{aligned} \tag{3}$$

Putting things together. Putting Eq. 2 and Eq. 3 back into Eq. 1, we get the following:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_{\mathcal{A}}}(k) &= \mathsf{Pr}[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_{\mathcal{A}}}(k) = 1 \wedge \exists i : \mathcal{A} \text{ solves puzzle } i] \\
&\quad + \mathsf{Pr}[\mathsf{Exp}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_{\mathcal{A}}}(k) = 1 \wedge \nexists i : \mathcal{A} \text{ solves puzzle } i] - \frac{1}{2} \\
&\leq m \cdot \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{B}_1,\varepsilon}(k) + \mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com},\mathcal{B}_2,\mathcal{D}_{\mathcal{B}_2}}(k) + \frac{1}{2} - \frac{1}{2} \\
&= m \cdot \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{B}_1,\varepsilon}(k) + \mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com},\mathcal{B}_2,\mathcal{D}_{\mathcal{B}_2}}(k)
\end{aligned}$$

Since Com is by assumption concurrently non-malleable, it holds that $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{Com},\mathcal{B}_1,\varepsilon}(k)$ and $\mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com},\mathcal{B}_2,D_{\mathcal{B}_2}}(k)$ are negligible. Thus, $\mathsf{Adv}^{\mathsf{cnm}}_{\mathsf{Com}',\mathcal{A},\mathcal{D}_{\mathcal{A}}}(k)$ is also negligible, which concludes the proof of the theorem. □

5 More Instantiations of the Puzzle-Solution Approach

In this section, we show how more separation results can be obtained by appropriate instantiations of the puzzle-solution approach. Therefore we illustrate how the puzzle-solution approach from Sec. 3 should be instantiated to show the respective result.

Using the same puzzle and very similar arguments as in the proof of theorem Thm. 1, one can proof that parallel non-malleability does not imply parallel CCA-security, that non-malleability does not imply one-one CCA-security, that concurrent non-malleability does not imply parallel CCA-security and that parallel non-malleability does not imply one-one CCA-security.

Theorem 2 (PNM $\not\Rightarrow$ PCCA). *If* Com *is a statistically binding, parallel non-malleable commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 1, \mathit{seq})$ *is also statistically binding and parallel non-malleable but not parallel CCA-secure.*

Theorem 3 (NM $\not\Rightarrow$ 1CCA). *If* Com *is a statistically binding, non-malleable commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 1, \mathit{seq})$ *is also statistically binding and non-malleable but not one-one CCA-secure.*

Theorem 4 (CNM $\not\Rightarrow$ PCCA). *If* Com *is a statistically binding, concurrent non-malleable commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 1, \mathit{seq})$ *is also statistically binding and concurrent non-malleable but not parallel CCA-secure.*

Theorem 5 (PNM $\not\Rightarrow$ 1CCA). *If* Com *is a statistically binding, parallel non-malleable commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 1, \mathit{seq})$ *is also statistically binding and parallel non-malleable but not one-one CCA-secure.*

We can proof additional separations using other puzzles.

Theorem 6 (1CCA $\not\Rightarrow$ PCCA). *If* Com *is a statistically binding, one-one CCA-secure commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 2, \mathit{par})$ *is also statistically binding and one-one CCA-secure but not parallel CCA-secure.*

Proof Idea. The puzzle consists of two parallel commitments. It is thus solvable with a parallel CCA-oracle but not with a one-one CCA-oracle. The probability that in the reduction of the first case of the second claim the oracle query can be answered is at least $1/2 - 1/2^k$ (with k the tag length). □

Theorem 7 (PCCA $\not\Rightarrow$ CCA). *If* Com *is a statistically binding, parallel CCA-secure commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, 2, \mathit{seq})$ *is also statistically binding and parallel CCA-secure but not CCA-secure.*

Proof Idea. The puzzle consists of two sequentially sent commitments. It it thus solvable with a CCA-oracle but not with a parallel CCA-oracle. The probability that in the reduction of the first case of the second claim the oracle query can be answered is at least $1/2 - m/2^k$ (with m the number of commitments in the oracle query and k the tag length). □

Theorem 8 (qCCA $\not\Rightarrow$ $(q+1)$CCA). *Let* $q \geq 1$ *be a positive integer. If* Com *is a statistically binding, q-bounded CCA-secure commitment scheme, then* $\mathsf{Com}' \leftarrow \mathsf{PComGen}(\mathsf{Com}, q+1, \mathit{seq})$ *is also statistically binding and q-bounded CCA-secure but not* $(q+1)$*-bounded CCA-secure.*

Proof Idea. The puzzle consists of $q+1$ sequentially sent commitments. It it thus solvable with a $(q+1)$-bounded CCA-oracle but not with a q-bounded CCA-oracle. The probability that in the reduction of the first case of the second claim the oracle query can be answered is at least $1/q+1 - 1/2^k$ (with k the tag length). □

6 The Second Transformation: Sharing Approach

In this section, we settle the remaining separations. Up to now we have been able to prove our separations using the puzzle-solution approach. However, in order to prove the remaining separations, we cannot use the puzzle-solution approach anymore. This is because we need to construct commitment schemes that do not fulfill a certain *variant of non-malleability* for the remaining separations. We can therefore no longer insert a puzzle into a given commitment scheme since an adversary (i.e., a man-in-the-middle) in a non-malleability-related experiment does not have a committed value oracle at his disposal that can be used to solve the puzzle.

We will therefore deviate from the puzzle-solution approach in the following way: Instead of sending a puzzle, i.e., commitments to random strings, we let the sender commit to *shares* of the message to be committed to using two different random tags. This way, the man-in-the-middle will be able to forward the commitments to the shares to the receiver in his experiment. After the commit phase is over, these shares will then be opened by the implicit oracle in the experiment. The distinguisher will then be able to reconstruct the message and win in the experiment.

Using the above approach, we will first show that parallel CCA-security does not imply concurrent non-malleability. To this end, consider the following scheme Com', given a commitment scheme Com:

On input $v \in \{0,1\}^k$, $tag \in \{0,1\}^k$, the sender generates message shares $s_0, s_1 \in \{0,1\}^k$ such that $s_0 \oplus s_1 = v$. He then sends $\mathsf{Com}_{tag_0}(s_0)$ and $\mathsf{Com}_{tag_1}(s_1)$ to the receiver in a *sequential* order using random tags $tag_0, tag_1 \in \{0,1\}^k$. Afterwards, the sender sends $\mathsf{Com}_{tag}(v)$ to the receiver. The unveil phase is the same as in Com (notice that the shares are never unveiled).

First note that, in general, the above construction Com' does not yield a separation between concurrent non-malleability and parallel CCA-security, even if Com is parallel CCA-secure. This is because Com' may fulfill *neither* of these two security notions. For instance, assuming Com is *non-interactive*, an adversary against the parallel CCA-security of Com' can simply forward the two commitments to the shares to his oracle and thereby easily win in his experiment.

In order to obtain a separation, we therefore additionally assume that Com is *extractable*. Note that if a statistically binding, parallel CCA-secure commitment scheme exists then there also exists a statistically binding, parallel CCA-secure commitment scheme that is additionally extractable. This is because one-way functions can be constructed from commitment schemes [21] (in a black-box way) and [23] showed how to construct an extractable CCA-secure commitment scheme from one-way functions (in a black-box way).

For the proof of the separation between concurrent non-malleability and parallel CCA-security, we use the following experiment as an auxiliary tool:

Definition 5 (RepeatPCCA). *RepeatPCCA is like the ordinary parallel CCA-security game except that the adversary can "reset" the experiment at any given moment.*

More specifically, the adversary first chooses two strings (v_0, v_1) such that $|v_0| = |v_1|$ and a tag tag *and sends (v_0, v_1, tag) to the experiment. The experiment then chooses a random bit $b \leftarrow \{0,1\}$ and commits to v_b using the tag* tag*. The adversary can then send* `reset` *to the experiment or a bit b'. If the adversary sends* `reset` *then the experiment commits to v_b again using the same tag* tag *as before. The adversary may reset the experiment polynomially many times. If the adversary sends a bit b' then the experiment outputs 1 if $b = b'$ and 0 otherwise.*

Throughout the experiment, the adversary may send a single parallel query to $\mathcal{O}_{\mathsf{pcca}}$ on tags that are different from the challenge tag tag*.*

Denote by $\mathsf{Exp}^{\mathsf{rpcca}}_{\mathsf{Com},\mathcal{A}(z)}(k)$ *the output of the above experiment. We say that a tag-based commitment scheme* Com *is* RepeatPCCA-secure *if for every* PPT*-adversary* $\mathcal{A}$ *and all* $z \in \{0,1\}^*$ *the following holds:*

$$\Pr[\,\mathsf{Exp}^{\mathsf{rpcca}}_{\mathsf{Com},\mathcal{A}(z)}(k) = 1\,] \leq \frac{1}{2} + \mathsf{negl}(k)$$

for a negligible function negl*.*

The following lemma follows from a standard reduction argument:

Lemma 1. *If a commitment scheme is parallel CCA-secure then it is also RepeatPCCA-secure.*

We are now ready to prove the following theorem:

Theorem 9 (PCCA $\not\Rightarrow$ CNM). *If there exists a statistically binding, parallel CCA-secure commitment scheme, then there also exists a statistically binding and parallel CCA-secure commitment scheme that is not concurrently non-malleable.*

Proof. Let Com' be as above with a parallel CCA-secure and extractable commitment scheme Com as its base commitment scheme (as noted above, such a Com exists if a statistically binding, parallel CCA-secure commitment scheme exists).

The statistical binding property of Com' follows readily from the statistical binding property of the underlying commitment scheme Com. In the following, we prove that Com' is parallel CCA-secure but not concurrently non-malleable.[6]

Claim 1: Com' is not concurrently non-malleable. A man-in-the-middle-adversary in the concurrent non-malleability game first sends $\big((v_1^0, \ldots, v_m^0)$, $(v_1^1, \ldots, v_m^1)$, $(tag_1, \ldots, tag_m)\big)$ to the sender, who randomly selects a bit b. The sender then commits for each $i \in \{1, \ldots, m\}$ to the shares $s^b_{i_0}$ and $s^b_{i_1}$ using random tags and to v_i^b using tag tag_i to the adversary (with $s^b_{i_0} \oplus s^b_{i_1} = v_i^b$). Let $h := \lfloor \frac{m}{2} \rfloor$. For each $j \in \{1, \ldots, h\}$ the adversary forwards the commitments to $s^b_{j_0}$ and $s^b_{j_1}$ to the sender (as shares for these commitments he just uses commitments to 0^k).[7] If m is odd, he chooses 0^k as his last message to commit to (he also uses commitments to 0^k as shares). The distinguisher is then given $(s^b_{1_0}, s^b_{1_1}, \ldots, s^b_{h_0}, s^b_{h_1})$ as input (and possibly 0^k) and can thus reconstruct $(v_1^b, \ldots, v_h^b)$, which suffices to deduce the correct b if the challenge messages are chosen appropriately.

[6] For ease of notation, we again omit the (non-uniform) input z of the adversary and distinguisher. The proof can be easily adapted to include this input.

[7] Note that the receiver in Com' does not "examine" the commitments to the shares. This would, of course, not work anyway. Since we assume that the commitment schemes Com is hiding, it is impossible for the receiver to learn the values of the shares by any efficient procedure.

Claim 2: Com' is parallel CCA-secure. Let $\mathcal{A}$ be a PPT-adversary against the parallel CCA-security of Com'. Consider the following hybrids for the commitment scheme Com': H_0 is the ordinary parallel CCA-security game, H_1 is like H_0 except that the sender now commits to two random and *independently distributed* strings s_0, t (that therefore do not fulfill $s_0 \oplus t = v$ in general) and finally H_2 that is like H_1 except that the sender commits to 0^k instead of (his input) v.

Let out_i be the output of the hybrid H_i.

Sub-Claim 1: $|\mathsf{Pr}[out_0 = 1] - \mathsf{Pr}[out_1 = 1]| \leq \mathsf{negl}(k)$. Consider the following adversary $\mathcal{B}$ against Com in the RepeatPCCA game: $\mathcal{B}$ simulates the experiment H_0 for $\mathcal{A}$. After $\mathcal{A}$ has sent (v_0, v_1, tag), $\mathcal{B}$ chooses a random bit $b \leftarrow \{0,1\}$ and generates shares s_0, s_1 such that $s_0 \oplus s_1 = v_b$ and a random string $t \in \{0,1\}^k$. $\mathcal{B}$ then sends (s_1, t, tag_1), where tag_1 is a random tag of length k, to his experiment. $\mathcal{B}$ then randomly selects one of the two (sequentially ordered) commit sessions to the shares of v_b in the commit phase of Com' and inserts his challenge C^* into the selected session and $\mathsf{Com}_{tag_0}(s_0)$ into the other session (for a randomly chosen tag $tag_0 \in \{0,1\}^k$). If the adversary $\mathcal{A}$ starts his (parallel) oracle query *during the challenge phase of* $\mathcal{B}$ (i.e., during the session in which $\mathcal{B}$ has inserted his challenge C^*), then $\mathcal{B}$ resets his experiment and repeats the aforementioned strategy, i.e., randomly inserts the new C^* into one of the two commit sessions (and a new commitment to s_0 into the other session).

Otherwise, $\mathcal{B}$ answers $\mathcal{A}$'s oracle query in the following way:

Case 1: If the challenge C^* was inserted into the *second* session *and* $\mathcal{A}$'s query does not use $\mathcal{B}$'s challenge tag tag_1, then $\mathcal{B}$ forwards $\mathcal{A}$'s queries to his own parallel CCA-oracle (if $\mathcal{A}$'s query uses $\mathcal{B}$'s challenge tag then $\mathcal{B}$ aborts).

Case 2: If the challenge C^* was inserted into the *first* session, then $\mathcal{B}$ answers the query by extracting $\mathcal{A}$ (note that Com is extractable and $\mathcal{B}$'s challenge phase is already over in this case).[8]

Afterwards, $\mathcal{B}$ continues simulating the experiment H_0 for $\mathcal{A}$. After the simulated experiment is over, $\mathcal{B}$ outputs what the simulated experiment outputs. $\mathcal{B}$ repeats the experiment at most $k-1$ times (and aborts if the k^{th} iteration leads to another reset).

Denote by **BadQuery** the event that the adversary $\mathcal{A}$ queries the parallel CCA-oracle during the challenge phase of $\mathcal{B}$ in *all* iterations.

[8] Note that, in general, $\mathcal{B}$ cannot use his own oracle in case 1. This is because, in this case, $\mathcal{A}$ queries his parallel CCA-oracle *after* the first session is over (since we assume that $\mathcal{A}$ has not started his oracle query during $\mathcal{B}$'s challenge phase, which was inserted into the first session in case 1). $\mathcal{A}$ may therefore know *both* tags (tag_0 and tag_1) and query his parallel CCA-oracle using these two tags. Therefore, $\mathcal{B}$ cannot simply forward these queries to his own parallel CCA-oracle since they may contain $\mathcal{B}$'s challenge tag. Furthermore, $\mathcal{B}$ cannot use the extractability property in case 2 since the messages of $\mathcal{A}$'s oracle query and the messages of $\mathcal{B}$'s challenge phase may *overlap* in this case. Hence $\mathcal{B}$ cannot extract $\mathcal{A}$ since this may require "rewinding" the experiment of $\mathcal{B}$ to a specific point in $\mathcal{B}$'s challenge phase.

Let $j \in \{1, 2\}$ be the session into which $\mathcal{B}$ has chosen to insert his challenge C^*. Since $\mathcal{B}$ chooses j *randomly* in each iteration and $\mathcal{A}$'s view is *independent* of j in each iteration, it holds that $\mathsf{Pr}[\mathbf{BadQuery}] \leq {}^{1}/_{2^k}$.

Denote by **GuessTag** the probability that $\mathcal{A}$ queries his parallel CCA-oracle during the *first* commit session *using the tag of the second session* in one of the iterations.

Since the sender chooses the tag of the second session randomly (from the set of strings of length k) *after the first session is over*, it holds that $\mathsf{Pr}[\mathbf{GuessTag}] \leq {}^{k \cdot i}/_{2^k}$, where $i = poly(k)$ is the number of commitments in the parallel oracle query.

Now it holds that *conditioned on* ***BadQuery*** *and* ***GuessTag*** *both* not *occurring*, the output of $\mathcal{B}$ is either identically distributed to the output of H_0 (this holds if $\mathrm{C}^* = \mathsf{Com}_{tag_1}(s_1)$) or identically distributed to the output of H_1 (this holds if $\mathrm{C}^* = \mathsf{Com}_{tag_1}(t)$).

Let $E = \mathbf{BadQuery} \vee \mathbf{GuessTag}$ and let $\mathrm{Output_b}(\mathcal{B})$ denote the output of $\mathcal{B}$ in the RepeatPCCA-experiment if the challenge bit b was chosen. Then we have the following:

$$\begin{aligned} |\mathsf{Pr}[out_0 = 1] - \mathsf{Pr}[out_1 = 1]| &\leq \mathsf{Pr}[E] + |\mathsf{Pr}[out_0 = 1|\neg E] - \mathsf{Pr}[out_1 = 1|\neg E]| \\ &= \mathsf{Pr}[E] + |\mathsf{Pr}[\mathrm{Output}_0(\mathcal{B}) = 1|\neg E] \\ &- \mathsf{Pr}[\mathrm{Output}_1(\mathcal{B}) = 1|\neg E]| \\ &\leq \frac{k \cdot i + 1}{2^k} + \mathsf{negl}(k) \\ &= \mathsf{negl}'(k) \end{aligned}$$

Note that $|\mathsf{Pr}[\mathrm{Output}_0(\mathcal{B}) = 1|\neg E] - \mathsf{Pr}[\mathrm{Output}_1(\mathcal{B}) = 1|\neg E]| \leq \mathsf{negl}(k)$ holds because Com is RepeatPCCA-secure by Lemma 1 and $\mathsf{Pr}[\neg E] = 1 - {}^{k \cdot i+1}/_{2^k}$ is overwhelming in k (see Appendix B).

Sub-Claim 2: $|\mathsf{Pr}[out_1 = 1] - \mathsf{Pr}[out_2 = 1]| \leq \mathsf{negl}(k)$**.** This follows from a standard reduction argument to the parallel CCA-security of Com. Consider an adversary $\mathcal{B}'$ against the parallel CCA-security of Com. $\mathcal{B}'$ simulates the experiment H_1 for $\mathcal{A}$. After $\mathcal{A}$ has sent (v_0, v_1, tag), $\mathcal{B}'$ chooses a random bit $b \leftarrow \{0, 1\}$ and sends $(v_b, 0^k, tag)$ to his experiment. $\mathcal{B}'$ then forwards his challenge C^* to $\mathcal{A}$ as $\mathcal{A}$'s challenge. If $\mathcal{A}$ queries his oracle then $\mathcal{B}'$ forwards this query to his own oracle. After the simulated experiment is over, $\mathcal{B}'$ outputs what the simulated experiment outputs. It holds that the output of $\mathcal{B}'$ is identically distributed to the output of H_1 if $\mathrm{C}^* = \mathsf{Com}_{tag}(v_b)$ and identically distributed to the output of H_2 if $\mathrm{C}^* = \mathsf{Com}_{tag}(0^k)$. Sub-Claim 2 now follows from the parallel CCA-security of Com.

Sub-Claim 3: $\mathsf{Pr}[out_2 = 1] = {}^{1}/_{2}$**.** This follows from the fact that the view of $\mathcal{A}$ in the hybrid H_2 is independent of the challenge bit.

In conclusion, $|\mathsf{Pr}[out_0 = 1] - {}^{1}/_{2}]| \leq \mathsf{negl}(k)$. Hence, Com' is parallel CCA-secure. □

Using the transformation implied by [21, 23] described earlier and the proof of Thm. 9, we also get the following separation:

Theorem 10 (PNM $\not\Rightarrow$ CNM). *If there exists a statistically binding, parallel non-malleable commitment scheme, then there also exists a statistically binding and parallel non-malleable commitment scheme that is not concurrently non-malleable.*

Using similar arguments as in the proof of Thm. 9, one can also show that one-one CCA-security does not imply parallel non-malleability.

Theorem 11 (1CCA $\not\Rightarrow$ PNM). *If there exists a statistically binding, one-one CCA-secure commitment scheme, then there also exists a statistically binding and one-one CCA-secure commitment scheme that is not parallel non-malleable.*

Proof Idea. This separation follows by adapting the techniques used for the separation in Thm. 9. In the commitment scheme Com' the sender commits to the shares s_0 and s_1 *in parallel* instead of sequentially. The experiment Repeat1CCA is like RepeatPCCA except that the adversary may now query $\mathcal{O}_{\mathsf{1cca}}$ instead of $\mathcal{O}_{\mathsf{pcca}}$. Note that Com does not have to be extractable for this separation because the shares are sent in parallel. □

Remark 2. We remark that all separations in this work also hold for *bit* commitment schemes. This can be shown by a similar arguments as in the proofs of Thm. 1 and Thm. 9. The main difference for the proofs using the puzzle-solution approach is that the puzzle consists of k parallel (bit) commitments. The constructions based on the sharing approach still work in the same way.

Remark 3. Note that if one-way functions exist, all base commitment schemes required for this work exist. In all results one can use, e.g., the commitment scheme from [9] that is based on one-way functions as base commitment scheme Com. This scheme is CCA-secure and therefore fulfills all other desired security notions.

References

1. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Advances in Cryptology – CRYPTO 1998. Proceedings. pp. 26–45. Springer (1998)
2. Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard security does not imply security against selective-opening. In: Advances in Cryptology – EUROCRYPT 2012. Proceedings. pp. 645–662. Springer (2012)
3. Bellare, M., Sahai, A.: Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In: Advances in Cryptology – CRYPTO 1999. Proceedings. pp. 519–536. Springer (1999)
4. Böhl, F., Hofheinz, D., Kraschewski, D.: On definitions of selective opening security. In: Public Key Cryptography - PKC 2012. pp. 522–539 (2012)

5. Broadnax, B., Döttling, N., Hartung, G., Müller-Quade, J., Nagel, M.: Concurrently composable security with shielded super-polynomial simulators. In: Advances in Cryptology – EUROCRYPT 2017. Proceedings Part I. pp. 351–381. Springer (2017)
6. Canetti, R.: Universally composable security: A new paradigm for cryptographic protocols. In: 42th Annual IEEE Symposium on Foundations of Computer Science. pp. 136–145. FOCS '01, IEEE (2001)
7. Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) Advances in Cryptology – CRYPTO 2001: 21st Annual International Cryptology Conference, Proceedings. pp. 19–40. Springer (2001)
8. Canetti, R., Kushilevitz, E., Lindell, Y.: On the limitations of universally composable two-party computation without set-up assumptions. In: Biham, E. (ed.) Advances in Cryptology – EUROCRYPT 2003: 22nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings. pp. 68–86. Springer (2003)
9. Canetti, R., Lin, H., Pass, R.: Adaptive hardness and composable security in the plain model from standard assumptions. In: The 51st Annual IEEE Symposium on Foundations of Computer Science, 2010. Proceedings. pp. 541–550. FOCS 2010, IEEE (2010)
10. Canetti, R., Lin, H., Pass, R.: From unprovability to environmentally friendly protocols. In: The 54th Annual IEEE Symposium on Foundations of Computer Science, 2013. Proceedings. pp. 70–79. FOCS 2013, IEEE (2013)
11. Cao, Z., Visconti, I., Zhang, Z.: Constant-round concurrent non-malleable statistically binding commitments and decommitments. In: Public Key Cryptography – PKC 2010. pp. 193–208. Springer (2010)
12. Choi, S.G., Dachman-Soled, D., Malkin, T., Wee, H.: A black-box construction of non-malleable encryption from semantically secure encryption. Journal of Cryptology pp. 1–30 (2017)
13. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Concurrent non-malleable commitments (and more) in 3 rounds. In: Advances in Cryptology – CRYPTO 2016. Proceedings. pp. 270–299. Springer (2016)
14. Cramer, R., Hanaoka, G., Hofheinz, D., Imai, H., Kiltz, E., Pass, R., Shelat, A., Vaikuntanathan, V.: Bounded cca2-secure encryption. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 502–518. ASIACRYPT 2007, Springer (2007)
15. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing. pp. 542–552. STOC 1991, ACM (1991)
16. Garg, S., Mukherjee, P., Pandey, O., Polychroniadou, A.: The exact round complexity of secure computation. In: Advances in Cryptology – EUROCRYPT 2016. Proceedings Part II. pp. 448–476. Springer (2016)
17. Goldreich, O.: Foundations of Cryptography: Volume 1, Basic Tools. Cambridge University Press (2001)
18. Goldreich, O., Lindell, Y.: Session-key generation using human passwords only. In: Advances in Cryptology – CRYPTO 2001. Proceedings. pp. 408–432. Springer (2001)
19. Goyal, V.: Constant round non-malleable protocols using one way functions. In: Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing. pp. 695–704. STOC 2011, ACM (2011)

20. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing. pp. 1128–1141. STOC 2016 (2016)
21. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography. In: Foundations of Computer Science, 1989., 30th Annual Symposium on. pp. 230–235. IEEE (1989)
22. Katz, J., Ostrovsky, R., Smith, A.: Round efficiency of multi-party computation with a dishonest majority. In: Advances in Cryptology – EUROCRYPT 2003. Proceedings. pp. 578–595. Springer (2003)
23. Kiyoshima, S.: Round-efficient black-box construction of composable multi-party computation. In: Advances in Cryptology – CRYPTO 2014. Proceedings. pp. 351–368. Springer (2014)
24. Kiyoshima, S., Manabe, Y., Okamoto, T.: Constant-round black-box construction of composable multi-party computation protocol. In: Theory of Cryptography Conference. pp. 343–367. TCC 2014, Springer (2014)
25. Lin, H., Pass, R.: Black-box constructions of composable protocols without set-up. In: Advances in Cryptology – CRYPTO 2012. Proceedings. pp. 461–478. Springer (2012)
26. Lin, H., Pass, R., Venkitasubramaniam, M.: Concurrent non-malleable commitments from any one-way function. In: Theory of Cryptography Conference. pp. 571–588. TCC 2008, Springer (2008)
27. Lin, H., Pass, R., Venkitasubramaniam, M.: A unified framework for concurrent security: Universal composability from stand-alone non-malleability. In: Proceedings of the 41st annual ACM symposium on Theory of computing. pp. 179–188. STOC 2009, ACM (2009)
28. Pandey, O., Pass, R., Vaikuntanathan, V.: Adaptive one-way functions and applications. In: Advances in Cryptology – CRYPTO 2008. Proceedings. pp. 57–74. Springer (2008)
29. Pass, R., Shelat, A., Vaikuntanathan, V.: Construction of a non-malleable encryption scheme from any semantically secure one. In: Advances in Cryptology – CRYPTO 2006. Proceedings. pp. 271–289. Springer (2006)
30. Prabhakaran, M., Sahai, A.: New notions of security: Achieving universal composability without trusted setup. In: Proceedings of the 36th Annual ACM Symposium on Theory of Computing. pp. 242–251. STOC '04, ACM (2004)
31. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: The 51st Annual IEEE Symposium on Foundations of Computer Science, 2010. Proceedings. pp. 531–540. FOCS 2010, IEEE (2010)

A The Implication Results

Here we prove all our implication results (cf. Fig. 1). The results in themselves are not surprising, but are included for the sake of completeness. To this end, we adapt proof techniques from Bellare and Sahai [3]. They show that for public-key encryption schemes the notions of parallel CCA-security and parallel non-malleability are equivalent. We show that for *non-interactive* commitment schemes this is also the case. However, for *interactive* commitment schemes the situation is different, as we have constructed in Sec. 5 a commitment scheme that separates the two notions.

Theorem 12 (PCCA $\Rightarrow$ PNM). *Let* Com *be a parallel CCA-secure commitment scheme. Then* Com *is also parallel non-malleable.*

Proof Idea. By taking the strategy from Bellare and Sahai (cf. proof of Thm. 5.3 in [3] and adapting the proof to commitment schemes the proof of this theorem is straightforward. The general strategy for the parallel CCA-adversary is to forward his challenge to the parallel non-malleability adversary and use his parallel CCA-oracle to decommit the messages the parallel non-malleability adversary send to the receiver. □

Note that this proof holds for general commitment schemes, regardless of whether they are interactive or non-interactive. With a very similar proof one can show that one-one CCA-security implies non-malleability.

Theorem 13 (1CCA $\Rightarrow$ NM). *Let* Com *be a one-one CCA-secure commitment scheme. Then* Com *is also non-malleable.*

In contrast to public-key encryption schemes, the theorem that parallel non-malleability implies parallel CCA-security only holds for non-interactive commitment schemes.

Theorem 14 (PNM $\xRightarrow{\text{n.-i.}}$ PCCA). *Let* Com *be a non-interactive parallel non-malleable commitment scheme. Then* Com *is also parallel CCA-secure.*

Proof Idea. We again adapt the strategy from Bellare and Sahai (cf. proof of Thm. 5.2 in [3]). The general strategy for the parallel non-malleability adversary is to forward his challenge to the parallel CCA-adversary and to forward the oracle query of the parallel CCA-adversary to the receiver. Then the distinguisher gets what is effectively the oracle answer as input (via the implicit committed value oracle of the experiment) and can continue the simulation of the parallel CCA-adversary until he outputs his solution. □

With essentially the same proof one can show that non-malleability implies one-one CCA-security for non-interactive commitment schemes .

Theorem 15 (NM $\xRightarrow{\text{n.-i.}}$ 1CCA). *Let* Com *be a non-interactive non-malleable commitment scheme. Then* Com *is also one-one CCA-secure.*

B A Technical Detail

The statement $|\mathsf{Pr}[\mathrm{Output}_0(\mathcal{B}) = 1|\neg E] - \mathsf{Pr}[\mathrm{Output}_1(\mathcal{B}) = 1|\neg E]| \leq \mathsf{negl}(k)$ holds (see proof of sub-claim 1 in Thm. 9).

Proof.

$$|\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1]| =$$

$$|\mathsf{Pr}[\neg E] \cdot (\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|\neg E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|\neg E])$$
$$+ \mathsf{Pr}[E] \cdot (\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|E])|$$

$$\geq \mathsf{Pr}[\neg E] \cdot |\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|\neg E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|\neg E]|$$
$$- \mathsf{Pr}[E] \cdot |\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|E]|$$
(because $|x + y| \geq |x| - |y|$)

$$\geq \mathsf{Pr}[\neg E] \cdot |\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|\neg E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|\neg E]| - \mathsf{Pr}[E] \cdot 1$$
(because $|\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|E]| \leq 1$)

$$\geq (1/2) \cdot |\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|\neg E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|\neg E]| - \mathsf{Pr}[E] \cdot 1$$
(This holds for sufficiently large k because $\mathsf{Pr}[\neg E]$ is overwhelming.
Note that 1/2 is arbitrary, any constant works.)

Since $|\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1]|$ and $\mathsf{Pr}[E]$ are negligible, $|\mathsf{Pr}[\text{Output}_0(\mathcal{B}) = 1|\neg E] - \mathsf{Pr}[\text{Output}_1(\mathcal{B}) = 1|\neg E]|$ must also be negligible. □