Overcoming Impossibility Results in Composable Security using Interval-Wise Guarantees

Daniel Jost and Ueli Maurer

Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland.
{dajost, maurer}@inf.ethz.ch

Abstract. Composable security definitions, sometimes called simulation-based definitions, provide very strong security guarantees. In particular, they assure that the guarantees hold in any context. However, they are also met with some skepticism because of many impossibility results; goals such as commitments and zero-knowledge that are achievable in a stand-alone sense were shown to be unachievable composably (without a setup) since provably no efficient simulator exists. In particular, in the context of adaptive security, the so-called "simulator commitment problem" arises: once a party gets corrupted, an efficient simulator is unable to be consistent with its pre-corruption outputs. A natural question is whether such impossibility results are unavoidable or only artifacts of frameworks being too restrictive.

In this work, we propose a new type of composable security statement that evades the commitment problem by a specific instantiation of the concept of system specifications in the Constructive Cryptography (CC) framework, capturing the intersection of several interval-wise guarantees, each specifying the guarantees between two events. We develop the required theory within the CC framework and present the corresponding new composition theorem.

We present three applications of our notion. First, we show in the context of symmetric encryption with adaptive corruption how our notion naturally captures the expected confidentiality guarantee—the messages remain confidential until either party gets corrupted—and that it can be achieved by any standard semantically secure scheme (negating the need for non-committing encryption). Second, we present a composable formalization of commitments that is instantiable without a trusted setup like a CRS, and show its application to coin tossing over the telephone, one of the early intuitive applications of commitments. Third, we reexamine a result by Hofheinz, Matt, and Maurer [Asiacrypt'15] implying that IND-ID-CPA security is not the right notion for identity-based encryption, unmasking this claim as an unnecessary framework artifact.

1 Introduction

1.1 A Plea for Composable Security

Common security definitions found in the literature are game-based, i.e., they require that an adversary cannot win a game that exports certain oracles to the

adversary. The goal of such a security game is thereby to capture the adversary's potential attacks in a minimal manner. However, the mapping of the oracles the game provides to potential attacks in the real-world use of the cryptographic protocol is commonly not straightforward. Thus, it is often a-priori unclear which game-based security notion is required in order for the protocol to be secure in a specific application. Rather, that aspect is often informally considered and passed down outside the security definitions, becoming "folklore" over the years.

Composable frameworks, such as [7, 24, 19, 14], on the other hand, provide operational security definitions instead. The way they formalize security is based around comparing the execution of the protocol in the real world to an idealized world that intrinsically has the desired security properties. Importantly, this definition is with respect to any environment, thereby ensuring that the security guarantees not only do not exclude any attacks but also hold irrespective of other protocols (or multiple instances of the same one) being executed. For instance, the composable security definition of a symmetric encryption scheme *is* the construction of a secure communication channel from an authentic one, with different assumed and constructed channels leading to different notions (e.g., whether replaying is possible). Hence, having a concrete goal in mind, and a particular assumed channel, it is now easy to select the right scheme.

Finally, composable frameworks facilitate modularity. First, they are based on defining components with clean abstraction boundaries (e.g., a secure channel) that abstract away the details of how that module has been constructed (or otherwise obtained). This idealized module can then be used by a higher-level protocol with the security of the combined overall protocol following directly from the composition theorem. Thus, the security of complex protocols can be neatly proven by composing it from smaller sub-protocols.

1.2 Obstacles for Composable Security

While the clear semantics, modularity, and high security guarantees suggest that all protocols should be proven secure in a composable framework rather than in an ad-hoc game-based manner, they are still not prevalent with the majority of new research still carried out using game-based definitions.

One of the main reasons hindering adoption might be that many primitives are known to be impossible to achieve in the plain UC model, such as zero knowledge [7] and commitments [8]. Furthermore, Lindell has shown [15] that impossibility results are not specific to the UC model but inherent to any kind of similar model based around the existence of an efficient simulator. As a consequence, respective protocols have to rely on additional setup assumptions, such as a common reference string, and are also generally less efficient.

One particular obstacle composable definitions often face is the so-called "simulator commitment problem", which mainly arises when considering adaptive security. In a nutshell, it describes the simulator's inability to explain some of its previous choices the moment a party gets corrupted. More concretely, consider the example of two parties securing their communication using symmetric encryption. The intuition is that the adversary does not learn the messages until either of the

parties gets corrupted, thereby revealing the key. Before, the adversary should learn at most the length. As a result, the simulator, in the first phase, has to output a fake ciphertext not containing the real message. For any semantically secure encryption scheme he can actually do so. This, however, commits him on those fake ciphertexts. At the moment a party gets corrupted, the simulator then needs to be able to explain those ciphertexts by outputting a corresponding encryption key. Even if he learns all the previous messages, he will not be able to do so for regular encryption schemes. Note, however, that the commitment problem is not restricted to adaptive corruptions only. Similar issues also arise, for instance, in the context of password-based security [11] or identity-based encryption [12], where it has been shown that due to this commitment problem the standard game-based notions do not induce the excepted corresponding composable statement.

On a general level, this raises the fundamental question whether such impossibility results actually indicate a security issue, and hence protocols not satisfying the stronger composable definitions should not be used, or whether they present an artifact of the framework. Especially for the commitment problem, the common understanding is that the latter is true. Furthermore, the obstacles are often dealt with by either reverting to composable security with static corruptions only, or by simpling retracting to game-based definitions. As a result, there is a clear need for a better composable security notion that lets us settle this question and remedy the issue of the spurious impossibilities.

1.3 Existing Attempts to Overcome the Obstacles

A number of approaches have been proposed in order to circumvent the aforementioned issues of composable security.

First, Canetti and Krawczyk proposed the notion of non-information oracles [10] within the UC-framework. A non-information oracle is essentially a game-based definition embedded into an ideal functionality. For instance, rather than saying that an encryption scheme should realize a secure channel that only leaks the length, the respective functionality leaks the output of the non-information oracle, which is required to satisfy a CPA-like definition. While this circumvents the commitment problem, there are two drawbacks. First, it weakens composition by requiring explicit reductions to the embedded games in the security proof of the higher-level protocols using the functionality. Second, for each ideal functionality a different type non-information oracle needs to be defined, without providing any generic template. As a consequence, the question of the "right" non-information oracle re-arises, just like when defining a security game.

Second, a line of work considers super-polynomial simulators [23, 25, 6]. The initial proposal by Pass [23] considered sub-exponential simulators and polynomially bounded environments. This implies, however, that the simulator cannot be absorbed into the environment, ceding some of the most fundamental composition properties of the UC-framework. The later works by Prabhakaran and Sahai [25] and Broadnax et al. [6] empower the simulator in a more controlled manner, preserving most natural composition properties. Their adoption, however,

still suffers from being rather technical, and moreover, still quite limited in the number of issues a more powerful simulator can overcome. For instance, when considering a PRG whose seed might leak, even an all powerful simulator will with high probability not be able to explain a truly randomly chosen output with an appropriate seed.

Finally, Backes, Dürmuth, Hofheinz, and Küsters [1] proposed an approach where the real-world resource would just disallow certain activation sequences by the environment that were otherwise impossible to simulate. While this avoids the complications of the other approaches, it scarifies the evident semantics of composable security notions by excluding certain—deemed artificial—attacks. The same approach has recently been used by Jost, Maurer and Mularczyk in [13].

1.4 Contributions

A novel security notion. As the first contribution, we introduce a novel composable security notion, in the Constructive Cryptography (CC) framework, formalizing guarantees that hold in a certain interval (between two events).

At its heart, the CC framework is a theory about resource specifications. Proving a protocol $\boldsymbol{\pi}$ to be secure corresponds to modeling the assumed real-world specification $\mathcal{R}$, and showing that the resulting specification $\boldsymbol{\pi}\mathcal{R}$ is contained in an ideal-world specification $\mathcal{S}$, i.e, $\boldsymbol{\pi}\mathcal{R} \subseteq \mathcal{S}$. Such a statement, by definition, is environment-agnostic with the transitivity of set inclusion being the fundamental composition property. While a specification is in principle an arbitrary set of resources, one usually restricts oneself to specific specification types, for the following reasons: First, such a type provides a clear interpretation—or semantics—of how the guarantees asserted by $\mathcal{S}$ should be interpreted. Second, restricting to a specific types allows to formalize syntactical derivation rules on how to combine multiple statements. For a predefined type of specifications, the induced space of statements $\boldsymbol{\pi}\mathcal{R} \subseteq \mathcal{S}$ is then usually called a *construction notion* and the set of corresponding derivation rules is called *composition theorem*.

In this work, we consider two novel types of specifications. First, we inherently consider specifications consisting of the intersection of multiple ones, modeling the conjunction of their respective guarantees. Second, we formalize a type of specification that naturally can be interpreted as providing *interval-wise* guarantees, i.e., providing guarantees only between two events. We formalize this type of specification using a so-called relaxation, that waives all guarantees before and after the respective events. In the spirit of modularity, we actually introduce two atomic and independent relaxations—one waiving the guarantees before an event, and one after—and then show how they can be appropriately combined. We then study how those relaxations interact with the existing building blocks of CC and, finally, present the respective syntactic composition theorem, that actually supersedes all the existing ones.

Applications. As a second contribution, we apply our methodology to several examples. First, we consider the encrypt-then-MAC paradigm in a setting where

the keys can adaptively leak to the adversary, stylizing adaptive passive corruptions. Using our interval-wise guarantees, we obtain a simple composable security definition thereof without the need for non-committing encryption. More concretely, we consider the following three properties. First, we require the messages to be confidential as long as neither the encryption nor the authentication key leaked. (An IND-CPA secure scheme cannot guarantee confidentiality without authenticity.) In our definition, this is phrased as the construction of a secure channel *up to that point*. Second, between the exposure of the encryption key and the authentication key, we require communication to still be authentic, i.e., an authenticated channel to be constructed. Finally, after the encryption key has been exposed, we still require correctness, i.e., the parties can communicate as long as the adversary does not actively interfere.

As a second application, we present a composable formalization of information-theoretically binding commitment schemes realizable in the plain model. We then show how, based on such a commitment scheme, Blum's protocol constructs a composable coin-toss notion. Applying composition then directly implies that this formalization can be achieved in the plain model as well. While the resulting specification is obviously too weak to serve as a common reference string, it guarantees unbiasedness, in particular validating the intuitive-level argumentation about flipping a coin over the telephone of the corresponding papers of that time.

Finally, we consider the composable guarantees of identity-based encryption. We revisit the result by Hofheinz, Matt, and Maurer [12] that shows the standard ind-id-cpa notion to be too weak when considering a traditional composable statement, even when considering *static corruptions*, due to the commitment problem. Furthermore, the authors have shown that the same weaker construction that actually can be achieved, could also be achieved by a weaker game-based notion ind-id1-cpa, modeling so-called lunch-time attacks. We refute their results in the following way: Based on interval-wise guarantees we formalize a composable specification of IBE that corresponds exactly to the standard ind-id-cpa notion.

1.5 Outline

In Section 3, we introduce our new type of composable security statement. As a running example, we consider encrypt-then-MAC with symmetric keys that might get leaked to the adversary. In Section 4 we present a novel composable definition of perfectly binding commitments and its application to coin tossing. Finally, in Appendix A we revisit composable security of identity-based encryption.

2 Preliminaries: Constructive Cryptography

Ever since its initial proposal [19, 16], the Constructive Cryptography (CC) framework has undergone significant changes and refinements. Not only did the actual definitions evolve—such as in the shift from asymptotic security definitions in the initial version towards finite statements with explicit reductions

in subsequent work—but also the *interpretation* of the basic elements, such as the simulator, fundamentally changed over the course of the years.

Since this work inherently builds upon some more recent aspect of the framework, we provide an extensive summary of CC in this section. The presentation in large parts follows the exposition introduced in [19], with some adaptations from [13].

2.1 Resources, Converters, and the Interaction Model

At its heart, the Constructive Cryptography framework views cryptography as a resource[1] theory, in which parties use certain resources—e.g., communication channels and a public-key infrastructure—to construct via a protocol other resources. A resource thereby formalizes a module that exports a well-defined interface in a black-box manner to the rest of the world.

Global events. In this work, we use the version of Constructive Cryptography introduced in [13] that enriches the interaction model by a notion of globally observable events. Formally, events are a generalization of monotone binary outputs (MBO) introduced by Maurer et al. [18]. Roughly, an MBO is a value that can change from 0 to 1 but not back, which can be interpreted as a single event happening once the MBO changes to 1. An event then just corresponds to a named MBO and the *global event history* $\mathcal{E}$ is a list of event names without duplicates (to model that every event can occur at most once). For an event name n, we denote by $\mathcal{E} \xleftarrow{+} \mathcal{E}_n$ the act of appending n to $\mathcal{E}$ (or leaving it unchanged if it is already contained). Moreover, we use $\mathcal{E}_n$ as a short-hand notation to denote that n is in the list $\mathcal{E}$, and say that the event happened. Finally, we denote by $\mathcal{E}_{n_1} \prec \mathcal{E}_{n_2}$ that the event n_1 precedes the event n_2 in the event history.

Resources. A resource R is a reactive system that interacts in the following two ways with the rest of the world: First, it allows interaction at one or several named *communication interfaces*, in the following just called interfaces, at which it can be activated by an input x, and is expected to return activation by answering with a single output y (at the same interface). Second, while activated by an input, the resource R can depend on the global event history $\mathcal{E}$, and can furthermore append events from a predefined set of names. We call this set of names the events controlled by R. Note that another resource triggering an event does not activate R, only inputs do.

Formally, resources are modeled as random systems [17], where the interface address, the actual input x, and the current state of the event history are encoded as part of the input. Analogously, the answer y and the new state of the event history are encoded as part of the output, under the constraint that the old state of the event history is a prefix of the new one. For the sake of this paper, a reader unfamiliar with the CC framework might however just think of a resource

[1] The analogon to *functionalities* in the UC framework [7].

as the behavior of an oracle machine, where each interface corresponds to an oracle and the event history being similar to the "directory" ITI used in the recent version (as of December 2018) of UC [7]. Note, however, that formally a resource only defines the behavior of the system and not its description, i.e., two different (pseudo-code descriptions of) ITMs having the same input-output behavior denote the same resource.

A set of resources can be viewed as a single one. The interface set of the composed resource corresponds to the union of the ones from the composed resources. For resources $\mathsf{R}_1, \ldots, \mathsf{R}_n$ (with disjoint interface sets) we denote by $[\mathsf{R}_1, \ldots, \mathsf{R}_n]$ the *parallel composition*.

Converters and protocols. In the Constructive Cryptography framework, *converters* express the local action executed by one party. A converter expects to be connected to a given number of interfaces at the "inside", and emulates a certain set of interfaces at the "outside". Upon an input at an outside interface, the converter is allowed to make a bounded number of oracle queries to the inside interfaces (recall that a resource always returns at the same interface it was queried), before returning a value at the queried interface.

For a converter π and a resource R, let $\mathcal{I}$ denote a tuple describing an injective mapping from π's inside interfaces to interfaces of R. We then denote by $\mathsf{R}' \coloneqq \pi^{\mathcal{I}}\mathsf{R}$ the resource obtained from connecting the converter accordingly. The resource R' no longer exposes those interfaces to the world, but the ones emulated by π instead. Converter attachment satisfies the natural property of *composition order independence*, stating that on the term algebra level the composition order does not matter—only the final system. This is summarized by the following proposition.

Proposition 1. *Let π_1 and π_2 be two converters, let R be a resource and let $\mathcal{I}_1$ and $\mathcal{I}_2$ be such that they assign* disjoint *interfaces. Then,*

$$\pi_1^{\mathcal{I}_1}\pi_2^{\mathcal{I}_2}\mathsf{R} = \pi_2^{\mathcal{I}_2}\pi_1^{\mathcal{I}_1}\mathsf{R}.$$

Moreover, if S is another resource such that the interface sets of R and S are disjoint, then we have

$$\pi_1^{\mathcal{I}_1}\big[\mathsf{R}, \mathsf{S}\big] = \big[\pi_1^{\mathcal{I}_1}\mathsf{R}, \mathsf{S}\big].$$

We define a *protocol* to be a set of converter-connection pairs, i.e,. $\boldsymbol{\pi} \coloneqq \{(\pi_1, \mathcal{I}_1), \ldots, (\pi_n, \mathcal{I}_n)\}$ with pairwise disjoint $\mathcal{I}_i$'s. Moreover, we say that $\boldsymbol{\pi}$ is a protocol for a resource R, if R has all the required interfaces for the protocol application to be well-defined, and write $\boldsymbol{\pi}\mathsf{R}$ to denote its application. (By composition order independence it is irrelevant in which order the converters are attached.)

The environment (distinguisher). The distinguisher D is a special type of environment that first interacts with a resource R by making queries to the resource's interfaces. Between two such queries it can access the global

event history and append events it, except the ones controlled by R. Note that activations are atomic, i.e., at any moment in time either the resource or the distinguisher is activated, but not both. Finally, the distinguisher ends the interaction with the resource by outputting a bit. The advantage of D is then defined as

$$\Delta^{\mathsf{D}}(\mathsf{R},\ \mathsf{S}) \coloneqq \Pr\left[\mathsf{D}^{\mathcal{E}}(\mathsf{S}) = 1\right] - \Pr\left[\mathsf{D}^{\mathcal{E}}(\mathsf{R}) = 1\right],$$

where we use the syntax $\mathsf{D}^{\mathcal{E}}(\cdot)$ to make explicit that the distinguisher has oracle access to the global event history $\mathcal{E}$.

2.2 Constructions

Specifications. It is natural to consider only certain desired (or assumed) properties of a system and deliberately not specify others. Some of those choices are intrinsic to the mathematical model we use, such as only considering the input-output behavior and ignoring the physical aspects. Other properties can be purposefully ignored by considering specifications of systems that simply leave out those aspects, focusing only on the relevant properties. Following [20], we model specifications as sets of resources $\mathcal{R}$ that all have the same interface set. For each property, such as confidentiality, one has in mind, one can consider the set $\mathcal{R}$ of resources satisfying that property. Vice versa, each set of resources $\mathcal{R}$ can be interpreted as the set of properties common to all elements. For instance, authenticated communication might be modeled as the set of all communication channels that are authentic—not specifying the level of confidentiality by including both confidential as well as non-confidential channels.

Over the years, specifications have turned into a cornerstone of Constructive Cryptography. They have been used in various works, such as [3, 2], to cope with situations where a complete description of the construction would be unnecessary and undesirably complicated. This is a more direct approach than the one commonly used in the literature on UC, where such details are usually delegated to the adversary. Note that thereby both approaches cover the worst case. While the UC approach generally models the worst case only (decided by the adversary), the specification view of CC captures every case, including the worst one.

Constructions as subsets. In provable security one typically considers the execution of a protocol $\boldsymbol{\pi}$ that makes use of some assumed specification $\mathcal{R}$, such as a communication network or a public-key infrastructure. In short, one wants to show that the specification $\boldsymbol{\pi}\mathcal{R}$ satisfies the desired security properties. As explained in the previous section, those properties are formalized as a specification $\mathcal{S}$, and thus proving security means proving $\boldsymbol{\pi}\mathcal{R} \subseteq \mathcal{S}$. Note that obviously the guarantees given by $\mathcal{S}$ are generally weaker than the ones by $\boldsymbol{\pi}\mathcal{R}$. The purpose of such a statement is, however, that the security properties are in $\mathcal{S}$ both more explicit and simpler to analyze. In other words, the goal is to distill out the relevant properties and abstract away the others.

Traditionally, the statement $\boldsymbol{\pi}\mathcal{R} \subseteq \mathcal{S}$ is read as the protocol $\boldsymbol{\pi}$ constructing the specification $\mathcal{S}$ from the specification $\mathcal{R}$, or in UC-jargon the protocol securely

realizing the specification $\mathcal{S}$ in the $\mathcal{R}$-hybrid model. Hence, as a shorthand notation we introduce the following construction notion.

Definition 1. *Let $\mathcal{R}$ and $\mathcal{S}$ be arbitrary specifications, and let $\boldsymbol{\pi}$ be an arbitrary protocol for $\mathcal{R}$. Then, we say that $\boldsymbol{\pi}$* constructs *$\mathcal{S}$ from $\mathcal{R}$, denoted $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S}$, if and only if $\boldsymbol{\pi}\mathcal{R} \subseteq \mathcal{S}$, i.e.,*

$$\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S} \;:\Longleftrightarrow\; \boldsymbol{\pi}\mathcal{R} \subseteq \mathcal{S}.$$

This construction notion is associated with the usual composition properties of Constructive Cryptography: sequential and parallel composition—which form the equivalence of the universal composition theorem of the UC-framework.

Theorem 1. *Let $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$ be arbitrary specifications, and let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ be arbitrary protocols for $\mathcal{R}$ and $\mathcal{S}$, respectively. Then, we have*

1. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S} \wedge \mathcal{S} \xrightarrow{\boldsymbol{\pi}'} \mathcal{T} \implies \mathcal{R} \xrightarrow{\boldsymbol{\pi}'\circ\boldsymbol{\pi}} \mathcal{T}$,
2. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S} \implies [\mathcal{R}, \mathcal{T}] \xrightarrow{\boldsymbol{\pi}} [\mathcal{S}, \mathcal{T}]$.

Proof. The first property follows directly from the transitivity of the subset relation, $\boldsymbol{\pi}'(\boldsymbol{\pi}\mathcal{R}) \subseteq \boldsymbol{\pi}'\mathcal{S} \subseteq \mathcal{T}$, and the second property follows from Proposition 1: $\boldsymbol{\pi}[\mathcal{R}, \mathcal{T}] = [\boldsymbol{\pi}\mathcal{R}, \mathcal{T}] \subseteq [\mathcal{S}, \mathcal{T}]$.

The specifications $\boldsymbol{\pi}\mathcal{R}$ and $\mathcal{S}$ are often referred to as real- and ideal-world, respectively, according to the so-called real-world/ideal-world paradigm on which most composable frameworks [7, 24, 19, 14] are based. Following that paradigm, security statement affirm that the real word is "just-as-good" as the ideal world, meaning that for all parties, no matter whether honest or adversarial, it does not make a difference whether they live in the real (where an arbitrary element of $\boldsymbol{\pi}\mathcal{R}$ is present), or in the ideal world (where some element of $\mathcal{S}$ is present). Hence, if the honest parties are content with the guarantees they get from the ideal specification, they can safely execute the protocol in the real world instead. While such security definitions traditionally hard-code a particular security flavor, our basic definition, on the other hand, is absolute. It does not require the choice of a particular computational model, an (asymptotic) efficiency notion, a particular type of indistinguishability (computational vs. statistical), or even the existence of a simulator. Those aspects are treated independently (as explained in the next sections).

The (in)existence of a simulator. Simulation-based security turned out to be one of the most fundamental concepts in cryptography and is closely linked with the real-world / ideal-world paradigm. It not only forms the foundation of semantic security, zero knowledge, and the security of MPC, but also of virtually every composable framework. Whereas the former definitions tend to require an after-the-fact simulation of the transcript, composable frameworks get their stronger guarantees from requiring on-line simulation, where an adaptive environment interacts with the simulator. The common understanding of those

security definitions is then that the simulator "translates" the attacks from the real-world adversary to the ideal world such that they achieve the same effect.

While the initial version of the Constructive Cryptography framework also hard-coded the existence of a simulator (with respect to the dummy adversary), starting form [20], the simulator is no longer an integral part of the construction notion. Rather, employing a simulator is just one way of defining an ideal specification, $\boldsymbol{\sigma}\mathcal{S}$. Thus, showing that $\boldsymbol{\pi}\mathcal{R} \subseteq \boldsymbol{\sigma}\mathcal{S}$ (where the simulator $\boldsymbol{\sigma}$ is connected to adversarially controlled interfaces) is really just a way of expressing the specification in a decomposed manner that makes the achieved security properties obvious. For instance, the specification of confidential channels can then be written as the specification $\mathcal{S}$ of channels that only leak the message length, combined with an arbitrary simulator. From this description it is then apparent that for any resource in the combined specification $\boldsymbol{\sigma}\mathcal{S}$ the adversary does not learn more about the message than the length.

If one restricts oneself to specifications of this type, then the following more specific composition theorem can be deduced. In short, it states that the simulator of one construction statement can be soundly ignored in further steps, improving the abstraction provided by a construction.

Proposition 2. *Let $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$ be specifications, and let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ be protocols for $\mathcal{R}$ and $\mathcal{S}$, respectively. For any simulators $\boldsymbol{\sigma}$ (for $\mathcal{S}$) and $\boldsymbol{\sigma}'$ (for $\mathcal{T}$), such that the set of interfaces controlled by the simulators are disjoint from the ones controlled by the protocols, we have*

1. $\mathcal{S} \xrightarrow{\boldsymbol{\pi}'} \boldsymbol{\sigma}'\mathcal{T} \implies \boldsymbol{\sigma}\mathcal{S} \xrightarrow{\boldsymbol{\pi}'} \boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T}$,
2. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \boldsymbol{\sigma}\mathcal{S} \wedge \mathcal{S} \xrightarrow{\boldsymbol{\pi}'} \boldsymbol{\sigma}'\mathcal{T} \implies \mathcal{R} \xrightarrow{\boldsymbol{\pi}'\circ\boldsymbol{\pi}} \boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T}$,
3. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \boldsymbol{\sigma}\mathcal{S} \implies [\mathcal{R}, \mathcal{T}] \xrightarrow{\boldsymbol{\pi}} \boldsymbol{\sigma}[\mathcal{S}, \mathcal{T}]$.

Proof. By composition order invariance we have $\boldsymbol{\pi}'\boldsymbol{\sigma}\mathcal{S} = \boldsymbol{\sigma}\boldsymbol{\pi}'\mathcal{S} \subseteq \boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T}$, implying the first property. The second property follows directly from combining the first one with Theorem 1. The third property follows from Theorem 1 and Proposition 1 as well:

$$\boldsymbol{\pi}[\mathcal{R}, \mathcal{T}] = [\boldsymbol{\pi}\mathcal{R}, \mathcal{T}] \subseteq [\boldsymbol{\sigma}\mathcal{S}, \mathcal{T}] = \boldsymbol{\sigma}[\mathcal{S}, \mathcal{T}],$$

concluding the proof. □

2.3 Relaxations

The basic construction notion does not take into account errors or computational assumptions. Those aspects are formalized by so-called relaxations, as introduced in [20]. Treating relaxations as orthogonal to the basic construction notion allows the framework to consider different security notions within the same overall framework and, more generally, enables more flexible security statements.

On an abstract level, a relaxation is a mapping from specifications to weaker, so-called relaxed, specifications. For our purpose, where we instantiate specifications by sets of resources, we can define a relaxation as a function mapping a single resource to a set of resources.

Definition 2. *Let Θ denote the set of all resources. A* relaxation *ϕ is a function $\phi\colon \Theta \to 2^{\Theta}$ (where 2^{Θ} denotes the power set of Θ) such that $\mathsf{R} \in \phi(\mathsf{R})$ for all $\mathsf{R} \in \Theta$. In addition, for a specification $\mathcal{R}$, we define $\mathcal{R}^{\phi} \coloneqq \bigcup_{\mathsf{R}\in\mathcal{R}} \phi(\mathsf{R})$ as a shorthand notation.*

A concrete relaxation thereby formalizes some notion of resources being "almost-as-good" in some context. That is, if we were happy with constructing a resource specification $\mathcal{S}$, then we should also be happy with $\mathcal{S}^{\phi}$, if we believe the weakening ϕ to be justifiable in the given context. For instance, one could consider the relaxation that maps the resource R to the set of all computationally indistinguishable resources from R under some computational assumption. Then, if we consider the ideal specification modeling a secure channel, its relaxation consists of the set of all systems indistinguishable from a secure channel. Thus, showing that the real-world system is contained therein, asserts that it behaves like a secure channel unless the computational assumption is broken. Hence, if we believe the computational assumption to be valid, we should be as content with the relaxed specification as with secure channel.

Abstracting away irrelevant properties is a core paradigm of any modular analysis. Applied to Constructive Cryptography, this means that ideally we should be able to "forget" relaxations. That is, if one shows that one protocol constructs $\mathcal{S}^{\phi}$ (from some assumed resources), one should be able to compose it with another statement that assumes $\mathcal{S}$ instead. Hence, as part of the theory we need to distill out how the concrete relaxations interact with each other, as well as the other elements of the theory such as converter attachment. On the most abstract level, it is easy to see that the following rules apply to any relaxation.

Proposition 3. *For any specifications $\mathcal{R}$ and $\mathcal{S}$, and any relaxation ϕ, we have*

1. $\mathcal{R} \subseteq \mathcal{R}^{\phi}$,
2. $\mathcal{R} \subseteq \mathcal{S} \Longrightarrow \mathcal{R}^{\phi} \subseteq \mathcal{S}^{\phi}$,
3. $(\mathcal{R} \cap \mathcal{S})^{\phi} \subseteq \mathcal{R}^{\phi} \cap \mathcal{S}^{\phi}$,
4. $(\mathcal{R} \cup \mathcal{S})^{\phi} = \mathcal{R}^{\phi} \cup \mathcal{S}^{\phi}$.

Proof. All properties follow directly from basic set theory and the fact that by definition $\mathsf{R} \in \phi(\mathsf{R})$.

The reduction relaxation. We now introduce the most fundamental relaxation, which captures computational security based on explicit reductions. This is defined as a function ϵ that maps distinguishers to their respective performance in $[0, 1]$, where $\epsilon(\mathsf{D})$ typically refers to the winning probability of a modified distinguisher D' (the reduction) on the underlying computational problem.

Definition 3. *Let ϵ be a function that maps distinguishers to a value in $[0, 1]$. Then, the induced relaxation on a resource R, denoted R^{ϵ}, is defined as*

$$\mathsf{R}^{\epsilon} \coloneqq \left\{ \mathsf{S} \;\middle|\; \forall \mathsf{D} : |\Delta^{\mathsf{D}}(\mathsf{R}, \mathsf{S})| \leq \epsilon(\mathsf{D}) \right\}.$$

We call such a relaxation generally an ϵ-relaxation *or reduction relaxation.*

We now discuss several properties that ϵ-relaxations have. First, the errors just add up, as expressed by the following theorem.

Theorem 2. *Let $\mathcal{R}$ be an arbitrary specification, and let ϵ_1 and ϵ_2 be arbitrary ϵ-relaxations. Then we have $\left(\mathcal{R}^{\epsilon_1}\right)^{\epsilon_2} \subseteq \mathcal{R}^{\epsilon_1+\epsilon_2}$.*

Proof. This follows directly from the triangle inequality of the distinguishing advantage. A more detailed proof is given in Appendix B.1.

Second, they naturally commute with protocol application and parallel composition of additional resources, i.e., the relaxation can be "pulled out". In such a step, however, the additional resource or converter has to be explicitly accounted for in the reduction.

Theorem 3. *The ϵ-relaxation is* compatible *with protocol application in the following sense:*

$$\boldsymbol{\pi}\left(\mathcal{R}^{\epsilon}\right) \subseteq \left(\boldsymbol{\pi}\mathcal{R}\right)^{\epsilon_{\boldsymbol{\pi}}},$$

for $\epsilon_{\boldsymbol{\pi}}(\mathsf{D}) \coloneqq \epsilon(\mathsf{D}\boldsymbol{\pi}(\cdot))$, where $\mathsf{D}\boldsymbol{\pi}(\cdot)$ denotes the distinguisher that first attaches $\boldsymbol{\pi}$ to the given resource and then executes D. Moreover, the ϵ-relaxation is compatible with parallel composition, i.e.,

$$\left[\mathcal{R}^{\epsilon}, \mathcal{S}\right] \subseteq \left[\mathcal{R}, \mathcal{S}\right]^{\epsilon_{\mathcal{S}}},$$

for $\epsilon_{\mathcal{S}}(\mathsf{D}) \coloneqq \sup_{\mathsf{S}\in\mathcal{S}} \epsilon(\mathsf{D}[\,\cdot\,,\mathsf{S}])$, where $\mathsf{D}[\,\cdot\,,\mathsf{S}]$ denotes the distinguisher that emulates S in parallel to the given resource and then lets D interact with them.

Proof. The proof is given in Appendix B.2.

The composition theorem with ϵ-relaxations then follows directly from these compatibility results. The following corollary phrases the corresponding result—which in older version of Constructive Cryptography used to be called *the* composition theorem, thereby hard-coding computational security.

Corollary 1. *For any specifications $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$, any protocols $\boldsymbol{\pi}$ and $\boldsymbol{\pi'}$, and any ϵ-relaxation ϵ and ϵ', we have*

1. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S} \implies \mathcal{R}^{\epsilon} \xrightarrow{\boldsymbol{\pi}} \mathcal{S}^{\epsilon_{\boldsymbol{\pi}}}$,
2. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S}^{\epsilon} \implies [\mathcal{R}, \mathcal{T}] \xrightarrow{\boldsymbol{\pi}} [\mathcal{S}, \mathcal{T}]^{\epsilon_{\mathcal{T}}}$,
3. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S}^{\epsilon} \wedge \mathcal{S} \xrightarrow{\boldsymbol{\pi'}} \mathcal{T}^{\epsilon'} \implies \mathcal{R} \xrightarrow{\boldsymbol{\pi'}\circ\boldsymbol{\pi}} \mathcal{T}^{\epsilon_{\boldsymbol{\pi}'}+\epsilon'}$,

where $\epsilon_{\boldsymbol{\pi}}$ and $\epsilon_{\mathcal{S}}$ are defined as in Theorem 3, respectively.

Proof. The proof can be found in Appendix B.3.

3 Interval-Wise Guarantees

In this section, we introduce our novel composable security statement that expresses the guarantees in a certain interval. We proceed in several steps. First, the motivating running example of encrypt-then-MAC with key exposure is formalized. Second, we introduce two novel relaxations: one that allows to give up all guarantees after a certain event (to formalize that they only need to hold until then), and the complementary one that gives up all guarantees before a certain event. Third, we combine those relaxations and show that it fits well into the existing theory. Finally, we present the resulting construction notion and phrase the motivating example therein.

3.1 The Running Example: Encrypt-then-MAC

Consider two parties, Alice and Bob, who have pre-shared keys that they want to use to communicate securely over the Internet by using the encrypt-then-MAC paradigm. If both parties are honest and the keys are secure, then this can easily be formalized and proven in a composable framework [10, 21]. Unfortunately, once we take into account that the adversary might learn the encryption key, we face the so-called commitment problem of simulation-based security.

To this end, consider an insecure communication channel $\mathsf{InsecCh}$ from Alice to Bob and two keys that can leak[2], $\mathsf{AuthKey}$ and EncKey, for authentication and encryption, respectively, as formally defined in Figure 1. Note that whether the key leaked or not is modeled by the events $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}$ and $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$, respectively, which are triggered by the environment any point it wants. This models that we do not make any assumptions why or when the keys leak, i.e, whether it is the adversary to decide or not, but it does enforce that it happens the same moment in both the real and ideal world.

Naturally, we would like to prove that authentication is guaranteed until the authentication key leaks, i.e., the event $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}$ occurs, and confidentiality is guaranteed until either of the key leaks. Furthermore, correctness should hold unconditionally. Following the paradigm of modularity, one might try to formalize and prove this in two steps and first consider authentication only.

Proposition 4. *Let* AuthCh *denote the authenticated channel that degrades its security once the respective key is leaked, as formally defined in Figure 1. Then, there exists a simulator* $\boldsymbol{\sigma}_{\mathsf{MAC}}$ *such that*

$$\left[\mathsf{AuthKey}, \mathsf{InsecCh}\right] \xrightarrow{\boldsymbol{\pi}_{\mathsf{MAC}}} \left(\boldsymbol{\sigma}_{\mathsf{MAC}}\mathsf{AuthCh}\right)^{\epsilon_{\mathsf{MAC}}},$$

where ϵ_{MAC} *denotes a simple reduction to the MAC-forgery game.*

[2] In Constructive Cryptography, the adversary by definition only has access to interfaces statically assigned to him. Hence, adaptive corruptions are modeled by introducing explicit memory and computation resources with an adversarial interface, granting the adversary access once the party is corrupted. For simplicity, we here consider directly leaking key resources instead. Assuming secure erasure, this is equivalent to passive corruptions.

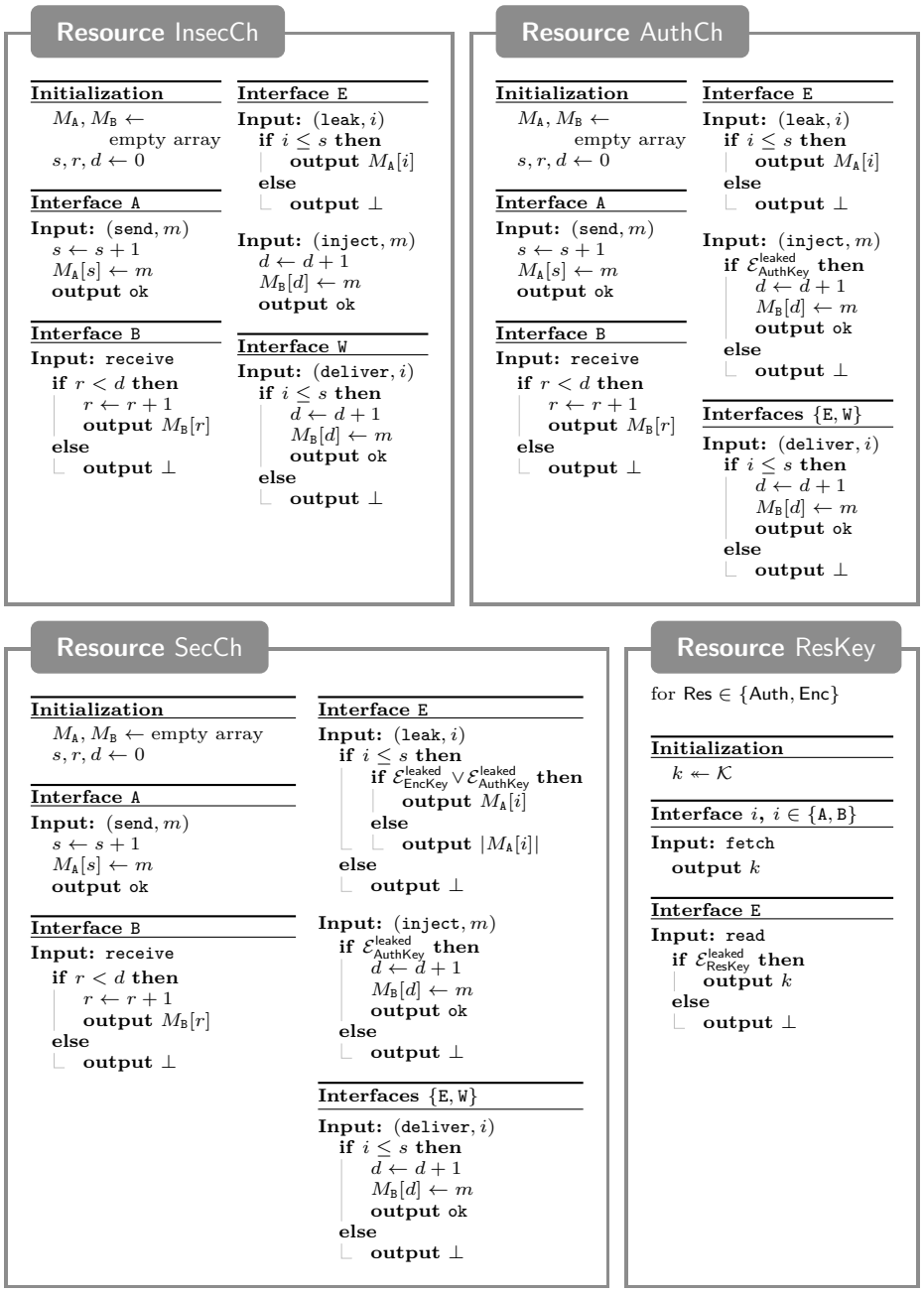

Fig. 1: The resources involved in the encrypt-then-MAC example. Observe how the authenticated and the secure channel degrade their guarantees once the respective keys have been leaked.

Proof. This is a well-known result, which has for instance been sketched in [16].

Then, as a second step one would naïvely like to prove that using such a key, one can construct a corresponding secure channel using an IND-CPA secure encryption scheme. Using the standard simulation-based construction notion, this is however not true, as expressed by the following proposition.

Proposition 5. *Let* SecCh *denote a secure channel that degrades the respective guarantees once the keys have been exposed, as depicted in Figure 1. For any (efficient) simulator* $\boldsymbol{\sigma}_{\mathsf{ENC}}$*, and reduction* ϵ_{CPA} *to the IND-CPA game, we have*

$$\left[\mathsf{EncKey}, \mathsf{AuthCh}\right] \xrightarrow{\not\boldsymbol{\pi}_{\mathsf{ENC}}} \left(\boldsymbol{\sigma}_{\mathsf{ENC}}\mathsf{SecCh}\right)^{\epsilon_{\mathsf{CPA}}},$$

i.e., IND-CPA security does not suffice.

Proof (Sketch). In the first phase, the simulator has to produce, without knowing the message, fake ciphertexts $c_1, c_2, \ldots, c_n$ that look indistinguishable from the real one. For an IND-CPA secure scheme, he can easily do so by encrypting an arbitrary message of the correct length. The moment the encryption key leaks in the real world, he however has to output a uniformly looking key that makes his ciphertexts decrypt to the correct messages. Even knowing the messages by now, this is infeasible unless we assume a so-called non-committing encryption scheme. Furthermore, as long as the requested key is shorter than n, Nielsen [22] showed that NCE cannot be achieved in the standard model by non-interactive protocols.

In the remaining of the section, we introduce a novel type of specification that circumvents this seemingly spurious impossibility result, as intuitively CPA-security should imply that the messages remain confidential until one of the keys leaks.

Remark 1. Insisting that the simulator can explain the ciphertexts (in the traditional notion) formalizes that the ciphertexts are never revealed, even after the key exposure, of any value—in a broader sense than confidentiality. For instance, consider a scenario in which an adversary wants to prove to another party that he managed to wiretap the channel before the transmitted message and the corresponding encryption key are publicly announced. Phrasing that no adversary can succeed requires the simulator to work beyond the public announcement, and achieving it requires non-committing encryption. Otherwise, committing to the ciphertext ahead of the public announcement should convince the other party.

3.2 Guarantees up to Some Point

As we have seen in the motivational example, the confidentiality of the messages should be guaranteed *until* the key is leaked. To phrase this, we, on a high level, only require that the simulator works up to this event. More concretely, we cap the interaction between the distinguisher and the system the moment the corresponding event is triggered.

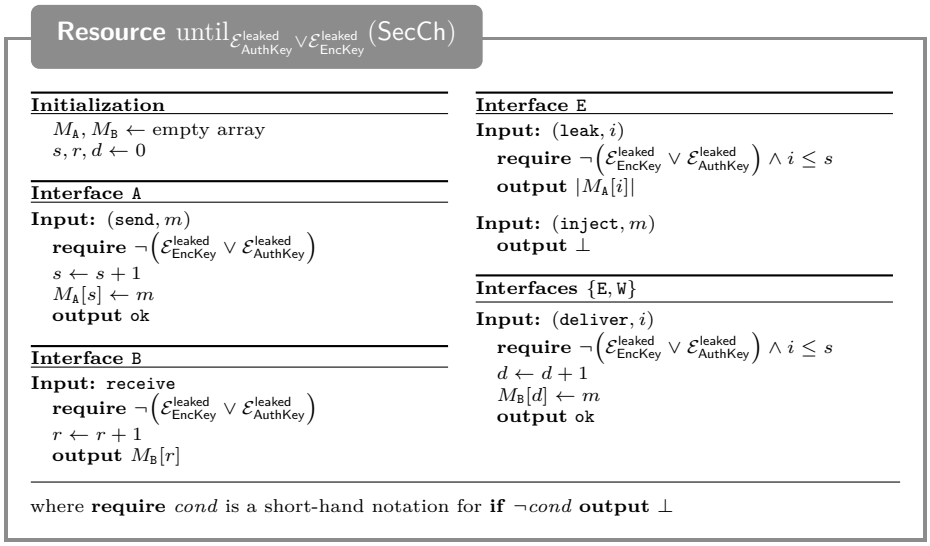

Fig. 2: The secure channel from Figure 1 when halted once either key leaks. In contrast to the original one, this resource never leaks the actual messages.

We formalize this as a novel type of relaxation. To this end, for a resource R, we consider the modified resource that halts once a certain predicate on the global event history is satisfied.

Definition 4. *Let* R *denote a resource, and let* $P(\mathcal{E})$ *denote a monotone predicate on the global event history. That is, if* $\mathcal{E}$ *is a prefix of* $\mathcal{E}'$ *then* $P(\mathcal{E}) \to P(\mathcal{E}')$*. Then, we denote by* $\text{until}_P(\mathsf{R})$ *the resource that behaves like* R *but* halts *the moment* $P(\mathcal{E})$ *becomes true. That is, it no longer triggers any further events and all subsequent (including the one for the query that triggered the condition) answers are the special symbol* $\perp$ *.*

Getting back to our example, consider the resource $\text{until}_P(\mathsf{SecCh})$ for $P(\mathcal{E}) \coloneqq \mathcal{E}^{\text{leaked}}_{\text{AuthKey}} \vee \mathcal{E}^{\text{leaked}}_{\text{EncKey}}$, depicted in Figure 2. Since this resource no longer produces any output once either event occurred, it clearly never leaks the messages to Eve and removes Eve's capability of injecting messages. Hence, the resulting resource closely matches the expected secure channel when ignoring key exposures.

Based on this projection of resources we now define the according relaxation, which maps a system to the set of all systems that behave equivalently up to some event.

Definition 5. *Let* P *be a monotone predicate on the global event history, indicating until when the behavior must be the same as the one of the resource* R*. Then, the induced relaxation on a resource* R*, denoted* $\mathsf{R}^{P]}$*, is defined as*

$$\mathsf{R}^{P]} \coloneqq \left\{\mathsf{S} \;\middle|\; \text{until}_P(\mathsf{R}) = \text{until}_P(\mathsf{S})\right\}$$

We call such a relaxation generally an until-relaxation.

As with the ϵ-relaxation, the statements only become reusable and thus truly composable if we understand how the until-relaxation interacts with the other elements of the framework. For this, first observe that equality up to some point is monotone, i.e., if two resources are equivalent up to some point, they are also equivalent up to every earlier point. This furthermore implies that two until-relaxations add up in the natural manner, as follows.

Theorem 4. *Let* R *and* S *be two resources, and let* P_1 *and* P_2 *be two monotone predicates. Then, we have*

$$\operatorname{until}_{P_1}(\mathsf{R}) = \operatorname{until}_{P_1}(\mathsf{S}) \implies \operatorname{until}_{P_1 \vee P_2}(\mathsf{R}) = \operatorname{until}_{P_1 \vee P_2}(\mathsf{S}).$$

In particular, for every specification $\mathcal{R}$*, we have* $\mathcal{R}^{P_1]} \subseteq \left(\mathcal{R}^{P_1]}\right)^{P_2]} \subseteq \mathcal{R}^{P_1 \vee P_2]}$.

Proof. The first property follows directly from the definition of the until() projection. In order to prove the second property, let $\mathsf{S} \in \left(\mathcal{R}^{P_1]}\right)^{P_2]}$. Then, there exists $\mathsf{T} \in \mathcal{R}^{P_1]}$ such that $\operatorname{until}_{P_2}(\mathsf{S}) = \operatorname{until}_{P2}(\mathsf{T})$. Moreover, there exists $\mathsf{R} \in \mathcal{R}$ such that $\operatorname{until}_{P_1}(\mathsf{T}) = \operatorname{until}_{P1}(\mathsf{R})$. By the first property we, thus, obtain $\operatorname{until}_{P_1 \vee P_2}(\mathsf{S}) = \operatorname{until}_{P_1 \vee P2}(\mathsf{T}) = \operatorname{until}_{P_1 \vee P2}(\mathsf{R})$, concluding the proof. □

Furthermore, on a positive note, the relaxation is compatible with both protocol application and parallel composition, as expressed by the following theorem. Those compatibility properties—analogously to Corollary 1—also directly imply sequential and parallel composition properties. For the lack of use, we however omit explicitly stating them.

Theorem 5. *The until-relaxation is compatible with protocol attachment, i.e.,* $\boldsymbol{\pi}\left(\mathcal{R}^{P]}\right) \subseteq \left(\boldsymbol{\pi}\mathcal{R}\right)^{P]}$ *and with parallel composition, i.e.,* $\left[\mathcal{R}^{P]}, \mathcal{S}\right] \subseteq \left[\mathcal{R}, \mathcal{S}\right]^{P]}$.

Proof. For the first property, consider an arbitrary element of $\boldsymbol{\pi}\left(\mathcal{R}^{P]}\right)$, i.e., $\boldsymbol{\pi}\mathsf{T}$ for an arbitrary $\mathsf{T} \in \mathcal{R}^{P]}$. Hence, there must exist a $\mathsf{R} \in \mathcal{R}$ such that $\operatorname{until}_P(\mathsf{R}) = \operatorname{until}_P(\mathsf{T})$. Using that $\operatorname{until}_P(\boldsymbol{\pi} \operatorname{until}_P(\mathsf{R})) = \operatorname{until}_P(\boldsymbol{\pi}\mathsf{R})$ for any resource R, implies

$$\operatorname{until}_P(\boldsymbol{\pi}\mathsf{R}) = \operatorname{until}_P(\boldsymbol{\pi} \operatorname{until}_P(\mathsf{R})) = \operatorname{until}_P(\boldsymbol{\pi} \operatorname{until}_P(\mathsf{T})) = \operatorname{until}_P(\boldsymbol{\pi}\mathsf{T})$$

and, thus, $\boldsymbol{\pi}\mathsf{T} \in \left(\boldsymbol{\pi}\mathsf{R}\right)^{P]} \subseteq \left(\boldsymbol{\pi}\mathcal{R}\right)^{P]}$. The second property follows analogously observing that $\operatorname{until}_P([\operatorname{until}_P(\mathsf{R}), \mathsf{S}]) = \operatorname{until}_P([\mathsf{R}, \mathsf{S}])$ for all R and S. □

Unfortunately, however, the until-relaxation does not commute directly with the ϵ-relaxation, as expressed by the following theorem.

Theorem 6. *There exist specifications* $\mathcal{R}$ *and* $\mathcal{S}$*, a monotone predicate* P*, and a function* ϵ *mapping distinguishers to values in* $[0, 1]$ *such that*

$$\left(\mathcal{R}^{P]}\right)^{\epsilon} \not\subseteq \left(\mathcal{R}^{\epsilon}\right)^{P]} \qquad \text{and} \qquad \left(\mathcal{S}^{\epsilon}\right)^{P]} \not\subseteq \left(\mathcal{S}^{P]}\right)^{\epsilon}.$$

Proof. The proof can be found in Appendix C.2.

This not only raises the question which order actually corresponds to the intuitive interpretation of such a combination—the set of all systems which behave equally until the condition is triggered assuming the assumption of ϵ is valid—but also restricts reuse of such statement. That is if one construction step assumes $\mathcal{S}^{P]}$ in order to construct some specification $\mathcal{T}$, and another one constructs $\mathcal{S}^{\epsilon}$, then adjusting them to compose is non-trivial, conflicting the goal of a composable framework. As a consequence, we will introduce a combined relaxation in Section 3.4, resolving both issues.

3.3 Guarantees From Some Point On

In this section, we now consider the complementing type of guarantees: guarantees that only hold from a certain point on. Formalizing such guarantees in a model where an adaptive environment interacts with the resource is, however, quite delicate. In this work, we thus opt for a rather simple (and restricted) version of it, where we use again a monotone condition on the global event history. We then define the projection that disables access to a system R before that condition is met. Clearly, the condition must rely on "external" events only (the ones not controlled by R), i.e., satisfying it must not require accessing the resource itself.

Definition 6. *Let $P(\mathcal{E})$ denote a monotone predicate on the global event history. For a resource R, let $\mathrm{from}_P(\mathsf{R})$ denote the resource that behaves like R, except that it only accepts queries once $P(\mathcal{E})$ is true (and before only returns $\bot$).*

For instance, the resource $\mathrm{from}_{\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}}(\mathsf{SecCh})$ only answers queries once the environment triggered the event $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$. Thus, in contrast to SecCh, this resource always leaks the full message of the adversary, in line with our intuition that it describes the behavior after the key has been exposed. A formal definition of the resource is depicted in Appendix C.1.

Based on this projection, we now introduce the corresponding relaxation.

Definition 7. *Let $P(\mathcal{E})$ be a monotone predicate, indicating from which point on the behavior must be the same as the one of the resource R. Then, the induced relaxation on a resource R, denoted $\mathsf{R}^{[P}$, is defined as*

$$\mathsf{R}^{[P} \coloneqq \left\{ \mathsf{S} \;\middle|\; \mathrm{from}_P(\mathsf{R}) = \mathrm{from}_P(\mathsf{S}) \right\}$$

We call such a relaxation generally a from-relaxation.

The way the from-relaxation interacts with the other elements of the theory is analogous to the until-relaxation. First, two from-relaxations add up naturally: if we relax the guarantees offered by a specification to only hold from the moment P_1 is satisfied, and then further relax them to only hold once P_2 is satisfied, then the guarantees only hold once $P_1 \wedge P_2$ is satisfied.

Theorem 7. *Let* R *and* S *be two resources, and let* P_1 *and* P_2 *be monotone predicates on the global event history. Then, we have*

$$\mathrm{from}_{P_1}(\mathsf{R}) = \mathrm{from}_{P_1}(\mathsf{S}) \implies \mathrm{from}_{P_1 \wedge P_2}(\mathsf{R}) = \mathrm{from}_{P_1 \wedge P_2}(\mathsf{S}).$$

In particular, for every specification $\mathcal{R}$*, we have* $\mathcal{R}^{[P_1} \subseteq \left(\mathcal{R}^{[P_1}\right)^{[P_2} \subseteq \mathcal{R}^{[P_1 \wedge P_2}$*.*

Proof. The proof is analogous to the one of Theorem 4.

Second, the relaxation is compatible with protocol application and parallel composition, which moreover implies that it graciously interacts with the basic construction notion.

Theorem 8. *The from-relaxation is compatible with protocol application, i.e.,* $\boldsymbol{\pi}\left(\mathcal{R}^{[P}\right) \subseteq \left(\boldsymbol{\pi}\mathcal{R}\right)^{[P}$ *and with parallel composition, i.e.,* $\left[\mathcal{R}^{[P}, \mathcal{S}\right] \subseteq \left[\mathcal{R}, \mathcal{S}\right]^{[P}$*.*

Proof. The proof is analogous to the one of Theorem 5.

Analogously to the until-relaxation, the from-relaxation, however, does not commute with the ϵ-relaxation.

Theorem 9. *There exist specifications* $\mathcal{R}$ *and* $\mathcal{S}$*, a monotone predicate* $P(\mathcal{E})$*, and a function* ϵ *mapping distinguishers to values in* $[0, 1]$ *such that*

$$\left(\mathcal{R}^{[P}\right)^{\epsilon} \not\subseteq \left(\mathcal{R}^{\epsilon}\right)^{[P} \qquad \text{and} \qquad \left(\mathcal{S}^{\epsilon}\right)^{[P} \not\subseteq \left(\mathcal{S}^{[P}\right)^{\epsilon}.$$

Proof. The proof is presented in Appendix C.3.

Finally, consider the interaction between the from- and the until-relaxation. While the from-projection and the until-projection commute, i.e.,

$$\mathrm{from}_{P_1}(\mathrm{until}_{P_2}(\mathsf{R})) = \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{R})),$$

it is an interesting open questions whether the two respective relaxations actually commute. As a consequence, we introduce a combined from-until relaxation in the next subsection.

3.4 The Interval-Wise Relaxation

As we have seen, the ϵ-relaxation, the until-relaxation, and the from-relaxation do not mutually commute. This impedes modularity and reusability of the statements. Furthermore, it also deteriorates the intuitive semantics of the statements: if for instance we want to express that a systems behaves like a certain ideal up to some point, and under certain computational assumptions, which order of the relaxations is the right one and should be proven? To alleviate those issues, in this section, we introduce two combined relaxations that build on the atomic ones introduced in the previous section. We then show that they both have natural semantics and clean properties.

First, we consider a relaxation that combines the from- and until-relaxation, thereby alleviating the issue that those relaxations might not commute.

Definition 8. *Let $P_1(\mathcal{E})$ and $P_2(\mathcal{E})$ be two monotone predicates, indicating from when until when the resource must behave like* R*. We then define the following relaxation*

$$\mathsf{R}^{[P_1,P_2]} \coloneqq \left\{\mathsf{S} \;\middle|\; \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{R})) = \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{S}))\right\}.$$

While this combined relaxation apparently neither corresponds to $\left(\mathsf{R}^{[P_1}\right)^{P_2]}$ nor $\left(\mathsf{R}^{P_2]}\right)^{[P_1}$, it interestingly corresponds to the transitive closure thereof. Taking the transitive closure, moreover, also restores symmetry, i.e., $\mathsf{S} \in \mathsf{R}^{[P_1,P2]} \Leftrightarrow \mathsf{R} \in \mathsf{S}^{[P_1,P2]}$, lost by each of the two individual combinations. Since a relaxation should intuitively correspond to some sort of "almost-as-good" relation (each of the elements just be of equal preference), this strongly indicates that the combined relaxation is the right one to use.

Theorem 10. *For any resource* R *and any monotone predicates P_1 and P_2, we have*

$$\begin{aligned}\mathsf{R}^{[P_1,P_2]} &= \bigcup_{n\in\mathbb{N}} \left(\bigcup \left\{\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \mid \forall i \le n : \phi_i \in \{P_2],[P_1\}\right\}\right) \\ &= \left(\left(\mathsf{R}^{[P_1}\right)^{P_2]}\right)^{[P_1} = \left(\left(\mathsf{R}^{P_2]}\right)^{[P_1}\right)^{P_2]},\end{aligned}$$

where $\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_n}$ is a shorthand notation for first applying ϕ_1, then ϕ_2, until ϕ_n.

Proof. The proof can be found in Appendix C.4.

We can now leverage this alternative definition to directly derive properties about the combined relaxations based on the proven properties of the two underlying ones. In particular, we can show that two such relaxations add up in the expected manner and are compatible with both protocol application as well as parallel composition.

Theorem 11. *For every specification $\mathcal{R}$, and all monotone predicates P_1, P_2, P_1', and P_2', we have $\left(\mathcal{R}^{[P_1,P_2]}\right)^{[P_1',P_2']} \subseteq \mathcal{R}^{[P_1\wedge P_1', P_2\vee P_2']}$.*

Proof. This follows directly by combining Theorems 4, 7 and 10. □

Theorem 12. *The combined relaxation is both compatible with protocol application, i.e., $\boldsymbol{\pi}\left(\mathcal{R}^{[P_1,P_2]}\right) \subseteq \left(\boldsymbol{\pi}\mathcal{R}\right)^{[P_1,P_2]}$ and with parallel composition, i.e., $\left[\mathcal{R}^{[P_1,P_2]},\mathcal{S}\right] \subseteq \left[\mathcal{R},\mathcal{S}\right]^{[P_1,P_2]}$.*

Proof. By Theorem 10 we have that $\mathcal{R}^{[P_1,P_2]} = \left(\left(\mathsf{R}^{[P_1}\right)^{P_2]}\right)^{[P_1}$. Using the compatibility of the from-relaxation and until-relaxation, i.e., Theorems 5 and 8, directly implies the desired properties. □

As we have seen, neither the until- nor the from-relaxation commute with the computational ϵ-relaxation, and the same holds true for the from-until-relaxation as well. As a consequence, neither $(\mathsf{R}^{[P_1,P_2]})^{\epsilon}$ nor $(\mathsf{R}^{\epsilon})^{[P_1,P_2]})$ seems to capture the set of all systems that behave like $\mathcal{R}$ in the interval $[P_1, P_2]$ assuming that the computational problem encoded in ϵ is hard. In the spirit of the combined from-until relaxation, we solve this issue by introducing a combined relation. Since the ϵ relaxation is not idempotent, but the epsilons add up, taking the transitive closure, however, does not match the desired relaxation but the following restricted version of transitive closure does.

Definition 9. *For two monotone predicates P_1 and P_2, and a function ϵ mapping distinguishers to values in $[0,1]$, we define the following relaxation:*

$$\mathsf{R}^{[P_1,P_2]:\epsilon} \coloneqq \left(\left(\mathsf{R}^{[P_1,P_2]}\right)^{\epsilon}\right)^{[P_1,P_2]},$$

and call such a relaxation an interval-wise relaxation.

We now prove that the interval-wise relaxation has all the desired properties.

Theorem 13. *Let P_1 and P_2 be two monotone predicates, and let ϵ be a function mapping distinguishers to values in $[0,1]$. Then, for any specification $\mathcal{R}$ we have*

$$\left(\mathcal{R}^{[P_1,P_2]:\epsilon}\right)^{[P_1',P_2']:\epsilon'} \subseteq \mathcal{R}^{[P_1\wedge P_1', P_2\vee P_2']:\epsilon_{[P_1\wedge P_1', P_2\vee P_2']}+\epsilon'_{[P_1\wedge P_1', P_2\vee P_2']}},$$

where $\epsilon_{[P_1\wedge P_1', P_2\vee P_2']}(D) \coloneqq \epsilon(\mathsf{D}\circ \mathrm{until}_{P_2\vee P_2'} \circ \mathrm{from}_{P_1\wedge P_1'})$, i.e., the performance of the distinguisher interacting with the projected resource, and analogously for $\epsilon'_{[P_1\wedge P_1', P_2\vee P_2']}$.

Proof. The proof can be found in Appendix C.5.

Theorem 14. *The interval-wise relaxation is compatible with protocol application, i.e., $\boldsymbol{\pi}\left(\mathcal{R}^{[P_1,P_2]:\epsilon}\right) \subseteq \left(\boldsymbol{\pi}\mathcal{R}\right)^{[P_1,P_2]:\epsilon_{\boldsymbol{\pi}}}$ and with parallel composition, i.e., $\left[\mathcal{R}^{[P_1,P_2]:\epsilon}, \mathcal{S}\right] \subseteq \left[\mathcal{R},\mathcal{S}\right]^{[P_1,P_2]:\epsilon_{\mathcal{S}}}$.*

Proof. By definition we have $\mathsf{R}^{[P_1,P_2]:\epsilon} \coloneqq \left(\left(\mathsf{R}^{[P_1,P_2]}\right)^{\epsilon}\right)^{[P_1,P_2]}$. Using the compatibility of the ϵ-relaxation and the from-until-relaxation, i.e., Theorems 3 and 12, directly implies the result.

3.5 The Resulting Construction Notion

Based on the interval-wise relaxation, we now introduce our new construction notion. We thereby want to formalize guarantees for several intervals. For instance, in our running example, we want to express that the adversary does not learn anything about the messages *until* the keys leak and that authenticity holds until the MAC-key leaks. We thereby simply phrase them separately and then consider the conjunction: if the specification $\boldsymbol{\sigma}_1\mathcal{S}_1$ formalizes the confidentiality guarantee

and $\boldsymbol{\sigma}_2\mathcal{S}_2$ the authenticity guarantee, then proving that $\mathcal{R} \subseteq (\boldsymbol{\sigma}_1\mathcal{S}_1) \cap (\boldsymbol{\sigma}_2\mathcal{S}_2)$ simply states that both guarantees hold. In particular, there is no need for the two simulators to be the same one.

Let Ω denote a set of tuples $(P_1, P_2, \epsilon, \boldsymbol{\sigma})$, where P_1 and P_2 are monotone predicates on the global event history, ϵ is a function mapping distinguishers to values in $[0, 1]$, and $\boldsymbol{\sigma}$ denotes a simulator. We then consider constructions of the following type:

$$\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \bigcap_{(P_1,P_2,\epsilon,\boldsymbol{\sigma})\in\Omega} (\boldsymbol{\sigma}\mathcal{S})^{[P_1,P_2]:\epsilon}.$$

That is, each element in Ω describes a time-interval in which the elements in $\boldsymbol{\pi}\mathcal{R}$ can be abstracted as elements in $\mathcal{S}$—with respect to the simulator $\boldsymbol{\sigma}$ and error ϵ.

Application to the running example. In our example, we want to phrase that the symmetric encryption protocol constructs the secure channel from the authenticated one in the corresponding intervals.

Proposition 6. *Let $\boldsymbol{\pi}_{\mathsf{ENC}} = (\pi_{\mathsf{enc}}, \pi_{\mathsf{dec}})$ denote the protocol securing communication using a symmetric encryption scheme. Then, for the resources in Figure 1, there exist (efficient) simulators $\boldsymbol{\sigma_1}$, $\boldsymbol{\sigma_2}$, and $\boldsymbol{\sigma_3}$ such that*

$$\big[\mathsf{EncKey}, \mathsf{AuthCh}\big] \xrightarrow{\boldsymbol{\pi}_{\mathsf{ENC}}} \bigcap_{(P_1,P_2,\epsilon,\boldsymbol{\sigma})\in\Omega} (\boldsymbol{\sigma}\mathsf{SecCh})^{[P_1,P_2]:\epsilon}$$

for

$$\Omega \coloneqq \big\{\big(\mathtt{true}, \mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}} \vee \mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}, \epsilon_{\mathsf{CPA}}, \boldsymbol{\sigma}_1\big), \big(\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}, \mathtt{false}, 0, \boldsymbol{\sigma}_2\big), \big(\mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}, \mathtt{false}, 0, \boldsymbol{\sigma}_3\big)\big\},$$

where ϵ_{CPA} denotes a simple reduction from distinguishing the secure and authenticated channel (without key leakage) to the IND-CPA game.

Proof (Sketch). Let $\boldsymbol{\sigma}_1$ be the usual simulator that creates fake ciphertexts by encrypting $0^{|m|}$ instead of m, when queried for the leakage of the channel. By definition, we have $\mathrm{until}_{\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}} \vee \mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}}(\boldsymbol{\sigma}_1\mathsf{SecCh}) \in (\boldsymbol{\sigma}_1\mathsf{SecCh})^{[\mathtt{true}, \mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}} \vee \mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}]}$. Thus, if the encryption scheme is IND-CPA secure, then it is easy to see that

$$\mathrm{until}_{\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}} \vee \mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}}(\boldsymbol{\pi}_{\mathsf{ENC}}\big[\mathsf{EncKey}, \mathsf{AuthCh}\big]) \in \big((\boldsymbol{\sigma}_1\mathsf{SecCh})^{[\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}} \vee \mathcal{E}^{\mathsf{leaked}}_{\mathsf{AuthKey}}}\big)^{\epsilon_{\mathsf{CPA}}},$$

Daniel: clarify what condition means here!

which proves the construction inside the first interval. For the second interval, consider the simulator $\boldsymbol{\sigma}_2$ that encrypts the real messages. Using correctness of the encryption scheme, it is easy to see that

$$\mathrm{from}_{\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}}(\boldsymbol{\pi}_{\mathsf{ENC}}\big[\mathsf{EncKey}, \mathsf{AuthCh}\big]) = \mathrm{from}_{\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}}(\boldsymbol{\sigma}_2\mathsf{SecCh}),$$

and analogous for the third interval using the same simulator $\boldsymbol{\sigma}_3 = \boldsymbol{\sigma}_2$. □

Composition. Finally, we finish the section by stating the composition guarantees of this type of construction statement. It follows directly from the properties proven about the interval-wise relaxation in Theorems 13 and 14.

Theorem 15. *Let $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$ be arbitrary specifications, let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ be arbitrary protocols, and let Ω and Ω' be arbitrary interval-wise guarantees. Then, we have*

$$\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \bigcap_{(P_1,P_2,\epsilon,\boldsymbol{\sigma})\in\Omega} (\boldsymbol{\sigma}\mathcal{S})^{[P_1,P_2]:\epsilon} \quad \wedge \quad \mathcal{S} \xrightarrow{\boldsymbol{\pi}'} \bigcap_{(P_1,P_2,\epsilon,\boldsymbol{\sigma})\in\Omega'} (\boldsymbol{\sigma}\mathcal{T})^{[P_1,P_2]:\epsilon}$$
$$\implies \quad \mathcal{R} \xrightarrow{\boldsymbol{\pi}'\circ\boldsymbol{\pi}} \bigcap_{\substack{(P_1,P_2,\epsilon,\boldsymbol{\sigma})\in\Omega \\ (P_1',P_2',\epsilon',\boldsymbol{\sigma}')\in\Omega'}} (\boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T})^{[P_1\wedge P_1',P_2\vee P_2']:\tilde{\epsilon}},$$

where $\tilde{\epsilon} := (\epsilon_{\boldsymbol{\pi}'})_{[P_1\wedge P_1',P_2\vee P_2']} + (\epsilon'_{\boldsymbol{\sigma}})_{[P_1\wedge P_1',P_2\vee P_2']}$. Furthermore, we have

$$\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \bigcap_{(P_1,P_2,\epsilon,\boldsymbol{\sigma})\in\Omega} (\boldsymbol{\sigma}\mathcal{S})^{[P_1,P_2]:\epsilon} \implies [\mathcal{R},\mathcal{T}] \xrightarrow{\boldsymbol{\pi}} \bigcap_{(P_1,P_2,\epsilon,\boldsymbol{\sigma})\in\Omega} (\boldsymbol{\sigma}[\mathcal{S},\mathcal{T}])^{[P_1,P_2]:\epsilon_{\mathcal{T}}}.$$

Proof. The proof is stated in Appendix C.6.

Note that this construction notion subsumes all those previously introduced in this work. In particular, instantiating $P_1 = \texttt{true}$, $P_2 = \texttt{false}$, $\epsilon(\mathsf{D}) = 0$, and $\boldsymbol{\sigma} = \mathsf{id}$, i.e., the identity converter, yields $(\mathsf{id}\mathcal{S})^{[\texttt{true},\texttt{false}]:0} = \mathcal{S}$. As a consequence, the above composition theorem also allows to combine constructions according to each of the notions introduced in this work. For instance, in our example, we can compose the construction of AuthCh from Proposition 4 (according to the standard notion) with the interval-wise construction of SecCh from Proposition 6.

4 Application to Commitments and Coin-Tossing

In this section, we present a composable formalization of (information-theoretically binding) commitments that can be constructed in the plain model. To this end, we take advantage of Constructive Cryptography allowing us to formalize the properties of commitment schemes—correctness, binding, and hiding—each as individual specifications. Especially, we formalize hiding using the interval-wise guarantees introduced in the previous section. We then apply Blum's coin-tossing protocol on top of it. While, obviously, the resulting specifications are not sufficient to be used as a CRS, we show that it is unbiased, and thus can for instance be used for lotteries.

4.1 Information-Theoretically Binding Commitments

While UC commitments [8] provide clean and strong guarantees, unfortunately they intrinsically require setup assumptions such as a common reference string.

Nevertheless, for many protocols, regular commitments only satisfying the classical game-based properties seem to suffice. This raises the question: can we formalize a weaker yet composable security notion for (non-interactive) commitments?

In Constructive Cryptography, the security of a commitment scheme is phrased using three different constructions [19], for each set of potentially dishonest parties (ignoring the case of both parties being dishonest). Typically, this is presented as one construction parametrized in the set of honest parties, where the ideal specification consists of a filtered resource. That is, for each party P, a filter ϕ_{P} is specified that when connected to the resource limits the honest party's capabilities. However, akin to the way as an ideal specification can in principle consist of the intersection of specifications that are expressed in different ways, there is no fundamental reason for those three construction statements' specifications to be of some unified type. As a result, we henceforth focus on specifying each property—hiding, binding, and correctness—individually.

Obviously, correctness is the easiest to formalize.

Definition 10. *Let* $\boldsymbol{\pi}_{\mathsf{com}} = (\pi^{\mathtt{A}}_{\mathsf{com}}, \pi^{\mathtt{B}}_{\mathsf{com}})$ *denote a non-interactive commitment protocol where* $\mathtt{A}$ *commits a value* $m \in \mathcal{M}$ *towards* $\mathtt{B}$*. Then, the scheme is said to be (perfectly)* correct *if*

$$\left[\mathsf{Ch}_1^{\mathtt{A}\to\mathtt{B}}, \mathsf{Ch}_2^{\mathtt{A}\to\mathtt{B}}\right] \xrightarrow{\boldsymbol{\pi}_{\mathsf{com}}} \mathsf{Com}_{\mathcal{M}}^{\mathtt{A}\to\mathtt{B}},$$

where $\mathsf{Com}_{\mathcal{M}}^{\mathtt{A}\to\mathtt{B}}$ *denotes the commitment resource defined in Figure 3, and* $\mathsf{Ch}_1^{\mathtt{A}\to\mathtt{B}}$ *and* $\mathsf{Ch}_2^{\mathtt{A}\to\mathtt{B}}$ *denote two single-message communications channels from* $\mathtt{A}$ *to* $\mathtt{B}$*.*

Now we proceed to formalize the hiding property. On an intuitive level, (computational) hiding of a non-interactive commitment scheme requires that the commitment string must not reveal any information about the committed value to the receiver $\mathtt{B}$, until the commitment is opened. Clearly, we can directly apply our notion from Section 3 and formalize this using an interval-wise relaxation.

Definition 11. *Let* $\boldsymbol{\pi}_{\mathsf{com}} = (\pi^{\mathtt{A}}_{\mathsf{com}}, \pi^{\mathtt{B}}_{\mathsf{com}})$ *denote a non-interactive commitment protocol. Then, the scheme is said to be (computationally)* hiding *if*

$$\left[\mathsf{Ch}_1^{\mathtt{A}\to\mathtt{B}}, \mathsf{Ch}_2^{\mathtt{A}\to\mathtt{B}}\right] \xrightarrow{\pi^{\mathtt{A}}_{\mathsf{com}}} \left(\sigma^{\mathtt{B}}_{\mathsf{com}}\mathsf{Com}_{\mathcal{M}}^{\mathtt{A}\to\mathtt{B}}\right)^{[\mathtt{true},\mathcal{E}^{\mathsf{opened}}]:\epsilon},$$

for some simulator $\sigma^{\mathtt{B}}_{\mathsf{com}}$ *and some computational assumption encoded in* ϵ*.*

The situation is more challenging with binding. The UC formalization, and analogously $\mathsf{Com}_{\mathcal{M}}^{\mathtt{A}\to\mathtt{B}}$, requires that the adversary inputs the value to which it commits to in the initial phase, in order to formalize that it then cannot be altered anymore. This, however additionally enforces that the value is known to the adversary during the commit phase. On a technical level, since in the real-world the adversary can input an arbitrary commitment string, the simulator needs to be able to extract the value from it. Without setup, this fundamentally contradicts the hiding property. Note, however, that enforcing the simulator to input the committed value is just one, albeit convenient, manner to specify that the value is fixed at this point.

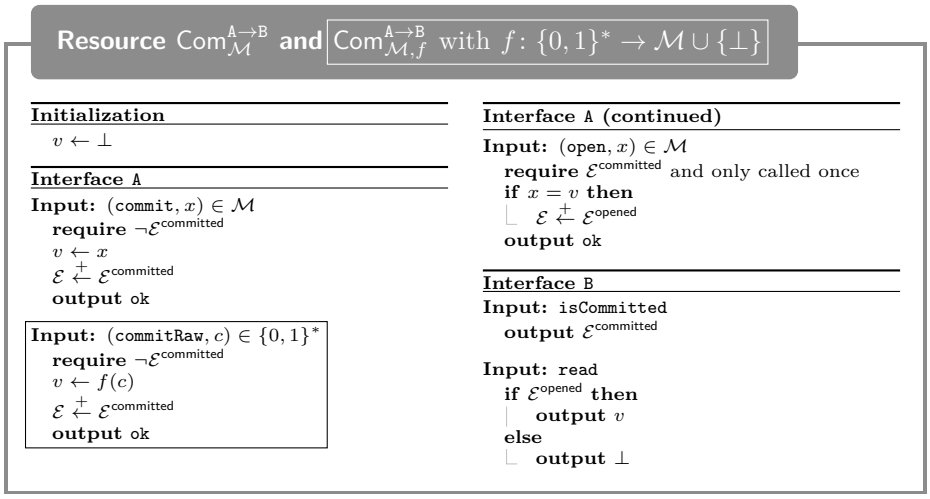

Fig. 3: The commitment resources for message space $\mathcal{M}$. In the basic version, Alice has to specify the value at the time of commitment, whereas in the unfiltered version she additionally has the ability to commit to $f(c)$, where f is a parameter of the resource.

In the following, we consider perfect (or information-theoretically secure) commitments only. The binding property is then formalized in a straight-forward manner: The commitment string of a non-interactive commitment scheme for the message space $\mathcal{M}$ must uniquely determine the committed value. This is encoded as a function $f\colon \{0,1\}^* \to \mathcal{M} \cup \{\bot\}$ that maps the commitment string either to a message m, or to $\bot$ indicating that it is malformed. Then, we consider the resource $\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\mathcal{M},f}$, depicted in Figure 3, which allows the dishonest A to input an arbitrary string x in order to commit $v = f(x)$, and only at the time of opening the value v has to be known. Clearly, no matter how f is defined, $f(c)$ is a fixed value that cannot be altered anymore. Hence, the following definition faithfully captures perfect hiding.

Definition 12. *Let $\boldsymbol{\pi}_{\mathsf{com}} = (\pi^{\mathtt{A}}_{\mathsf{com}}, \pi^{\mathtt{B}}_{\mathsf{com}})$ denote a non-interactive commitment protocol where A commits a value $m \in \mathcal{M}$ towards B. Then, the scheme is said to be* perfectly hiding *if there exists an efficient simulator $\sigma^{\mathtt{A}}_{\mathsf{com}}$ such that*

$$\left[\mathsf{Ch}^{\mathtt{A}\to\mathtt{B}}_1, \mathsf{Ch}^{\mathtt{A}\to\mathtt{B}}_2\right] \xrightarrow{\pi^{\mathtt{B}}_{\mathsf{com}}} \left\{\sigma^{\mathtt{A}}_{\mathsf{com}}\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\mathcal{M},f} \;\middle|\; f\colon \{0,1\}^* \to \mathcal{M} \cup \{\bot\}\right\},$$

where $\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\mathcal{M},f}$ denotes the extended commitment resource defined in Figure 3.

As a side note, observe that the resource $\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\mathcal{M}}$ can trivially be expressed as a filtered version of $\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\mathcal{M},f}$, where the filter $\phi_{\mathtt{A}}$ removes access to the commitRaw oracle. That is, we obviously have $\phi_{\mathtt{A}}\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\mathcal{M},f} = \mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\mathcal{M}}$ for every function f.

Remark 2. Observe that in the ideal specification of the binding property, the resources $\mathsf{Com}^{\mathsf{A}\to\mathsf{B}}_{\mathcal{M},f}$, the function f is not necessary efficiently computable. Actually, for a hiding scheme, f cannot be efficiently computable. This, however, does not imply that the overall specification has to contain resources that are not efficiently implementable, as clearly the real-world resource is efficient, and yet corresponds to $\sigma^{\mathsf{A}}\mathsf{Com}^{\mathsf{A}\to\mathsf{B}}_{\mathcal{M},f}$ for an inefficient f. In other words, the decomposition, containing the inefficient resource, is just one way of describing an overall efficient resource. The overall specifications could, thus, be required to consist of only efficient resources by intersecting them with a set of efficiently implementable resources, which we did not make explicit here focusing only on the security properties. This is somewhat reminiscent of the solution proposed by Broadnax et al. [6] to deal with inefficient simulators in a manner that retains the expected composition guarantees.

ElGamal commitments. We briefly consider a variant of ElGamal commitments as a concrete instantiation of the above formalized notion. Let $\mathbb{G} = \langle g \rangle$ denote a cyclic group of order n with generator g.

- To commit to a message $m \in \mathbb{G}$, $((g^a, g^b, m \cdot g^{ab}), (a, b)) \leftarrow \mathrm{Commit}(m)$ for $a, b \in \mathbb{Z}_n$ uniformly at random. That is, the commitment string is $(g^a, g^b, m \cdot g^{ab})$ and the opening value (a, b).
- $\mathrm{Open}(c, (a, b)) \coloneqq c \cdot g^{-ab}$

Proposition 7. *Let $\boldsymbol{\pi}_{\mathsf{ElG\text{-}com}}$ denote the pair of converters implementing the aforementioned ElGamal commitment scheme. Then, $\boldsymbol{\pi}_{\mathsf{ElG\text{-}com}}$ satisfies correctness, hiding (under the DDH assumption), and binding according to Definitions 10 to 12, respectively.*

Proof (Sketch). It is easy to see that the our correctness condition holds. Furthermore, with the simulator $\sigma^{\mathsf{B}}_{\mathsf{com}}$ outputting a random triple of group elements as commitment string, hiding holds under the DDH assumption, i.e., for ϵ encoding an appropriate reduction to the DDH problem. Finally, consider the function f that maps $(U, V, W) \in \mathbb{G}^3$ to $W \cdot g^{-\mathrm{DL}_g(U)\cdot\mathrm{DL}_g(V)}$ and all other bit-strings to $\perp$. For this function, it is easy to see that a simulator $\sigma^{\mathsf{A}}_{\mathsf{com}}$ exists such that the construction that formalizes binding holds. □

4.2 Coin-Tossing

In this section, we consider Blum's simple coin-tossing protocol [4]. The protocol assumes to have a commitment resource from Alice to Bob, and a communication channel in the reverse direction, at its disposal. It then proceeds as follows: Alice chooses $X \in \{0, 1\}$ uniformly at random and commits to it. Once Bob is sure that Alice committed, he chooses $Y \in \{0, 1\}$ uniformly at random and sends it over to Alice (in clear). Finally, Alice opens the commitment and both parties output $Z = X \oplus Y$.

Clearly, this protocol does not provide fairness—even when instantiated with a UC-secure commitment. This is due to the fact that both parties can always

choose to abort the protocol by not responding, and in particular Alice can do so *after* she has seen the result. When instantiating the commitment with the resource constructed in the last section, one even obtains a weaker resource. Note that this is inherent for our construction being in the plain model, as otherwise it could be used as the bit of a CRS, contradicting well-known impossibility results.

In a nutshell, the resource obtained by our construction guarantees that the output is not biased, but does not exclude that during the opening phase, one of the parties learns some trapdoor allowing it to distinguish it from a uniformly random value. For example, our formalization would allow the resulting bit to be the first bit of a PRG's output, while leaking the seed during the opening phase. Note that such a coin toss resource is still useful, for instance for lotteries. First, if the resulting bit is just used to determine which party gets some good, then bias-resistance is obviously good enough irrespective of the fact that the parties might be aware that the result is only pseudo-random. Second, in a simple lottery where people's preferences are obvious, fairness can be achieved by declaring the party that caused the abort to have lost.

The coin-toss resource. The ideal specification is expressed in terms of the resources $\mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\mathcal{M}}$ and $\mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\mathcal{M},f}$, where the former denotes a restricted version of the latter. The resource $\mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\mathcal{M}}$ initially draws an element $Z \in \mathcal{M}$ uniformly at random. In order for the coin-toss Z to become available to the parties, $\mathtt{A}$ has to initiate it, and $\mathtt{B}$ has to respond afterwards. From this point on, $\mathtt{A}$ can obtain Z and then decide whether the value should also be released to $\mathtt{B}$. In the resource $\mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\mathcal{M},f}$, $\mathtt{A}$ furthermore can query once a leakage $f(c)$, of some potentially inefficient function f. A formal definition of the resources can be found in Figure 4.

The constructions. First, consider correctness. It is easy to see that the following construction holds, i.e., two honest parties actually get to agree on a uniform random bit.

Proposition 8. *Let $\boldsymbol{\pi}_{\mathsf{CT}} \coloneqq (\pi^{\mathtt{A}}_{\mathsf{CT}}, \pi^{\mathtt{B}}_{\mathsf{CT}})$ denote the pair of converters implementing Blum's protocol (c.f. Appendix D.1 for a formal definition). Then, we have*

$$\left[\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\{0,1\}}, \mathsf{Ch}^{\mathtt{B}\to\mathtt{A}}\right] \xrightarrow{\boldsymbol{\pi}_{\mathsf{CT}}} \mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\{0,1\}},$$

and thus

$$\left[\mathsf{Ch}^{\mathtt{A}\to\mathtt{B}}_{1}, \mathsf{Ch}^{\mathtt{A}\to\mathtt{B}}_{2}, \mathsf{Ch}^{\mathtt{B}\to\mathtt{A}}\right] \xrightarrow{\boldsymbol{\pi}_{\mathsf{CT}} \circ \boldsymbol{\pi}_{\mathsf{com}}} \mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\{0,1\}},$$

for any commitment scheme $\boldsymbol{\pi}_{\mathsf{com}}$ satisfying Definition 10 (correctness).

Second, consider the guarantee for an honest initiator $\mathtt{A}$.

Proposition 9. *Let $\boldsymbol{\pi}_{\mathsf{CT}} \coloneqq (\pi^{\mathtt{A}}_{\mathsf{CT}}, \pi^{\mathtt{B}}_{\mathsf{CT}})$ denote the pair of converters implementing Blum's protocol. Then, there exists an efficient simulator $\sigma^{\mathtt{B}}_{\mathsf{CT}}$ such that*

$$\left[\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\{0,1\}}, \mathsf{Ch}^{\mathtt{B}\to\mathtt{A}}\right] \xrightarrow{\pi^{\mathtt{A}}_{\mathsf{CT}}} \sigma^{\mathtt{B}}_{\mathsf{CT}} \mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\{0,1\}},$$

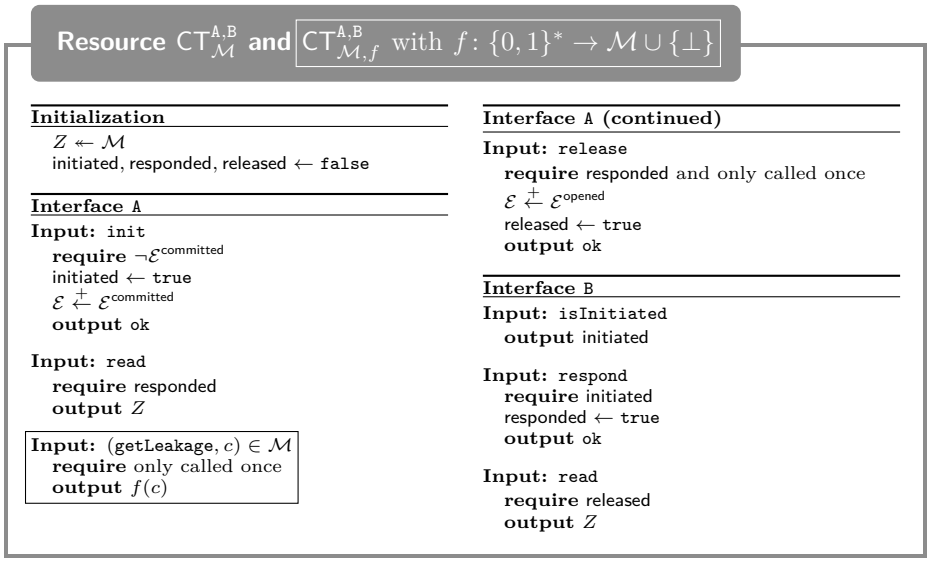

Fig. 4: The coin-toss resources for coin space $\mathcal{M}$. In the unfiltered version, Alice additionally has the capability to once obtain a leakage to $f(c)$, where f is a parameter of the resource. Note that neither version provides fairness, as Alice can always chooses to not release the value after having seen it.

and thus, for any commitment scheme $\boldsymbol{\pi}_{\mathsf{com}}$ *satisfying Definition 11 (hiding), we have*

$$\left[\mathsf{Ch}_1^{\mathtt{A}\to\mathtt{B}}, \mathsf{Ch}_2^{\mathtt{A}\to\mathtt{B}}, \mathsf{Ch}^{\mathtt{B}\to\mathtt{A}}\right] \xrightarrow{\pi^{\mathtt{A}}_{\mathsf{CT}}\circ\pi^{\mathtt{A}}_{\mathsf{com}}} \left(\sigma^{\mathtt{B}}_{\mathsf{com}}\sigma^{\mathtt{B}}_{\mathsf{CT}}\mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\{0,1\}}\right)^{[\mathtt{true},\mathcal{E}^{\mathsf{opened}}]:\tilde{\epsilon}},$$

with $\tilde{\epsilon} \coloneqq \left(\epsilon_{\sigma^{\mathtt{B}}_{\mathsf{CT}}}\right)_{[\mathtt{true},\mathcal{E}^{\mathsf{opened}}]}$.

Proof. Recall that $\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\{0,1\}}$ only reveals the value X to Bob after he sent his value Y. Hence, X and Y are independent and with X chosen uniform at random by Alice, implying that $Z = X \oplus Y$ is a uniform random value. Hence, using the simple simulator $\sigma^{\mathtt{B}}_{\mathsf{CT}}$ that simulates the output of the commitment resource as $X \coloneqq Z \oplus Y$ (see Appendix D.2 for a formal definition), it is easy to see that the construction actually achieves the coin-toss resource perfectly. □

Note that this implies that the output Z that Alice obtains looks indistinguishable from a uniform random value until the value is released for the dishonest party. Hence, while it is not guaranteed that the dishonest party does not learn some trapdoor afterwards, the value Z is at least unbiased.

Finally, consider the security guarantees for an honest party B against a potentially dishonest party A. To this end, we turn to the unfiltered resources $\mathsf{Com}^{\mathtt{A}\to\mathtt{B}}_{\{0,1\},f}$ and $\mathsf{CT}^{\mathtt{A},\mathtt{B}}_{\{0,1\},f}$, where the latter once allows Alice to obtain $f(c)$ for a c of her choice.

Proposition 10. *Let* $\boldsymbol{\pi}_{\mathsf{CT}} \coloneqq (\pi^{\mathsf{A}}_{\mathsf{CT}}, \pi^{\mathsf{B}}_{\mathsf{CT}})$ *denote the pair of converters implementing Blum's protocol. Then, there exists an efficient simulator* $\sigma^{\mathsf{A}}_{\mathsf{CT}}$ *such that*

$$\begin{aligned}\left\{\left[\mathsf{Com}^{\mathsf{A}\to\mathsf{B}}_{\{0,1\},f}, \mathsf{Ch}^{\mathsf{B}\to\mathsf{A}}\right] \mid f\colon \{0,1\}^* \to \{0,1,\bot\}\right\}\\ \xrightarrow{\pi^{\mathsf{B}}_{\mathsf{CT}}} \left\{\sigma^{\mathsf{A}}_{\mathsf{CT}}\mathsf{CT}^{\mathsf{A},\mathsf{B}}_{\{0,1\},f} \mid f\colon \{0,1\}^* \to \{0,1,\bot\}\right\},\end{aligned}$$

and thus, for any commitment scheme $\boldsymbol{\pi}_{\mathsf{com}}$ *satisfying Definition 12 (binding), we have*

$$\left[\mathsf{Ch}^{\mathsf{A}\to\mathsf{B}}_1, \mathsf{Ch}^{\mathsf{A}\to\mathsf{B}}_2, \mathsf{Ch}^{\mathsf{B}\to\mathsf{A}}\right] \xrightarrow{\pi^{\mathsf{B}}_{\mathsf{CT}}\circ\pi^{\mathsf{B}}_{\mathsf{com}}} \left\{\sigma^{\mathsf{A}}_{\mathsf{com}}\sigma^{\mathsf{A}}_{\mathsf{CT}}\mathsf{CT}^{\mathsf{A},\mathsf{B}}_{\{0,1\},f} \mid f\colon \{0,1\}^* \to \{0,1,\bot\}\right\}.$$

Proof. Consider the real-world system resulting from attaching Bob's converter only, for some function f. Interacting with this resource, the environment can input a commitment string C at Alice's interface, then see Bob's bit Y at Alice's channel interface, and finally see the resulting bit $Z = f(C) \oplus Y$ as the output of Bob's converter. In the following, consider the ideal-world system with the same function f as in the real world. It is now easy to see that a simulator can easily replicate the real-world behavior by getting Z from the resource, querying the leakage-oracle on C getting $f(C)$, and then setting $Y = Z \oplus f(C)$. A formal definition of the simulator $\sigma^{\mathsf{A}}_{\mathsf{com}}$ can be found in Appendix D.2. □

As a final note, observe that formalizing Bob's security guarantees for the commitment resource in terms of an interval-wise relaxation, rather than introducing the unfiltered resource $\mathsf{CT}^{\mathsf{A},\mathsf{B}}_{\{0,1\},f}$, would not work. This is due to the fact that in the real world Y (requiring the additional capabilities to simulate) is output at Alice's interface before Bob sees Z. Hence, simulating only until Alice sends Y would not give any guarantees on Bob's output. In summary, this demonstrates the advantage of being able to phrase different types of statements within one (meta-)framework rather than hard-coding a single type of the Constructive Cryptography framework.

5 Conclusion and Future Work

We have demonstrated that considering new types of resource specifications can lead to security notions that are composable, yet do not suffer the artificial impossibilities exhibited by the classical simulation-based definitions. We have introduced a type of specification that formalizes guarantees that hold in a certain time-interval (between two events), which has clean semantics, comes with a natural syntactical composition theorem, and integrates well with the existing Constructive Cryptography framework. The usefulness of this type of security guarantees has been shown by revisiting several examples where a composable statement previously either needed additional trusted setup or less efficient protocols compared to the corresponding game-based notions.

While our novel type of relaxation does not resolve every issue of composable security, we ultimately believe that all (meaningful) security statements can be

expressed as an as a construction from an assumed specification being contained in an ideal one. Further work is, hence, needed to identify additional types of specifications that allow to express more properties—while retaining strong syntactical composition rules and clear semantics.

References

[1] Backes, M., Dürmuth, M., Hofheinz, D., Küsters, R.: Conditional reactive simulatability. In: Gollmann, D., Meier, J., Sabelfeld, A. (eds.) Computer Security – ESORICS 2006. pp. 424–443. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

[2] Badertscher, C., Maurer, U.: Composable and robust outsourced storage. In: Topics in Cryptology — CT-RSA 2018. LNCS, vol. 10808, pp. 354–373. Springer (Apr 2018)

[3] Badertscher, C., Maurer, U., Tackmann, B.: On composable security for digital signatures. In: Abdalla, M., Dahab, R. (eds.) Public-Key Cryptography — PKC 2018. pp. 494–523. Springer International Publishing, Cham (2018)

[4] Blum, M.: Coin flipping by telephone a protocol for solving impossible problems. SIGACT News **15**(1), 23–27 (Jan 1983). https://doi.org/10.1145/1008908.1008911

[5] Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) Advances in Cryptology — CRYPTO 2001. pp. 213–229. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)

[6] Broadnax, B., Döttling, N., Hartung, G., Müller-Quade, J., Nagel, M.: Concurrently composable security with shielded super-polynomial simulators. In: Coron, J.S., Nielsen, J.B. (eds.) Advances in Cryptology – EUROCRYPT 2017. pp. 351–381. Springer International Publishing, Cham (2017)

[7] Canetti, R.: Universally Composable Security: A New Paradigm for Cryptographic Protocols. In: 42nd IEEE Symposium on Foundations of Computer Science – FOCS 2001. pp. 136–145. IEEE Computer Society (2001)

[8] Canetti, R., Fischlin, M.: Universally composable commitments. In: Kilian, J. (ed.) Advances in Cryptology — CRYPTO 2001. pp. 19–40. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)

[9] Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) Advances in Cryptology — EUROCRYPT 2003. pp. 255–271. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)

[10] Canetti, R., Krawczyk, H.: Universally composable notions of key exchange and secure channels. In: Knudsen, L.R. (ed.) Advances in Cryptology – EUROCRYPT 2002. pp. 337–351. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)

[11] Demay, G., Gaži, P., Maurer, U., Tackmann, B.: Per-session security: Password-based cryptography revisited. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security – ESORICS 2017. pp. 408–426. Springer International Publishing, Cham (2017)

[12] Hofheinz, D., Matt, C., Maurer, U.: Idealizing identity-based encryption. In: Iwata, T., Cheon, J.H. (eds.) Advances in Cryptology – ASIACRYPT 2015. Lecture Notes in Computer Science, vol. 9452, pp. 495–520. Springer Berlin Heidelberg (2015)

[13] Jost, D., Maurer, U., Mularczyk, M.: A unified and composable take on ratcheting. In: Theory of Cryptography — TCC 2019. Springer (Dec 2019), to appear

[14] Kuesters, R., Tuengerthal, M., Rausch, D.: The iitm model: a simple and expressive model for universal composability. Cryptology ePrint Archive, Report 2013/025 (2013), https://eprint.iacr.org/2013/025

[15] Lindell, Y.: General composition and universal composability in secure multiparty computation. Journal of Cryptology **22**(3), 395–428 (Jul 2009)
[16] Maurer, U.: Constructive Cryptography–A New Paradigm for Security Definitions and Proofs. In: Theory of Security and Applications – TOSCA 2011, pp. 33–56. Springer Berlin Heidelberg (2011)
[17] Maurer, U.: Indistinguishability of random systems. In: Knudsen, L.R. (ed.) Advances in Cryptology – EUROCRYPT 2002. pp. 110–132. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
[18] Maurer, U., Pietrzak, K., Renner, R.: Indistinguishability amplification. In: Menezes, A. (ed.) Advances in Cryptology – CRYPTO 2007. pp. 130–149. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
[19] Maurer, U., Renner, R.: Abstract cryptography. In: In Innovations in Computer Science – ICS 2011, pp. 1–21. Tsinghua University (2011)
[20] Maurer, U., Renner, R.: From Indifferentiability to Constructive Cryptography (and Back). In: Theory of Cryptography – TCC 2016, pp. 3–24. Springer Berlin Heidelberg (2016)
[21] Maurer, U., Tackmann, B.: On the soundness of authenticate-then-encrypt: Formalizing the malleability of symmetric encryption. In: Keromytis, A.D., Shmatikov, V. (eds.) Proceedings of the 17th ACM Conference on Computer and Communication Security. pp. 505–515. ACM, ACM (Oct 2010)
[22] Nielsen, J.B.: Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In: Yung, M. (ed.) Advances in Cryptology — CRYPTO 2002. pp. 111–126. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
[23] Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) Advances in Cryptology — EUROCRYPT 2003. pp. 160–176. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
[24] Pfitzmann, B., Waidner, M.: A model for asynchronous reactive systems and its application to secure message transmission. In: Proceedings 2001 IEEE Symposium on Security and Privacy – S&P 2001. pp. 184–200 (May 2001). https://doi.org/10.1109/SECPRI.2001.924298
[25] Prabhakaran, M., Sahai, A.: New notions of security: Achieving universal composability without trusted setup. In: Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing. pp. 242–251. STOC '04, ACM, New York, NY, USA (2004). https://doi.org/10.1145/1007352.1007394

A Revisiting Composable Identity-Based Encryption

In this section, we reexamine a result by Hofheinz, Matt, and Maurer [12] implying that IND-ID-CPA security is not the right notion for identity-based encryption, unmasking this claim as an unnecessary framework artifact.

A.1 Background and Motivation

Identity-based encryption (IBE) is a generalization of public-key encryption that allows to encrypt messages using a master public key and the identity of the receiver, e.g., the e-mail address. This stands in contrast to a regular public-key encryption scheme, where the encryption needs the receiver's public key, suggesting IBE as a solution to the key-distribution problem.

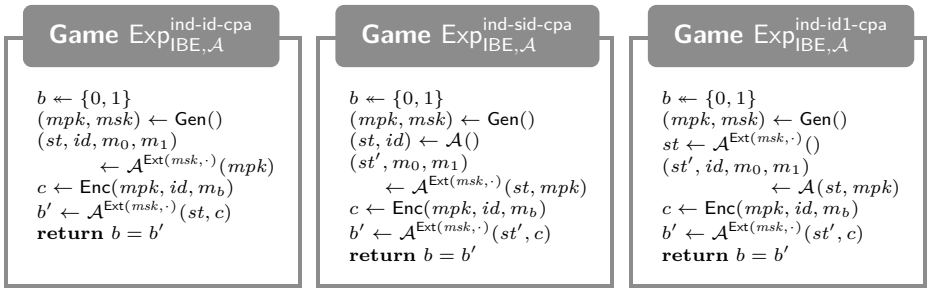

Fig. 5: Three game-based security definition of identity-based encryption against passive attacks. The standard security definition (left) has been put forward by Boneh and Franklin [5]. Later, the weaker static security notion (middle) has been introduced in [9] by Canetti, Halevi, and Katz. Finally, Hofheinz, Matt, and Maurer proposed a version that loosely corresponds to the standard definition under "lunchtime attack" (right). Note that in all of them, $\mathcal{A}$ is not allowed to query the Ext oracle on the identity it output.

An IBE scheme $\mathsf{IBE} := (\mathsf{Gen}, \mathsf{Ext}, \mathsf{Enc}, \mathsf{Dec})$ consists of four algorithms. The key generation algorithm $(mpk, msk) \leftarrow \mathsf{Gen}(1^\lambda)$ outputs a master public and a master secret key (given the security parameter as input). The extraction algorithm $sk_{id} \leftarrow \mathsf{Ext}(msk, id)$ outputs a user secret key given the master secret key and the user's identity. Encryption $c \leftarrow \mathsf{Enc}(mpk, id, m)$ outputs a ciphertext, and $m' \leftarrow \mathsf{Dec}(sk_{id}, id, c)$ the corresponding plain-text. For correctness, it is required that for all $(mpk, msk) \leftarrow \mathsf{Gen}()$, all identities id, all messages m, and all $sk_{id} \leftarrow \mathsf{Ext}(msk, id)$, we always have $\mathsf{Dec}(sk_{id}, id, \mathsf{Enc}(mpk, id, m)) = m$. Security, on the other hand is classically formalized via game-based definitions. For security against passive attacks, the standard notion is ind-id-cpa, as depicted in Figure 5. Weaker notions have been proposed as well, such as a version ind-sid-cpa, depicted in Figure 5 as well, where the adversary has to choose the identity under attack without knowing the master public key.

Hofheinz, Matt, and Maurer [12] investigated the composable security of IBE in Constructive Cryptography from an application centric point of view. To this end, they considered non-interactive communication, the apparent standard application of IBE. Assume that there is a trusted party that stores the master secret key and handles registration, i.e, hands out the user secret keys to the correct users. A set of users, each knowing the master public key and his user secret key(s), can now confidentially send messages to each other by encrypting it under the receiver's identity. To obvious security goal is that only the legitimate receiver can read the message. In turned out, however, that the standard ind-id-cpa security does not imply such a construction in the standard simulation-based construction notion—even when considering *static corruptions* only—due to the commitment problem. They, however, managed to show that such a scheme suffices in a weaker setting where it is guaranteed that all identities are registered before the first

ciphertext is ever sent. Furthermore, the authors also introduced a new weaker security notion ind-id1-cpa, that essentially considers "lunchtime attack" and proved that it suffices for the same construction. They hence concluded, that the standard notion is at the same time both too strong (to achieve the weaker construction) and too weak (to achieve the desired construction).

Since the weaker construction is not very realistic, e.g., in a company it is natural that new employees join long after the first ciphertext has been sent, the question about the right security definition remained open. In the remainder of this section, we devise a natural composable formalization based on interval-wise guarantees whose security exactly corresponds to the standard ind-id-cpa notion—resolving the issue.

A.2 The Real and Ideal Worlds

On a high level, Hofheinz et al. considered non-interactive secure communication in a setting with one honest sender $\mathtt{A}$, n potentially dishonest receivers $\mathtt{B}_i$, and one honest party $\mathtt{C}$ deriving and distribution the user secret keys. We here consider the same setting with essentially the same resources as in the original work.

In the real world, we assume that the sender $\mathtt{A}$ has a broadcast channel BCAST_n available, through which the ciphertexts are sent. For simplicity, we will assume guaranteed delivery throughout the rest of the example. Furthermore, we assume the existence of an authenticated channel $\mathsf{AUTH}_{\mathcal{P}}^{\mathtt{C}\to\mathtt{A}}$ from $\mathtt{C}$ to $\mathtt{A}$ to transmit the master public key Alice needs for encryption, and n secure channels $\mathsf{SEC}_{\mathcal{P}}^{\mathtt{C}\to\mathtt{B}_i}$ from $\mathtt{C}$ to $\mathtt{B}_i$ to transmit the user secret keys. A formal description of the corresponding resources can be found in Figure 6. Recall that we consider static corruptions here. Hence, the set of corrupted parties $\mathcal{P}$ appears as an explicit parameter of the resources.

The protocol securing the communication works as expected: whenever Alice wants to send a message to a certain identity, she encrypts it under the given *id* and broadcasts the ciphertext together with the identity. Each honest receiver then checks whether he has the corresponding decryption key, i.e., has been registered for this identity, and either decrypts the message or discards it. Finally, Charlie's protocol not only sends the master public key to Alice, but also allows to register identities for each receiver. Note that each identity can in principle be registered to many interface, i.e, many parties can posses the same user secret key. Furthermore, the assignment is not fixed but chosen by the environment, modeling an arbitrary or even adversarially chosen assignment. For completeness, a formal description of the corresponding converters $\pi_{\mathsf{Enc},t}^{\mathtt{A}}$, $\pi_{\mathsf{Dec}}^{\mathtt{B}_i}$, and $\pi_{\mathsf{Ext}}^{\mathtt{C}}$ is given in Figure 7.

Finally, consider the ideal resource $\mathsf{DCC}_{n,t,\mathcal{P}}^{\mathcal{ID}}$, standing from *delivery controlled channel*. The name derives from the fact that the encryption can be seen as a mechanism with which the sender can specify to whom the message should be delivered. Hence, the resource acts like the broadcast channel, except that each message is only received by parties registered for the specified identity. Other dishonest parties may learn the message length, while honest parties ignore messages not intended for them. Furthermore, dishonest party might also learn

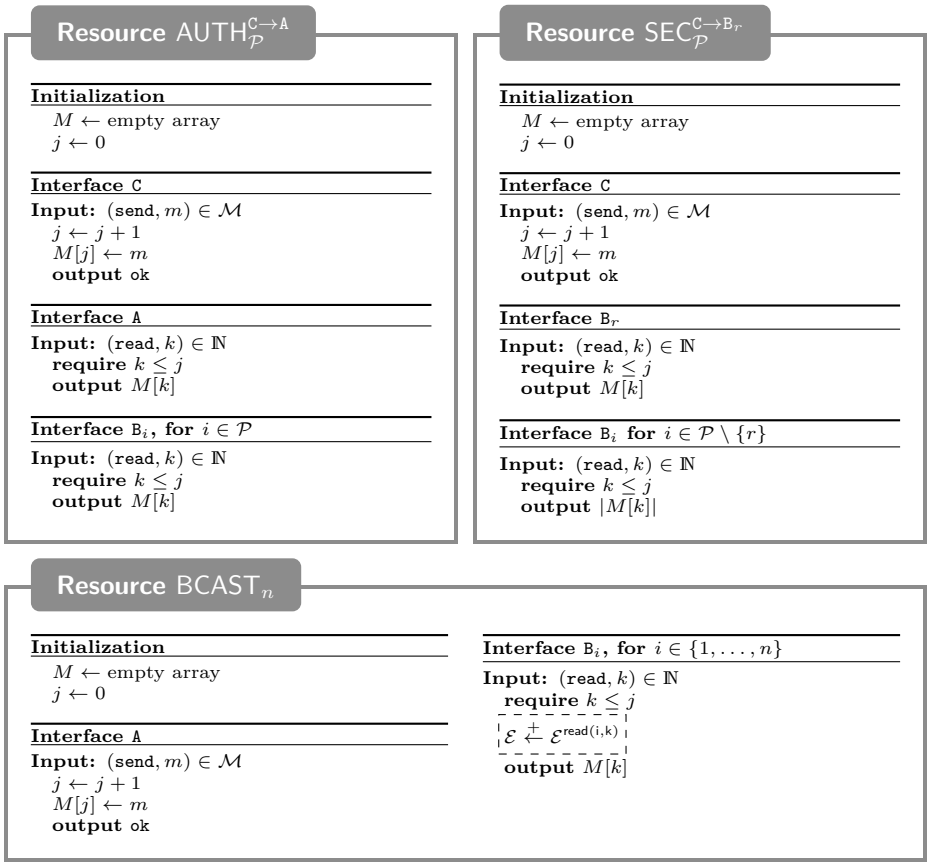

Fig. 6: Description of the assumed resources.The set $\mathcal{P} \subseteq \{1, \ldots, n\}$ thereby specifies the set of statically corrupted receivers $\mathtt{B}_i$, while the sender A and registrar C are assumed to be honest.

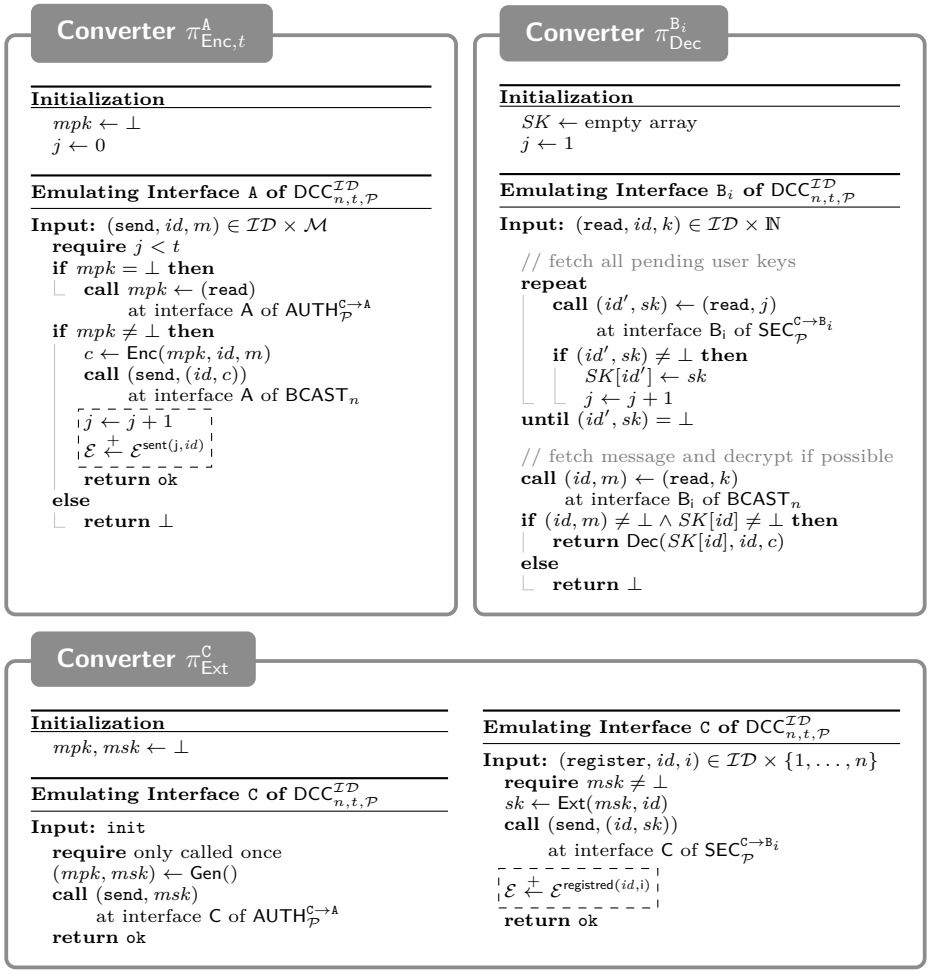

Fig. 7: A formal description of the converters using IBE to achieve confidentiality.

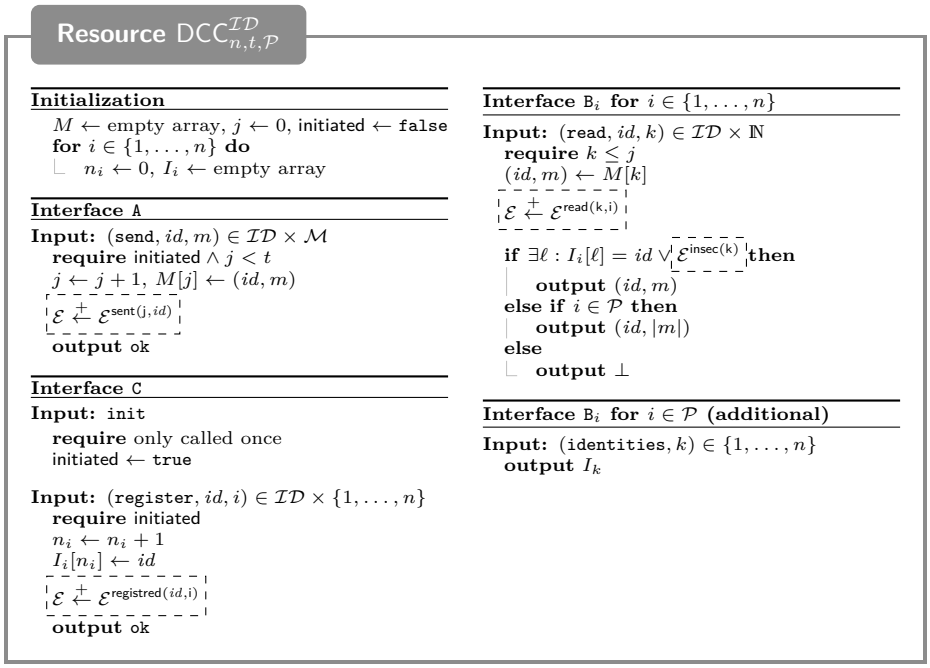

Fig. 8: The delivery controlled channel with an honest sender A, an honest registrar C, and n receivers $\mathtt{B}_i$, with $\mathcal{P} \subseteq \{1,\ldots,n\}$ denoting the statically corrupted ones.

which receiver is registered for identities (as the secure channel leaks the length of the user keys, which might depend on the identity). A formal definition of the resource is depicted in Figure 8. Note that compared to [12], we will use global simulators rather than local ones, what allowed us to simplify the resource a bit.

A.3 The Composable Statement

One of the main results in [12] has been showing that, due to the commitment problem, an IBE scheme does not construct the delivery controlled channel even under static corruptions.

Proposition 11 (Theorem 5.1 in [12]). *Let $\mathcal{P} \subseteq \{1,\ldots,n\}$ denote the set of statically corrupted receivers. For every efficient simulator $\boldsymbol{\sigma}^{\mathcal{P}}_{\mathsf{IBE}}$, we have*

$$\left[\mathsf{BCAST}_n, \mathsf{AUTH}^{\mathtt{C}\to\mathtt{A}}_{\mathcal{P}}, \mathsf{SEC}^{\mathtt{C}\to\mathtt{B}_1}_{\mathcal{P}}, \ldots, \mathsf{SEC}^{\mathtt{C}\to\mathtt{B}_n}_{\mathcal{P}}\right] \xrightarrow{\boldsymbol{\pi}^{n,t,\mathcal{P}}_{\mathsf{IBE}}} \left(\boldsymbol{\sigma}^{\mathcal{P}}_{\mathsf{IBE}} \mathsf{DCC}^{\mathcal{ID}}_{n,t,\mathcal{P}}\right)^{\epsilon_{\mathsf{ind\text{-}id\text{-}cpa}}},$$

where $\epsilon_{\mathsf{ind\text{-}id\text{-}cpa}}$ denotes a reduction to the ind-id-cpa *game and*

$$\boldsymbol{\pi}^{n,t,\mathcal{P}}_{\mathsf{IBE}} := \left\{\pi^{\mathtt{A}}_{\mathsf{Enc},t}, \pi^{\mathtt{C}}_{\mathsf{Ext}}, \right\} \cup \bigcup_{i\in\{1,\ldots,n\}\setminus\mathcal{P}} \pi^{\mathtt{B}_i}_{\mathsf{Dec}}$$

denotes the protocol where all honest parties apply their converter.

In the end, one is however not interested in such a strong guarantee. Rather, the above construction was just intended as a proxy to formalize confidentiality of the messages. Using our approach, hence immediately yields a better statement based on interval-wise guarantees. More concretely, we consider each message individually and guarantee its confidentiality *until it trivially leaks* because a dishonest party is registered for the corresponding receiver's identity.

In the following, let $\mathcal{E}^{\mathsf{sent}(\mathsf{j},id)}$ indicate that the j-th message has been encrypted for identity id, $\mathcal{E}^{\mathsf{registred}(id,\mathsf{i})}$ indicate that the i-th receiver has been registered for identity id, and $\mathcal{E}^{\mathsf{read}(\mathsf{j},\mathsf{i})}$ that the i-th receiver read the j-th message. Moreover, let $\mathcal{E}^{\mathsf{insec}(\mathsf{j})}$ indicate that the j-th message is not confidential. See the dashed boxed in Figure 8 for a formal definition of when those events get triggered in the ideal world. Note that in the real world the former two events get triggered in the converters $\pi^{\mathsf{A}}_{\mathsf{Enc},t}$ and $\pi^{\mathsf{C}}_{\mathsf{Ext}}$, respectively (c.f. Figure 7), whereas the third one gets triggered in the underlying broadcast channel BCAST_n (c.f. Figure 6). This is due to the fact that Alice and Charlie are assumed to always apply their converter, while the dishonest receivers can access the broadcast channel directly.

Theorem 16. *Let $t \in \mathbb{N}$ denote an upper bound on the number of messages, let $\mathcal{P} \subseteq \{1,\dots,n\}$ denote an arbitrary set of statically corrupted receivers, and let $\boldsymbol{\pi}^{n,t,\mathcal{P}}_{\mathsf{IBE}}$ denote the protocol where all honest parties apply their converter as in Proposition 11. For each $\mathcal{P}$, there exists a sequence of efficient simulators $\boldsymbol{\sigma}^{\mathcal{P},j}_{\mathsf{IBE}}$ and a reduction $\epsilon_{\mathsf{ind\text{-}id\text{-}cpa}}$ to the* ind-id-cpa *game, such that*

$$\left[\mathsf{BCAST}_n, \mathsf{AUTH}^{\mathsf{C}\to\mathsf{A}}_{\mathcal{P}}, \mathsf{SEC}^{\mathsf{C}\to\mathsf{B}_1}_{\mathcal{P}}, \dots, \mathsf{SEC}^{\mathsf{C}\to\mathsf{B}_n}_{\mathcal{P}}\right] \xrightarrow{\boldsymbol{\pi}^{n,t,\mathcal{P}}_{\mathsf{IBE}}} \bigcap_{j\in\{1,\dots,t\}} \left(\boldsymbol{\sigma}^{\mathcal{P},j}_{\mathsf{IBE}} \mathsf{DCC}^{\mathcal{ID}}_{n,t,\mathcal{P}}\right)^{[P_{\mathsf{only}(\mathsf{j})}, P_{\mathsf{leaked}(\mathsf{j})}]:\epsilon_{\mathsf{ind\text{-}id\text{-}cpa}}}.$$

where

$$P_{\mathsf{leaked}(\mathsf{j})}(\mathcal{E}) \coloneqq \exists i \in \mathcal{P} : \mathcal{E}^{\mathsf{sent}(\mathsf{j},id)} \wedge \mathcal{E}^{\mathsf{registred}(id,\mathsf{i})} \wedge \mathcal{E}^{\mathsf{read}(\mathsf{j},\mathsf{i})}$$

formalizes the event that the j-th message inherently leaked, and

$$P_{\mathsf{only}(\mathsf{j})}(\mathcal{E}) \coloneqq \bigwedge_{\ell\in\{1,\dots,t\}\setminus\{j\}} \mathcal{E}^{\mathsf{insec}(\ell)}$$

formalizes that we do not consider confidentiality of the other messages.

Proof (Sketch). The simulator initially generates a master secret-key and master public-key pair. Observe that for all messages except the j-th one, the ideal-world resource outputs the actual message (together with the corresponding identity). Hence, the simulator can simply encrypt it himself under the correct identity. For the j-th message, observe that the simulator only has to work if none of the dishonest receivers obtained the corresponding user key, and this key will then never get revealed. Hence, the simulator can encrypt the zero-string of the appropriate length. See Figure 9 for a formal definition of the simulator.

Observe that the reduction from distinguishing the real- and idea-world to the ind-id-cpa game is straight-forward. For all but the j-th message, the reduction

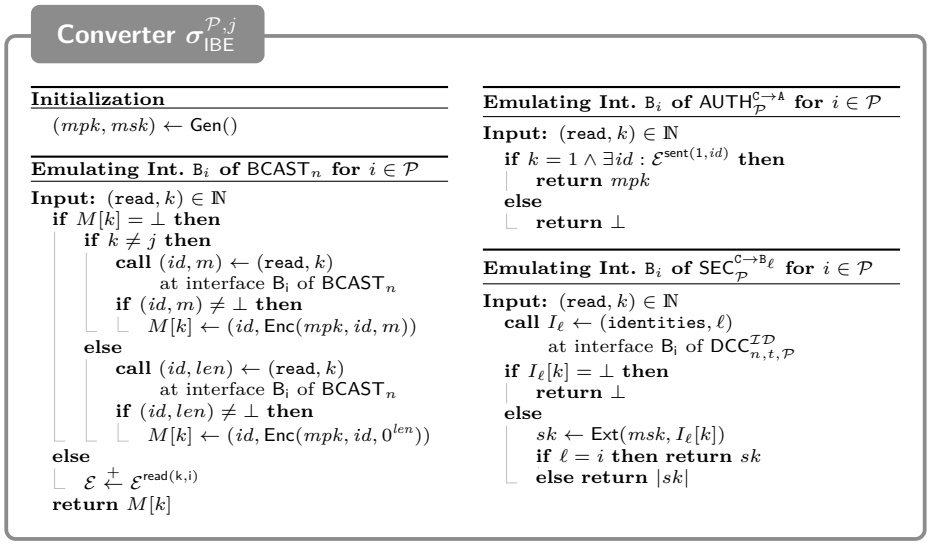

Fig. 9: The simulator from Theorem 16.

can simply query the game for the corresponding user key, and the challenge can be chosen as $(m_j, 0^{|m_j|})$. If $b = 0$, this corresponds to the real-world behavior, and if $b = 1$ to the ideal-world behavior. □

Remark 3 (On interpreting conjunction specifications). Observe that phrasing our statement for a bounded number of messages is without loss of generality in any actual application. We, however, have chosen to do so for good reasons. Assume for the moment that ϵ is simply a constant in $[0, 1]$. If we have a specification of the form $\bigcap_{i\in\mathcal{J}} \mathcal{S}_i{}^{\epsilon}$, then this can be read as the guarantee given by S_i holding with probability $1 - \epsilon$. In the end, we are however interested to know the probability such that all guarantees hold. In general, the best we can do is to apply the union bound, implying that the error is bounded by $\epsilon|\mathcal{J}|$. As a result, it is important that the number of conjunctions is small for the statement to be meaningful. Especially, while we could prove an analogous statement with intervals terminated by the leakage of each user key (rather than the messages), this would only make sense if the identity-space is small, in which case ind-sid-cpa security would suffice.

B Details of Section 2

B.1 Proof of Theorem 2

Theorem 2. *Let $\mathcal{R}$ be an arbitrary specification, and let ϵ_1 and ϵ_2 be arbitrary ϵ-relaxations. Then we have $\left(\mathcal{R}^{\epsilon_1}\right)^{\epsilon_2} \subseteq \mathcal{R}^{\epsilon_1+\epsilon_2}$.*

Proof. Let $\mathsf{T} \in (\mathcal{R}^{\epsilon_1})^{\epsilon_2}$ be arbitrary. By the definition of the ϵ-relaxation,

$$(\mathcal{R}^{\epsilon_1})^{\epsilon_2} = \bigcup_{\mathsf{S} \in \mathcal{R}^{\epsilon_1}} \left\{ \mathsf{T} \;\middle|\; \forall \mathsf{D} : |\Delta^{\mathsf{D}}(\mathsf{S}, \mathsf{T})| \leq \epsilon_2(\mathsf{D}) \right\},$$

there exists a $\mathsf{S} \in \mathcal{R}^{\epsilon_1}$, such that for all D, $\Delta^{\mathsf{D}}(\mathsf{S}, \mathsf{T}) \leq \epsilon_2(\mathsf{D})$. Analogously, invoking the definition of the ϵ-relaxation once more, there moreover must exist a $\mathsf{R} \in \mathcal{R}$, such that that for all D, $\Delta^{\mathsf{D}}(\mathsf{R}, \mathsf{S}) \leq \epsilon_1(\mathsf{D})$. Using the triangle inequality we, thus, obtain

$$\Delta^{\mathsf{D}}(\mathsf{R}, \mathsf{T}) \leq \Delta^{\mathsf{D}}(\mathsf{R}, \mathsf{S}) + \Delta^{\mathsf{D}}(\mathsf{S}, \mathsf{T}) \leq \epsilon_1(\mathsf{D}) + \epsilon_1(\mathsf{D}),$$

and, hence, $\mathsf{T} \in (\mathsf{R}^{\epsilon_1 + \epsilon_2}) \subseteq (\mathcal{R}^{\epsilon_1 + \epsilon_2})$. □

B.2 Proof of Theorem 3

Theorem 3. *The ϵ-relaxation is* compatible *with protocol application in the following sense:*

$$\boldsymbol{\pi}\,(\mathcal{R}^{\epsilon}) \subseteq (\boldsymbol{\pi}\mathcal{R})^{\epsilon_{\boldsymbol{\pi}}},$$

for $\epsilon_{\boldsymbol{\pi}}(\mathsf{D}) \coloneqq \epsilon(\mathsf{D}\boldsymbol{\pi}(\cdot))$, where $\mathsf{D}\boldsymbol{\pi}(\cdot)$ denotes the distinguisher that first attaches $\boldsymbol{\pi}$ to the given resource and then executes D. Moreover, the ϵ-relaxation is compatible with parallel composition, i.e.,

$$\left[\mathcal{R}^{\epsilon}, \mathcal{S}\right] \subseteq [\mathcal{R}, \mathcal{S}]^{\epsilon_{\mathcal{S}}},$$

for $\epsilon_{\mathcal{S}}(\mathsf{D}) \coloneqq \sup_{\mathsf{S} \in \mathcal{S}} \epsilon(\mathsf{D}[\,\cdot\,, \mathsf{S}])$, where $\mathsf{D}[\,\cdot\,, \mathsf{S}]$ denotes the distinguisher that emulates S in parallel to the given resource and then lets D interact with them.

Proof. Let $\mathsf{S} \in \boldsymbol{\pi}\,(\mathcal{R}^{\epsilon})$ be arbitrary. Then, by definition, there exists a $\mathsf{T} \in \mathcal{R}^{\epsilon}$ such that $\mathsf{S} = \boldsymbol{\pi}\mathsf{T}$. Observe that $\mathsf{T} \in \mathcal{R}^{\epsilon}$ directly implies $|\Delta^{\mathsf{D}}(\mathsf{R}, \mathsf{S})| \leq \epsilon$, and thus

$$|\Delta^{\mathsf{D}}(\boldsymbol{\pi}\mathsf{R}, \mathsf{S})| = |\Delta^{\mathsf{D}}(\boldsymbol{\pi}\mathsf{R}, \boldsymbol{\pi}\mathsf{T})| = |\Delta^{\mathsf{D}\boldsymbol{\pi}(\cdot)}(\mathsf{R}, \mathsf{T})| \leq \epsilon(\mathsf{D}\boldsymbol{\pi}(\cdot)) = \epsilon_{\boldsymbol{\pi}}(\mathsf{D}),$$

implying $\mathsf{S} \in (\boldsymbol{\pi}\mathcal{R})^{\epsilon_{\boldsymbol{\pi}}}$, i.e., compatibility with protocol application.

Now, let $\mathsf{U} \in \left[\mathcal{R}^{\epsilon}, \mathcal{S}\right]$ be arbitrary. Then, by definition there exists $\mathsf{V} \in \mathcal{R}^{\epsilon}$ and $\mathsf{S} \in \mathcal{S}$ such that $\mathsf{U} = [\mathsf{V}, \mathsf{S}]$. Moreover, by definition there exists $\mathsf{R} \in \mathcal{R}$ such that $|\Delta^{\mathsf{D}}(\mathsf{R}, \mathsf{V})| \leq \epsilon$. As a consequence we obtain

$$|\Delta^{\mathsf{D}}([\mathsf{R}, \mathsf{S}], \mathsf{U})| = |\Delta^{\mathsf{D}}([\mathsf{R}, \mathsf{S}], [\mathsf{V}, \mathsf{S}]| = |\Delta^{\mathsf{D}[\,\cdot\,, \mathsf{V}]}(\mathsf{R}, \mathsf{V})| \leq \epsilon(\mathsf{D}[\,\cdot\,, \mathsf{V}]) \leq \epsilon_{\mathcal{S}}(\mathsf{D}),$$

and thus $\mathsf{U} \in [\mathcal{R}, \mathcal{S}]^{\epsilon_{\mathcal{S}}}$, concluding the proof. □

B.3 Proof of Corollary 1

Corollary 1. *For any specifications $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$, any protocols $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$, and any ϵ-relaxation ϵ and ϵ', we have*

1. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S} \implies \mathcal{R}^{\epsilon} \xrightarrow{\boldsymbol{\pi}} \mathcal{S}^{\epsilon_{\boldsymbol{\pi}}}$,

2. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S}^{\epsilon} \implies [\mathcal{R}, \mathcal{T}] \xrightarrow{\boldsymbol{\pi}} [\mathcal{S}, \mathcal{T}]^{\epsilon_{\mathcal{T}}}$,

3. $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S}^{\epsilon} \wedge \mathcal{S} \xrightarrow{\boldsymbol{\pi}'} \mathcal{T}^{\epsilon'} \implies \mathcal{R} \xrightarrow{\boldsymbol{\pi}' \circ \boldsymbol{\pi}} \mathcal{T}^{\epsilon_{\boldsymbol{\pi}'} + \epsilon'}$,

where $\epsilon_{\boldsymbol{\pi}}$ and $\epsilon_{\mathcal{S}}$ are defined as in Theorem 3, respectively.

Proof. The first property directly follows from Theorem 3, i.e.,

$$\boldsymbol{\pi}\left(\mathcal{R}^{\epsilon}\right) \subseteq \left(\boldsymbol{\pi}\mathcal{R}\right)^{\epsilon_{\boldsymbol{\pi}}} \subseteq \mathcal{S}^{\epsilon_{\boldsymbol{\pi}}},$$

where in the second step we used Definition 1, i.e., $\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \mathcal{S} :\iff \boldsymbol{\pi}\mathcal{R} \subseteq \mathcal{S}$, and Proposition 3. Analogously, the second property follows from Theorem 3 as well:

$$\boldsymbol{\pi}\left[\mathcal{R}, \mathcal{T}\right] = \left[\boldsymbol{\pi}\mathcal{R}, \mathcal{T}\right] \subseteq \left[\mathcal{S}^{\epsilon}, \mathcal{T}\right] \subseteq [\mathcal{S}, \mathcal{T}]^{\epsilon_{\mathcal{S}}}.$$

Finally, for the third property observe that by Theorem 2, we have $\left(\mathcal{T}^{\epsilon'}\right)^{\epsilon_{\boldsymbol{\pi}'}} \subseteq \mathcal{T}^{\epsilon_{\boldsymbol{\pi}'} + \epsilon'}$. Thus, using the first property we obtain $\mathcal{S}^{\epsilon} \xrightarrow{\boldsymbol{\pi}'} \mathcal{T}^{\epsilon_{\boldsymbol{\pi}'} + \epsilon'}$ and, hence, the property then follows by Theorem 1.

C Details of Section 3

C.1 Example: The Behavior of a Resource After an Event

Consider the following resource from$_{\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}}(\mathsf{SecCh})$, which only answers queries once the environment triggered the event $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$. Thus, in contrast to SecCh, this resource always leaks the full message of the adversary, in line with our intuition that it describes the behavior after the key has been exposed.

Resource from$_{\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}}(\mathsf{SecCh})$

Initialization

$M_{\mathtt{A}}, M_{\mathtt{B}} \leftarrow$ empty array
$s, r, d \leftarrow 0$

Interface A

Input: $(\mathtt{send}, m)$
require $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$
[rest as in SecCh]

Interface B

Input: `receive`
require $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$
[rest as in SecCh]

Interface E

Input: $(\mathtt{leak}, i)$
require $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$
if $i \leq s$ **then**
  output $M_{\mathtt{A}}[i]$
else
  output $\perp$

Input: $(\mathtt{inject}, m)$
require $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$
[rest as in SecCh]

Interfaces $\{\mathtt{E}, \mathtt{W}\}$

Input: $(\mathtt{deliver}, i)$
require $\mathcal{E}^{\mathsf{leaked}}_{\mathsf{EncKey}}$
[rest as in SecCh]

C.2 Proof of Theorem 6

Theorem 6. *There exist specifications $\mathcal{R}$ and $\mathcal{S}$, a monotone predicate P, and a function ϵ mapping distinguishers to values in $[0,1]$ such that*

$$\left(\mathcal{R}^{P]}\right)^{\epsilon} \not\subseteq \left(\mathcal{R}^{\epsilon}\right)^{P]} \qquad \text{and} \qquad \left(\mathcal{S}^{\epsilon}\right)^{P]} \not\subseteq \left(\mathcal{S}^{P]}\right)^{\epsilon}.$$

Proof. In the following, let R denote the following resource that provides two interfaces: R initially chooses a seed $s \in \{0,1\}^m$ uniformly at random. Upon a trigger input at the first interface it outputs $\mathrm{PRG}(s)$ (for some m-bits to n-bits PRG). Upon a trigger input at the second interface, it triggers a $\mathcal{E}^{\mathsf{leaked}}$ event and outputs s. Let the resource S work analogously except that it outputs a uniformly distributed an independent value $t \in \{0,1\}^n$ instead of the PRG output. Furthermore, let $P(\mathcal{E}) \coloneqq \mathcal{E}^{\mathsf{leaked}}$ and let ϵ denote the reduction to the security of the PRG.

Using $\mathcal{R} \coloneqq \{\mathsf{R}\}$ it is now easy to see that $\mathsf{S} \in (\mathcal{R}^{P]})^{\epsilon}$: Consider a hybrid system H that outputs $\mathrm{PRG}(s')$ and s for independent and s and s' u.a.r. Then, we have $\mathsf{T} \in \mathcal{R}^{P]}$, since the first output is equally distributed and the second blinded. Moreover, distinguishing T from S boils down to breaking the PRG-security, and thus $\mathsf{S} \in \mathsf{T}^{\epsilon}$. On the other hand, one can show that $\mathsf{S} \notin \left(\mathcal{R}^{\epsilon}\right)^{P]}$. To see this, observe that if the PRG is secure, then $\mathcal{R}^{\epsilon} \approx \mathcal{R}$. Especially, leaking the seed prevents us from replacing the first output by a truly random string. Thus, applying the until-relaxation still does not contain S, concluding the first part of the proof. For the other direction consider $\mathcal{S} \coloneqq \{\mathsf{S}\}$. Using an analogous argument one can show that $\mathsf{R} \in \left(\mathcal{S}^{\epsilon}\right)^{P]}$ but $\mathsf{R} \notin \left(\mathcal{S}^{P]}\right)^{\epsilon}$. □

C.3 Proof of Theorem 9

Theorem 9. *There exist specifications $\mathcal{R}$ and $\mathcal{S}$, a monotone predicate $P(\mathcal{E})$, and a function ϵ mapping distinguishers to values in $[0,1]$ such that*

$$\left(\mathcal{R}^{[P}\right)^{\epsilon} \not\subseteq \left(\mathcal{R}^{\epsilon}\right)^{[P} \qquad \text{and} \qquad \left(\mathcal{S}^{\epsilon}\right)^{[P} \not\subseteq \left(\mathcal{S}^{[P}\right)^{\epsilon}.$$

Proof. The counter example works similarly to the one from Theorem 6. Consider a resource R with two interfaces that initially chooses a PRG-seed s uniformly at random. At the first interface it outputs $\mathrm{PRG}(s)$. At the second interface it outputs s, but only before some event $\mathcal{E}^{\mathsf{secure}}$. The second resource S works analogously except that it outputs an independent uniform random string at the first interface instead of the PRG output. Let now $\mathcal{R} \coloneqq \{\mathsf{R}\}$, $\mathcal{S} \coloneqq \{\mathsf{S}\}$, and $P \coloneqq \{\mathcal{E}^{\mathsf{secure}}\}$. Analogous to the proof from Theorem 6, one can now show that $\mathsf{S} \in \left(\mathcal{R}^{[P}\right)^{\epsilon}$ but $\mathsf{S} \notin \left(\mathcal{R}^{\epsilon}\right)^{[P}$, and furthermore that $\mathsf{R} \in \left(\mathcal{S}^{\epsilon}\right)^{[P}$ but $\mathsf{S} \notin \left(\mathcal{S}^{[P}\right)^{\epsilon}$. □

C.4 Proof of Theorem 10

Theorem 10. *For any resource* R *and any monotone predicates* P_1 *and* P_2*, we have*

$$\begin{aligned}\mathsf{R}^{[P_1,P_2]} &= \bigcup_{n\in\mathbb{N}}\Big(\bigcup\{\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \mid \forall i \le n : \phi_i \in \{P_2],[P_1\}\}\Big)\\ &= \Big(\big(\mathsf{R}^{[P_1}\big)^{P_2]}\Big)^{[P_1} = \Big(\big(\mathsf{R}^{P_2]}\big)^{[P_1}\Big)^{P_2]},\end{aligned}$$

where $\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_n}$ *is a shorthand notation for first applying* ϕ_1*, then* ϕ_2*, until* ϕ_n*.*

Proof. We prove the theorem in three steps:

Claim. $\mathcal{R}^{[P_1,P_2]} = \Big(\big(\mathcal{R}^{[P_1}\big)^{P_2]}\Big)^{[P_1}$

Proof (of claim). Let $\mathsf{S} \in \Big(\big(\mathcal{R}^{[P_1}\big)^{P_2]}\Big)^{[P_1}$ be arbitrary. This means that there exist $\mathsf{T} \in \big(\mathcal{R}^{[P_1}\big)^{P_2]}$, $\mathsf{U} \in \mathcal{R}^{[P_1}$ and $\mathsf{R} \in \mathcal{R}$ such that

$$\mathrm{from}_{P_1}(\mathsf{S}) = \mathrm{from}_{P_1}(\mathsf{T}), \tag{1}$$
$$\mathrm{until}_{P_2}(\mathsf{T}) = \mathrm{until}_{P_2}(\mathsf{U}), \tag{2}$$
$$\mathrm{from}_{P_1}(\mathsf{U}) = \mathrm{from}_{P_1}(\mathsf{R}). \tag{3}$$

Combining these properties with the commutativity of from and until, we obtain

$$\begin{aligned}\mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{R})) &= \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{U})) = \mathrm{from}_{P_1}(\mathrm{until}_{P_2}(\mathsf{U}))\\ &= \mathrm{from}_{P_1}(\mathrm{until}_{P_2}(\mathsf{T})) = \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{T}))\\ &= \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{S})),\end{aligned}$$

implying that $\mathsf{S} \in \mathsf{R}^{[P_1,P_2]}$ and thus $\mathcal{R}^{[P_1,P_2]} \supseteq \Big(\big(\mathcal{R}^{[P_1}\big)^{P_2]}\Big)^{[P_1}$.

Now, let $\mathsf{S} \in \mathcal{R}^{[P_1,P_2]}$ be arbitrary. By definition, we thus know that there exists a $\mathsf{R} \in \mathcal{R}$ such that $\mathrm{from}_{P_1}(\mathsf{S}) \in \big(\mathrm{from}_{P_1}(\mathsf{R})\big)^{P_2]}$. Combining this with the basic fact that $\mathrm{from}_{P_1}(\mathsf{R}) \in \mathsf{R}^{[P_1}$, we obtain $\mathrm{from}_{P_1}(\mathsf{S}) \in \big(\mathsf{R}^{[P_1}\big)^{P_2]}$. Combining this in turn with the basic fact that $\mathsf{S} \in \mathsf{S}^{[P_1}$ then yields $\mathsf{S} \in \Big(\big(\mathsf{R}^{[P_1}\big)^{P_2]}\Big)^{[P_1}$ and thus $\mathcal{R}^{[P_1,P_2]} \subseteq \Big(\big(\mathcal{R}^{[P_1}\big)^{P_2]}\Big)^{[P_1}$. ◊

Claim. $\mathcal{R}^{[P_1,P_2]} = \Big(\big(\mathcal{R}^{P_2]}\big)^{[P_1}\Big)^{P_2]}$

Proof (of claim). Observing that

$$\mathcal{R}^{[P_1,P_2]} = \big\{\mathsf{S} \;\big|\; \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{R})) = \mathrm{until}_{P_2}(\mathrm{from}_{P_1}(\mathsf{S}))\big\}$$

(using the commutativity), it is easy to see that the proof follows analogously. ◊

Claim. $\mathcal{R}^{[P_1,P_2]} = \bigcup_{n\in\mathbb{N}}\Big(\bigcup\{\mathcal{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \mid \forall i \leq n : \phi_i \in \{P_2],[P_1\}\}\Big)$

Proof (of claim). Using the previous claims, it is trivial to see that

$$\mathcal{R}^{[P_1,P_2]} \subseteq \bigcup_{n\in\mathbb{N}}\Big(\bigcup\{\mathcal{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \mid \forall i \leq n : \phi_i \in \{P_2],[P_1\}\}\Big).$$

For the other direction, we proceed by induction over n. Note that without loss of generality (by Theorems 4 and 7), we can assume the order of the relaxations to strictly alternate. Furthermore, the previous claims already prove it for $n = 3$. Hence, the relation is also trivial for $n < 3$, as adding a further relaxation only enlarges the set. Assume now as the induction hypothesis that for some $n \geq 3$

$$\mathcal{R}^{[P_1,P_2]} \supseteq \bigcup\{\mathcal{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \mid \forall i \leq n : \phi_i \in \{P_2],[P_1\}\}.$$

We want to show that $\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_{n+1}} \in \mathsf{R}^{[P_1,P_2]}$ as well. Assume w.l.o.g. that $\phi_{n+1} = P_2]$. By the induction hypothesis, we have that $\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \in \mathsf{R}^{[P_1,P_2]}$ and thus by the second claim $\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \in \Big(\big(\mathsf{R}^{P_2]}\big)^{[P_1}\Big)^{P_2]}$ and thus

$$\mathsf{R}^{\phi_1\cdot\phi_2\cdots\phi_n} \in \Big(\Big(\big(\mathsf{R}^{P_2]}\big)^{[P_1}\Big)^{P_2]}\Big)^{P_2]} = \Big(\Big(\big(\mathsf{R}^{P_2]}\big)^{[P_1}\Big)^{P_2]}\Big)^{P_2]},$$

where the second step follows from Theorem 4. ◊

C.5 Proof of Theorem 13

Theorem 13. *Let P_1 and P_2 be two monotone predicates, and let ϵ be a function mapping distinguishers to values in $[0,1]$. Then, for any specification $\mathcal{R}$ we have*

$$\big(\mathcal{R}^{[P_1,P_2]:\epsilon}\big)^{[P_1',P_2']:\epsilon'} \subseteq \mathcal{R}^{[P_1\wedge P_1',P_2\vee P_2']:\epsilon_{[P_1\wedge P_1',P_2\vee P_2']}+\epsilon'_{[P_1\wedge P_1',P_2\vee P_2']}},$$

where $\epsilon_{[P_1\wedge P_1',P_2\vee P_2']}(D) \coloneqq \epsilon(\mathsf{D}\circ \mathrm{until}_{P_2\vee P_2'}\circ \mathrm{from}_{P_1\wedge P_1'})$, i.e., the performance of the distinguisher interacting with the projected resource, and analogously for $\epsilon'_{[P_1\wedge P_1',P_2\vee P_2']}$.

Proof. Observe that it suffices to show

$$\Big(\Big(\big(\mathcal{R}^{[P_1\wedge P_1',P_2\vee P_2']}\big)^{\epsilon}\Big)^{[P_1\wedge P_1',P_2\vee P_2']}\Big)^{\epsilon'} \subseteq \Big(\big(\mathcal{R}^{[P_1\wedge P_1',P_2\vee P_2']}\big)^{\epsilon_{[P_1,P_2]}+\epsilon'_{[P_1,P_2]}}\Big)^{[P_1\wedge P_1',P_2\vee P_2']}.$$

The rest then follows trivially by Theorems 2 and 11. To this end, consider an arbitrary $\mathsf{S} \in \Big(\Big(\big(\mathcal{R}^{[P_1\wedge P_1',P_2\vee P_2']}\big)^{\epsilon}\Big)^{[P_1\wedge P_1',P_2\vee P_2']}\Big)^{\epsilon'}$. Hence, there must exist

a $\mathsf{T} \in \left(\left(\mathcal{R}^{[P_1 \wedge P_1', P_2 \vee P_2']}\right)^{\epsilon}\right)^{[P_1 \wedge P_1', P_2 \vee P_2']}$, a $\mathsf{U} \in \left(\mathcal{R}^{[P_1 \wedge P_1', P_2 \vee P_2']}\right)^{\epsilon}$, and a $\mathsf{V} \in \mathcal{R}^{[P_1 \wedge P_1', P_2 \vee P_2']}$ such that

$$\left|\Delta^{\mathsf{D}}(\mathsf{S}, \mathsf{T})\right| \leq \epsilon'(\mathsf{D}) \tag{4}$$

$$\mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{T})) = \mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{U})) \tag{5}$$

$$\left|\Delta^{\mathsf{D}}(\mathsf{U}, \mathsf{V})\right| \leq \epsilon(\mathsf{D}). \tag{6}$$

Using those properties, we obtain

$$\begin{aligned}
&\left|\Delta^{\mathsf{D}}\left(\mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{V})), \mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{S}))\right)\right| \\
&\leq \left|\Delta^{\mathsf{D}}(\mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{V})), \mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{U})))\right| \\
&\qquad + \left|\Delta^{\mathsf{D}}(\mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{T})), \mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{S})))\right| \\
&\leq \left|\Delta^{\mathsf{D} \circ \mathrm{until}_{P_2 \vee P_2'}(\cdot) \circ \mathrm{from}_{P_1 \wedge P_1'}(\cdot)}(\mathsf{V}, \mathsf{U})\right| + \left|\Delta^{\mathsf{D} \circ \mathrm{until}_{P_2 \vee P_2'}(\cdot) \circ \mathrm{from}_{P_1 \wedge P_1'}(\cdot)}(\mathsf{T}, \mathsf{S})\right| \\
&\leq \epsilon_{[P_1 \wedge P_1', P_2 \vee P_2']}(\mathsf{D}) + \epsilon'_{[P_1 \wedge P_1', P_2 \vee P_2']}(\mathsf{D}).
\end{aligned}$$

Now observe that $\mathsf{V} \in \mathcal{R}^{[P_1 \wedge P_1', P_2 \vee P_2']}$ implies

$$\mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{V})) \in \mathcal{R}^{[P_1 \wedge P_1', P_2 \vee P_2']},$$

and thus,

$$\mathrm{until}_{P_2 \vee P_2'}(\mathrm{from}_{P_1 \wedge P_1'}(\mathsf{S})) \in \left(\mathcal{R}^{[P_1 \wedge P_1', P_2 \vee P_2']}\right)^{\epsilon_{[P_1 \wedge P_1', P_2 \vee P_2']} + \epsilon'_{[P_1 \wedge P_1', P_2 \vee P_2']}}.$$

As a result, we have

$$\mathsf{S} \in \left(\left(\mathcal{R}^{[P_1 \wedge P_1', P_2 \vee P_2']}\right)^{\epsilon_{[P_1 \wedge P_1', P_2 \vee P_2']} + \epsilon'_{[P_1 \wedge P_1', P_2 \vee P_2']}}\right)^{[P_1 \wedge P_1', P_2 \vee P_2']},$$

concluding the proof. □

C.6 Proof of Theorem 15

Theorem 15. *Let $\mathcal{R}$, $\mathcal{S}$, and $\mathcal{T}$ be arbitrary specifications, let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ be arbitrary protocols, and let Ω and Ω' be arbitrary interval-wise guarantees. Then, we have*

$$\begin{aligned}
\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \bigcap_{(P_1, P_2, \epsilon, \boldsymbol{\sigma}) \in \Omega} (\boldsymbol{\sigma}\mathcal{S})^{[P_1, P_2]:\epsilon} \quad \wedge \quad \mathcal{S} \xrightarrow{\boldsymbol{\pi}'} \bigcap_{(P_1, P_2, \epsilon, \boldsymbol{\sigma}) \in \Omega'} (\boldsymbol{\sigma}\mathcal{T})^{[P_1, P_2]:\epsilon} \\
\implies \quad \mathcal{R} \xrightarrow{\boldsymbol{\pi}' \circ \boldsymbol{\pi}} \bigcap_{\substack{(P_1, P_2, \epsilon, \boldsymbol{\sigma}) \in \Omega \\ (P_1', P_2', \epsilon', \boldsymbol{\sigma}') \in \Omega'}} (\boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T})^{[P_1 \wedge P_1', P_2 \vee P_2']:\tilde{\epsilon}},
\end{aligned}$$

where $\tilde{\epsilon} \coloneqq (\epsilon_{\boldsymbol{\pi}'})_{[P_1 \wedge P_1', P_2 \vee P_2']} + (\epsilon'_{\boldsymbol{\sigma}})_{[P_1 \wedge P_1', P_2 \vee P_2']}$. Furthermore, we have

$$\mathcal{R} \xrightarrow{\boldsymbol{\pi}} \bigcap_{(P_1, P_2, \epsilon, \boldsymbol{\sigma}) \in \Omega} (\boldsymbol{\sigma}\mathcal{S})^{[P_1, P_2]:\epsilon} \implies [\mathcal{R}, \mathcal{T}] \xrightarrow{\boldsymbol{\pi}} \bigcap_{(P_1, P_2, \epsilon, \boldsymbol{\sigma}) \in \Omega} (\boldsymbol{\sigma}[\mathcal{S}, \mathcal{T}])^{[P_1, P_2]:\epsilon_{\mathcal{T}}}.$$

Proof. We first prove sequential composition. In the following, let $(P_1, P_2, \epsilon, \boldsymbol{\sigma}) \in \Omega$ and $(P_1', P_2', \epsilon', \boldsymbol{\sigma}') \in \Omega'$. From the first part of the assumption, Theorem 14, and composition order invariance we obtain

$$\boldsymbol{\pi}'\boldsymbol{\pi}\mathcal{R} \subseteq \boldsymbol{\pi}'\big((\boldsymbol{\sigma}\mathcal{S})^{[P_1,P_2]:\epsilon}\big) \subseteq (\boldsymbol{\pi}'\boldsymbol{\sigma}\mathcal{S})^{[P_1,P_2]:\epsilon_{\boldsymbol{\pi}'}} \subseteq (\boldsymbol{\sigma}\boldsymbol{\pi}'\mathcal{S})^{[P_1,P_2]:\epsilon_{\boldsymbol{\pi}'}}.$$

Moreover, using the second part of the assumption and Theorem 14 yields

$$\boldsymbol{\sigma}\boldsymbol{\pi}'\mathcal{S} \subseteq \boldsymbol{\sigma}\big((\boldsymbol{\sigma}'\mathcal{T})^{[P_1',P_2']:\epsilon'}\big) \subseteq (\boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T})^{[P_1',P_2']:\epsilon'_{\boldsymbol{\sigma}}}.$$

Combining the two statements, and using the monotonicity of relaxations, we get

$$\boldsymbol{\pi}'\boldsymbol{\pi}\mathcal{R} \subseteq \big((\boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T})^{[P_1',P_2']:\epsilon'_{\boldsymbol{\sigma}}}\big)^{[P_1,P_2]:\epsilon_{\boldsymbol{\pi}'}} \subseteq (\boldsymbol{\sigma}\boldsymbol{\sigma}'\mathcal{T})^{[P_1\wedge P_1', P_2\vee P_2']:\tilde{\epsilon}},$$

where in the last step we used Theorem 13, directly implying the sequential composition property.

Now consider parallel composition. We directly obtain

$$\begin{aligned}
\boldsymbol{\pi}\big[\mathcal{R},\mathcal{T}\big] &= \big[\boldsymbol{\pi}\mathcal{R},\mathcal{T}\big] \\
&\subseteq \big[(\boldsymbol{\sigma}\mathcal{S})^{[P_1,P_2]:\epsilon},\mathcal{T}\big] \\
&\subseteq \big[\boldsymbol{\sigma}\mathcal{S},\mathcal{T}\big]^{[P_1,P_2]:\epsilon_{\mathcal{T}}} \\
&= \big(\boldsymbol{\sigma}\big[\mathcal{S},\mathcal{T}\big]\big)^{[P_1,P_2]:\epsilon_{\mathcal{T}}},
\end{aligned}$$

where in the first and the last step we used composition order invariance, in the second step the assumption, and in the third step Theorem 14. □

D Details of Section 4

D.1 The Coin-Tossing Converters

In this section, we provide the formal definition of the two converters of Blum's protocol for constructing a single-bit coin-toss resource. Note that in the protocol Alice acts as the initiator, and Bob as the responder, respectively. The pseudo-code description is presented in Figure 10.

D.2 The Simulators

A formal description of the two simulators involved in the construction statements of the coin-toss resource is depicted in Figure 11. The left $\sigma_{\mathsf{CT}}^{\mathsf{A}}$ is used to formalize security against a potentially dishonest Alice (initiator), and the right $\sigma_{\mathsf{CT}}^{\mathsf{B}}$ is used to formalize security against a potentially dishonest Bob (responder), respectively.

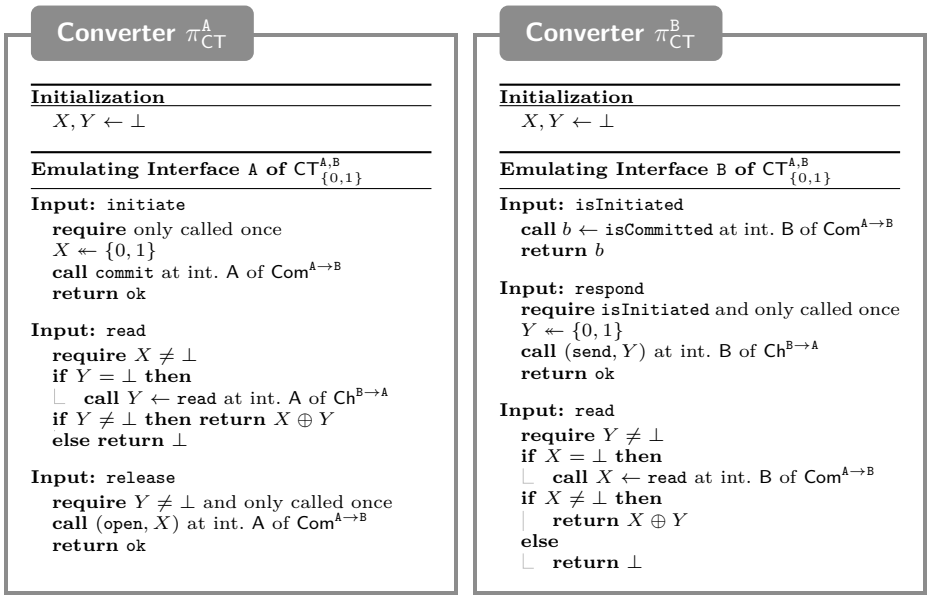

Fig. 10: A formal description of the coin-tossing converters.

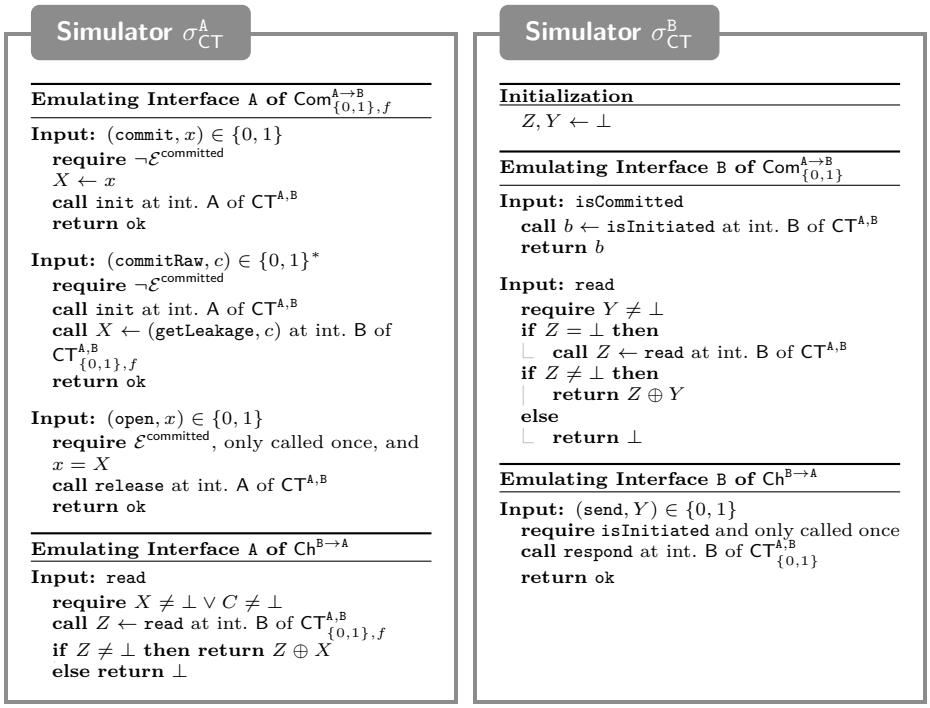

Fig. 11: A formal description of the respective coin-tossing simulators.