

Concurrently Composable Security With Shielded Super-polynomial Simulators

Brandon Broadnax^{1,*}, Nico Döttling^{2,**}, Gunnar Hartung¹, Jörn Müller-Quade¹, and Matthias Nagel^{1,*}

¹ Karlsruhe Institute of Technology, Germany

{brandon.broadnax,gunnar.hartung,joern.mueller-quade,matthias.nagel}@kit.edu

² University of California Berkeley (nico.doettling@gmail.com)

Abstract. We propose a new framework for concurrently composable security that relaxes the security notion of UC security. As in previous frameworks, our notion is based on the idea of providing the simulator with super-polynomial resources. However, in our new framework simulators are only given *restricted access* to the results computed in super-polynomial time. This is done by modeling the super-polynomial resource as a stateful oracle that may directly interact with a functionality without the simulator seeing the communication. We call these oracles “shielded oracles”.

Our notion is fully compatible with the UC framework, i. e., protocols proven secure in the UC framework remain secure in our framework. Furthermore, our notion lies strictly between SPS and Angel-based security, while being closed under protocol composition.

Shielding away super-polynomial resources allows us to apply new proof techniques where we can replace super-polynomial entities by indistinguishable polynomially bounded entities. This allows us to construct secure protocols in the plain model using weaker primitives than in previous Angel-based protocols. In particular, we only use non-adaptive-CCA-secure commitments as a building block in our constructions.

We show that constant-round general MPC in the plain model based on standard polynomial-time hardness assumptions is feasible in our framework. We present two constructions, one of which is fully black-box and therefore the *first* black-box constant-round concurrently secure general MPC protocol in the plain model based on polynomial-time hardness assumptions.

* This work was supported by the German Federal Ministry of Education and Research within the framework of the project “Sicherheit vernetzter Infrastrukturen (SVI)” in the Competence Center for Applied Security Technology (KASTEL).

** This work was supported by the DAAD (German Academic Exchange Service) under the postdoctoral program (57243032) and in part supported by European Research Council Starting Grant 279447. Research supported in part from a DARPA/ARL SAFEWARE award, AFOSR Award FA9550-15-1-0274, and NSF CRII Award 1464397. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

1 Introduction

Cryptographic protocols typically run in a network where multiple protocols interact with each other. Some of them may even act in an adversarial manner. This makes designing protocols that are secure in such a general setting a complicated task. The universal composability (UC) framework [Can01] provides means for designing and analyzing cryptographic protocols in this concurrent setting. More specifically, it captures a security notion that implies two major properties: *general concurrent security* and *modular analysis*. The former means that a protocol remains secure even when run in an environment with multiple instances of arbitrary protocols. The latter implies that one can deduce the security of a protocol from its components. Unfortunately, there exist strong impossibility results [CF01; CKL03; Lin03; PR08; KL11] regarding the realizability of cryptographic tasks in the UC framework: One requires trusted setup assumptions in order to design UC-secure protocols for many cryptographic tasks. UC-secure protocols have thus been constructed based on various setup assumptions [Can+02; Bar+04; Can+07; KLP07; Kat07; CPS07; LPV09; Dac+13]. However, if the trusted setup is compromised, all security guarantees are lost. In general, one would like to base the security of cryptographic protocols on as little trust as possible.

In order to drop the requirement for trusted setup, relaxed notions of security have been developed. One of the most prominent solutions is “UC security with super-polynomial time simulators” (SPS), introduced in [Pas03]. In this model, the simulator is allowed to run in *super-polynomial time*, thereby overcoming the impossibility results. Various multi-party computation protocols without trusted setup that satisfy this notion have been constructed, e. g., [Pas03; BS05; LPV09; LPV12; Gar+12; Dac+13; Ven14]. SPS security weakens the security of the UC framework because the simulator, being able to run in super-polynomial time, may now be able to carry out stronger attacks in the ideal setting. Still, this security notion is meaningful, since for many cryptographic tasks the ideal setting has an information-theoretic nature. Contrary to UC security, however, security in this model is not closed under protocol composition. As a consequence, this notion neither implies general concurrent security nor modular analysis.

“Angel-based security” [PS04] overcomes these issues. In this model, both the adversary and the simulator have access to an oracle called “(Imaginary) Angel” that provides super-polynomial resources for *specific* computational problems. Many general MPC protocols without setup have been constructed in the Angel-based framework [PS04; MMY06; CLP10; LP12; KMO14; Kiy14; Goy+15; HV16]. Like UC-security, this notion is closed under protocol composition. Furthermore, Angel-based security implies SPS security. In fact, it provides a stronger security notion since the simulator has only access to specific super-polynomial computations. [CLP10] later recast the Angel-based security model in the extended UC (EUC) framework [Can+07] and dubbed their notion “UC with super-polynomial helpers”. In contrast to the non-interactive and stateless Angels in previous works, the “helpers” in [CLP10] are highly interactive and stateful.

In this work, we take this framework a step further. In our new framework, simulators only have *restricted access* to the results computed in super-polynomial

time. More specifically, we model the super-polynomial resources as stateful oracles that are “glued” to an ideal functionality. These oracles may directly interact with the functionality without the simulator observing the communication. The outputs of these oracles are therefore “shielded away” from the simulator. As with Angel-based security, our notion implies SPS security. Moreover, it can be shown that our notion is in fact strictly weaker than Angel-based security. Furthermore, our notion comes with a composition theorem guaranteeing general concurrent security. While modular analysis is not directly implied for technical reasons, using our composition theorem one can achieve modular analysis by constructing protocols with strong composition features. Protocols with these features can be “plugged” into large classes of UC-secure protocols in such a way that the composed protocol is secure in our framework. As a proof of concept, we construct a constant-round commitment scheme with such features.

In order to obtain a composable security notion, environments are “augmented” in our framework, i. e., they may invoke additional (ideal) protocols that include shielded oracles. Since the super-poly computations in these protocols are hidden away, these augmented environments have the unique property that they do not “hurt” protocols proven secure in the UC framework. Therefore, our notion is in fact fully compatible with the UC framework. Moreover, our concept of “shielding away” super-polynomial resources allows us to apply new proof techniques not applicable in previous frameworks: We are able to replace entities involving super-polynomial resources in our proofs by indistinguishable polynomially bounded entities. This allows us to construct (constant-round) protocols using weaker primitives than in previous Angel-based protocols.

1.1 Our results

We propose a new framework that is based on the idea of granting simulators only restricted access to the results of a super-polynomial oracle. We have the following results:

- *New Composable Security Notion:* Our notion of security is closed under general composition, it implies SPS security and is strictly weaker than Angel-based security. (Theorem 9, Proposition 8, Theorem 17)
- *UC-compatibility:* Protocols proven secure in the UC framework are also secure in our new framework. (Theorem 12, Corollary 13)
- *Modular Composition:* As a proof of concept, we present a constant-round commitment scheme in the plain model based on OWPs that is secure in our framework and can be “plugged” into a large class of UC-secure protocols, such that the composite protocol is secure in our framework. Furthermore, this construction can be made fully black-box based on homomorphic commitment schemes. To the best of our knowledge, this is the first constant-round (black-box) commitment scheme in the plain model based on a standard polynomial-time hardness assumption with such a composition feature. (Theorem 21, Corollary 22, Corollary 23, Theorem 26, Corollary 30)

- *Constant-round (black-box) MPC*: We show that constant-round general MPC in the plain model based on standard polynomial-time hardness assumptions is feasible in our framework. We present two constructions: a non-black-box protocol based on enhanced trapdoor permutations and a black-box construction based on homomorphic commitment schemes and PKE with oblivious public-key generation. The latter is the first black-box construction of a constant-round concurrently secure general MPC protocol with SPS security in the plain model based on polynomial-time hardness assumptions. (Theorem 31)
- *Building on non-adaptive CCA-commitments*: Our constructions require weaker primitives than previous Angel-based protocols. Specifically, it suffices to use non-adaptive CCA-secure commitment schemes as a building block in our constructions instead of CCA-secure commitment schemes used previously. (Theorem 21, Theorem 26)

2 Related Work

The frameworks most related to ours are SPS and Angel-based security.

SPS security, introduced by [Pas03], provides a meaningful security notion for many cryptographic tasks such as commitment schemes or oblivious transfer. However, SPS security does not come with a composition theorem. There exist many constructions (in the plain model) satisfying this notion, e.g., [Pas03; BS05; LPV09; LPV12; Gar+12; Dac+13; Ven14]. Notably, [LPV12; Gar+12] constructed (non-black-box) constant-round general MPC protocols based on standard polynomial-time hardness assumptions.

Angel-based security [PS04] implies SPS security and comes with a composition theorem. Various general MPC protocols without setup have been constructed in the Angel-based setting [PS04; MMY06; CLP10; LP12; KMO14; Kiy14; Goy+15; HV16]. Some rely on non-standard or super-polynomial time assumptions [PS04; MMY06; KMO14]. The construction in [CLP10] is the first one to rely on standard polynomial-time assumptions, but has non-constant round complexity. Later works [Goy+15; Kiy14] have improved the round-complexity, while also relying on standard assumptions. Notably, the construction of [Kiy14] requires $\tilde{O}(\log^2 n)$ rounds and makes only black-box use of the underlying cryptographic primitive. Some Angels in the literature, e.g., [CLP10; KMO14; Kiy14; Goy+15] come with a feature called “robustness” which guarantees that any attack mounted on a constant-round protocol using this angel can be carried out by a polytime adversary with no angels. Protocols proven secure for robust Angels can be “plugged” into UC-secure protocols, resulting in Angel-secure protocols. All known constructions for robust Angels based on standard polytime assumptions require a super-constant number of rounds. Moreover, [CLP13] construct a (super-constant-round) protocol that is secure in the Angel-based setting and additionally preserves certain security properties of other protocols running in the system. They call such protocols “environmentally friendly”.

We want to note that other security notions in the concurrent setting have been proposed that are not based on the idea of simulators with super-polynomial resources. The “multiple ideal query model” [GJO10; GJ13; GGJ13; CGJ15] considers simulators that are allowed to make more than one output query per session to the ideal functionality. Another (not simulation-based) notion is “input indistinguishability” [MPR06; Gar+12] which guarantees that an adversary cannot decide which inputs have been used by the honest protocol parties. We note that this security notion is incomparable to ours.

3 Shielded Oracles

3.1 Definition of the Framework

Our model is based on the universal composability framework (UC). In this model, a protocol π carrying out a given task is defined to be secure by comparing it to an *ideal functionality* \mathcal{F} , which is a trusted and incorruptible party that carries out a given task in an ideally secure way. π is said to be secure if it “emulates” \mathcal{F} . For a more detailed description of the UC framework, see Appendix A.

While the plain UC model leaves open how session identifiers and corruptions are organized, we follow the convention that both must be consistent with the hierarchical order of the protocols: The session identifier (*sid*) of a sub-protocol must be an extension of the session id of the calling protocol. Likewise, in order to corrupt a sub-party, an adversary must corrupt all parties that are above that sub-party in the protocol hierarchy. Again, see Appendix A for more details.

We relax the UC security notion by introducing a super-polynomial time machine that may aid the simulator. This machine is modeled as a *stateful oracle* \mathcal{O} that is “glued” to an the ideal functionality \mathcal{F} . \mathcal{O} may freely interact with the simulator and \mathcal{F} . However, the simulator does not “see” the communication between between \mathcal{O} and \mathcal{F} . Since the output of the oracle is partially hidden from the simulator, we call \mathcal{O} a *shielded oracle*.

Definition 1 (Shielded oracles). *A shielded oracle is a stateful oracle \mathcal{O} that can be implemented in super-polynomial time. By convention, the outputs of a shielded oracle \mathcal{O} are of the form (output-to-funct, y) or (output-to-adv, y).*

The simulator is allowed to communicate with the functionality *only* via the shielded oracle. This way, the shielded oracle serves as an interface that carries out specific tasks the simulator could not do otherwise. The communication between the shielded oracle and the functionality is hidden away from the simulator. The actions of the shielded oracle may depend on the session identifier (*sid*) of the protocol session as well as the party identifiers of the corrupted parties.

Definition 2 (\mathcal{O} -adjoined functionalities). *Given a functionality \mathcal{F} and a shielded oracle \mathcal{O} , define the interaction of the \mathcal{O} -adjoined functionality $\mathcal{F}^{\mathcal{O}}$ in an ideal protocol execution with session identifier *sid* as follows (See Fig. 1, p. 43 for a graphical depiction):*

- $\mathcal{F}^\mathcal{O}$ internally runs an instance of \mathcal{F} with session identifier sid
- When receiving the first message x from the adversary, $\mathcal{F}^\mathcal{O}$ internally invokes \mathcal{O} with input (sid, x) .
All subsequent messages from the adversary are passed to \mathcal{O} .
- Messages between the honest parties and \mathcal{F} are forwarded.
- Corruption messages are forwarded to \mathcal{F} and \mathcal{O} .
- When \mathcal{F} sends a message y to the adversary, $\mathcal{F}^\mathcal{O}$ passes y to \mathcal{O} .
- The external write operations of \mathcal{O} are treated as follows:
 - If \mathcal{O} sends `(output-to-funct, y)`, $\mathcal{F}^\mathcal{O}$ sends y to \mathcal{F} .
 - If \mathcal{O} sends `(output-to-adv, y)`, $\mathcal{F}^\mathcal{O}$ sends y to the adversary.

Let $\text{IDEAL}(\mathcal{F}^\mathcal{O})$ be the ideal protocol with functionality $\mathcal{F}^\mathcal{O}$ as defined in [Can01].

In order to obtain a composable security notion, we introduce the notion of *augmented environments*. Augmented environments are UC environments that may invoke, apart from the challenge protocol, polynomially many instances of $\text{IDEAL}(\mathcal{F}^\mathcal{O})$ for a given functionality $\mathcal{F}^\mathcal{O}$. The only restriction is that the session identifiers of these instances as well as the session identifier of the challenge protocol are not extensions of one another.

Augmented environments may send inputs to and receive outputs from any invoked instance of $\text{IDEAL}(\mathcal{F}^\mathcal{O})$. In addition, augmented environments can play the role of any adversary via the adversary’s interface of the functionality. In particular, augmented environments may corrupt parties sending the corresponding corruption message as input to the functionality.

In what follows we give a definition of an execution experiment with an $\mathcal{F}^\mathcal{O}$ -augmented environment. For simplicity and due to space constraints, the description is kept informal.

Definition 3 (The $\mathcal{F}^\mathcal{O}$ -execution experiment). *An execution of a protocol σ with adversary \mathcal{A} and an $\mathcal{F}^\mathcal{O}$ -augmented environment \mathcal{Z} on input $a \in \{0, 1\}^*$ and with security parameter $n \in \mathbb{N}$ is a run of a system of interactive Turing machines (ITMs) with the following restrictions (See Fig. 2, p. 43 for a graphical depiction):*

- First, \mathcal{Z} is activated on input $a \in \{0, 1\}^*$.
- The first ITM to be invoked by \mathcal{Z} is the adversary \mathcal{A} .
- \mathcal{Z} may invoke a single instance of a challenge protocol, which is set to be σ by the experiment. The session identifier of σ is determined by \mathcal{Z} upon invocation.
- \mathcal{Z} may pass inputs to the adversary or the protocol parties of σ .
- \mathcal{Z} may invoke, send inputs to and receive outputs from instances of $\text{IDEAL}(\mathcal{F}^\mathcal{O})$ as long as the session identifiers of these instances as well as the session identifier of the instance of σ are not extensions of one another.
- The adversary \mathcal{A} may send messages to protocol parties of σ as well as to the environment.
- The protocol parties of σ may send messages to \mathcal{A} , pass inputs to and receive outputs from subparties and give outputs to \mathcal{Z} .

Denote by $\text{Exec}(\sigma, \mathcal{A}, \mathcal{Z}[\mathcal{F}^\mathcal{O}])(n, a)$ the output of the $\mathcal{F}^\mathcal{O}$ -augmented environment \mathcal{Z} on input $a \in \{0, 1\}^*$ and with security parameter $n \in \mathbb{N}$ when interacting with σ and \mathcal{A} according to the above definition.

Define $\text{Exec}(\sigma, \mathcal{A}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) = \{\text{Exec}(\sigma, \mathcal{A}, \mathcal{Z}[\mathcal{F}^\mathcal{O}])(n, a)\}_{n \in \mathbb{N}, a \in \{0, 1\}^*}$

We will now define security in our framework in total analogy to the UC framework:

Definition 4 ($\mathcal{F}^\mathcal{O}$ -emulation). Let π and ϕ be protocols. π is said to emulate ϕ in the presence of $\mathcal{F}^\mathcal{O}$ -augmented environments, denoted by $\pi \geq_{\mathcal{F}^\mathcal{O}} \phi$, if for any PPT adversary \mathcal{A} there exists a PPT adversary (called “simulator”) \mathcal{S} such that for every $\mathcal{F}^\mathcal{O}$ -augmented PPT environment \mathcal{Z} it holds that

$$\text{Exec}(\pi, \mathcal{A}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \stackrel{c}{\equiv} \text{Exec}(\phi, \mathcal{S}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \quad (1)$$

Throughout this paper, we only consider *static* corruptions.

3.2 Basic Properties and Justification

In this section, we show that that our security notion is transitive and that the dummy adversary is complete within this notion. As a justification for our notion, we show that it implies super-polynomial time simulator (SPS) security.

Definition 5 ($\mathcal{F}^\mathcal{O}$ -emulation with respect to the dummy adversary). The dummy adversary \mathcal{D} is an adversary that when receiving a message (sid, pid, m) from the environment, sends m to the party with party identifier pid and session identifier sid , and that, when receiving m from the party with party identifier pid and session identifier sid , sends (sid, pid, m) to the environment.

Let π and ϕ be protocols. π is said to emulate ϕ in the presence of $\mathcal{F}^\mathcal{O}$ -augmented environments with respect to the dummy adversary, if

$$\exists \mathcal{S}_{\mathcal{D}} \forall \mathcal{Z} : \text{Exec}(\pi, \mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \stackrel{c}{\equiv} \text{Exec}(\phi, \mathcal{S}_{\mathcal{D}}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \quad (2)$$

Proposition 6 (Completeness of the dummy adversary). Let π and ϕ be protocols. Then, π emulates ϕ in the presence of $\mathcal{F}^\mathcal{O}$ -augmented environments if and only if π emulates ϕ in the presence of $\mathcal{F}^\mathcal{O}$ -augmented environments with respect to the dummy adversary.

The proof is almost exactly the same as in [Can01], and therefore deferred to Appendix B.1. The proof for transitivity is straightforward and can be found in Appendix B.2.

Proposition 7 (Transitivity). Let π_1, π_2, π_3 be protocols. If $\pi_1 \geq_{\mathcal{F}^\mathcal{O}} \pi_2$ and $\pi_2 \geq_{\mathcal{F}^\mathcal{O}} \pi_3$ then it holds that $\pi_1 \geq_{\mathcal{F}^\mathcal{O}} \pi_3$.

In order to justify our new notion, we prove that security with respect to $\mathcal{F}^\mathcal{O}$ -emulation implies security with respect to SPS-emulation which we will denote by \geq_{SPS} . For a formal definition of $\pi \geq_{\text{SPS}} \phi$ see Definition 32 in Appendix B.3. The proof is straightforward: View the oracle as part of the simulator. This simulator runs in super-polynomial time, hence can be simulated by an SPS-simulator (cf. Fig. 4, p. 44).

Proposition 8 ($\mathcal{F}^\mathcal{O}$ -emulation implies SPS-emulation). *Let \mathcal{O} be a shielded oracle. Assume $\pi \geq_{\mathcal{F}^\mathcal{O}} \mathcal{F}^\mathcal{O}$. Then it holds that $\pi \geq_{\text{SPS}} \mathcal{F}$.*

3.3 Universal Composition

A central property of the UC framework is the universal composition theorem. This theorem guarantees that the security of a protocol is *closed* under protocol composition. This means that security guarantees can be given for a UC-secure protocol even if multiple other protocols interact with this protocol in a potentially adversarial manner. We prove a similar theorem in our framework. More specifically, we generalize the universal composition theorem to also include $\mathcal{F}^\mathcal{O}$ -hybrid protocols.

Theorem 9 (Composition theorem). *Let \mathcal{O} be a shielded oracle, \mathcal{F} and \mathcal{G} functionalities.*

1. (Polynomial hybrid protocols) *Let $\pi, \rho^\mathcal{G}$ be protocols. Assume $\pi \geq_{\mathcal{F}^\mathcal{O}} \mathcal{G}$. Then it holds that $\rho^\pi \geq_{\mathcal{F}^\mathcal{O}} \rho^\mathcal{G}$.*
2. ($\mathcal{F}^\mathcal{O}$ -hybrid protocols) *Let π be a protocol, $\rho^{\mathcal{F}^\mathcal{O}}$ a protocol in the $\mathcal{F}^\mathcal{O}$ -hybrid model. Assume $\pi \geq_{\mathcal{F}^\mathcal{O}} \mathcal{F}^\mathcal{O}$. Then it holds that $\rho^\pi \geq_{\mathcal{F}^\mathcal{O}} \rho^{\mathcal{F}^\mathcal{O}}$.*

Proof (of the second statement). For single instance composition (where ρ calls only a single instance of π), treat ρ as part of the environment and use the premise that $\pi \geq_{\mathcal{F}^\mathcal{O}} \mathcal{F}^\mathcal{O}$.

For the general case iteratively apply the single instance composition theorem. In each iteration a new instance of $\text{IDEAL}(\mathcal{F}^\mathcal{O})$ is replaced by an instance of π , and the remaining instances of π , $\text{IDEAL}(\mathcal{F}^\mathcal{O})$ and ρ are treated as part of the augmented environment. The claim then follows using transitivity. (See Fig. 3, p. 44 for a graphical depiction.) \square

The universal composition theorem in the UC framework has two important implications: general concurrent security and modular analysis. The former means that a protocol remains secure even when run in an environment with multiple instances of arbitrary protocols. The latter implies that one can deduce the security of a protocol from its components.

Theorem 9 directly implies general concurrent security (with super-polynomial time simulators). However, modular analysis is not directly implied by Theorem 9. This is because the oracle \mathcal{O} may contain all “complexity” of the protocol π , i. e., proving security of $\rho^{\mathcal{F}^\mathcal{O}}$ may be as complex as proving security of ρ^π .

Still, one can use Theorem 9 to achieve modular analysis by constructing secure protocols with strong composition features. A protocol π with such composition features allows analyzing the security of a large class of protocols $\rho^{\mathcal{F}}$ in the UC framework and achieve security in our framework when replacing \mathcal{F} with π . As a proof of concept, we will show, using Theorem 9, that a large a class of protocols in the \mathcal{F}_{com} -hybrid model can be composed with a commitment protocol presented in this paper (Theorem 26).

The following is a useful extension of Theorem 9 for multiple oracles. (See Appendix B.4 for a proof.)

Corollary 10 (Composition theorem for multiple oracles). *Let $\mathcal{O}, \mathcal{O}'$ be shielded oracles. Assume that $\pi \geq_{\mathcal{F}^{\mathcal{O}}} \mathcal{F}^{\mathcal{O}}$ and $\rho^{\mathcal{F}^{\mathcal{O}}} \geq_{\mathcal{F}^{\mathcal{O}}, \mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$. Then there exists a shielded oracle \mathcal{O}'' such that $\rho^{\pi} \geq_{\mathcal{G}^{\mathcal{O}''}} \mathcal{G}^{\mathcal{O}''}$.*

3.4 Polynomial Simulatability

We show a unique feature of our framework: For appropriate oracles to be defined below, augmented environments do not “hurt” UC-secure protocols. This means that a protocol that was proven secure in the UC framework is secure in our framework, too. This makes our security notion fully compatible with UC security.

Definition 11 (Polynomial simulatability). *Let \mathcal{O} be a shielded oracle, \mathcal{F} a functionality. Say that \mathcal{O} adjoined to \mathcal{F} is polynomially simulatable if there exists a (PPT) functionality \mathcal{M} such that for all $\mathcal{F}^{\mathcal{O}}$ -augmented environments \mathcal{Z} it holds that*

$$\mathcal{F}^{\mathcal{O}} \geq_{\mathcal{F}^{\mathcal{O}}} \mathcal{M} \quad (3)$$

If a functionality $\mathcal{F}^{\mathcal{O}}$ is polynomially simulatable then the super-polynomial power of the oracle \mathcal{O} is totally “shielded away” from the environment. Note that in Definition 11, indistinguishability must hold for *augmented* environments not only for polynomial environments.

As a consequence, $\mathcal{F}^{\mathcal{O}}$ -augmented environments can be replaced by *efficient* environments if $\mathcal{F}^{\mathcal{O}}$ is polynomially simulatable.

Theorem 12 (Reduction to polynomial time environments). *Let \mathcal{O} be a shielded oracle and \mathcal{F} a functionality such that $\mathcal{F}^{\mathcal{O}}$ is polynomially simulatable. Let π, ϕ be protocols that are PPT or in the $\mathcal{F}^{\mathcal{O}}$ -hybrid model. It holds that*

$$\pi \geq_{\mathcal{F}^{\mathcal{O}}} \phi \iff \pi \geq_{\text{poly}} \phi \quad (4)$$

where the right-hand side means that π emulates ϕ in the presence of all $\mathcal{F}^{\mathcal{O}}$ -augmented environments that never invoke an instance of $\text{IDEAL}(\mathcal{F}^{\mathcal{O}})$.

Proof. *Poly-emulation implies $\mathcal{F}^{\mathcal{O}}$ -emulation:* Replace all instances of $\text{IDEAL}(\mathcal{F}^{\mathcal{O}})$ with instances of \mathcal{M} using the fact that $\mathcal{F}^{\mathcal{O}}$ is polynomially simulatable. Treat

all instances of \mathcal{M} as part of the environment. This new environment runs in polynomial time. Substitute π by ϕ using the premise. Replace all instances of \mathcal{M} with instances of $\text{IDEAL}(\mathcal{F}^\mathcal{O})$ again. The statement follows.

The converse is trivial. \square

As augmented environments that never invoke instances of $\text{IDEAL}(\mathcal{F}^\mathcal{O})$ are identical to an UC-environment, the following corollary immediately follows.

Corollary 13 (Compatibility with the UC framework). *Let \mathcal{O} be a shielded oracle and \mathcal{F} a functionality such that $\mathcal{F}^\mathcal{O}$ is polynomially simulatable. It holds that*

$$\pi \underset{\mathcal{F}^\mathcal{O}}{\geq} \phi \iff \pi \underset{\text{UC}}{\geq} \phi \quad (5)$$

Note that this does not contradict the classical impossibility results for the plain UC framework (cp. [CF01]): If $\pi \underset{\mathcal{F}^\mathcal{O}}{\geq} \mathcal{F}^\mathcal{O}$ for a polynomially simulatable $\mathcal{F}^\mathcal{O}$, then this only means that $\pi \underset{\text{UC}}{\geq} \mathcal{F}^\mathcal{O}$, but it does not follow that $\pi \underset{\text{UC}}{\geq} \mathcal{F}$. Although the super-polynomial power of \mathcal{O} is shielded away from the outside, it is indeed necessary.

Replacing augmented environments with efficient environments will be a key property in various proofs later in this paper. In particular, it will allow us to prove the security of protocols in our framework using relatively weak primitives such as *non-adaptively-secure-CCA* commitments as opposed to CCA-secure commitments, which are commonly used in Angel-based protocols.

Next, we show that by suitably tweaking a given oracle \mathcal{O} one can make $\mathcal{F}^\mathcal{O}$ polynomially simulatable while preserving the security relation.

Lemma 14 (Derived oracle). *Let \mathcal{O} be a shielded oracle such that $\pi \underset{\mathcal{F}^\mathcal{O}}{\geq} \mathcal{F}^\mathcal{O}$. Then there exists a shielded oracle \mathcal{O}' such that $\pi \underset{\mathcal{F}^{\mathcal{O}'}}{\geq} \mathcal{F}^{\mathcal{O}'}$ and additionally \mathcal{O}' adjoined to \mathcal{F} is polynomially simulatable.*

Proof. (See Fig. 5, p. 45 for a graphical depiction of the proof.) Since π emulates $\mathcal{F}^\mathcal{O}$, there exists a simulator $\mathcal{S}_\mathcal{D}$ for the dummy adversary \mathcal{D} . Define the shielded oracle \mathcal{O}' as follows: \mathcal{O}' internally simulates $\mathcal{S}_\mathcal{D}$ and \mathcal{O} , passes each message $\mathcal{S}_\mathcal{D}$ sends to \mathcal{F} to \mathcal{O} , sends each `output-to-funct` output from \mathcal{O} to \mathcal{F} and each `output-to-adv` output from \mathcal{O} to $\mathcal{S}_\mathcal{D}$, and forwards the communication between $\mathcal{S}_\mathcal{D}$ and the environment. By construction, for all $\mathcal{F}^\mathcal{O}$ -augmented environments \mathcal{Z} it holds that

$$\text{Exec}(\pi, \mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \stackrel{c}{\equiv} \text{Exec}(\mathcal{F}^\mathcal{O}, \mathcal{S}_\mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \equiv \text{Exec}(\mathcal{F}^{\mathcal{O}'}, \mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \quad (6)$$

It follows from Proposition 6 that $\pi \underset{\mathcal{F}^\mathcal{O}}{\geq} \mathcal{F}^{\mathcal{O}'}$ and $\mathcal{F}^{\mathcal{O}'} \underset{\mathcal{F}^\mathcal{O}}{\geq} \pi$. Since $\mathcal{S}_\mathcal{D}$ runs in polynomial time, $\mathcal{F}^\mathcal{O}$ -augmented environments can simulate $\mathcal{F}^{\mathcal{O}'}$ -augmented environments. Therefore, $\pi \underset{\mathcal{F}^{\mathcal{O}'}}{\geq} \mathcal{F}^{\mathcal{O}'}$ and $\mathcal{F}^{\mathcal{O}'} \underset{\mathcal{F}^{\mathcal{O}'}}{\geq} \pi$. The theorem follows by defining \mathcal{M} to be the functionality that internally simulates the protocol π . \square

The following corollary shows that UC-secure protocols can be used as sub-protocols in protocols proven secure in our framework, while preserving security.

Corollary 15 (Composition with UC-secure protocols). *Let $\pi, \rho^{\mathcal{F}}$ be protocols such that $\pi \geq_{UC} \mathcal{F}$ and $\rho^{\mathcal{F}} \geq_{\mathcal{G}^{\mathcal{O}}} \mathcal{G}^{\mathcal{O}}$. Then there exists a shielded oracle \mathcal{O}' such that*

$$\rho^{\pi} \underset{\mathcal{G}^{\mathcal{O}'}}{\geq} \mathcal{G}^{\mathcal{O}'} \quad (7)$$

Proof. Since $\rho^{\mathcal{F}}$ is PPT there exists a shielded oracle \mathcal{O}' such that $\mathcal{G}^{\mathcal{O}'}$ is polynomially simulatable and $\rho^{\mathcal{F}} \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$ by Lemma 14. From Corollary 13 it follows that $\pi \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{F}$. The statement then follows from the composition theorem and the transitivity of $\mathcal{G}^{\mathcal{O}'}$ -emulation. \square

The last result demonstrates the compatibility of our framework with the UC framework again. While it is much more desirable to “plug” a protocol proven secure in our framework into a UC secure protocol—in order to obtain a secure protocol in the *plain model* (this will be addressed in Theorem 26 and Corollary 30)—doing it the other way around is still a convenient property. For instance, it allows one to instantiate “auxiliary” functionalities such as authenticated channels $\mathcal{F}_{\text{auth}}$ or secure channels \mathcal{F}_{SMT} , while preserving security.

3.5 Relation with Angel-based Security

A natural question that arises is how our security notion compares to Angel-based security. We will prove that for a large class of Angels (which to the best of our knowledge includes all Angels that can be found in the literature), Angel-based security implies our security notion. However, assuming the existence of one-way functions, the converse does not hold. Thus, our notion is *strictly weaker* than Angel-based security.

In the following, we denote by $\pi \geq_{\Gamma\text{-Angel}} \phi$ if π securely realizes ϕ with respect to an angel Γ . Note that the the following results also hold for “UC with super-polynomial helpers” put forward by [CLP10].

Definition 16 (Session-respecting Angel (informal)). *(See Appendix B.6, Definition 34 for a formal treatment.) An Angel is called session-respecting if its internal state can be regarded as a vector with independent components for each session the Angel is queried for.*

Theorem 17 (Relation between angels and shielded oracles).

1. Assume $\pi \geq_{\Gamma\text{-Angel}} \mathcal{F}$ for an imaginary Angel Γ . If Γ is session-respecting, then there exists a shielded oracle \mathcal{O} such that $\pi \geq_{\mathcal{F}^{\mathcal{O}}} \mathcal{F}^{\mathcal{O}}$.
2. Assume the existence of one-way functions. Then there exists a protocol ρ (in the $\mathcal{F}_{\text{auth}}$ -hybrid model), a functionality \mathcal{G} and a shielded oracle \mathcal{O} s. t. $\rho \geq_{\mathcal{G}^{\mathcal{O}}} \mathcal{G}^{\mathcal{O}}$ but no imaginary angel Γ can be found such that $\rho \geq_{\Gamma\text{-Angel}} \mathcal{G}$ holds.

The proof is also informal; for a more formal treatment see again Appendix B.6. For an illustration of the second part of the theorem also see Fig. 6

Proof (Idea of proof).

1. Since $\pi \geq_{\Gamma\text{-Angel}} \mathcal{F}$ we have that there exists a simulator $\mathcal{S}_{\mathcal{D}}$ such that for all environments \mathcal{Z}^{Γ} it holds that $\text{Exec}(\pi, \mathcal{D}^{\Gamma}, \mathcal{Z}^{\Gamma}) \equiv \text{Exec}(\mathcal{F}, \mathcal{S}_{\mathcal{D}}^{\Gamma}, \mathcal{Z}^{\Gamma})$.
Now, we consider the experiment with shielded oracle $\mathcal{O} = \mathcal{S}_{\mathcal{D}}^{\Gamma}$, ideal functionality $\mathcal{F}^{\mathcal{O}}$ and simulator $\mathcal{S} = \mathcal{S}_{\mathcal{D}}$. Since Γ is session-respecting the operation of the Angel is split between \mathcal{O} , which internally runs a copy of the Angel for all queries within the challenge session, and the simulator \mathcal{S} , which handles all remaining queries having access to the global Angel Γ . It follows that $\text{Exec}(\mathcal{F}, \mathcal{S}_{\mathcal{D}}^{\Gamma}, \mathcal{Z}^{\Gamma}) \equiv \text{Exec}(\mathcal{F}^{\mathcal{O}}, \mathcal{S}^{\Gamma}, \mathcal{Z}^{\Gamma})$ for all environments \mathcal{Z}^{Γ} . Hence, $\text{Exec}(\pi, \mathcal{D}, \mathcal{Z}^{\Gamma}) \equiv \text{Exec}(\mathcal{F}^{\mathcal{O}}, \mathcal{S}, \mathcal{Z}^{\Gamma})$ for all environments \mathcal{Z}^{Γ} . Since $\mathcal{F}^{\mathcal{O}}$ -augmented environments can be simulated by Γ -Angel environments, the statement follows.
2. Let $\tilde{\rho}$ be a commitment protocol such that $\tilde{\rho} \geq_{\mathcal{F}_{\text{com}}^{\mathcal{O}}} \mathcal{F}_{\text{com}}^{\mathcal{O}}$ and \mathcal{O} adjoined to \mathcal{F}_{com} is poly-simulatable. One can find such a protocol using the Angel-based protocol in [CLP10], part 1 of this theorem and Lemma 14, assuming the existence of one-way functions. Define the protocol ρ to be identical to $\tilde{\rho}$ except for the following instruction:

Before the actual commit phase begins, the receiver chooses a_1, \dots, a_n uniformly at random (n is the security parameter) and sends $\text{Commit}(a_i)$ ($i = 1, \dots, n$) to the sender (by running the program of the honest sender in $\tilde{\rho}$ with the pid of the sender). The sender replies with $(1, \dots, 1) \in \{0, 1\}^n$. The receiver then checks if the values he received from the sender equal (a_1, \dots, a_n) . If yes, the receiver outputs “11” (2-bit string). Otherwise, the protocol parties execute the protocol $\tilde{\rho}$.

By construction, it holds that $\rho \geq_{\mathcal{F}_{\text{com}}^{\mathcal{O}}} \mathcal{F}_{\text{com}}^{\mathcal{O}}$. This follows from the fact that every $\mathcal{F}^{\mathcal{O}}$ -augmented environment can be replaced by an efficient environment (since \mathcal{O} attached to \mathcal{F} is polynomially simulatable) and efficient environments can guess the correct a_i only with negligible probability (otherwise $\tilde{\rho}$ would be insecure, contradicting $\tilde{\rho} \geq_{\mathcal{F}_{\text{com}}^{\mathcal{O}}} \mathcal{F}_{\text{com}}^{\mathcal{O}}$).

Assume for the sake of contradiction that there exists an imaginary angel Γ s. t. $\rho \geq_{\Gamma\text{-Angel}} \mathcal{F}_{\text{com}}$ holds. Let the sender be corrupted. Since the adversary has access to Γ , he can run the program of the simulator. The simulator must be able to extract commitments (because $\rho \geq_{\Gamma\text{-Angel}} \mathcal{F}_{\text{com}}$). This enables the adversary to extract all a_i (by relaying the commitments from the receiver each to a different internal copy of the simulator), forcing the receiver to output “11” in the real model experiment. This cannot be simulated in the ideal model experiment, however. We have thus reached a contradiction. \square

Theorem 17 raises the question if it is possible to construct secure protocols with “interesting properties” in our framework that are not (known to be) secure in the Angel-based setting. We will answer this question in the affirmative, presenting a modular construction of a general MPC protocol in the plain model that is constant-round (and black-box) and based only on standard polynomial-time hardness assumptions (Theorem 31).

We would like to briefly note that by Theorem 17 we can already conclude that we can realize every (well-formed) functionality in our framework by importing the results of [CLP10].

Proposition 18 (General MPC in the plain model). *Assume the existence of enhanced trapdoor permutations. For every (well-formed)³ functionality \mathcal{F} , there exists an extraction oracle \mathcal{O} and a protocol ρ (in the plain model⁴) such that*

$$\rho \underset{\mathcal{F}^{\mathcal{O}}}{\geq} \mathcal{F}^{\mathcal{O}} \quad (8)$$

4 A Constant-Round Commitment Scheme

In this section we will construct a constant-round commitment scheme that is secure in our framework. We note that we assume authenticated channels and implicitly work in the $\mathcal{F}_{\text{auth}}$ -hybrid model.

Let $\langle C, R \rangle$ be a commitment scheme that we will use a building block for our bit commitment scheme Π later. We require $\langle C, R \rangle$ to be tag-based. In a tag-based commitment scheme the committer and receiver additionally use a “tag”—or identity—as part of the protocol [PR05; DDN00]. Moreover we require $\langle C, R \rangle$ to be “immediately committing” as in the following definition.

Definition 19 (Immediately committing). *A commitment scheme $\langle C, R \rangle$ is called immediately committing if the first message in the protocol comes from the sender and already perfectly determines the value committed to.*

The above definition implies that the commitment scheme is perfectly binding and super-polynomially extractable, i. e., given the transcript an extractor can find the unique message of the commitment by exhaustive search.

For the discussion of our commitment scheme, we settle the following notation. Let $s = ((s_{i,b})) \in \{0, 1\}^{2n}$ for $i \in [n]$ and $b \in \{0, 1\}$ be a $2n$ -tuple of bits. For an n -bit string $I = b_1 \cdots b_n$, we define $s_I := (s_{1,b_1}, \dots, s_{n,b_n})$. Thus I specifies a selection of n of the $s_{i,b}$, where one of these is selected from each pair $s_{i,0}, s_{i,1}$.

Construction 1. *The bit commitment scheme Π is defined as follows. Whenever the basic commitment scheme $\langle C, R \rangle$ is used, the committing party uses its pid and the sid as its tag. Let $m \in \{0, 1\}$*

- Commit(m):
 - R: Choose a random n -bit string I and commit to I using $\langle C, R \rangle$
 - S: Pick n random bits $s_{i,0}$ and compute $s_{i,1} = s_{i,0} \oplus m$ for all $i \in [n]$.
 - S and R run $2n$ sessions of $\langle C, R \rangle$ in parallel in which S commits to the s_{i,b_i} ($i \in [n], b_i \in \{0, 1\}$).
- Unveil:
 - S: Send all $s_{i,b_i} \in \{0, 1\}$ ($i \in [n], b_i \in \{0, 1\}$) to R.

³ See [Can+02] for a definition of well-formed functionalities.

⁴ A model without any trusted setup except for authenticated communication channels.

- R: Check if $s_{1,0} \oplus s_{1,1} = \dots = s_{n,0} \oplus s_{n,1}$. If this holds, unveil the string I to S.
- S: If R unveiled the string correctly, then unveil all s_I .
- R: Check if S unveiled correctly. If yes, let s'_1, \dots, s'_n be the unveiled values. Check if $s'_i = s_{i,b_i}$ for all $i \in [n]$. If so, output $m := s_{1,0} \oplus s_{1,1}$.

The above construction is reminiscent of [DS13] who presented a compiler that transforms any ideal straight-line extractable commitment scheme into an extractable and equivocal commitment scheme.

Note that if an attacker is able to learn the index set I in the commit phase then he can easily open the commitment to an arbitrary message m' by sending “fake” shares $t_{i,b}$, such that $t_I = s_I$, and $t_{-I} = s_I \oplus (m', \dots, m')$. (Here \oplus is interpreted element-wise.) Hence Π is equivocal for super-polynomial machines.

We claim that this protocol securely realizes $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ for a certain shielded oracle \mathcal{O} . We first describe \mathcal{O} , before we move to the theorem.

Construction 2. We define the actions of the shielded oracle \mathcal{O} as follows.⁵
If the sender is corrupted

- \mathcal{O} chooses a random n -bit string I , and commits to the string I to the adversary \mathcal{A} using $\langle C, R \rangle$.
- \mathcal{O} acts as honest receiver in $2n$ sessions of $\langle C, R \rangle$ in parallel. After these sessions have completed, \mathcal{O} extracts each instance of $\langle C, R \rangle$, obtaining the shares $(s_{i,b}$ for $i \in [n]$) and $b \in \{0, 1\}$. (If a commitment cannot be extracted, the corresponding share is set to \perp)
- \mathcal{O} computes $m_i := s_{i,0} \oplus s_{i,1}$ for all $i \in [n]$. (Indices i where one or both of the $s_{i,b}$ is \perp are ignored.) Let $m \in \{0, 1\}$ be the most frequently occurring m_i . (If there are multiple m_i occurring with the highest frequency, m chooses $m = 0$). \mathcal{O} relays (Commit, m) to \mathcal{F}_{com}
- When \mathcal{A} sends shares $s'_{1,0}, s'_{1,1}, \dots, s'_{n,0}, s'_{n,1}$ in the unveil phase of Π , \mathcal{O} acts as an honest receiver, unveiling I .
- Finally, if \mathcal{A} 's unveil is accepting, \mathcal{O} instructs \mathcal{F}_{com} to unveil the message.

If the receiver is corrupted

- \mathcal{O} acts as the sender in an execution of Π , engaging in a commit session of $\langle C, R \rangle$ with the adversary. If the adversary's commitment is accepting, \mathcal{O} extracts this instance of $\langle C, R \rangle$ obtaining a string I (If parts of this string cannot be extracted they are set to \perp).
- \mathcal{O} picks n random bits $s_{i,0}$, and lets $s_{i,1} = s_{i,0}$ for all $i \in [n]$, as if it were honestly committing to $m = 0$. Next, it runs $2n$ instances of Π in parallel, committing to the $s_{i,b}$.
- In the unveil phase, when \mathcal{O} learns the message m , it computes “fake” shares $t_{i,b}$ as follows: $t_I = s_I$ and $t_{-I} = s_{-I} \oplus (m, \dots, m)$ (\oplus is interpreted element-wise.). \mathcal{O} sends these shares $t_{i,b}$ to the adversary.

⁵ For ease of notation, we drop the prefixes **output-to-funct** and **output-to-adv** in the messages output by \mathcal{O} .

- \mathcal{O} acts as the honest sender in the unveil phase of Π . If \mathcal{A} 's unveil of I is accepting, then \mathcal{O} honestly executes the unveil phase for all bit shares t_I . (Otherwise, \mathcal{O} outputs nothing and ignores all further inputs.)

If **no parties are corrupted**, \mathcal{O} simulates an honest execution of protocol Π on input 0, forwarding all messages to the adversary. Since \mathcal{O} knows the index string I (because \mathcal{O} has created it itself) it can create fake shares just like in the case of a corrupted receiver.

If **both parties are corrupted**, \mathcal{O} just executes the dummy adversary \mathcal{D} internally. (Note that \mathcal{Z} only interacts with \mathcal{D} in the real experiment if both parties are corrupted).

This concludes the description of the shielded oracle \mathcal{O} . Observe that \mathcal{O} can be implemented in super-polynomial time. Also note that in the case of *both or no* party being corrupted, \mathcal{O} can be implemented in polynomial time.

Before we can state our theorem, we need another assumption about the commitment scheme $\langle C, R \rangle$.

Definition 20 (pCCA-secure commitment schemes). Let $\langle C, R \rangle$ be a tag-based commitment scheme. A pCCA-decommitment oracle \mathcal{E} interacts with an adversary \mathcal{A} in polynomial many parallel sessions of $\langle C, R \rangle$ as an honest receiver with tags chosen by the adversary. For each session that is accepting, \mathcal{E} returns a committed value compatible with that session to \mathcal{A} if such a value exists, otherwise \mathcal{E} returns \perp . (Note that when a session has multiple compatible committed values, \mathcal{E} reveals only one of them. Hence, there might exist many decommitment oracles.)

Consider the probabilistic experiment $\text{IND}_b(\langle C, R \rangle, \mathcal{A}^\mathcal{E}, 1^n, z)$ with $b \in \{0, 1\}$:

On input 1^n and auxiliary input z , the adversary \mathcal{A} adaptively chooses a pair of challenge values $v_0, v_1 \in \{0, 1\}$ together with a tag and sends them to the challenger. The challenger commits to v_b using $\langle C, R \rangle$ with that tag. The output of the experiment is the output of $\mathcal{A}^\mathcal{E}$. If any of the tags used by \mathcal{A} for queries to the pCCA-decommitment oracle equals the tag of the challenge, the output of the experiment is replaced by \perp .

$\langle C, R \rangle$ is said to be parallel-CCA-secure if there exists an \mathcal{E} s. t. for all PPT adversaries \mathcal{A} it holds that:⁶

$$\text{IND}_0(\langle C, R \rangle, \mathcal{A}^\mathcal{E}, 1^n, z) \stackrel{c}{\equiv} \text{IND}_1(\langle C, R \rangle, \mathcal{A}^\mathcal{E}, 1^n, z)$$

Note that previous protocols proven secure in the Angel-based framework required (adaptive) CCA-secure commitments schemes [CLP10; Goy+15; Kiy14]. For our notion it suffices to assume parallel-CCA-secure (i. e. non-adaptive) commitment schemes as a building block.

Theorem 21. Assume that $\langle C, R \rangle$ is parallel-CCA-secure and immediately committing. Then $\Pi \geq_{\mathcal{F}_{com}^\mathcal{O}} \mathcal{F}_{com}^\mathcal{O}$, where Π is as defined in Construction 1 and \mathcal{O} is the shielded oracle as defined in Construction 2.

⁶ Note that in our special case the decommitment oracle \mathcal{E} is unique since we assume an immediately committing commitment scheme.

Proof. By Proposition 6 it suffices to find a simulator for the dummy adversary. By construction of \mathcal{O} the simulator in the ideal experiment can be chosen to be identical to the dummy adversary.

The main idea of the proof is to consider a sequence of hybrid experiments for a PPT environment \mathcal{Z} that may externally invoke polynomially many $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ -sessions and iteratively replace those sessions by the real protocol Π . This is done in a specific order utilizing the fact that the super-polynomial computations of \mathcal{O} are hidden away and thus the replacements are unnoticeable by \mathcal{Z} , or otherwise we would obtain a PPT adversary against the hiding property of $\langle C, R \rangle$.

Step 1: Let \mathcal{Z} be a PPT environment that may externally invoke polynomial many $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ -sessions. We denote the output of this experiment by the random variable $\text{Exec}(\mathcal{F}_{\text{com}}^{\mathcal{O}}, \mathcal{Z})$. Let $\text{Exec}(\Pi, \mathcal{Z})$ be the output of \mathcal{Z} if all instances of $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ sessions are replaced by the instances of the protocol Π . We show that for all environments \mathcal{Z} it holds that

$$\text{Exec}(\mathcal{F}_{\text{com}}^{\mathcal{O}}, \mathcal{Z}) \stackrel{c}{\equiv} \text{Exec}(\Pi, \mathcal{Z}) \quad (9)$$

Let \mathcal{Z} be an environment. By a standard averaging argument we can fix some random coins r for \mathcal{Z} . Thus we can assume henceforth that \mathcal{Z} is deterministic.

We call instances of $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ (or Π) where the sender or receiver is corrupted *sender sessions* or *receiver sessions*, respectively. Since in the cases where both or no party is corrupted, the \mathcal{O} -adjoined functionalities in this case can be treated as part of the environment. We therefore only need to consider $\mathcal{F}^{\mathcal{O}}$ -augmented environments that only invoke either sender sessions or receiver sessions.

We say a *discrepancy* occurred, if in any ideal sender session of $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ the oracle \mathcal{O} extracts a value m , but later \mathcal{Z} correctly unveils a value $m' \neq m$. First notice that unless a discrepancy happens, the output of an ideal sender session is identically distributed to the output of the real protocol Π .

We will now distinguish two cases.

1. The probability that \mathcal{Z} causes a discrepancy is negligible.
2. The probability that \mathcal{Z} causes a discrepancy is non-negligible.

Case 1: We replace all sender sessions with instances of Π , incurring only a negligible statistical distance. We are left with a hybrid experiment in which only the receiver sessions are still ideal. We will now iteratively replace ideal receiver sessions with the real protocol, beginning with the *last* session that is started.

Assume that there are at most q receiver sessions. Define hybrids $\mathbf{H}_0, \dots, \mathbf{H}_q$ as follows. Hybrid \mathbf{H}_i is the experiment where the first i receiver sessions are ideal and the remaining $q - i$ receiver sessions are replaced by instances of Π (in which the receiver is corrupted). Clearly, \mathbf{H}_q is identical to the experiment where all receiver sessions are ideal, whereas \mathbf{H}_0 is the experiment where all receiver sessions are real. The experiment \mathbf{H}_i outputs whatever \mathcal{Z} outputs. Let $P_i = \Pr[\mathbf{H}_i = 1]$ denote the probability that \mathcal{Z} outputs 1 in the hybrid game \mathbf{H}_i . Assume now that $\epsilon := |P_0 - P_q|$ is non-negligible, i. e., \mathcal{Z} has non-negligible

advantage ϵ in distinguishing H_0 from H_q . We will now construct an adversary \mathcal{A}_Π that breaks the hiding property of Π with advantage ϵ/q .

By the averaging principle, there must exist an index $i^* \in [q]$ such that $|P_{i^*-1} - P_{i^*}| \geq \epsilon/q$. By a standard coin-fixing argument, we can now fix the coins that are used in the first $i^* - 1$ receiver sessions up until the point where session i^* starts, while maintaining \mathcal{Z} 's distinguishing advantage. Since we fixed \mathcal{Z} 's coins before, the experiment is now deterministic until the start of session i^* . Since \mathcal{Z} is fully deterministic up until this point, the first message of \mathcal{Z} in session i^* , which is a commitment on a bitstring I , is also computed deterministically.

We can now construct the non-uniform adversary \mathcal{A} against the hiding property of $\langle C, R \rangle$. (We note that we do not construct an adversary \mathcal{A} for the standard hiding game but for a multi-instance variant.) As a non-uniform advice, \mathcal{A} receives a complete trace of all messages sent until this point. This includes all bit strings I_1, \dots, I_{i^*} to which \mathcal{Z} committed to in all receiver sessions $1, \dots, i^*$ (it also includes \mathcal{Z} 's input). Note that all messages come from a deterministic process, and the corresponding I_i are uniquely determined by the first messages of each session i since $\langle C, R \rangle$ is immediately committing.

\mathcal{A} now proceeds as follows. \mathcal{A} internally simulates \mathcal{Z} and all sessions invoked by \mathcal{Z} . This simulation can be done in *polynomial time*, since all sender sessions and the subsequent receiver sessions $i^* + 1$ through q have been replaced by instances of Π , and \mathcal{A} knows the index strings I_i that are used in the (ideal) receiver sessions 1 through i^* .

Let m^* be the message that \mathcal{Z} chooses as input for the sender in session i^* . \mathcal{A} reads $I \stackrel{\text{def}}{=} I_{i^*}$ from its non-uniform advice and samples a tuple s_I of n random strings. It then computes $s_{-I} = s_I \oplus (m^*, \dots, m^*)$ and $s'_{-I} = s_I$ for all $i \in [n]$. \mathcal{A} sends the messages (s_{-I}, s'_{-I}) to the hiding experiment. It now forwards all the messages between the hiding experiment and \mathcal{Z} and simultaneously commits honestly on all values s_I to \mathcal{Z} . When \mathcal{Z} requires that the commitments for all s_I be opened, \mathcal{A} honestly unveils these. When \mathcal{Z} terminates, \mathcal{A} outputs whatever \mathcal{Z} output in the experiment. This concludes the description of \mathcal{A} .

We will now analyze \mathcal{A} 's advantage. If the challenger of the hiding game picks the messages s'_{-I} , \mathcal{Z} obtains a commitment on the all-zero string in \mathcal{A} 's simulation. Therefore, in this case the view of \mathcal{Z} is distributed identically to the view inside the hybrid H_{i^*} . If the challenger of the hiding game picks the messages s_{-I} , \mathcal{Z} obtains a commitment to the message m which is identical to the view of \mathcal{Z} inside the hybrid H_{i^*-1} . It follows

$$\text{Adv}(\mathcal{A}) = |\Pr[H_{i^*} = 1] - \Pr[H_{i^*-1} = 1]| = |P_{i^*} - P_{i^*-1}| \geq \epsilon/q, \quad (10)$$

Hence, \mathcal{A} breaks the hiding property of $\langle C, R \rangle$ with advantage ϵ/q .

Case 2: We now turn to case 2. A first observation is that we only need to consider augmented environments that invoke exactly *one* external session where the sender is corrupted. This is because if a (general) environment \mathcal{Z} causes a discrepancy with non-negligible probability, then there exists a session j^* in which a discrepancy happens *for the first time*. An environment \mathcal{Z}' that invokes

only one session where the sender is corrupted can then simulate \mathcal{Z} , guess j^* and simulate all the other sessions where the sender is corrupted with the real protocol. It holds that \mathcal{Z}' also causes a discrepancy with non-negligible probability.

So we henceforth assume that \mathcal{Z} invokes at most q sessions and only one session where the sender is corrupted. In what follows, we will replace all ideal sessions where the receiver is corrupted with real protocols using the same strategy as in case 1. Define the hybrids H_0, \dots, H_q as in case 1 except that now \mathcal{Z} can additionally invoke exactly one sender session in all these hybrids. Clearly, H_q is identical to the experiment where all sessions are ideal, whereas H_0 is the experiment where all receiver sessions are real. Let $P_i = \Pr[H_i = 1]$ again.

Assume that \mathcal{Z} can distinguish between H_0 and H_q with non-negligible advantage ϵ . Then there exists an index i^* such that $|P_{i^*-1} - P_{i^*}| \geq \epsilon/q$. We can now fix the coins that are used in the first $i^* - 1$ receiver sessions and the one sender session up until the point where session i^* starts, while maintaining \mathcal{Z} 's distinguishing advantage.

We will construct a non-uniform adversary \mathcal{A}' that breaks the parallel-CCA-security of $\langle C, R \rangle$ with advantage ϵ/q . As in case 1, \mathcal{A}' receives as a non-uniform advice a trace of a run of \mathcal{Z} which also includes all index sets I_i to which \mathcal{Z} committed in all sessions until session i^* and possibly the shares to which \mathcal{Z} committed in the only sender session (again, it also includes \mathcal{Z} 's input).

\mathcal{A}' now proceeds the same way as in case 1. It internally runs \mathcal{Z} and simulates either hybrid H_{i^*-1} or H_{i^*} for \mathcal{Z} by embedding the challenge of the hiding game into the simulated session i^* . The adversary \mathcal{A}' simulates all ideal receiver sessions for $i \leq i^*$ with the help of its advice while all subsequent *receiver* sessions for $i > i^*$ have already been replaced by Π . If \mathcal{Z} has already started to commit to the shares in the only sender session then (by definition) these shares are also part of \mathcal{A}' 's advice and \mathcal{A}' can simulate the sender session. (Note that $\langle C, R \rangle$ is immediately committing, hence the first message of (the parallel executions of) $\langle C, R \rangle$ uniquely determines the shares). If \mathcal{Z} has not yet started to commit to the shares in the sender session then \mathcal{A}' can use its parallel-CCA oracle to extract them by forwarding the corresponding messages between the oracle and \mathcal{Z} . After the experiment terminates, \mathcal{A}' outputs whatever \mathcal{Z} outputs.

The analysis of \mathcal{A}' is the same as in case 1 and we end up with the conclusion that \mathcal{A}' breaks the parallel-CCA security of protocol $\langle C, R \rangle$ with advantage ϵ/q .

Hence, it remains to consider environments that invoke exactly one sender session. Assume that such an environment \mathcal{Z} causes a discrepancy with non-negligible probability.

We will now construct a non-uniform adversary \mathcal{A}'' that breaks the hiding property of $\langle C, R \rangle$. \mathcal{A}'' takes part in a partial one-way hiding experiment where the challenger picks a random (choice) string $I = b_1 \dots b_n$ and commits to I using $\langle C, R \rangle$ and a tag selected by the adversary. \mathcal{A}'' then sends a vector (a_1, \dots, a_n) to the experiment where $a_l \in \{0, 1, \perp\}$. Let $M = \{l \mid a_l \neq \perp\}$. \mathcal{A}'' wins if $\text{card}(M) \geq n/2$ and $a_l = b_l$ for all $l \in M$. Throughout the experiment, \mathcal{A}'' may query the parallel-CCA oracle of $\langle C, R \rangle$. It holds that since $\langle C, R \rangle$ is parallel-CCA-secure, \mathcal{A}'' can win in this experiment only with negligible probability.

\mathcal{A}'' receives as non-uniform advice the input of \mathcal{Z} . \mathcal{A}'' now proceeds as follows: \mathcal{A}'' forwards the commitment it receives in the experiment to \mathcal{Z} as in the commit phase of the one sender session that \mathcal{Z} can invoke. When \mathcal{Z} sends the commitments on the shares $s_{l,b}$, \mathcal{A}'' forwards them to its parallel-CCA oracle, thus learning the values $s_{l,b}$ that \mathcal{Z} committed to. \mathcal{A} can now simulate the oracle \mathcal{O} and reconstruct the message m defined by these shares (by defining m to be the most frequent value that occurs in $\{s_{i,0} \oplus s_{i,1}\}_{i \in [n]}$ just like \mathcal{O}). When \mathcal{Z} sends the shares $s'_{l,b}$ in the unveil phase of the sender session, \mathcal{A}'' compares them to the originally extracted shares $s_{l,b}$ and defines the vector (a_1, \dots, a_n) as

$$a_l := \begin{cases} b_l & \text{if } \exists b_l \in \{0, 1\} : s_{l,b_l} = s'_{l,b_l} \wedge s_{l,-b_l} \neq s'_{l,-b_l} \\ \perp & \text{else (if no such } b_l \text{ exists)} \end{cases} \quad (\star) \quad (11)$$

and sends (a_1, \dots, a_n) to the experiment.

We will now analyze \mathcal{A}'' 's success probability. Let M be the set of indices l for that condition (\star) holds. If \mathcal{Z} causes a discrepancy, it holds that all tuples of shares $(s'_{l,0}, s'_{l,1})$ define the same message m' that is different from m , the most frequently occurring value of $\{m_l = s_{l,0} \oplus s_{l,1}\}_{i \in [n]}$, i. e. $\text{card}(M) \geq n/2$. Moreover, for each $l \in M$, b_l equals the l th bit of I . Hence, by construction, \mathcal{A}'' wins with non-negligible probability if \mathcal{Z} causes a discrepancy with non-negligible probability.

Step 2: We will now prove that for every $\mathcal{F}^\mathcal{O}$ -augmented environment

$$\text{Exec}(\Pi, \mathcal{D}, \mathcal{Z}[\mathcal{F}_{\text{com}}^\mathcal{O}]) \stackrel{c}{=} \text{Exec}(\mathcal{F}_{\text{com}}^\mathcal{O}, \mathcal{D}, \mathcal{Z}[\mathcal{F}_{\text{com}}^\mathcal{O}]) .$$

If the *sender is corrupted* then nothing needs to be shown, as in this case the real and ideal experiment are statistically close. This follows from the fact that by step 1, case 2, an $\mathcal{F}_{\text{com}}^\mathcal{O}$ -augmented environment can cause a discrepancy only with negligible probability.

If the *receiver is corrupted* then by step 1 the real and ideal experiment are both indistinguishable to an experiment where all ideal sender sessions and ideal receiver sessions been replaced by instances of the real protocol. Hence the outputs of the real and ideal experiment are indistinguishable.

If *no party is corrupted* then one can first replace all sender sessions and receiver sessions with the real protocol using step 1, obtaining a polynomial time environment. Then one can prove indistinguishability by using a very similar reduction to the hiding property as in step 1, case 1.

If *both parties are corrupted* then the real and ideal experiment are identically distributed. \square

The premise of Theorem 21 can be further relaxed by using only a *weakly* pCCA oracle instead of a standard pCCA oracle. A weakly pCCA oracle returns \perp everywhere in case that at least one commitment is not accepting. Weakly pCCA suffices because a shielded oracle in a sender session (acting as the honest receiver) aborts if at least one commitment is not accepting in the commit phase.

The underlying commitment scheme $\langle C, R \rangle$ can be instantiated with the 8-round construction in [Goy+14]. It is straightforward to see that this scheme is pCCA secure by using the extractor in its security proof. The Zero-Knowledge Argument of Knowledge inside [Goy+14] is instantiated with the Feige-Shamir protocol [FS90] and—deviating from the original work—the basic commitment scheme is instantiated by the Blum commitment [Blu81] because we require an immediately committing protocol. Since this scheme is constant-round, we obtain the following result:

Corollary 22. *Assume the existence of one-way permutations. Then there is a constant-round protocol Π_{com} and a shielded oracle \mathcal{O} such that $\Pi_{com} \geq_{\mathcal{F}_{com}^{\mathcal{O}}} \mathcal{F}_{com}^{\mathcal{O}}$.*

The above construction is non-black-box since [Goy+14] (instantiated this way) is non-black-box. However, recall that the only non-black-box part of [Goy+14] is a ZK proof for proving knowledge of committed values and that these values satisfy linear relations. As already pointed out in [Goy+14], this can both be done making only black-box use of a homomorphic commitment scheme. Instantiating [Goy+14] with a perfectly binding homomorphic commitment scheme thus yields a fully black-box construction. Since we need an immediately committing scheme in the plain model for our protocol we let the sender (and not a trusted setup) generate the commitment key of the homomorphic commitment. This construction can be used as a building block in [Goy+14] if the homomorphic commitment scheme is “verifiable”. A verifiable homomorphic commitment scheme allows one to (non-interactively) verify that a commitment key is well-formed. For instance, the ElGamal commitment scheme [ELG84] (which is based on the DDH assumption) is a verifiable perfectly binding homomorphic commitment scheme [AIR01]. The Linear Encryption scheme [BBS04] (which is based on the DLin assumption) can also be viewed as a commitment scheme with these properties.

Corollary 23. *Assume the existence of verifiable perfectly binding homomorphic commitment schemes. Then there exists a constant-round black-box protocol Π_{com}^{BB} and a shielded oracle \mathcal{O} such that $\Pi_{com}^{BB} \geq_{\mathcal{F}_{com}^{\mathcal{O}}} \mathcal{F}_{com}^{\mathcal{O}}$.*

5 A Modular Composition Theorem for Π

We show that we can plug the protocol Π from Construction 1 into a large class of UC-secure protocols in the \mathcal{F}_{com} -hybrid model in such a way that the composite protocol is secure in our framework. We first define Commit-Compute protocols and parallel-CCA-UC-emulation.

Definition 24 (Commit-Compute protocols). *Let $\rho^{\mathcal{F}_{com}}$ be a protocol in the \mathcal{F}_{com} -hybrid model. We call $\rho^{\mathcal{F}_{com}}$ a commit-compute protocol or CC protocol if it can be broken down into two phases: An initial commit phase, where the only communication allowed is sending messages to instances of \mathcal{F}_{com} . After the commit phase is over,⁷ a compute phase begins where sending messages*

⁷ We implicitly assume that each party notifies all the other parties after it has finished sending all its commitments. A protocol party will not start the compute phase before it has received a notification by all the other parties.

to instances of \mathcal{F}_{com} except for **unveil-messages** is prohibited, but all other communication is allowed.⁸

Definition 25 (pCCA-UC-emulation). We write $\rho \geq_{\mathcal{E}\text{-pCCA}} \phi$ if a protocol ρ UC-emulates a protocol ϕ in the presence of (non-uniform) environments that may interact with a pCCA-decommitment oracle \mathcal{E} as defined in Definition 20 for tags that are not extensions of the session identifier of the challenge protocol.

In the following, let Π be the protocol as in Construction 1 with an immediately committing and parallel-CCA-secure commitment scheme $\langle C, R \rangle$. Let \mathcal{E} be the (uniquely defined) pCCA-decommitment oracle of $\langle C, R \rangle$.

We are now ready to state the theorem:

Theorem 26. Let $\rho^{\mathcal{F}_{\text{com}}}$ be a CC protocol and \mathcal{G} a functionality. If $\rho^{\mathcal{F}_{\text{com}}} \geq_{\mathcal{E}\text{-pCCA}} \mathcal{G}$ then there exists a shielded oracle \mathcal{O}' such that

$$\rho^{\Pi} \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$$

Proof. Since $\rho^{\mathcal{F}_{\text{com}}} \geq_{\mathcal{E}\text{-pCCA}} \mathcal{G}$ there exists a dummy adversary simulator $\mathcal{S}_{\mathcal{D}}$. Let \mathcal{O} be the shielded oracle from Construction 2, s. t. $\Pi \geq_{\mathcal{F}_{\text{com}}^{\mathcal{O}}} \mathcal{F}_{\text{com}}^{\mathcal{O}}$. We define the shielded oracle \mathcal{O}' as follows. (For a graphical depiction see Fig. 7, p. 46). \mathcal{O}' internally simulates multiple instances of \mathcal{O} (one for each instance of \mathcal{F}_{com} in ρ) and $\mathcal{S}_{\mathcal{D}}$, and forwards messages as follows.

- Messages from the adversary addressed to an instance of \mathcal{F}_{com} are forwarded to the corresponding internal instance of \mathcal{O} .
- Messages from an internal instance of \mathcal{O} to an instance of \mathcal{F}_{com} are forwarded to the dummy adversary simulator $\mathcal{S}_{\mathcal{D}}$.
- Messages between $\mathcal{S}_{\mathcal{D}}$ and the functionality \mathcal{G} are forwarded.
- Messages from the dummy adversary simulator $\mathcal{S}_{\mathcal{D}}$ addressed as coming from an instance of \mathcal{F}_{com} are forwarded to the respective instance of \mathcal{O} .
- Messages from the dummy adversary simulator $\mathcal{S}_{\mathcal{D}}$ not addressed as coming from an instance of \mathcal{F}_{com} are output to the adversary (without forwarding them to an internal instance of \mathcal{O}).

We claim that for this oracle $\rho^{\Pi} \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$ holds. By Proposition 6 it is sufficient to find a simulator for the dummy adversary. The simulator will be the dummy adversary in the ideal world.

Recall that we call instances of $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ (or Π) where the sender or receiver is corrupted *sender sessions* or *receiver sessions*, respectively.

We denote by $\rho^{\Pi_{\text{S}}, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$ the protocol $\rho^{\mathcal{F}_{\text{com}}^{\mathcal{O}}}$ where all ideal sender sessions have been replaced by the real protocol. Let $\text{Exec}(\rho^{\Pi_{\text{S}}, \mathcal{F}_{\text{com}}^{\mathcal{O}}}, \mathcal{Z})$ denote an execution of

⁸ For technical reasons, a protocol party may not carry out local computations that *randomly* determine if the party sends a commit message to \mathcal{F}_{com} or not. Otherwise, a party would be able to transmit arbitrary (random) messages during the commit phase. For instance, a party could locally compute a bitstring $b = (b_1, \dots, b_m)$ and at the i -th activation only commit if the bit b_i equals 1.

an environment \mathcal{Z} with (polynomially many) instances of $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$. Furthermore, denote by $\text{Exec}(\mathcal{G}^{\mathcal{O}'}, \mathcal{Z})$ an execution of an environment \mathcal{Z} where all instances of $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$ have been replaced by instances of $\mathcal{G}^{\mathcal{O}'}$.

Let \mathcal{Z} be an environment. By a standard averaging argument we can fix some random coins r for \mathcal{Z} . Thus we can assume henceforth that \mathcal{Z} is deterministic.

In the following hybrid argument, we will have to globally order the main sessions by the *ending* of their commit-phase and (adaptively) invoke instances of $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$, $\rho^{\mathcal{F}_{\text{com}}^{\mathcal{O}}}$ or $\mathcal{G}^{\mathcal{O}'}$ based on this order. Since the message scheduling may be random, however, this order is not determined a-priori.

In the following, we will therefore have the experiment in the hybrids implement the commit-phases of all invoked protocols “obliviously”, i. e., interact with the environment by running the programs of the shielded oracles and store the inputs of the honest parties without following their instructions in the commit-phases. Note that the only communication that is *visible* to the environment in the commit-phase is its interaction with the shielded oracles or the receiver in an instance of H_S . The latter interaction is identical to an interaction with the shielded oracle in a sender session. The inputs of the honest parties have no effect on the messages the shielded oracles output to the environment in a commit phase.

Once the commit phases of an instance of $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$ has ended, the experiment in the hybrids will invoke an instance of $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$, $\rho^{\mathcal{F}_{\text{com}}^{\mathcal{O}}}$ or $\mathcal{G}^{\mathcal{O}'}$ depending on the position within the global order of sessions. The experiment will then invoke the honest parties with their respective inputs and follow their instructions (it will also invoke the simulator $\mathcal{S}_{\mathcal{D}}$ with the extracted values if this session is $\mathcal{G}^{\mathcal{O}'}$). Messages from $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ or $\mathcal{S}_{\mathcal{D}}$ to instances of \mathcal{O} (which are “ok” messages) are suppressed. This way, the emulation is consistent with the messages in the commit phase and distributed identically as if one of the protocols $\mathcal{G}^{\mathcal{O}'}$, $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$, or $\rho^{\mathcal{F}_{\text{com}}^{\mathcal{O}}}$ was executed from the beginning.

Step 1. We show that

$$\text{Exec}(\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}, \mathcal{Z}) \stackrel{c}{\equiv} \text{Exec}(\mathcal{G}^{\mathcal{O}'}, \mathcal{Z}) \quad (12)$$

Let $q(n)$ be an upper bound on the number of instances of $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$ that \mathcal{Z} invokes. Consider the $2q(n) + 1$ hybrids $H_{00}, H_{01}, H_{10}, H_{11}, H_{20}, \dots, H_{q(n)0}$ which are constructed as follows:

Definition of hybrid H_{ij} : Execute the commit phases of each session “without running the code of the parties” by invoking instances of \mathcal{O} . Follow the instruction of each instance of \mathcal{O} . Parties are only there as placeholders for the environment in the commit phases. Their instructions will be execute after the commit phase of the respective session is over. Note that this can be done since the actions of the parties (i.e. choice of input values) in the commit phases have no effect on the view of the environment in these phases. Messages output from an instance of \mathcal{O} are stored as well. After the commit phase of a session is over do the following:

(See Fig. 8, p. 47 for an illustration of the sequence of the hybrid games.)

1. If this is the k th session in which the commit phase has ended and $k \leq i$ then invoke an instance of the dummy adversary simulator and the functionality \mathcal{G} . Hand the dummy parties their respective inputs and the dummy adversary simulator the messages output by the instances of \mathcal{O} . Follow the instructions of the dummy adversary simulator and \mathcal{G} . Ignore messages of the dummy adversary simulator to the environment if these messages are coming from an instance of \mathcal{F}_{com} in the commit phase (i.e. an “ok” message). In the unveil phase, messages from the dummy adversary simulator mimicking an interaction with \mathcal{F}_{com} (which are messages of the form (unveil, b)) are forwarded to the respective instance of \mathcal{O} (with the same SID). Messages from the dummy adversary simulator not mimicking an interaction with an instance of \mathcal{F}_{com} are output (without forwarding them to an \mathcal{O} -instance).
2. If $k = i + 1$ and $j = 0$ or $k > i + 1$ then run the protocol parties of $\rho^{\mathcal{F}_{\text{com}}}$ with their inputs and follow their instructions. For all subsessions where the sender is corrupted invoke instances of $\Pi_{\mathcal{S}}$ and “replay” the commit phase of $\Pi_{\mathcal{S}}$ using the same randomness for the receiver as the respective oracle and the messages the environment has sent. For all subsessions where the receiver or both or no party has been corrupted invoke instances of \mathcal{F}_{com} and adjoin the respective oracle. Ignore “ok” messages from the instances of \mathcal{F}_{com} .
3. If $k = i + 1$ and $j = 1$ then run the parties of $\rho^{\mathcal{F}_{\text{com}}}$ with their inputs in the commit phase and follow their instructions. For all subsessions invoke an instance of \mathcal{F}_{com} and adjoin the respective oracle. Send the extracted committed values of the \mathcal{O} -instances in sender sessions to the respective \mathcal{F}_{com} -instance. Ignore “ok” messages from the instances of \mathcal{F}_{com} .

Observe that $H_{00} = \text{Exec}(\rho^{\Pi_{\mathcal{S}}, \mathcal{F}_{\text{com}}^{\mathcal{O}}}, \mathcal{Z})$ and $H_{q(n)0} = \text{Exec}(\mathcal{G}^{\mathcal{O}'}, \mathcal{Z})$.

Let P_{ij} denote the probability that \mathcal{Z} outputs 1 in hybrid H_{ij} . Assume $|P_{00} - P_{q(n)0}|$ is non-negligible. Then there exists an index i^* such that either $|P_{i^*1} - P_{(i^*+1)0}|$ or $|P_{i^*0} - P_{i^*1}|$ is also non-negligible.

Case 1: $|P_{i^*1} - P_{(i^*+1)0}|$ is non-negligible. In this case, these neighboring hybrids are equal except that in the $(i^* + 1)$ th session $\rho^{\mathcal{F}_{\text{com}}^{\mathcal{O}'}}$ is replaced by $\mathcal{G}^{\mathcal{O}'}$.

We can now fix the coins that are used in the experiment up until the point where the $(i^* + 1)$ th commit phase has ended, while maintaining \mathcal{Z} 's distinguishing advantage.

We can now construct an environment \mathcal{Z}' that distinguishes $\rho^{\mathcal{F}_{\text{com}}}$ from \mathcal{G} . As a non-uniform advice, \mathcal{Z}' receives a complete trace of all messages sent until this point, including all shares s_i and strings I that \mathcal{Z} committed to until the point where the $(i^* + 1)$ th commit phase has ended (note that since $\langle C, R \rangle$ is immediately committing, the first message in a receiver session uniquely determines the string I . Also note that the interaction with all \mathcal{O} -instances for the case of a corrupted sender mimicking a commit phase of Π is over until this point.) \mathcal{Z}' internally simulates the execution experiment with \mathcal{Z} using its advice. Messages to the $(i^* + 1)$ th session are sent to the challenge protocol. \mathcal{Z}' may (tentatively) also invoke ideal receiver sessions in order to simulate ideal receiver sessions that are invoked after the point where the $(i^* + 1)$ th commit phase has ended.

Observe that the real execution corresponds to hybrid H_{i^*1} and the ideal execution to hybrid $H_{(i^*+1)0}$. By construction, \mathcal{Z}' distinguishes $\rho^{\mathcal{F}_{\text{com}}}$ from \mathcal{G} . Since \mathcal{O} adjoined to \mathcal{F}_{com} is polynomially simulatable, \mathcal{Z}' can be replaced by a polynomial time environment that also distinguishes $\rho^{\mathcal{F}_{\text{com}}}$ from \mathcal{G} , using Theorem 12. This is a contradiction (to the def. of the dummy adversary simulator).

Case 2: $|P_{i^*0} - P_{i^*1}|$ is non-negligible. In this case, these neighboring hybrids are equal except that in the $(i^* + 1)$ th session $\rho^{H_S, \mathcal{F}_{\text{com}}^{\mathcal{O}}}$ is replaced by $\rho^{\mathcal{F}_{\text{com}}^{\mathcal{O}}}$. Since \mathcal{Z} distinguishes these hybrids it holds that with non-negligible probability \mathcal{Z} causes a *discrepancy* in hybrid H_{i^*1} as otherwise these hybrids would be statistically close. Let $\tilde{\mathcal{Z}}$ be the environment that internally runs \mathcal{Z} and outputs 1 as soon as a discrepancy occurs.⁹ By construction, $\tilde{\mathcal{Z}}$ outputs 1 with non-negligible probability in H_{i^*1} . We will now consider $i^* + 1$ new hybrids h_0, \dots, h_{i^*} .

Definition of hybrid h_j : (See Fig. 9, p. 48 for a graphical explanation of the hybrids h_j .) Execute the commit phases of each session “without running the code of the parties” as described in the description of the hybrids H_{ij} . After the commit phase of a session is over do the following (for a fixed $j \in \{0, \dots, i^*\}$):

1. If this is the k th session in which the commit phase has ended and $k \leq i^* - j$ then invoke an instance of the dummy adversary simulator and the functionality \mathcal{G} . (Marked as range (I) in Fig. 9.) Hand the dummy parties their respective inputs and the dummy adversary simulator the messages output by the instances of \mathcal{O} . Follow the instructions of the dummy adversary simulator and \mathcal{G} . Ignore messages of the dummy adversary simulator to the environment if these messages are coming from an instance of \mathcal{F}_{com} in the commit phase (i. e. an “ok” message). In the unveil phase, messages from the dummy adversary simulator mimicking an interaction with \mathcal{F}_{com} (which are messages of the form (unveil, b)) are forwarded to the respective instance of \mathcal{O} . Messages from the dummy adversary simulator not mimicking an interaction with an instance of \mathcal{F}_{com} are output (without forwarding them to an \mathcal{O} -instance).
2. If $i^* - j + 1 \leq k \leq i^* + 1$ then run the protocol parties of $\rho^{\mathcal{F}_{\text{com}}}$ with their inputs in the commit phase and follow their instructions. (Marked as range

⁹ To make the environment able to learn the committed value in a $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ -hybrid protocol, we redefine the shielded oracle \mathcal{O} for the case of a corrupted sender as follows: After the unveil phase is over, the oracle first outputs the extracted committed value to the simulator and after receiving a notification message from the simulator it sends an unveil message to the functionality. Denote this modified oracle by $\tilde{\mathcal{O}}$. Furthermore define $\tilde{\Pi}$ to be identical to Π , except that before outputting the committed value, the receiver sends the committed value to the sender. The sender then sends a notification message to the receiver, who then outputs the committed value. It follows from the exact same arguments as in the proof of Theorem 21 that $\tilde{\Pi} \geq_{\mathcal{F}_{\text{com}}^{\tilde{\mathcal{O}}}} \mathcal{F}_{\text{com}}^{\tilde{\mathcal{O}}}$ and that $\tilde{\mathcal{O}}$ adjoined to \mathcal{F}_{com} is polynomially simulatable. Using these modified versions in the above proof one obtains $\rho^{\tilde{\Pi}} \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$. Since Π unconditionally emulates $\tilde{\Pi}$ it holds that $\rho^{\Pi} \geq_{\mathcal{G}^{\mathcal{O}'}} \rho^{\tilde{\Pi}}$, hence $\rho^{\Pi} \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$.

- (II) in Fig. 9.) For all subsessions invoke an instance of \mathcal{F}_{com} and adjoin the respective oracle. Send the extracted committed values of the \mathcal{O} -instances in sender sessions to the respective \mathcal{F}_{com} -instance. Ignore “ok” messages from the instances of \mathcal{F}_{com} .
3. If $k \geq i^* + 2$ then run the protocol parties of $\rho^{\mathcal{F}_{\text{com}}}$ with their inputs and follow their instructions. For all subsessions where the sender is corrupted invoke instances of Π_S and “replay” the commit phase of Π_S using the same randomness for the receiver as the respective oracle and the messages the environment has sent. For all subsessions where the receiver or both or no party has been corrupted invoke instances of \mathcal{F}_{com} and adjoin the respective oracle. Ignore “ok” messages from the instances of \mathcal{F}_{com} .

Observe that $\mathbf{h}_0 = \mathbf{H}_{i^*1}$. Let j^* be the *largest index* such that $\tilde{\mathcal{Z}}$ causes a discrepancy in hybrid \mathbf{h}_{j^*} with non-negligible probability. j^* is well-defined, since there is an index for which this property holds (namely 0). Furthermore, $j^* \leq i^* - 1$. This follows from the following argument. Observe that the last hybrid \mathbf{h}_{i^*} only contains instances of $\rho^{\mathcal{F}_{\text{com}}}$ (since all instance of \mathcal{G} have been replaced). Because Π emulates $\mathcal{F}_{\text{com}}^{\mathcal{O}}$ and due to the composition theorem, $\text{Exec}(\rho^{\Pi}, \mathcal{Z})$ is indistinguishable from \mathbf{h}_{i^*} . Since no discrepancy occurs in $\text{Exec}(\rho^{\Pi}, \mathcal{Z})$ it follows that a discrepancy can occur in \mathbf{h}_{i^*} only with negligible probability.

By construction, $\tilde{\mathcal{Z}}$ distinguishes the hybrids \mathbf{h}_{j^*} and \mathbf{h}_{j^*+1} (in the first hybrid $\tilde{\mathcal{Z}}$ outputs 1 with non-negligible probability and in the second hybrid only with negligible probability).

We will now modify these hybrids. For $k \in \{j^*, j^* + 1\}$ define the hybrid hyb_{k-j^*} to be identical to \mathbf{h}_k except for the following: At the beginning, the experiment randomly selects one sender session. In all commit phases that end *after* the $(i^* - j^*)$ th commit phase the real protocol Π_S is invoked instead of $\mathcal{F}^{\mathcal{O}_s}$ in all sender sessions that have not been selected at the beginning. The one sender session that has been selected at the beginning remains ideal if and only if it is part of one of the commit phases $1, \dots, i^* + 1$.

It holds that $\tilde{\mathcal{Z}}$ also distinguishes hyb_0 from hyb_1 . This is because $\tilde{\mathcal{Z}}$ still causes a discrepancy in hyb_0 with non-negligible probability because with high probability ($1/\text{poly}$) the first session in which $\tilde{\mathcal{Z}}$ causes a discrepancy is selected. Furthermore, $\tilde{\mathcal{Z}}$ causes a discrepancy in hyb_1 with only negligible probability (otherwise $\tilde{\mathcal{Z}}$ would have already caused a discrepancy with non-negligible probability in \mathbf{h}_{j^*+1}).

We can now fix the coins that are used in the experiment up until the point where the $(i^* - j^*)$ th commit phase has ended, while maintaining $\tilde{\mathcal{Z}}$'s distinguishing advantage.

We can now construct an environment \mathcal{Z}'' that distinguishes $\rho^{\mathcal{F}_{\text{com}}}$ from \mathcal{G} . As a non-uniform advice, \mathcal{Z}'' receives a complete trace of all messages sent until this point, including all shares s_i and index sets I that $\tilde{\mathcal{Z}}$ committed to until the point where the $(i^* - j^*)$ th commit phase has ended. \mathcal{Z}'' proceeds as follows: \mathcal{Z}'' internally simulates the execution experiment with $\tilde{\mathcal{Z}}$ using its advice. Messages to the $(i^* - j^*)$ th session are sent to the challenge protocol. \mathcal{Z}'' can simulate the only instance of $\mathcal{F}^{\mathcal{O}_s}$ that may occur in a commit phase with its pCCA-oracle \mathcal{E} (note that if $\tilde{\mathcal{Z}}$ has already started to commit to shares in this session then

these shares are also part of \mathcal{Z}'' 's advice.) \mathcal{Z}'' may (tentatively) also invoke ideal receiver sessions in order to simulate ideal receiver sessions that are invoked after the point where the $(i^* - j^*)$ th commit phase has ended.

Observe that the real execution corresponds to hybrid hyb_1 and the ideal execution to hybrid hyb_0 . By construction, \mathcal{Z}'' distinguishes $\rho^{\mathcal{F}_{\text{com}}}$ from \mathcal{G} . With the same argument as in the proof of Theorem 21, step 1, case 2, one can replace all ideal receiver sessions that \mathcal{Z}'' invokes with instances of the real protocol (using the fact that $\langle C, R \rangle$ is parallel-CCA-secure). By construction, an environment \mathcal{Z}'' was found that can query the pCCA-oracle \mathcal{E} and distinguish $\rho^{\mathcal{F}_{\text{com}}}$ and \mathcal{D} from \mathcal{G} and $\mathcal{S}_{\mathcal{D}}$. We have thus reached a contradiction.

Step 2. We show that $\rho^{\Pi} \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$, completing the proof.

Let \mathcal{Z} be a $\mathcal{G}^{\mathcal{O}'}$ -augmented environments. By step 1, we can replace all instances of $\mathcal{G}^{\mathcal{O}'}$ with instances of $\rho^{\Pi_{\mathcal{S}}, \mathcal{F}_{\text{com}}^{\mathcal{O}'}}$. Since Π emulates $\mathcal{F}_{\text{com}}^{\mathcal{O}'}$, it follows from the composition theorem that we can replace (the challenge protocol) ρ^{Π} also with $\rho^{\Pi_{\mathcal{S}}, \mathcal{F}_{\text{com}}^{\mathcal{O}'}}$. Again by step 1, we can replace all instances of $\rho^{\Pi_{\mathcal{S}}, \mathcal{F}_{\text{com}}^{\mathcal{O}'}}$ back with instances of $\mathcal{G}^{\mathcal{O}'}$. The theorem follows. \square

If the following property holds for the commitment scheme $\langle C, R \rangle$, the premise $\rho^{\mathcal{F}_{\text{com}}} \geq_{\mathcal{E}\text{-pCCA}} \mathcal{G}$ is automatically fulfilled.

Definition 27 (*r*-non-adaptive robustness). *Let $\langle C, R \rangle$ be a tag-based commitment scheme and \mathcal{E} a pCCA-decommitment oracle for it as in Definition 20. For $r \in \mathbb{N}$, we say that $\langle C, R \rangle$ is *r*-non-adaptively-robust w. r. t. \mathcal{E} if for every PPT adversary \mathcal{A} , there exists a PPT simulator \mathcal{S} , such that for every PPT *r*-round interactive Turing machine \mathcal{B} , the following two ensembles are computationally indistinguishable:*

$$\begin{aligned} & - \{ \langle \mathcal{B}(y), \mathcal{A}^{\mathcal{E}}(z) \rangle(1^n) \}_{n \in \mathbb{N}, y \in \{0,1\}^*, z \in \{0,1\}^*} \\ & - \{ \langle \mathcal{B}(y), \mathcal{S}(z) \rangle(1^n) \}_{n \in \mathbb{N}, y \in \{0,1\}^*, z \in \{0,1\}^*} \end{aligned}$$

The above definition is a weakening of the (adaptive) robustness property put forward by [CLP10].

Corollary 28. *Let $\rho^{\mathcal{F}_{\text{com}}}$ be a constant-round protocol and \mathcal{G} a functionality such that $\rho^{\mathcal{F}_{\text{com}}} \geq_{\text{UC}} \mathcal{G}$. If the commitment scheme $\langle C, R \rangle$ in Π is also *r*-non-adaptively-robust for sufficiently large *r*, then there exists a shielded oracle \mathcal{O}' such that*

$$\rho^{\Pi} \geq_{\mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$$

Up to now we could instantiate $\langle C, R \rangle$ with a modified version of [Goy+14] as described above of Corollary 22. To additionally make this scheme *r*-non-adaptively-robust w. r. t. \mathcal{E} one can add “redundant slots” using the idea of [LP09] (the scheme needs to have at least $r + 1$ slots to be *r*-non-adaptively-robust).

In the following lemma we show that every UC-secure protocol $\rho^{\mathcal{F}_{\text{com}}}$ can be transformed into a UC-secure CC protocol.

Lemma 29 (CC compiler). *Let $\rho^{\mathcal{F}_{\text{com}}}$ be a protocol in the \mathcal{F}_{com} -hybrid model. Then there exists a CC protocol $\text{Comp}(\rho)^{\mathcal{F}_{\text{com}}}$ such that $\text{Comp}(\rho)^{\mathcal{F}_{\text{com}}} \geq_{UC} \rho^{\mathcal{F}_{\text{com}}}$. Furthermore, if $\rho^{\mathcal{F}_{\text{com}}}$ is constant-round then so is $\text{Comp}(\rho)^{\mathcal{F}_{\text{com}}}$.*

Proof (Idea of proof). Replace each instance of \mathcal{F}_{com} with a *randomized commitment* where the sender commits to a bit b by sending a random value a to \mathcal{F}_{com} and $a \oplus b$ to the receiver. Since the number of rounds of $\rho^{\mathcal{F}_{\text{com}}}$ is polynomially bounded, the number of commitments of each party is also polynomially bounded. Put all randomized calls to \mathcal{F}_{com} in a single commit phase. \square

Let Π_r be the constant-round protocol as in Construction 1 where $\langle C, R \rangle$ is instantiated with the immediately committing, parallel-CCA secure and r -non-adaptively-robust modified version of [Goy+14] as described above. Furthermore, let Π_r^{BB} be the same as Π_r , except that [Goy+14] is instantiated with a verifiable perfectly binding homomorphic commitment scheme, thus making the construction fully black-box. Applying Corollary 28 and Lemma 29 one obtains the following:

Corollary 30. *Assume the existence of one-way permutations. Let $\rho^{\mathcal{F}_{\text{com}}}$ be a constant-round protocol and \mathcal{G} a functionality. If $\rho^{\mathcal{F}_{\text{com}}} \geq_{UC} \mathcal{G}$ then there exists a shielded oracle \mathcal{O}' such that for sufficiently large r it holds that*

$$\text{Comp}(\rho)^{\Pi_r} \underset{\mathcal{G}^{\mathcal{O}'}}{\geq} \mathcal{G}^{\mathcal{O}'}$$

Furthermore, assuming the existence of verifiable perfectly binding homomorphic commitment schemes, the same property holds for Π_r^{BB} .

6 Constant-Round (Black-Box) General MPC

We can now apply Corollary 30 to obtain a constant-round general MPC protocol based on standard polynomial-time hardness assumptions that is secure in our framework. By [Can+02; IPS08] there exists a constant-round general MPC protocol in the \mathcal{F}_{com} -hybrid model based on enhanced trapdoor permutations. Plugging Π_r (for a sufficiently large r) into this protocol yields a constant-round general MPC protocol in the plain model (based on ETDPs). Furthermore, [HV15] constructed a constant-round black-box general MPC protocol in the \mathcal{F}_{CRS} -hybrid model based on PKE and constant-round semi-honest oblivious transfer. This protocol can be transformed into a protocol in the \mathcal{F}_{com} -hybrid model assuming PKE with oblivious public-key generation. Plugging Π_r^{BB} into this modified version of [HV15] yields a fully black-box construction of a constant-round general MPC protocol in the plain model based on polynomial-time hardness assumptions.

Theorem 31 (Constant-round general MPC in the plain model).

(a) *Assume the existence of enhanced trapdoor permutations. Then for every well-formed functionality \mathcal{F} , there exists a constant-round protocol $\pi_{\mathcal{F}}$ (in the plain model) and a shielded oracle \mathcal{O} such that*

$$\pi_{\mathcal{F}} \underset{\mathcal{F}^{\mathcal{O}}}{\geq} \mathcal{F}^{\mathcal{O}} \tag{13}$$

(b) Assume the existence of verifiable perfectly binding homomorphic commitment schemes and PKE with oblivious public-key generation.

Then for every well-formed functionality \mathcal{F} , there exists a constant-round protocol $\pi_{\mathcal{F}}^{BB}$ (in the plain model) and a shielded oracle \mathcal{O} such that

$$\pi_{\mathcal{F}}^{BB} \underset{\mathcal{F}^{\mathcal{O}}}{\geq} \mathcal{F}^{\mathcal{O}} \quad (14)$$

$\pi_{\mathcal{F}}^{BB}$ uses the underlying commitment scheme and public-key encryption scheme only in a black-box way.

7 Conclusion

Shielded super-polynomial resources allow for general concurrent composition in the plain model while being compatible with UC security. As an application a secure constant-round (black-box) general MPC protocol was modularly designed and future work will be needed to make this proof of concept a general principle.

Bibliography

- [AIR01] Bill Aiello, Yuval Ishai, and Omer Reingold. “Priced Oblivious Transfer: How to Sell Digital Goods”. In: *Advances in Cryptology – EUROCRYPT 2001: 20th Annual International Conference on the Theory and Application of Cryptographic Techniques, Proceedings*. (Innsbruck, Austria, May 6–10, 2001). Ed. by Birgit Pfitzmann. Berlin, Heidelberg: Springer, 2001, pp. 119–135.
- [Bar+04] Boaz Barak et al. “Universally Composable Protocols with Relaxed Set-Up Assumptions”. In: *45th Annual IEEE Symposium on Foundations of Computer Science*. (Rome, Italy, Oct. 17–19, 2004). FOCS ’04. IEEE. 2004, pp. 186–195.
- [BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. “Short Group Signatures”. In: *Advances in Cryptology – CRYPTO 2004: 24th Annual International Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA, Aug. 15–19, 2004). Berlin, Heidelberg: Springer, 2004, pp. 41–55.
- [Blu81] Manuel Blum. “Coin Flipping by Telephone”. In: *Advances in Cryptology – CRYPTO 1981: IEEE Workshop on Communications Security*. (Santa Barbara, CA, USA, Aug. 24–26, 1981). University of California, Santa Barbara, Department of Electrical and Computer Engineering, 1981, pp. 11–15.
- [BS05] Boaz Barak and Amit Sahai. “How to play almost any mental game over the net – concurrent composition via super-polynomial simulation”. In: *46th Annual IEEE Symposium on Foundations of Computer Science*. (Pittsburgh, PA, USA, Oct. 23–25, 2005). FOCS ’05. IEEE. 2005, pp. 543–552. Cryptology ePrint Archive (IACR): Report 2005/106. Version 2005-08-26.

- [Can+02] Ran Canetti et al. “Universally Composable Two-party and Multi-party Secure Computation”. In: *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*. (Montréal, QC, Canada, May 19–21, 2002). STOC '02. New York, NY, USA: ACM, 2002, pp. 494–503. Cryptology ePrint Archive (IACR): Report 2002/140. Version 2003-07-14.
- [Can+07] Ran Canetti et al. “Universally Composable Security with Global Setup”. In: *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Proceedings*. (Amsterdam, The Netherlands, Feb. 21–24, 2007). Ed. by Salil P. Vadhan. Berlin, Heidelberg: Springer, 2007, pp. 61–85. Cryptology ePrint Archive (IACR): Report 2006/432. Version 2007-10-02.
- [Can01] Ran Canetti. “Universally Composable Security: A New Paradigm for Cryptographic Protocols”. In: *42th Annual IEEE Symposium on Foundations of Computer Science*. (Las Vegas, NV, USA, Oct. 14–17, 2001). FOCS '01. IEEE. 2001, pp. 136–145. Cryptology ePrint Archive (IACR): Report 2000/067. Version 2013-07-16.
- [CF01] Ran Canetti and Marc Fischlin. “Universally composable commitments”. In: *Advances in Cryptology – CRYPTO 2001: 21st Annual International Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA, Aug. 19–23, 2001). Ed. by Joe Kilian. Berlin, Heidelberg: Springer, 2001, pp. 19–40. Cryptology ePrint Archive (IACR): Report 2001/055.
- [CGJ15] Ran Canetti, Vipul Goyal, and Abhishek Jain. “Concurrent Secure Computation with Optimal Query Complexity”. In: *Advances in Cryptology – CRYPTO 2015: 35th Annual Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA). Ed. by Rosario Gennaro and Matthew Robshaw. Berlin, Heidelberg: Springer, 2015, pp. 43–62. Cryptology ePrint Archive (IACR): Report 2015/559.
- [CKL03] Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. “On the Limitations of Universally Composable Two-Party Computation without Set-up Assumptions”. In: *Advances in Cryptology – EUROCRYPT 2003: 22nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*. (Warsaw, Poland, May 4–8, 2003). Ed. by Eli Biham. Berlin, Heidelberg: Springer, 2003, pp. 68–86. Cryptology ePrint Archive (IACR): Report 2004/116.
- [CLP10] Ran Canetti, Huijia Lin, and Rafael Pass. “Adaptive hardness and composable security in the plain model from standard assumptions”. In: *51st Annual IEEE Symposium on Foundations of Computer Science*. (Las Vegas, NV, USA, Oct. 23–26, 2010). FOCS '10. IEEE. 2010, pp. 541–550.
- [CLP13] Ran Canetti, Huijia Lin, and Rafael Pass. “From Unprovability to Environmentally Friendly Protocols”. In: *54th Annual IEEE Symposium on Foundations of Computer Science*. (Berkeley, CA, USA, Oct. 26–29, 2013). FOCS '13. IEEE. 2013, pp. 70–79.

- [CPS07] Ran Canetti, Rafael Pass, and Abhi Shelat. “Cryptography from Sunspots: How to Use an Imperfect Reference String”. In: *48th Annual IEEE Symposium on Foundations of Computer Science*. (Providence, RI, USA, Oct. 20–23, 2007). FOCS ’07. IEEE. 2007, pp. 249–259.
- [Dac+13] Dana Dachman-Soled et al. “Adaptive and Concurrent Secure Computation from New Adaptive, Non-malleable Commitments”. In: *Advances in Cryptology – ASIACRYPT 2013: 19th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*. (Bengaluru, India, Dec. 1–5, 2013). Ed. by Kazue Sako and Palash Sarkar. Berlin, Heidelberg: Springer, 2013, pp. 316–336. Cryptology ePrint Archive (IACR): Report 2011/611.
- [DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. “Nonmalleable Cryptography”. In: *SIAM Journal on Computing* 30.2 (2000), pp. 391–437.
- [DS13] Ivan Damgård and Alessandra Scafuro. “Unconditionally secure and universally composable commitments from physical assumptions”. In: *Advances in Cryptology – ASIACRYPT 2013: 19th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*. (Bengaluru, India, Dec. 1–5, 2013). Ed. by Kazue Sako and Palash Sarkar. Berlin, Heidelberg: Springer, 2013, pp. 100–119. Cryptology ePrint Archive (IACR): Report 2013/108. Version 2015-03-03.
- [ElG84] Taher ElGamal. “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”. In: *Advances in Cryptology – CRYPTO 1984: 4th Annual International Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA, Aug. 19–22, 1984). Ed. by George Robert Blakley and David Chaum. Berlin, Heidelberg: Springer, 1984, pp. 10–18.
- [FS90] Uriel Feige and Adi Shamir. “Witness Indistinguishable and Witness Hiding Protocols”. In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*. (Baltimore, MD, USA, May 13–17, 1990). STOC ’90. New York, NY, USA: ACM, 1990, pp. 416–426.
- [Gar+12] Sanjam Garg et al. “Concurrently Secure Computation in Constant Rounds”. In: *Advances in Cryptology – EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*. (Cambridge, UK, Apr. 15–19, 2012). Ed. by David Pointcheval and Thomas Johansson. Berlin, Heidelberg: Springer, 2012, pp. 99–116. Cryptology ePrint Archive (IACR): Report 2011/251.
- [GGJ13] Vipul Goyal, Divya Gupta, and Abhishek Jain. “What Information Is Leaked under Concurrent Composition?” In: *Advances in Cryptology – CRYPTO 2013: 33rd Annual Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA). Ed. by Ran Canetti and Juan A. Garay. Berlin, Heidelberg: Springer, 2013, pp. 220–238. Cryptology ePrint Archive (IACR): Report 2015/420.

- [GJ13] Vipul Goyal and Abhishek Jain. “On Concurrently Secure Computation in the Multiple Ideal Query Model”. In: *Advances in Cryptology – EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*. (Athens, Greece, May 26–30, 2013). Ed. by Thomas Johansson and Phong Q. Nguyen. Berlin, Heidelberg: Springer, 2013, pp. 684–701. Cryptology ePrint Archive (IACR): Report 2015/439.
- [GJO10] Vipul Goyal, Abhishek Jain, and Rafail Ostrovsky. “Password-Authenticated Session-Key Generation on the Internet in the Plain Model”. In: *Advances in Cryptology – CRYPTO 2010: 30th Annual Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA). Ed. by Tal Rabin. Berlin, Heidelberg: Springer, 2010, pp. 277–294. Cryptology ePrint Archive (IACR): Report 2015/420.
- [Goy+14] Vipul Goyal et al. “An Algebraic Approach to Non-malleability”. In: *55th Annual IEEE Symposium on Foundations of Computer Science*. (Philadelphia, PA, USA, Oct. 18–21, 2014). FOCS ’14. IEEE, 2014, pp. 41–50. Cryptology ePrint Archive (IACR): Report 2014/586. Version 2016-09-15.
- [Goy+15] Vipul Goyal et al. “Round-Efficient Concurrently Composable Secure Computation via a Robust Extraction Lemma”. In: *Theory of Cryptography: 12th Theory of Cryptography Conference, TCC 2015, Proceedings*. (Warsaw, Poland, May 23–25, 2015). Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Berlin, Heidelberg: Springer, 2015, pp. 260–289. Cryptology ePrint Archive (IACR): Report 2012/652.
- [HV15] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. “On Black-Box Complexity of Universally Composable Security in the CRS Model”. In: *Advances in Cryptology – ASIACRYPT 2015: 21st International Conference on the Theory and Application of Cryptology and Information Security, Proceedings, Part II*. (Auckland, New Zealand, Nov. 29–Dec. 3, 2015). Ed. by Tetsu Iwata and Jung Hee Cheon. Berlin, Heidelberg: Springer, 2015, pp. 183–209.
- [HV16] Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. “Composable Adaptive Secure Protocols without Setup under Polytime Assumptions”. In: *Theory of Cryptography: 14th Theory of Cryptography Conference, TCC 2016-B, Proceedings*. (Beijing, China, Oct. 31–Nov. 3, 2016). Printed version not yet published. 2016. Cryptology ePrint Archive (IACR): Report 2016/818. Version 2016-08-29.
- [IPS08] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. “Founding Cryptography on Oblivious Transfer – Efficiently”. In: *Advances in Cryptology – CRYPTO 2008: 28th Annual International Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA, Aug. 17–21, 2008). Ed. by David Wagner. Berlin, Heidelberg: Springer, 2008, pp. 572–591.
- [Kat07] Jonathan Katz. “Universally Composable Multi-party Computation Using Tamper-Proof Hardware”. In: *Advances in Cryptology*

- *EUROCRYPT 2007: 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*. (Barcelona, Spain, May 20–24, 2007). Ed. by Moni Naor. Berlin, Heidelberg: Springer, 2007, pp. 115–128.
- [Kiy14] Susumu Kiyoshima. “Round-Efficient Black-Box Construction of Composable Multi-Party Computation”. In: *Advances in Cryptology – CRYPTO 2014: 34th Annual Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA). Ed. by Juan A. Garay and Rosario Gennaro. Berlin, Heidelberg: Springer, 2014, pp. 351–368. Cryptology ePrint Archive (IACR): Report 2014/557.
- [KL11] Dafna Kidron and Yehuda Lindell. “Impossibility Results for Universal Composability in Public-Key Models and with Fixed Inputs”. In: *Journal of Cryptology* 24.3 (2011), pp. 517–544. Cryptology ePrint Archive (IACR): Report 2007/478. Version 2010-06-06.
- [KLP07] Tauman Yael Kalai, Yehuda Lindell, and Manoj Prabhakaran. “Concurrent Composition of Secure Protocols in the Timing Model”. In: *Journal of Cryptology* 20.4 (Oct. 2007), pp. 431–492.
- [KMO14] Susumu Kiyoshima, Yoshifumi Manabe, and Tatsuaki Okamoto. “Constant-Round Black-Box Construction of Composable Multi-Party Computation Protocol”. In: *Theory of Cryptography: 11th Theory of Cryptography Conference, TCC 2014, Proceedings*. (San Diego, CA, USA, Feb. 24–26, 2014). Ed. by Yehuda Lindell. Berlin, Heidelberg: Springer, 2014, pp. 343–367. Cryptology ePrint Archive (IACR): Report 2013/811.
- [Lin03] Yehuda Lindell. “General Composition and Universal Composability in Secure Multi-party Computation”. In: *44th Annual IEEE Symposium on Foundations of Computer Science*. (Cambridge, MA, USA, Oct. 11–14, 2003). FOCS ’03. IEEE, 2003, pp. 394–403. Cryptology ePrint Archive (IACR): Report 2004/116.
- [LP09] Huijia Lin and Rafael Pass. “Non-malleability Amplification”. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*. (Bethesda, MD, USA, May 31–June 2, 2009). STOC ’09. New York, NY, USA: ACM, 2009, pp. 189–198.
- [LP12] Huijia Lin and Rafael Pass. “Black-Box Constructions of Composable Protocols without Set-Up”. In: *Advances in Cryptology – CRYPTO 2012: 32nd Annual Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA). Ed. by Reihaneh Safavi-Naini and Ran Canetti. Berlin, Heidelberg: Springer, 2012, pp. 461–478.
- [LPV09] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkatasubramanian. “A Unified Framework for Concurrent Security: Universal Composability from Stand-alone Non-malleability”. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*. (Bethesda, MD, USA, May 31–June 2, 2009). STOC ’09. New York, NY, USA: ACM, 2009, pp. 179–188.

- [LPV12] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkatasubramanian. “A Unified Framework for UC from Only OT”. In: *Advances in Cryptology – ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Proceedings*. (Beijing, China, Dec. 2–6, 2012). Ed. by Xiaoyun Wang and Kazue Sako. Berlin, Heidelberg: Springer, 2012, pp. 699–717.
- [MMY06] Tal Malkin, Ryan Moriarty, and Nikolai Yakovenko. “Generalized Environmental Security from Number Theoretic Assumptions”. In: *Theory of Cryptography: 3rd Theory of Cryptography Conference, TCC 2006, Proceedings*. (New York, USA, May 4–7, 2006). Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer, 2006, pp. 343–359.
- [MPR06] Silvio Micali, Rafael Pass, and Alon Rosen. “Input-Indistinguishable Computation”. In: *47th Annual IEEE Symposium on Foundations of Computer Science*. (Berkeley, CA, USA, Oct. 21–24, 2006). FOCS ’06. IEEE. 2006, pp. 367–378.
- [Pas03] Rafael Pass. “Simulation in Quasi-Polynomial Time, and Its Application to Protocol Composition”. In: *Advances in Cryptology – EUROCRYPT 2003: 22nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*. (Warsaw, Poland, May 4–8, 2003). Ed. by Eli Biham. Berlin, Heidelberg: Springer, 2003, pp. 160–176.
- [PR05] Rafael Pass and Alon Rosen. “Concurrent non-malleable commitments”. In: *46th Annual IEEE Symposium on Foundations of Computer Science*. (Pittsburgh, PA, USA, Oct. 23–25, 2005). FOCS ’05. IEEE. 2005, pp. 563–572. Version 2005-08-26.
- [PR08] Manoj Prabhakaran and Mike Rosulek. “Cryptographic complexity of multi-party computation problems: Classifications and separations”. In: *Advances in Cryptology – CRYPTO 2008: 28th Annual International Cryptology Conference, Proceedings*. (Santa Barbara, CA, USA, Aug. 17–21, 2008). Ed. by David Wagner. Berlin, Heidelberg: Springer, 2008, pp. 262–279.
- [PS04] Manoj Prabhakaran and Amit Sahai. “New Notions of Security: Achieving Universal Composability Without Trusted Setup”. In: *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*. (Chicago, IL, USA, June 13–16, 2004). STOC ’04. New York, NY, USA: ACM, 2004, pp. 242–251. Cryptology ePrint Archive (IACR): Report 2004/139.
- [Ven14] Muthuramakrishnan Venkatasubramanian. “On Adaptively Secure Protocols”. In: *Security and Cryptography for Networks: 9th International Conference, SCN 2014, Proceedings*. (Amalfi, Italy, Sept. 3–5, 2014). Ed. by Michel Abdalla and Roberto De Prisco. Springer, 2014, pp. 455–475.

A Short Introduction into UC Security

The framework we built our security notion upon is the universal composability model as originally defined by [Can01]. For self-containment we give a brief overview over the framework.

The essential idea is to define security by means of the indistinguishability between an experiment in which the task at hand is carried out by dummy parties with the help of an ideal incorruptible entity and an experiment in which the parties must conduct the task themselves. In contrast to previous attempts to define security by simulation the indistinguishability must not only hold after the protocol execution has completed, but the distinguisher—called the environment \mathcal{Z} —takes part in the experiment, orchestrates all adversarial attacks, supplies the inputs to the parties running the challenge protocol and can observe the parties’ output as well as communication during the whole protocol execution.

The basic model of computation The basic model of computation consists of a set of (possibly polynomial many) instances (ITIs) of interactive Turing machines (ITMs). An interactive Turing machine (ITM) is the description of a Turing machine with additional tapes, namely the identity tape, tapes for subroutine input and output as well as tapes for incoming and outgoing network messages. The tangible instantiation of an ITM—the ITI—is identified by the content of its identity tape which consists of a session and a party identifier (SID/PID). The order of activation of the ITIs is completely asynchronous and message-driven. An ITI gets activated if either subroutine input or an incoming message is written onto its respective tape. If the ITI provides subroutine output or writes an outgoing message, the activation of the ITI completes and the ITI to whom the message has been delivered to gets activated next. Each experiment comprises two special ITIs the environment \mathcal{Z} and the adversary \mathcal{A} (in the real experiment) or the simulator \mathcal{S} (in the ideal experiment). The environment is the ITI that is initially activated. If during the execution any ITI completes its activation without giving any output, the environment is activated again as a fall-back. If the environment \mathcal{Z} conducts a subroutine output, the whole experiments stops. The output of the experiment is the output of \mathcal{Z} .

The adversary The adversary \mathcal{A} has the following capabilities. If any ITI writes an outgoing message the message is not directly delivered to the incoming tape of designated receiver but the adversary is responsible for all message transfers. To this end every message is implicitly copied to the incoming message tape of the adversary. The adversary can process the message arbitrarily. The adversary may decide to deliver to message (by writing the message on its own outgoing tape), the adversary may postpone or completely suppress the message, inject new messages or alter messages in any way including the recipient and/or alleged sender. This modeling reflects the idea of an unreliable and untrusted network. Please note twofold: (a) Only incoming/otgoing messages are under the control of the adversary, subroutine input/output between ITIs is immediate and trustworthy as long as the ITIs are *uncorrupted*. (b) As the sequence of

activations is message-driven the adversary also controls the scheduling and order of execution. Moreover the adversary can *corrupt* an ITI. In this case the adversary learns the complete entire state of the corrupted ITI and takes over its execution. This means whenever the corrupted ITI would have been activated (even due to subroutine input) the adversary gets activated with the same input.

The real experiment In the real experiment for a challenge protocol π , denoted by $\text{Exec}(\pi, \mathcal{A}, \mathcal{Z})$, the environment \mathcal{Z} is activated first. After the invocation of the adversary \mathcal{A} the environment \mathcal{Z} requests the creation of the challenge protocol. The main parties of π become subroutines of the environment and the environment freely choses their input and the SID of the challenge protocol. The experiment is executed as outlined above.

The ideal experiment In the ideal experiment, denoted by $\text{Exec}(\mathcal{F}, \mathcal{S}, \mathcal{Z})$, the challenge protocol is silently replaced by an instance of \mathcal{F} together with dummy parties. The dummy parties obtain a common session identifier (SID) and individual party identifiers (PIDs) from the environment as if they were the actual main parties of the protocol π in the real experiment, however they merely forward the subroutine input/output between the instance of the functionality \mathcal{F} and the environment. The ideal functionality \mathcal{F} is simultaneously a subroutine for each dummy party, holds the same SID but no PID, and conducts the prescribed task without the necessity to exchange any network messages. Moreover, in the ideal experiment the adversary is replaced by a simulator \mathcal{S} that mimics the adversarial behavior to the environment as if this was the real experiment with real parties carrying out the real protocol with real π -messages.

Definition of Security A real protocol π is said to securely UC-realize an ideal functionality \mathcal{F} , denoted by $\pi \geq_{\text{UC}} \mathcal{F}$, iff

$$\forall \mathcal{A} \exists \mathcal{S} \forall \mathcal{Z} : \text{Exec}(\pi, \mathcal{A}, \mathcal{Z}) \stackrel{c}{\equiv} \text{Exec}(\mathcal{F}, \mathcal{S}, \mathcal{Z}) \quad (15)$$

holds whereby the randomness on the left and on the right is taken over the initial input of \mathcal{Z} and all random tapes of all PPT machines.

This concludes the brief overview over the UC framework, but some final remarks are in order.

To prove security it suffices to consider only the so-called dummy adversary \mathcal{D} and to provide an simulator $\mathcal{S}_{\mathcal{D}}$ for that. The dummy adversary is merely an adversarial interface for the environment such that the environment remains agnostic if it runs in the real or ideal experiment but otherwise the dummy adversary \mathcal{D} is under complete control of the environment. If the environment requests \mathcal{D} to corrupt a party, \mathcal{D} does so and returns the state of the corrupted party to \mathcal{Z} , if the adversary needs to schedule a message, \mathcal{D} just forwards the message to \mathcal{Z} and leaves the decision how to proceed to the environment, and so on. This completeness of the dummy adversary allows to perceive the environment as the only adversarial entity.

The plain UC model does not pose any restriction on how session identifiers are generated. A very natural and often found convention is demand

that the session identifiers reflect the hierarchy of the ITIs. We also adhere to this restriction. More specifically, let the session identifier of a protocol be of the form $(string_1 \parallel \dots \parallel string_m)$. Then the session identifier of a sub-protocol is required to be of the form $(string_1 \parallel \dots \parallel string_{l+1})$. For a session identifier $sid = (string_1 \parallel \dots \parallel string_k)$, a session identifier of the form

$$sid' = (string_1 \parallel \dots \parallel string_k \parallel \dots \parallel string_m) \quad (16)$$

for $k \geq m$ is called an *extension* of sid .

Similar to the convention on session identifiers the order of corruptions must follow the hierarchy. In order to corrupt a sub-party, an adversary must corrupt all parties that are above that sub-party in the protocol hierarchy.

B Some Definitions and Proofs

This section contains definitions and proofs that were omitted in the main body of this work due to space restrictions.

B.1 Proof of Proposition 6

We prove Proposition 6, which shows that the dummy adversary is complete. The proof is almost exactly the same as in [Can01, Section 4.4.1]. Let us first repeat the exact statement.

Proposition 6 (Completeness of the dummy adversary). *Let π and ϕ be protocols. Then, π emulates ϕ in the presence of $\mathcal{F}^{\mathcal{O}}$ -augmented environments if and only if π emulates ϕ in the presence of $\mathcal{F}^{\mathcal{O}}$ -augmented environments with respect to the dummy adversary.*

Proof. Clearly, if π emulates ϕ in the presence of $\mathcal{F}^{\mathcal{O}}$ -augmented environments then π also emulates ϕ in the presence of $\mathcal{F}^{\mathcal{O}}$ -augmented environments with respect to the dummy adversary.

We now show the converse. Let π emulate ϕ in the presence of $\mathcal{F}^{\mathcal{O}}$ -augmented environments with respect to the dummy adversary. Let \mathcal{A} be an adversary interacting with a protocol π and an environment \mathcal{Z} . Define the simulator \mathcal{S} as follows: \mathcal{S} internally runs simulated copies of \mathcal{A} and the dummy adversary simulator $\mathcal{S}_{\mathcal{D}}$. \mathcal{S} relays the communication between \mathcal{A} and \mathcal{Z} and between $\mathcal{S}_{\mathcal{D}}$ and ϕ . When \mathcal{A} delivers a message m to a party with party identifier pid and session identifier sid , \mathcal{S} sends (sid, pid, m) to $\mathcal{S}_{\mathcal{D}}$. When $\mathcal{S}_{\mathcal{D}}$ outputs (sid, pid, m) , \mathcal{S} copies m to the incoming communication tape of \mathcal{A} as a message coming from the party with party identifier pid and session identifier sid .

Next define the environment $\mathcal{Z}_{\mathcal{D}}$ as follows: $\mathcal{Z}_{\mathcal{D}}$ internally runs simulated copies of \mathcal{Z} and \mathcal{A} . $\mathcal{Z}_{\mathcal{D}}$ relays the communication between \mathcal{A} and \mathcal{Z} and between \mathcal{Z} and the protocol parties. When \mathcal{A} delivers a message m the party with party identifier pid and session identifier sid , $\mathcal{Z}_{\mathcal{D}}$ sends (sid, pid, m) to the adversary. Likewise, when $\mathcal{Z}_{\mathcal{D}}$ receives a message (sid, pid, m) from the attacker, $\mathcal{Z}_{\mathcal{D}}$ copies m to the incoming communication tape of \mathcal{A} as a message coming from the party with party identifier pid and session identifier sid . $\mathcal{Z}_{\mathcal{D}}$ invokes the same instances of $\text{IDEAL}(\mathcal{F}^{\mathcal{O}})$ the environment \mathcal{Z} invokes. $\mathcal{Z}_{\mathcal{D}}$ outputs whatever \mathcal{Z} outputs. We have that

$$\begin{aligned} \text{Exec}(\pi, \mathcal{A}, \mathcal{Z}[\mathcal{F}^{\mathcal{O}}]) &\equiv \text{Exec}(\pi, \mathcal{D}, \mathcal{Z}_{\mathcal{D}}[\mathcal{F}^{\mathcal{O}}]) \\ &\stackrel{c}{\equiv} \text{Exec}(\phi, \mathcal{S}_{\mathcal{D}}, \mathcal{Z}_{\mathcal{D}}[\mathcal{F}^{\mathcal{O}}]) \\ &\equiv \text{Exec}(\phi, \mathcal{S}, \mathcal{Z}[\mathcal{F}^{\mathcal{O}}]) \quad , \end{aligned} \tag{17}$$

where the second step used the premise. The claim follows immediately. \square

B.2 Proof of Proposition 7

Proposition 7 (Transitivity). *Let π_1, π_2, π_3 be protocols. If $\pi_1 \geq_{\mathcal{F}^\mathcal{O}} \pi_2$ and $\pi_2 \geq_{\mathcal{F}^\mathcal{O}} \pi_3$ then it holds that $\pi_1 \geq_{\mathcal{F}^\mathcal{O}} \pi_3$.*

Proof. Let \mathcal{A} be an ITM. Since $\pi_1 \geq_{\mathcal{F}^\mathcal{O}} \pi_2$, there exists a simulator \mathcal{S}_1 such that $\text{Exec}(\pi_1, \mathcal{A}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \stackrel{c}{\equiv} \text{Exec}(\pi_2, \mathcal{S}_1, \mathcal{Z}[\mathcal{F}^\mathcal{O}])$, and since $\pi_2 \geq_{\mathcal{F}^\mathcal{O}} \pi_3$, there is \mathcal{S}_2 such that $\text{Exec}(\pi_2, \mathcal{S}_1, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \stackrel{c}{\equiv} \text{Exec}(\pi_3, \mathcal{S}_2, \mathcal{Z}[\mathcal{F}^\mathcal{O}])$ for all \mathcal{Z} .

Thus, $\text{Exec}(\pi_1, \mathcal{A}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \stackrel{c}{\equiv} \text{Exec}(\pi_3, \mathcal{S}_2, \mathcal{Z}[\mathcal{F}^\mathcal{O}])$, qed.

Since the SID mechanism is hierarchical, the class of $\mathcal{F}^\mathcal{O}$ -augmented environments that try to distinguish between π_1 and π_2 is the same as the class of $\mathcal{F}^\mathcal{O}$ -augmented environments that try to distinguish between π_2 and π_3 . \square

B.3 Definition of SPS Security

Definition 32 (Restatement of super-polynomial time simulator security). *Let π and ϕ be protocols. π is said to emulate ϕ with superpolynomial time simulator security, denoted by $\pi \geq_{\text{SPS}} \phi$, if*

$$\forall \mathcal{A} \exists (\text{SPS}) \mathcal{S} \forall \mathcal{Z} : \text{Exec}(\pi, \mathcal{A}, \mathcal{Z}) \stackrel{c}{\equiv} \text{Exec}(\phi, \mathcal{S}, \mathcal{Z}) \quad (18)$$

B.4 Proof of Corollary 10

Corollary 10 (Composition theorem for multiple oracles). *Let $\mathcal{O}, \mathcal{O}'$ be shielded oracles. Assume that $\pi \geq_{\mathcal{F}^\mathcal{O}} \mathcal{F}^\mathcal{O}$ and $\rho^{\mathcal{F}^\mathcal{O}} \geq_{\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}'}} \mathcal{G}^{\mathcal{O}'}$. Then there exists a shielded oracle \mathcal{O}'' such that $\rho^\pi \geq_{\mathcal{G}^{\mathcal{O}''}} \mathcal{G}^{\mathcal{O}''}$.*

Proof. Since $\rho^{\mathcal{F}^\mathcal{O}}$ emulates $\mathcal{G}^{\mathcal{O}'}$, there exists a simulator $\mathcal{S}_\mathcal{D}$ for the dummy adversary.

Define \mathcal{O}'' as follows: \mathcal{O}'' internally simulates $\mathcal{S}_\mathcal{D}$ and \mathcal{O}' , passes each message $\mathcal{S}_\mathcal{D}$ sends to \mathcal{G} to \mathcal{O}' , sends each `output-to-funct` output from \mathcal{O}' to \mathcal{G} and each `output-to-adv` output to $\mathcal{S}_\mathcal{D}$, and forwards the communication between $\mathcal{S}_\mathcal{D}$ and the environment. By construction, it holds that

$$\begin{aligned} \text{Exec}(\rho^{\mathcal{F}^\mathcal{O}}, \mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}'}]) &\stackrel{c}{\equiv} \text{Exec}(\mathcal{G}^{\mathcal{O}'}, \mathcal{S}_\mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}'}]) \\ &\equiv \text{Exec}(\mathcal{G}^{\mathcal{O}''}, \mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}''}]) \end{aligned} \quad (19)$$

Since $\mathcal{S}_\mathcal{D}$ runs in polynomial time, $(\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}'})$ -augmented environments can simulate $(\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}''})$ -augmented environments. Therefore, it follows from Proposition 6 that $\rho^{\mathcal{F}^\mathcal{O}} \geq_{\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}''}} \mathcal{G}^{\mathcal{O}''}$ and $\mathcal{G}^{\mathcal{O}''} \geq_{\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}''}} \rho^{\mathcal{F}^\mathcal{O}}$. By the composition theorem we have that $\rho^\pi \geq_{\mathcal{F}^\mathcal{O}} \rho^{\mathcal{F}^\mathcal{O}}$. Hence

$$\rho^\pi \geq_{\mathcal{F}^\mathcal{O}, \mathcal{G}^{\mathcal{O}''}} \mathcal{G}^{\mathcal{O}''} \quad (20)$$

Since it holds that $\mathcal{G}^{\mathcal{O}''} \geq_{\mathcal{F}^{\mathcal{O}}, \mathcal{G}^{\mathcal{O}''}} \rho^{\mathcal{F}^{\mathcal{O}}}$ one can iteratively replace all instances of $\text{IDEAL}(\mathcal{G}^{\mathcal{O}''})$ by instances of $\text{IDEAL}(\rho^{\mathcal{F}^{\mathcal{O}}})$ in a $\mathcal{G}^{\mathcal{O}''}$ -augmented environment, obtaining an $\mathcal{F}^{\mathcal{O}}$ -augmented. Therefore, it holds that $\rho^\pi \geq_{\mathcal{G}^{\mathcal{O}''}} \rho^{\mathcal{F}^{\mathcal{O}}}$. The statement follows from the transitivity of $\mathcal{G}^{\mathcal{O}''}$ -emulation. \square

B.5 Definition of Angel-based Security

Definition 33 (Restatement of Angel-based security). *Let π and ϕ be protocols, Γ an imaginary angel. π is said to emulate ϕ with Γ -angel-based security, denoted by $\pi \geq_{\Gamma\text{-Angel}} \phi$, if*

$$\forall \mathcal{A}^\Gamma \exists \mathcal{S}^\Gamma \forall \mathcal{Z}^\Gamma : \text{Exec}(\pi, \mathcal{A}^\Gamma, \mathcal{Z}^\Gamma) \stackrel{c}{\equiv} \text{Exec}(\phi, \mathcal{S}^\Gamma, \mathcal{Z}^\Gamma) \quad (21)$$

B.6 Restatement of Session-respecting Angels (Definition 16) and Detailed Proof of Theorem 17

Definition 34 (Session-respecting Angel (formal)). *Let Γ be an Angel and $\{id_1, id_2, \dots\}$ the set of all identifiers of all ITIs. We assume the identifiers to be of the form $id_i = (pid_i, sid_i)$. For the formulation of this definition we explicitly spell out the internal state of the Angel and consider the Angel to be a function*

$$\Gamma : (X, state, id, \mu) \mapsto (state', \sigma) \quad (22)$$

with

$$state' = \Gamma_s(X, state, id, \mu) \quad \sigma = \Gamma_\sigma(X, state, id, \mu) \quad (23)$$

whereby $X \subseteq \{id_1, id_2, \dots\}$ denotes the set of identifiers of corrupted ITIs, $\mu \in \{0, 1\}^*$ the inquiry, $\sigma \in \{0, 1\}^*$ the answer, $state \in \{0, 1\}^*$ the previous state before and $state' \in \{0, 1\}^*$ the new state after invocation.

An Angel is called session-respecting if the following conditions hold:

1. The state is of the form $state = (state_{sid_1}, state_{sid_2}, \dots)$
2. $\forall X \subseteq \{id_1, id_2, \dots\}, \forall state \in \{0, 1\}^*, \forall id \notin X, \forall \mu \in \{0, 1\}^* :$
 $\Gamma(X, state, id, \mu) = (state, \perp)$
3. Let $id_i \in X$ be the identifier of a corrupted ITI and sid_i the corresponding session identifier, i. e. $id_i = (pid_i, sid_i)$. Then there exists (possibly unbounded) TMs Γ'_s and Γ'_σ not depending on i nor sid_i such that

$$state' = (state_{sid_1}, state_{sid_2}, \dots, \Gamma'_s(X_{|sid_i}, state_{sid_i}, id_i, \mu), \dots) \quad (24)$$

$$\sigma = \Gamma'_\sigma(X_{|sid_i}, state_{sid_i}, id_i, \mu) \quad (25)$$

whereby $X_{|sid_i} \subseteq X$ denotes the subset of corrupted ITIs with session identifier sid_i .

The definition basically ensures that the global state of an session-respecting Angel can be split into independent components for each session. Especially, the Angel is still allowed to share some state between invocations as long as the state is only shared within the boundaries of the same session. This enables us to run the same code of the Angel on different, independent ITIs—one for each session.

Before we give the formal proof of the first part of Theorem 17 we repeat the statement here:

Theorem 17 (Relation between angels and shielded oracles).

1. Assume $\pi \geq_{\Gamma\text{-Angel}} \mathcal{F}$ for an imaginary Angel Γ . If Γ is session-respecting, then there exists a shielded oracle \mathcal{O} such that $\pi \geq_{\mathcal{F}^\mathcal{O}} \mathcal{F}^\mathcal{O}$.
2. Assume the existence of one-way functions. Then there exists a protocol ρ (in the $\mathcal{F}_{\text{auth}}$ -hybrid model), a functionality \mathcal{G} and a shielded oracle \mathcal{O} s. t. $\rho \geq_{\mathcal{G}^\mathcal{O}} \mathcal{G}^\mathcal{O}$ but no imaginary angel Γ can be found such that $\rho \geq_{\Gamma\text{-Angel}} \mathcal{G}$ holds.

Proof (Formal proof of claim 1). We consider the dummy adversary \mathcal{D} only. From the assumption $\pi \geq_{\Gamma\text{-Angel}} \mathcal{F}$ we have

$$\exists \mathcal{S}_1^\Gamma \forall \mathcal{Z}^\Gamma : \text{Exec}(\pi, \mathcal{D}^\Gamma, \mathcal{Z}^\Gamma) \equiv \text{Exec}(\mathcal{F}, \mathcal{S}_1^\Gamma, \mathcal{Z}^\Gamma) \quad (26)$$

We define the shielded oracle \mathcal{O} and a new simulator \mathcal{S}_2^Γ that both internally run an independent copy of \mathcal{S}_1^Γ . The functionality $\mathcal{F}^\mathcal{O}$ is defined as usual. If \mathcal{S}_1 that is internal to \mathcal{O} queries Γ , \mathcal{O} runs the code of Γ itself, if \mathcal{S}_1 that is internal to \mathcal{S}_2^Γ queries Γ , \mathcal{S}_2 relays the messages between \mathcal{S}_1 and the Angel that is given to \mathcal{S}_2 .

The routing of messages between the ITIs \mathcal{Z} , \mathcal{S}_2 and $\mathcal{F}^\mathcal{O}$ is as follows:

- If \mathcal{S}_2^Γ receives a message m from the environment that \mathcal{S}_2^Γ is supposed to sent on behalf of a corrupted party and the *sid* does not belong to the challenge protocol, then \mathcal{S}_2^Γ runs its internal \mathcal{S}_1^Γ on m and outputs whatever \mathcal{S}_1 would output.
- If \mathcal{S}_2^Γ receives a subroutine output m from an ideal functionality that is not a subsidiary of the challenge session for an corrupted party and, then \mathcal{S}_2^Γ runs its internal \mathcal{S}_1^Γ on m and outputs whatever \mathcal{S}_1 would output.
- If \mathcal{S}_2^Γ receives a message m from the environment that \mathcal{S}_2^Γ is supposed to sent on behalf of a corrupted party that is part of the challenge session, then \mathcal{S}_2^Γ forwards m to $\mathcal{F}^\mathcal{O}$ unmodified.
- If \mathcal{S}_2^Γ receives a simulated message m from $\mathcal{F}^\mathcal{O}$ on behalf of a corrupted party, then \mathcal{S}_2^Γ forwards m to the environment unmodified.
- If $\mathcal{F}^\mathcal{O}$ receives a forwarded message m from \mathcal{S}_2^Γ in the name a corrupted party, then $\mathcal{F}^\mathcal{O}$ internally routes the input to \mathcal{O} as usual, \mathcal{O} runs its internal copy of the original simulator \mathcal{S}_1^Γ and outputs whatever \mathcal{S}_1 would output. The internal output of \mathcal{O} is handled by $\mathcal{F}^\mathcal{O}$ as usual.
- If $\mathcal{F}^\mathcal{O}$ internally receives a subroutine output of \mathcal{F} that is intended for a corrupted party, $\mathcal{F}^\mathcal{O}$ internally routes the output to \mathcal{O} , \mathcal{O} runs its internal copy of the original simulator \mathcal{S}_1^Γ and outputs whatever \mathcal{S}_1 would output. The internal output of \mathcal{O} is handled by $\mathcal{F}^\mathcal{O}$ as usual.

The principle idea is that \mathcal{O} internally runs a copy of Γ for all queries that are related to the SID of the challenge protocol, while the simulator \mathcal{S}_2^Γ handles all remaining queries in virtue of the global Angel.

We now have to argue that

$$\text{Exec}(\mathcal{F}, \mathcal{S}_1^\Gamma, \mathcal{Z}^\Gamma) \equiv \text{Exec}(\mathcal{F}^\mathcal{O}, \mathcal{S}_2^\Gamma, \mathcal{Z}^\Gamma) \quad (27)$$

holds. The only difference between the both experiments is that in the left experiment only one ITI exists that runs the code of the Angel Γ while in the right experiment there are two ITIs that run Γ . Let $\mu^{(i)}, \sigma^{(i)}, \text{state}^{(i)}$ for $i \in \mathbb{N}$ denote the sequence of queries, responses and states of the Angel in the left experiment such that

$$\sigma^{(i)} = \Gamma_\sigma(\text{state}^{(i-1)}, \mu^{(i)}) \quad (28)$$

$$\text{state}^{(i)} = \Gamma_s(\text{state}^{(i-1)}, \mu^{(i)}) \quad (29)$$

holds. W.l.o.g we do not explicitly spell out X and id here (cp. Eqs. (22) and (23)).

Likewise let $\tilde{\mu}^{(i)}$ and $\tilde{\sigma}^{(i)}$ denote the sequence of all queries and responses in the right experiment. However, in the right experiment there are two Angels that mutually answers the queries, hence we have distinct state vectors. Let $\widetilde{\text{state}}^{(sid)}$ denote the state vector of the Angel inside the ITI that handles the session sid . Accordingly $\widetilde{\text{state}}_{sid}^{(sid)}$ denotes the sid 'th component of the state vector. We denote by $i(sid, j)$ the subsequence of query-response pairs for each SID and henceforth call i the global index and j the local index with respect to session sid . We stress that for every i there is exactly one unique pair (sid, j) such that $i = i(sid, j)$ and $i(sid, j)$ is monotone in j .

We show by induction that the sequences of queries and responses in both experiments are the same, i.e. that $\mu^{(i)} = \tilde{\mu}^{(i)}$ and $\sigma^{(i)} = \tilde{\sigma}^{(i)}$ holds for all i . To this end we fix the random coins and assume that \mathcal{Z} is deterministic. Along the way, we also show

$$\widetilde{\text{state}}_{sid}^{(sid, j)} = \text{state}_{sid}^{(i)} \quad (30)$$

for all i, j, sid under the restriction that $i = i(sid, j)$. Please note that we do not claim that the state vectors are completely equal but only on the relevant (“diagonal”) component. In general $\widetilde{\text{state}}^{(sid, j)} \neq \text{state}^{(i)}$ holds but the state vector is purely internal to the Angel and is never a part of the view of the environment.

Base clause of induction ($i = 1$): Until the first query both experiments are exactly identical. Hence, $\mu^{(1)} = \tilde{\mu}^{(1)}$ and $\widetilde{\text{state}}_{sid}^{(sid, 0)} = \text{state}_{sid}^{(0)}$ holds for all sid . The first claim of the base clause

$$\sigma^{(1)} = \Gamma'_\sigma(\text{state}_{sid}^{(0)}, \mu^{(1)}) = \Gamma'_\sigma(\widetilde{\text{state}}_{sid}^{(sid, 0)}, \tilde{\mu}^{(1)}) = \tilde{\sigma}^{(1)} \quad (31)$$

immediately follows. Let j, sid such that $1 = i = i(sid, j)$. Due to uniqueness and monotonicity of the subsequences $j = 1$ follows. This shows the second part

of the base clause

$$state_{sid}^{(1)} = \Gamma'_s(state_{sid}^{(0)}, \mu^{(1)}) = \Gamma'_s(\widetilde{state}_{sid}^{(sid,0)}, \tilde{\mu}^{(1)}) = \widetilde{state}_{sid}^{(sid,1)} \quad (32)$$

Induction step ($i-1 \rightsquigarrow i$): For all $i' < i$ the base clause of the induction holds, hence both experiments are identical up to query i . Hence, we have $\mu^{(i)} = \tilde{\mu}^{(i)}$. Let $i = i(sid, j)$ hold and define $\bar{i} := i(sid, j-1)$, i. e. \bar{i} is the global index of the previous query for the *same* SID. Obviously $\bar{i} < i$ holds. Let $i' \in \{\bar{i}, \dots, i\}$. Due to the definition of a session-respecting Angel we know that in the left experiment $state_{sid}^{(\bar{i})} = state_{sid}^{(\bar{i}+1)} = \dots = state_{sid}^{(i)}$ follows, because the sid 'th component of the state vector is never updated in between. Moreover, we have the equality $\widetilde{state}_{sid}^{(sid, j-1)} = state_{sid}^{(\bar{i})}$ between the left and the right experiment. By putting everything together, we obtain

$$\begin{aligned} \sigma^{(i)} &= \Gamma'_\sigma(state_{sid}^{(i-1)}, \mu^{(i)}) = \Gamma'_\sigma(state_{sid}^{(\bar{i})}, \mu^{(i)}) \\ &= \Gamma'_\sigma(\widetilde{state}_{sid}^{(sid, j-1)}, \tilde{\mu}^{(i)}) = \tilde{\sigma}^{(i)} \end{aligned} \quad (33)$$

and

$$\begin{aligned} state_{sid}^{(i)} &= \Gamma'_s(state_{sid}^{(i-1)}, \mu^{(i)}) = \Gamma'_s(state_{sid}^{(\bar{i})}, \mu^{(i)}) \\ &= \Gamma'_s(\widetilde{state}_{sid}^{(sid, j-1)}, \tilde{\mu}^{(i)}) = \widetilde{state}_{sid}^{(sid, j)} \end{aligned} \quad (34)$$

This completes the induction proof and we have shown (27).

Please note, that we are still in an Angel-based setting but \mathcal{S}_2^Γ does not need Γ for the challenge protocol. In order to prove $\pi \geq_{\mathcal{F}^\mathcal{O}} \mathcal{F}^\mathcal{O}$ we need to show

$$\exists \mathcal{S}^\Gamma \forall \mathcal{Z}^\Gamma : \text{Exec}(\pi, \mathcal{D}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \equiv \text{Exec}(\mathcal{F}^\mathcal{O}, \mathcal{S}, \mathcal{Z}[\mathcal{F}^\mathcal{O}]) \quad (35)$$

and we claim that $\mathcal{S} = \mathcal{S}_2$ suffices. Assume that (35) does not hold, i. e. there is a \mathcal{Z} that can distinguish between interacting with π and \mathcal{D} or with $\mathcal{F}^\mathcal{O}$ and \mathcal{S}_2 . Then the same environment \mathcal{Z} could also distinguish given direct access to Γ instead of being augmented by $\mathcal{F}^\mathcal{O}$ and thus contradicts (27). \square

C Illustrations

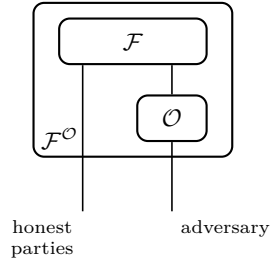


Fig. 1. Appended functionality $\mathcal{F}^\mathcal{O}$ internally runs \mathcal{F} and \mathcal{O} and enforces the correct routing of messages (cp. Definition 2)

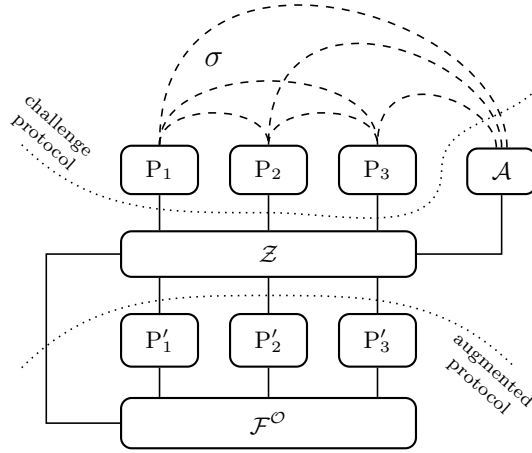


Fig. 2. Execution of the real experiment with challenge protocol and one additionally invoked $\mathcal{F}^\mathcal{O}$ -protocol (cp. Definition 3)

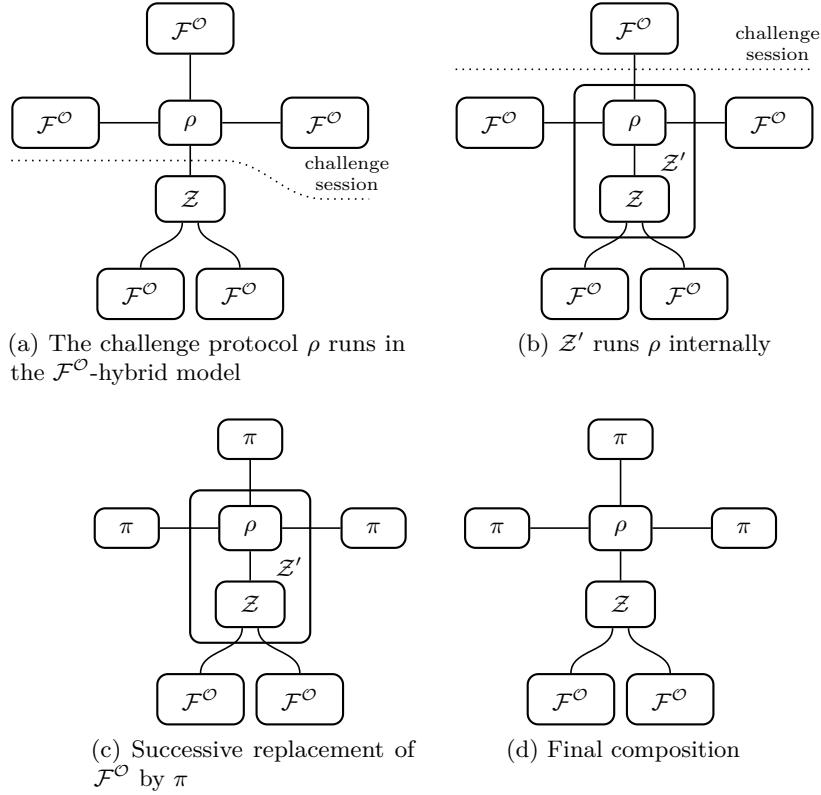


Fig. 3. The composition operation of $\mathcal{F}^{\mathcal{O}}$ -hybrid protocols in presence of $\mathcal{F}^{\mathcal{O}}$ -augmented environments (cp. proof of Theorem 9)

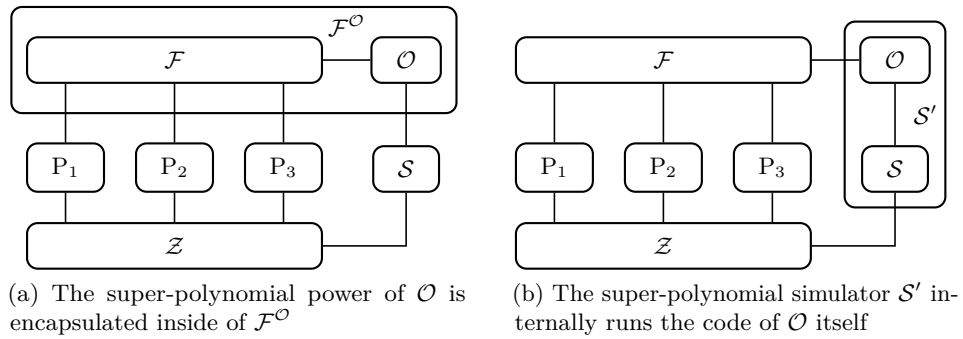


Fig. 4. Oracle-security implies SPS-security (cp. Proposition 8)

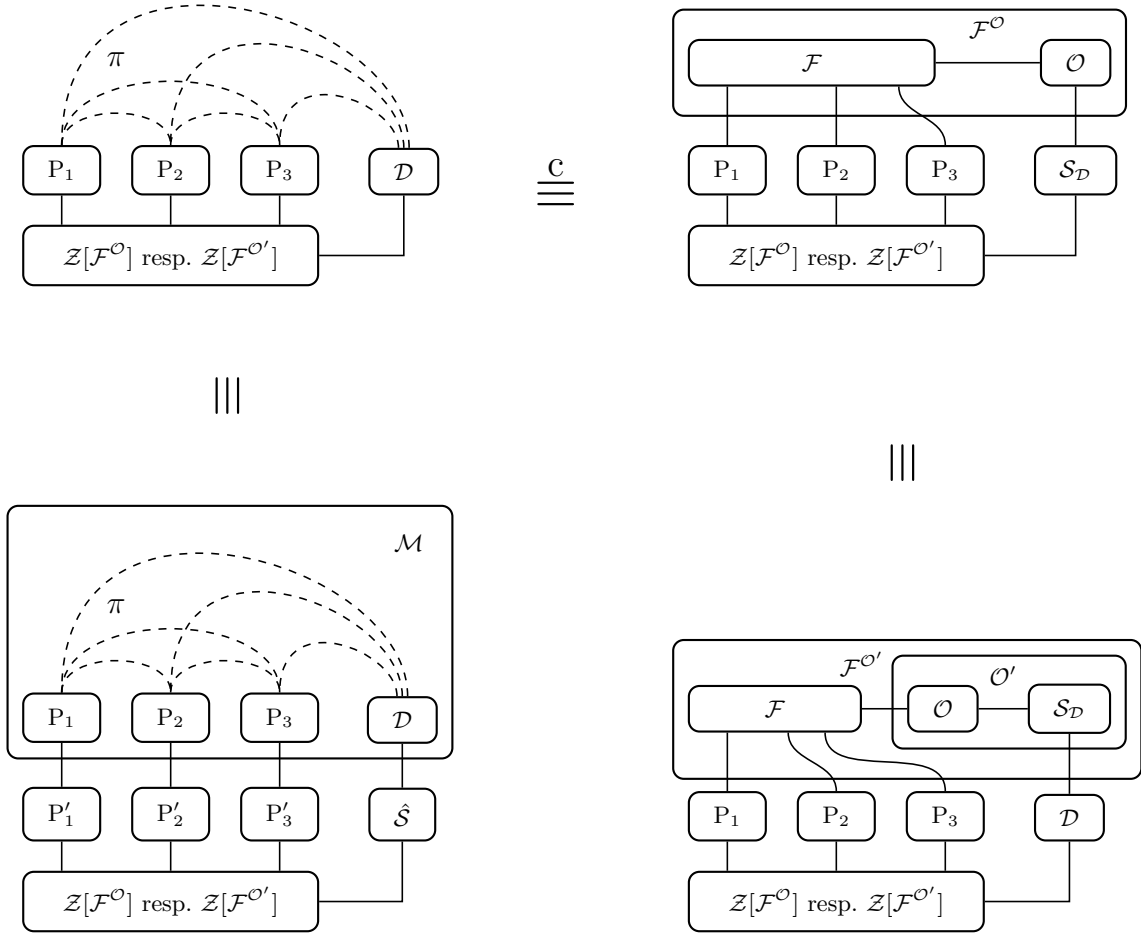


Fig. 5. Derived oracle (cp. Lemma 14)

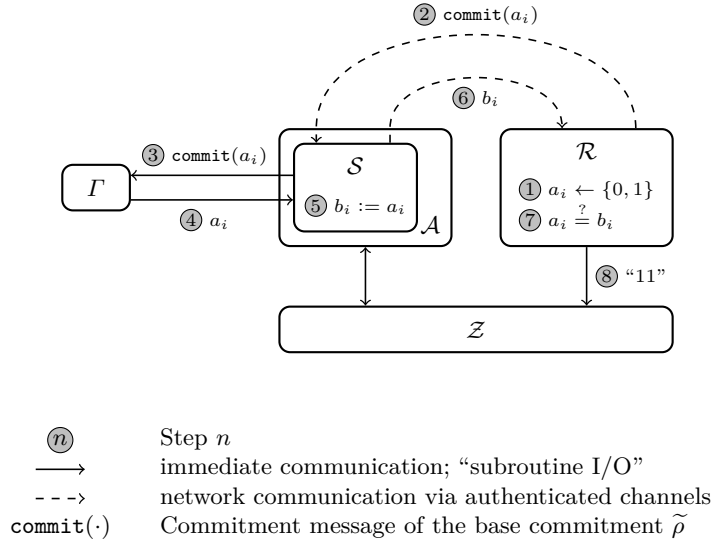


Fig. 6. Example of a protocol separating Angel-based security and Shielded-oracle security (cp. Theorem 17)

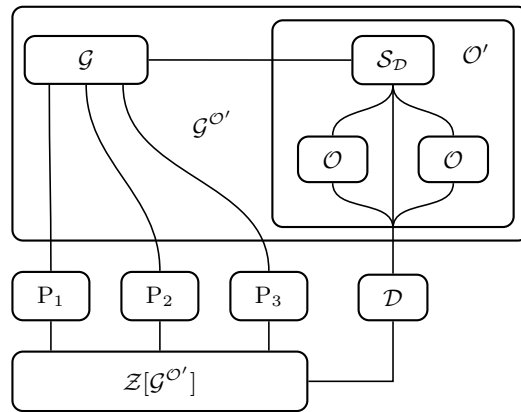


Fig. 7. The functionality \mathcal{G} with composed oracle \mathcal{O}'

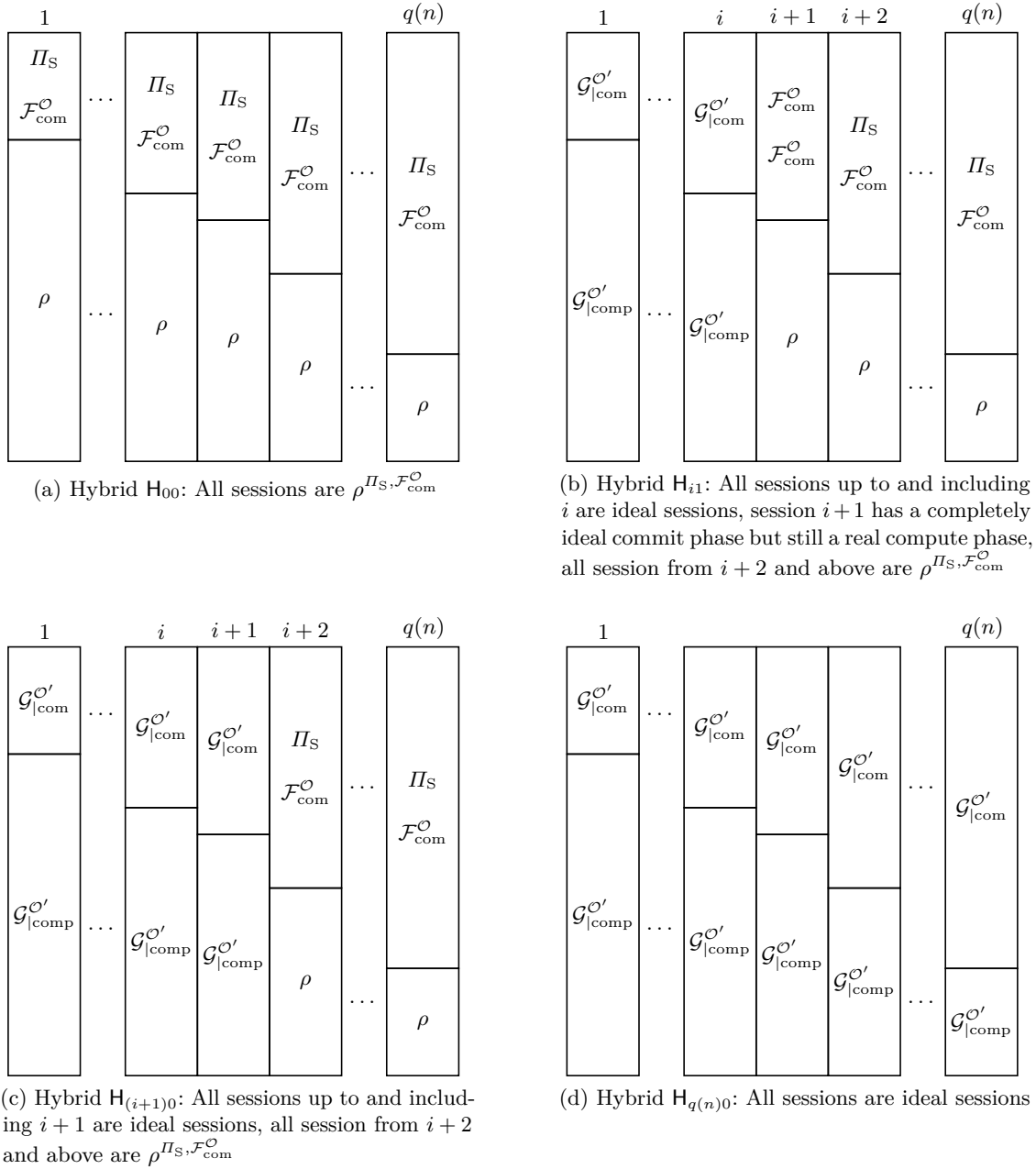
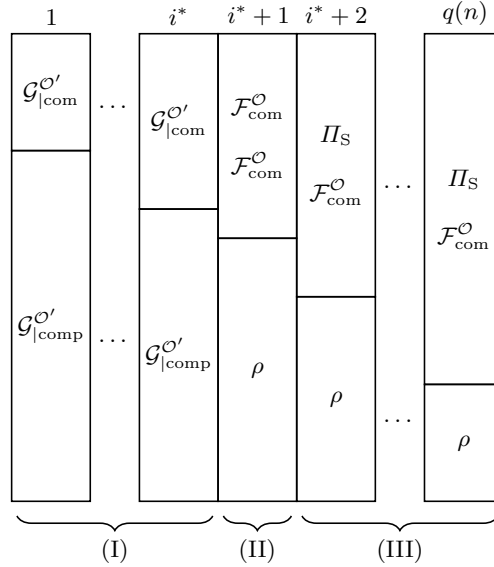
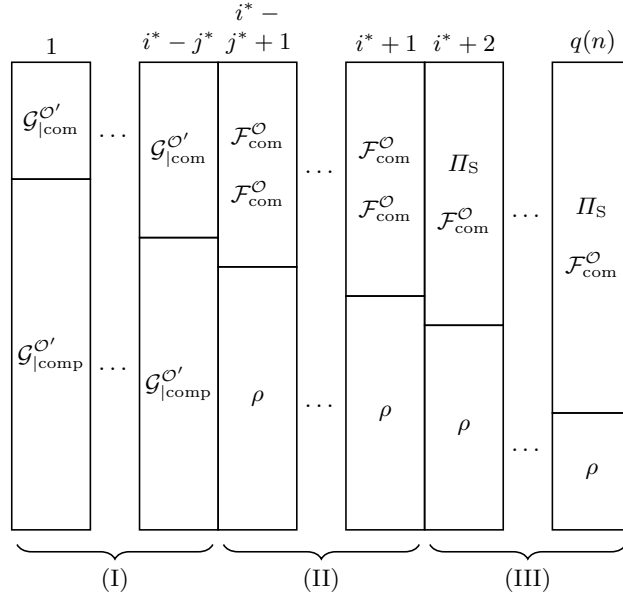


Fig. 8. The sequence of hybrids H_{ib} (see proof of Theorem 26): The notation $\mathcal{G}'_{\text{comp}}$ denotes the “compute phase” of an ideal \mathcal{G} -session and $\mathcal{G}'_{\text{com}}$ denotes the “commit phase” of an ideal \mathcal{G} -session



(a) Hybrid h_0 : The hybrid h_0 is identical to H_{i^*+1} (cp. Fig. 8b)



(b) Hybrid h_{j^*} : The substitution of ρ by \mathcal{G} has been reverted for sessions $k \in \{i^* - j^* + 1, \dots, i^* + 1\}$ (marked as range (II))

Fig. 9. The sequence of hybrids h_j (see proof of Theorem 26): The substitution conducted by the hybrids H_0 through H_{i^*+1} is partially reverted starting at session i^* and going j^* sessions backward. This results in three ranges: (I) These sessions remain purely ideal, (II) sessions that have been idealized before and are now reverted, (III) sessions that are still real and never have been modified