Simulation-Based Security with Inexhaustible Interactive Turing Machines*

Ralf Küsters

Institut für Informatik
Christian-Albrechts-Universität zu Kiel
24098 Kiel, Germany
kuesters@ti.informatik.uni-kiel.de

Abstract. Recently, there has been much interest in extending models for simulation-based security in such a way that the runtime of protocols may depend on the length of their input. Finding such extensions has turned out to be a non-trivial task. In this work, we propose a simple, yet expressive general computational model for systems of Interactive Turing Machines (ITMs) where the runtime of the ITMs may be polynomial per activation and may depend on the length of the input received. One distinguishing feature of our model is that the systems of ITMs that we consider involve a generic mechanism for addressing dynamically generated copies of ITMs. We study properties of such systems and, in particular, show that systems satisfying a certain acyclicity condition run in polynomial time. Based on our general computational model, we state different notions of simulation-based security in a uniform and concise way, study their relationships, and prove a general composition theorem for composing a polynomial number of copies of protocols, where the polynomial is determined by the environment. The simplicity of our model is demonstrated by the fact that many of our results can be proved by mere equational reasoning based on a few equational principles on systems.

1 Introduction

In the simulation-based security paradigm the security of protocols is defined in such a way that security is preserved even if the protocols are used as components of an arbitrary (polynomially bounded) distributed system. This strong composability property allows the modular design and analysis of protocols. The main idea behind simulation-based security is that the security of a protocol is defined in terms of an ideal protocol (also called, ideal functionality). A real protocol securely realizes the ideal protocol if every attack on the real protocol can be translated into an "equivalent" attack on the ideal protocol, where equivalence is specified based on an environment trying to distinguish the real attack from the ideal one.

Several related models for simulation-based security have been proposed [5, 19, 4, 3, 18, 10] (see [10] for a comparison of the models). In these models, systems

* This is the full version of [16].

of Interactive Turing Machines (ITMs) are considered. However, in the various models the ITMs can have different forms: rather standard ITMs, as used in [5], probabilistic I/O automata [4], and process calculus expressions [18, 10]. Depending on the kind of entities running in a system (environment, real/ideal adversary, simulator, real/ideal protocol) and the order of quantification over these entities, different security notions for simulation-based security are obtained, including strong [11, 10] (see also [5]), black-box [19], (dummy) universal [5, 19], and reactive simulatability [19].

The mentioned models have in common that the *total runtime* of the ITMs, i.e., the runtime summed over all activations, is bounded by a polynomial in the security parameter alone and may not depend on the length of the input that the ITMs receive from other ITMs. (We call these ITMs *exhaustible* in the following.) This is a mainly technically motivated restriction which guarantees that a system of ITMs runs in polynomial time in the security parameter. However, as explained below, it significantly limits the expressivity of the models and in some cases results in unintuitive behavior.

Recently, there has therefore been much interest in developing models for simulation-based security involving ITMs whose runtime may depend on the length of the input received from other ITMs. Canetti [7] and Hofheinz et al. [14] were the first to propose and study models for simulation-based security with such ITMs. Developing such models is a non-trivial task. As already pointed out in [4, 7, 14], naïve extensions of the existing models do not work: In a system of ITMs (whose runtime may depend on the length of their input), two ITMs can send messages back and forth among each other. Hence, such a system would not terminate, let alone perform a polynomially bounded computation, which is, however, required to guarantee the security of cryptographic primitives. Canetti [7] and Hofheinz et al. [14] have pointed out that globally bounding the runtime of an otherwise possibly non-terminating system by a polynomial, i.e., stopping the system after a polynomial bound has been reached, also does not yield a reasonable computational model for simulation-based security: an environment could (artificially) distinguish a real attack from an ideal one by measuring the overall number of steps taken in the different attacks.

Contribution of this Paper. In this paper, we propose a model for simulation-based security which extends and simplifies several aspects of previous models (see also the related work). More precisely, the main contributions of this work are twofold: First, we propose a simple, yet expressive general computational model for systems of what we call inexhaustible ITMs independent of the application to simulation-based security. A distinguishing feature of this model is a generic mechanism for addressing dynamically generated ITMs. Second, we demonstrate the flexibility and simplicity of our model by formalizing several notions of simulation-based security in it, along with a study of the relationships of the security notions and general composition theorems. In previous models, the formulations of the security notions were much more cumbersome or the security notions could not be formalized at all. Also, the composition theorems were more restricted in different respects. The simplicity of our model is also

reflected in the fact that many proofs can be carried out by mere equational reasoning on systems based on a few equational principles. Let us explain our general computational model and the application to simulation-based security in more detail.

The general computational model. The main building blocks of our computational model are the already mentioned *inexhaustible ITMs*. The runtime of these ITMs is only polynomially bounded per activation where the polynomial is in the length of the current input, the security parameter, and the size of the current configuration, i.e., the length of the current content written on the work tapes of the machine. This enables a machine to read every input and in every activation scan its entire current configuration. It also prevents machines from being exhausted by other machines sending useless messages; we note that in Canetti's model [7], ITMs can be exhausted. Inexhaustible ITMs have two main features that distinguish them from weakly polynomial machines [4, 14]: First, they may run in one of two modes (CheckAddress and Compute). These modes are used within a generic mechanism for addressing copies of ITMs. This avoids to fix specific details of an addressing mechanism (such as session IDs) in the general computational model. Second, we distinguish between enriching and consuming input tapes of an ITM and require that the output produced by a single ITM and the size of its current configuration (i.e., the length of the content written on the work tapes of the machine) is bounded by a polynomial in the security parameter plus the length of the input that has been received on *enriching* tapes so far.

The systems of ITMs that we consider may contain an unbounded number of copies of ITMs. In a run of a system, ITMs may create new copies of ITMs by invoking other machines. In other words, the number of copies of ITMs is determined dynamically. As mentioned, using the two modes in which ITMs may run, we employ a generic mechanism for addressing copies of ITMs. In the application for simulation-based security, this mechanism allows us to model multi-party protocols and to talk about different sessions of a protocol (as needed in the composition theorems). We identify semantic and syntactic conditions on systems of ITMs which guarantee that these systems run in polynomial time. The syntactic condition is an acyclicity condition on the way ITMs are connected via the mentioned enriching tapes.

We prove several equational properties of systems in our general computational model which in the application for simulation-based security allows us to carry out many of the proofs by mere equational reasoning on systems. We also show that any (sub-)system can be simulated by a single ITM (the simulation is independent of the environment in which the system may run). In particular, this is true for those systems describing an unbounded number of sessions of a protocol. This is the core of the joint state theorem [9, 7].

While, as we will see, our general computational model forms a solid and flexible basis for studying different forms of simulation-based security, we believe that our model and the properties shown are of interest independent of the application to simulation-based security.

Simulation-based security. Based on our computational model, we state and investigate different notions of simulation-based security and prove general composition theorems. One important feature of our model is that the security notions can be stated in a uniform and concise way and that many proofs can be carried out based on the mentioned equational principles for (general) systems of ITMs.

More precisely, we consider two classes of systems for describing (real/ideal) protocols, while the latter class is only briefly discussed due to space limitations: IO-enriching protocol systems and IO-network-enriching protocol systems.

In IO-enriching protocol systems tapes which are part of the I/O interface of the (real/ideal) protocol may be enriching while those that are part of the network interface are consuming. This enables parties to produce output whose length is only polynomially bounded in the security parameter plus the length of the workload received on their I/O interface, such as messages to be encrypted, signed, or securely transmitted. For this class of protocol systems we formulate the security notions strong, black-box, (dummy) universal, and reactive simulatability and identify sufficient conditions under which these notions are equivalent. We also prove a general composition theorem for composing a polynomial number of copies of protocols where the polynomial can be determined by the environment (and/or superior protocols).

In IO-network-enriching protocol systems not only the tapes of the I/O interface but also of the network interface may be enriching, yielding a more general class of protocol systems, but with somewhat restricted security notions.

Drawbacks of Models with Exhaustible ITMs. Models for simulation-based security based on exhaustible ITMs have several drawbacks:

First, their expressivity is limited. For example, when specifying an ideal protocol for modeling encryption (see, e.g., [5, 2]) the number and length of messages that can be encrypted using this protocol has to be bounded by some fixed polynomial in the security parameter; the same is true for other ideal protocols, such as those for modeling signatures [5, 8, 6, 1, 2] and secure message transmission [19]. Such a fixed bound is quite artificial and also restricts the security guarantees. Inexhaustible ITMs overcome these problems. Another example that illustrates the limited expressivity in models with exhaustible ITMs is the following: A party, modeled as an ITM, running a protocol is not able to block useless messages. It first has to examine the incoming message to decide whether to drop or to further process the message. This task consumes resources, and hence, by swamping an ITM with useless messages, external parties, including the adversary and the environment, can exhaust the total runtime available to a party and force it to stop. A partial solution to this problem of blocking useless messages is the length function in the model of Backes, Pfitzmann, and Waidner [4]. A more general approach is the concept of guards introduced in [10]. However, inexhaustible ITMs supersede such constructions.

Second, models with exhaustible ITMs exhibit in some cases unintuitive and unexpected behavior. For example, almost identical protocols may not be simulatable w.r.t. black-box simulatability, while they are with universal simulatability. More concretely, consider an ideal and real protocol which are identical

except that on the network interface the ideal protocol sends the bit-wise complement of messages the real protocol would send. While the real protocol realizes the ideal protocol w.r.t. universal simulatability, this is, in general, not the case w.r.t. black-box simulatability. The main reason for this peculiarity is that if the runtime of ITMs is polynomially bounded in the security parameter, then in general it is not possible to plug an entity, say $\mathcal{D}$, between two other entities, say $\mathcal{Q}_1$ and $\mathcal{Q}_2$, such that $\mathcal{D}$ can be chosen independently of at least one of the entities $\mathcal{Q}_1$ or $\mathcal{Q}_2$, and such that $\mathcal{D}$ forwards messages between $\mathcal{Q}_1$ and $\mathcal{Q}_2$ (this property is called FORWARDER property in [10]): $\mathcal{D}$ can be exhausted by these entities, i.e., the runtime available to $\mathcal{D}$ might not suffice to forward all messages between $\mathcal{Q}_1$ and $\mathcal{Q}_2$. As shown in [10], this property has a great impact on the relationships between the different security notions. When using ITMs where all tapes are enriching, the FORWARDER property can be satisfied. Another unexpected behavior is that when considering systems where the number of ITMs in the system is unbounded and is determined by the environment, then such a system cannot be simulated by a single ITM. For example, one cannot simulate an unbounded number of copies of protocols within one ITM. This is, however, what is required by the joint state theorem as stated in [7, 9]. In our model, every (sub-)system can be simulated by a single ITM.

Related Work. As mentioned above, Canetti [7] and Hofheinz et al. [14] were the first to study models for simulation-based security where the runtime of ITMs may depend on the length of their input. Let us discuss the main differences to the present work.

The results proved by Hofheinz et al. are most closely related to the results presented in this paper for IO-network-enriching protocol systems. These protocol systems are quite similar in expressivity to the polynomially shaped weakly polynomial collections considered in [14]. However, the way the security notions are defined is quite different from the definitions presented here. Hofheinz et al. do not consider IO-enriching protocol systems, for which in the present paper we have formulated and studied the different security notions (strong, black-box, (dummy) universal, and reactive simulatability), and proved a general composition theorem. Strong and black-box simulatability have not been investigated by Hofheinz et al. The computational model employed in [14] does not explicitly allow to talk about systems with an unbounded number of dynamically generated ITMs (such systems would have to be simulated within a fixed and finite number of ITMs). Hence, without further extending the model by Hofheinz et al., a composition theorem for composing a polynomial number of machines can not be stated (let alone proved) in their setting.

The model by Canetti [7] has been evolving over time and is still subject to change. Canetti's model has two main features that our model does not have: ITMs may generate the *code* of machines they invoke and a description of a system involves a control function which oversees whether or not an ITM is allowed to communiate with another ITM. Compared to Canetti's model, our model is much simpler, and yet, more expressive and flexible in different respects: The *total* runtime of the ITMs that Canetti employs is bounded by a polynomial in the

security parameter and the length of the input received on I/O tapes (minus the runtime provided to "'subroutine ITMs"'). In particular, the runtime of these ITMs may not be polynomial per activation, and in fact, these ITMs can be exhausted by other ITMs sending useless messages. This limits the expressivity of the model in the sense that certain protocols can not be formulated, e.g., protocols that, without consuming resources, simply ignore messages of unexpected format. Also, the exhaustion of ITM leads to much more involved constructions for the security notions and proofs. By using inexhaustible ITMs, we avoid these problems in our model. To guarantee that systems as defined by Canetti run in polynomial time, the length of the output of ITMs when invoking new ITMs must be strictly decreasing compared to the length of the input received. The number of ITMs that may be invoked by a machine is also restricted in a certain way. We do not have these restrictions in our model. Another difference between the two models is that Canetti explicitly defines as part of his computational model how session and party IDs are used to address specific copies of protocols. In the present work we instead have developed a more general mechanism for this purpose and do not fix details of the addressing mechanism in the general computational model. We also note that our composition theorems are more flexible in the way we allow protocols to connect to subprotocols. We finally mention that in the version of [7] from January 2005, Canetti considers ITMs whose runtime may depend on the input received on all tapes (he called such ITMs A-PPT). In our terminology, these are ITMs where all of the tapes are enriching. Canetti formulated different security notions using these ITMs. However, the proofs for establishing the relationships between these notions were flawed.

Structure of the Paper. In Section 2, we present our general computational model, including the definition of (inexhaustible) ITMs and systems of such ITMs, with important properties presented in Section 3. Based on the general computation model, we introduce the mentioned security notions for IO-enriching protocol systems in Section 4, with their relationships studied in Section 5. The composition theorems for this class of protocol systems are presented in Section 6. IO-network-enriching protocol systems are investigated in Section 7. We conclude in Section 8. More detailed definitions and full proofs are provided in the appendix.

Notation and Basic Terminology. For a bit string $a \in \{0,1\}^*$ we denote by $|a|$ the length of a. Following [7], a function $f : \{1\}^* \times \{0,1\}^* \to \mathbb{R}_{\geq 0}$ is *negligible* if for every polynomial p and q there exists k_0 such that for all $k > k_0$ and all bit strings $a \in \bigcup_{k' \leq q(k)} \{0,1\}^{k'}$ we have that $f(1^k, a) \leq \frac{1}{p(k)}$.

2 The General Computational Model

In this section, we define our general computational model independent of the application to simulation-based security. We introduce single interactive Turing machines and systems of such machines, define runs of systems, and state

basic properties. We also introduce further notation and terminology, used in subsequent sections.

2.1 Inexhaustible Interactive Turing Machines

We first introduce the syntax of (inexhaustible) interactive Turing machines and then the way these machines perform their computations.

Syntax. An *(inexhaustible) interactive Turing machine (ITM, for short) M* is a probabilistic Turing machine with the following tapes and a polynomial q associated with it where q will be used as a bound in computations of M: a read-only tape on which the mode the ITM M is supposed to run is written (the *mode tape*)—the possible modes are CheckAddress and Compute (see below)—, a read-only tape on which the security parameter is written (the *security parameter tape*), a write-only tape (the *address decision tape*, used in mode CheckAddress), zero or more *input* and *output tapes*, and *work tapes*. The input and output tapes have names and, in addition, input tapes have an attribute whose possible values are consuming or enriching (see below). We require that different tapes of M have different names. We allow M to randomly choose transitions. Alternatively, one could equip M with a random tape (see, e.g., [12]).

The set of input and output tapes of M is denoted by $\mathcal{T}(M)$, the set of input tapes by $\mathcal{T}_{in}(M)$, and the set of output tapes by $\mathcal{T}_{out}(M)$.

The names of input and output tapes will determine how ITMs are connected in a system of ITMs: If an ITM sends a message on an output tape named c, then only (a copy of) an ITM with an input tape named c can receive this message. We will use input tapes with attribute enriching (enriching input tapes, for short) to bound both the length of the output that may be produced by an ITM and the size of its current configuration.

Tapes named start and decision will serve a particular purpose. We require that only input tapes can be named start and only output tapes can be named decision. We will later use start to provide a system with external input and to trigger an ITM if no other ITM was triggered. An ITM is triggered by another ITM if the latter sends a message to the former. An ITM with an input tape named start will be called *master ITM*. On tapes named decision the final output of a system of ITMs will be written.

An ITM M runs in one of two modes, CheckAddress or Compute. The mode in which M is supposed to run is written on the mode tape of M.

Computation. We describe the computation of an ITM M in mode CheckAddress and Compute, respectively. Informally speaking, in mode CheckAddress an ITM M checks whether the incoming message is in fact addressed to it. Typically, this mode is used for the following purpose: In a system of ITMs there may be several copies of M (belonging to different parties in a multi-party protocol or to different copies of a protocol). To address the different copies one can prefix messages with identifiers (for example, session identifiers (SIDs) and/or party

identifiers (PIDs)). Now, in mode CheckAddress, M checks whether the incoming message is prefixed with the expected identifier, and either accepts or rejects that message. This allows to establish an unbounded number of *virtual channels* between ITMs. In mode Compute, the ITM actually processes an incoming message and possibly writes output on one of the output tapes, i.e., sends a message to another ITM. More formally, the computation in the two modes is defined as follows:

Mode CheckAddress: When M is activated in mode CheckAddress, it is the case that CheckAddress is written on the mode tape of M, the security parameter η is written on the security parameter tape, and one message, say m, is written on one of the input tapes, say c (the other input tapes and the output tapes are empty—or otherwise will be emptied before M starts to run—and the content on the work tapes represent the current configuration of M). We require that (i) at the end of the activation M has written accept or reject on the address decision tape; accordingly, we write $M(\mathsf{CheckAddress}, \eta, c, m) = \mathsf{accept}$ and $M(\mathsf{CheckAddress}, \eta, c, m) = \mathsf{reject}$, respectively, (ii) the computation performed by M in this mode is deterministic, i.e., is independent of internal coin tosses, and (iii) the number of transitions taken in the activation is bounded by $q(n)$ where q is the polynomial associated with M and n is the security parameter plus the length of the content of the input and work tapes at the beginning of the activation.

Mode Compute: To specify the computation in mode Compute, let l denote the length of the security parameter plus the accumulated length of all inputs written on *enriching* input tapes of M in mode Compute so far (i.e., the sum of the lengths of inputs written on enriching input tapes in the current activation in mode Compute and all previous activations in mode Compute).

When M is activated in mode Compute, it is the case that Compute is written on the mode tape of M, the security parameter η is written on the security parameter tape, and one message, say m, is written on one of the input tapes, say c (the other input tapes and the output tapes are empty—or otherwise will be emptied before M starts to run—and the content on the work tapes represent the current configuration of M). We require that the computation in every activation of M satisfies the following conditions: (i) Similar to other models [4, 7, 10], at the end of the activation, M has written *at most one* message on one of its output tapes (i.e., only one message can be sent to another ITM at a time), (ii) the number of transitions taken in the activation is bounded by $q(n)$ where q is the polynomial associated with M and n is the security parameter tape plus the length of the content of the input and work tapes at the beginning of the activation, (iii) the sum of the lengths of all outputs written on output tapes so far by M (in all activations) is bounded by $q(l)$, (iv) at the end of the current activation, the length of the content of the work tapes is bounded by $q(l)$.

We emphasize that in mode CheckAddress and Compute, M can not be exhausted: Whenever M is activated in one of the two modes, M is able to "scan" its complete current configuration, including the incoming message. Requirements

(iii) and (iv) in mode Compute bound the length of the output that can be produced and the size of the internal configuration. This will be used to guarantee that a system of ITMs runs in polynomial time. Note that the bounds in (iii) and (iv) may depend on the length of the input given to the machine on enriching input tapes. When modeling protocols, this will enable parties to produce output which is only polynomially bounded in the security parameter plus the length of the input received on the I/O interface (see Section 4 and 7), such as messages to be encrypted, signed, or transmitted.

Of course, inexhaustible ITMs can simulate exhaustible ITMs (which might be useful for modeling denial-of-service attacks) since inexhaustible ITMs can count the number of steps performed so far and halt if a certain bound has been reached.

2.2 Systems of ITMs

A *system* of ITMs can be built according to the following grammar where M ranges over (descriptions of) ITMs:

$$\mathcal{S} ::= M \;\mid\; (\mathcal{S} \,||\, \mathcal{S}) \;\mid\; !\,\mathcal{S}.$$

We require that the set of names of input tapes of different occurrences of ITMs in a system $\mathcal{S}$ are disjoint. This implies that in $\mathcal{S}$ only at most one ITM may be a master ITM, i.e., may have start as input tape. For example, if $\mathcal{S} = M_1 \,||\, M_2 \,||\, !\, M_3$, then the above restriction says that $\mathcal{T}_{in}(M_i) \cap \mathcal{T}_{in}(M_j) = \emptyset$ for every $i \neq j$.

Intuitively, $\mathcal{S}_1 \,||\, \mathcal{S}_2$ stands for the parallel composition of the systems $\mathcal{S}_1$ and $\mathcal{S}_2$, and $!\,\mathcal{S}$ stands for the parallel composition of an unbounded number of copies of (machines in) the system $\mathcal{S}$, where the actual number of copies is determined by the environment, i.e., external or internal machines invoking (machines of) $\mathcal{S}$. Following the common terminology of process calculus [13, 17], we call '!' the *bang operator*.

We say that an ITM M occurs in the scope of a bang in $\mathcal{S}$ if $\mathcal{S}$ contains a subexpression of the form $!\,\mathcal{S}'$ such that M occurs in $\mathcal{S}'$. It will be clear from the semantics of systems, i.e., the way a system of ITMs runs, that every system $\mathcal{S}$ can equivalently be written as $\mathcal{S} = M_1 \,||\, \cdots \,||\, M_k \,||\, !\, M'_1 \,||\, \cdots \,||\, !\, M'_{k'}$ where the M_i's and M'_i's are ITMs.

We will mainly be concerned with what we call well-formed systems. These systems are guaranteed to run in polynomial time (see Section 2.3) and they satisfy a certain acyclicity condition in the way ITMs are connected via enriching tapes. To define well-formed systems, we associate a graph with a system.

A system $\mathcal{S}$ induces a graph $G_{\mathcal{S}}$ which is defined as follows: The nodes of $G_{\mathcal{S}}$ are the ITMs occurring in $\mathcal{S}$. If M_1 and M_2 are two nodes in $G_{\mathcal{S}}$, then there is an edge from M_1 to M_2 in $G_{\mathcal{S}}$ if M_1 has an output tape c and M_2 has an *enriching* input tape c. For example, the graph $G_{\mathcal{S}}$ of $\mathcal{S} = M_1 \,||\, M_2 \,||\, !\, M_3$ has three nodes, M_1, M_2, M_3, and there is an edge from M_i to M_j if M_i has an output tape c and M_j an enriching input tape c.

Definition 1. *We call a system $\mathcal{S}$* well-formed *if $G_{\mathcal{S}}$ is acyclic and the master ITM (if any) occurring in $\mathcal{S}$ is not in the scope of a bang.*

2.3 Running a System

We now define how a system $\mathcal{S}$ runs given a security parameter η and a bit string a as external input. We denote such systems by $\mathcal{S}(1^\eta, a)$. More details and proofs can be found in Appendix A.

Informally speaking, in a run of $\mathcal{S}(1^\eta, a)$ at every time only one ITM is active and all other ITMs wait for new input. The active machine may write at most one message on one of its output tapes, say c. This message is then delivered to another ITM (which has an input tape named c). The previously active machine goes into a wait state and the receiver of the message is activated, resulting, after some internal computation, into a new output which is sent to another ITM, and so on. The first ITM to be activated in a run is the master ITM. It gets a as external input (on tape start). A run stops if the master ITM, after being activated, does not produce output or output was written on an output tape named $\mathsf{decision}$. If a message is sent on an output tape c but no previously activated ITM is willing to accept the message, then a new ITM (with input tape named c) might be created. If this is not possible, the master ITM will be triggered. More formally, a run of $\mathcal{S}(1^\eta, a)$ is defined as follows:

The current (global) configuration of a system in a run is described by a tuple (A, P) where A is a sequence of configurations of ITMs, the sequence of *(previously) activated machines*, and P is a system. The ITMs occurring in P are called *passive*. (In what follows, we often do not distinguish between an ITM and its current configuration.) We emphasize that the machines in A are not the ones that are currently active, i.e., currently performing some computation—only one of these machines was just active. The machines in A are rather those machines that were active at some point in the run so far. If a message is output without a machine in A willing to accept this message (this is tested by running the machines in mode $\mathsf{CheckAddress}$ starting with the first machine in A), it is tested if there is an ITM in P that would accept the message. If so, this machine will be copied from P to A. Also, it will be removed from P if it is not in the scope of a bang, and otherwise, it will stay in P. The intuition is that there is an unbounded supply of those ITMs in P that are in the scope of an ITM.

A run ρ of a system $\mathcal{S}(1^\eta, a)$ is a sequence of tuples of the form (A, P). The initial configuration is (A_0, P_0) where A_0 is the empty sequence—no machine has been activated yet—, and $P_0 = \mathcal{S}$. Roughly speaking, one gets from one configuration (A, P) to the next configuration (A', P') by one machine (either among A or a new machine obtained from P) reading a message from its input tape, thereby updating its current configuration, and possibly writing a message on one of its output tapes (which is then read in the next step by another ITM).

The first step in a run is to read the external input a, which is provided on tape start. Since initially A_0 is empty, it is checked whether P_0 contains a master ITM, i.e., an ITM with input tape start (recall that the master ITM is uniquely determined in $\mathcal{S}$). If this is not the case, the run stops. Otherwise, if

there is a master ITM, say M, and M accepts a in mode CheckAddress, i.e., $M(\mathsf{CheckAddress}, \eta, \mathsf{start}, a) = \mathsf{accept}$, then a (written on start) is processed by M in mode Compute and M (more precisely, the current configuration of M) is moved to A_0, and removed from P_0 if it is not in the scope of a bang. If M did not produce output, the run stops. If M produced output, say m was written on tape c, then in the next step this output is read by another ITM yielding a successor configuration.

More precisely, to define a successor configuration of a configuration, assume that the current configuration is (A, P) where A is the sequence of configurations $M_1, \ldots, M_n$ and P is some system, and that in the previous step the message m was written by some machine on an output tape c. (As explained above, after the first step of the run, we have that $A = M_1$ where M_1 is the current configuration of the master ITM and P coincides with P_0, except that possibly the master ITM might have been removed depending on whether it was in the scope of a bang in P.). We now describe how the successor configuration of (A, P) is obtained. We distinguish three cases:

1. There exists a machine M_i in A (where i is chosen to be minimal) with an input tape named c which accepts m on c, i.e., $M_i(\mathsf{CheckAddress}, \eta, c, m) = \mathsf{accept}$. Then, M_i is activated in mode Compute to process the input m on tape c. Now, A is updated with the new configuration of M_i (note that in this new configuration a new output message may be written on one of the output tapes); P remains unchanged.
2. No machine in A with an input tape named c accepts m on tape c. But there is a passive machine M (in P) with an input tape named c such that c is an *enriching* tape of M and M accepts m on tape c. We activate M in mode Compute to process m on tape c and add the new configuration of M at the end of A. (This new configuration may contain a new output message on one of the output tapes.) If M is not in the scope of a bang, then M is removed from P.
3. If neither 1. nor 2. is satisfied, the configuration does not change.

If in one step no output is produced (in 1. by M_i, in 2. by M), then in the next step the empty input ε is read from start, i.e., the master ITM is triggered.

A run immediately stops if the master ITM after being activated has not produced output or some machine wrote output on a tape named decision—this output is the overall output of the system.

We emphasize that a copy of an ITM can only be generated if a message is sent to an ITM via its enriching input tape (see 2. above). For simulation-based security, this requirement is not an essential restriction, but it is important to guarantee that well-formed systems run in polynomial time (see below).

Definition 2. *Let p be a polynomial p and ρ be a run of $\mathcal{S}(1^\eta, a)$. Then, ρ is* p-bounded *if the accumulated length of all outputs written on output tapes during the run is $\leq p(\eta + |a|)$. A system $\mathcal{S}$ is* p-bounded *if for all security parameters η and external inputs a all runs of $\mathcal{S}(1^\eta, a)$ are p-bounded. A system $\mathcal{S}$ is* (polynomially) bounded *if there exists a polynomial p such that $\mathcal{S}$ is p-bounded.*

We can prove that the length of every run of a bounded system $\mathcal{S}(1^\eta, a)$, the number of activated ITMs in such a run, the size of the configurations in a run, and the overall number of transitions taken by ITMs in a run of $\mathcal{S}(1^\eta, a)$ can be bounded by a polynomial in $\eta + |a|$. As a result one obtains:

Proposition 1. *Every bounded system can be simulated by a single ITM.*

We call a system $\mathcal{S}$ *almost p-bounded* if the probability Prob[run of $\mathcal{S}(1^\eta, a)$ is not p-bounded] as a function of η and a is negligible. For such systems, Proposition 1 also holds, except that a simulated run may deviate from a run in the original system with negligible probability.

We note that not all systems are (almost) bounded. For example, consider the system $\mathcal{S} = M_1 \,||\, M_2$ where M_1 and M_2 are connected via enriching tapes in both directions and one of the two machines is the master ITM. Then, M_1 and M_2 could send messages back and forth forever. Another example is the system $\mathcal{S} =! M$ where M is a master ITM. If M in mode $\mathsf{CheckAddress}$ only accepts a message in its first activation and in mode $\mathsf{Compute}$ always produces some fixed output, then after every activation of M a new copy of M will be generated and the run of the system does not terminate.

Note that the systems in the examples are not well-formed. We can prove:

Theorem 1. *Well-formed systems are bounded.*

We note that if new ITMs could be generated when invoked not only via enriching but also via consuming input tapes, then well-formed systems would not necessarily be bounded. Consider the following example:

Example 1. Let $\mathcal{S} = M_1 \,||\, ! M_2$ where M_1 is the master ITM (i.e., it has an input tape start) and has an enriching input tape c, M_2 has an output tape with the same name, M_2 has a consuming input tape c', and M_1 has an output tape with the same name. In mode $\mathsf{CheckAddress}$ M_1 accepts every message and in mode $\mathsf{Compute}$ M_1 writes a bit on output tape c'. In mode $\mathsf{CheckAddress}$, M_2 only accepts a message if it is activated for the first time. In mode $\mathsf{Compute}$, it outputs 1^η where η is the security parameter on output tape c. Now, in a run of $\mathcal{S}(\eta, a)$ M_1 keeps sending messages to copies of M_2 (on tape c'). Whenever M_1 has produced output, this output is sent to a new copy of M_2. This copy of M_2 outputs 1^η on c, and hence, since c is an enriching input tape of M_1, allows M_1 to produce more output. Consequently, runs of $\mathcal{S}(\eta, a)$ do not terminate and in these runs an unbounded number of copies of M_2 are generated.

If, however, we restricted ourselves to the simpler case where the number of copies of ITMs is bounded by a fixed polynomial in the security parameter (rather than determined by invoking machines), Theorem 1 would still hold. All other results (with appropriate reformulations) proved in this paper would also carry over to this simpler setting.

For bounded systems $\mathcal{S}$, we denote by

$$\mathsf{Prob}[\mathcal{S}(1^\eta, a) \rightsquigarrow 1]$$

the probability that a run of $\mathcal{S}(1^\eta, a)$ returns 1, i.e., 1 is written on $\mathsf{decision}$.

This definition can be extended to almost bounded systems. Basically, one only considers bounded runs and ignores all others: By definition, if $\mathcal{S}$ is almost bounded, there exists a polynomial p such that $\mathcal{S}$ is almost p-bounded. Now, we denote by $\mathsf{Prob}[\mathcal{S}(1^\eta, a) \rightsquigarrow 1]$ the probability that a run of $\mathcal{S}(1^\eta, a)$ is p-bounded *and* that it returns 1, i.e., we ignore runs that are not p-bounded. Strictly speaking, $\mathsf{Prob}[\mathcal{S}(1^\eta, a) \rightsquigarrow 1]$ depends on the specific p that is used. However, we will only consider the asymptotic behavior of a system, and therefore, the specific choice of the polynomial p does not matter as long as $\mathcal{S}$ is almost p-bounded.

2.4 Further Notation and Terminology

To state properties of systems and to apply our general computational model to simulation-based security, we now introduce some more notation and terminology.

Let $\mathcal{S}$ be a system and M be an ITM. Recall that $\mathcal{T}(M)$, $\mathcal{T}_{in}(M)$, and $\mathcal{T}_{out}(M)$ denote the set of (names of) input and output tapes, the set (of names) of input tapes, and the set (of names) of output tapes of M, respectively.

A tape c in $\mathcal{T}(\mathcal{S})$ is called *internal* if there exist two ITMs in $\mathcal{S}$, say M and M', such that $c \in \mathcal{T}_{out}(M) \cap \mathcal{T}_{in}(M')$. Otherwise, c is called *external*. The set of *internal tapes* of $\mathcal{S}$ is denoted by $\mathcal{T}_{int}(\mathcal{S})$ and the set of *external tapes* of $\mathcal{S}$ by $\mathcal{T}_{ext}(\mathcal{S})$. We call c an *(external) input tape* of $\mathcal{S}$ if $c \in \mathcal{T}_{ext}(\mathcal{S})$ and $c \in \mathcal{T}_{in}(M)$ for some ITM M in $\mathcal{S}$. Analogously, c is called an *(external) output tape* of $\mathcal{S}$ if $c \in \mathcal{T}_{ext}(\mathcal{S})$ and $c \in \mathcal{T}_{out}(M)$ for some ITM M in $\mathcal{S}$. The set of (external) input and output tapes of $\mathcal{S}$ is denoted by $\mathcal{T}_{in}(\mathcal{S})$ and $\mathcal{T}_{out}(\mathcal{S})$, respectively.

The set of external tapes of $\mathcal{S}$ is further partitioned into *network* and *I/O tapes*. We denote the set of external network tapes of $\mathcal{S}$ by $\mathcal{T}^{net}_{ext}(\mathcal{S})$ and the set of external I/O tapes of $\mathcal{S}$ by $\mathcal{T}^{io}_{ext}(\mathcal{S})$. Each of these sets is also partitioned into input and output tapes. We denote by $\mathcal{T}^{net}_{in}(\mathcal{S})$, $\mathcal{T}^{net}_{out}(\mathcal{S})$, $\mathcal{T}^{io}_{in}(\mathcal{S})$, and $\mathcal{T}^{io}_{out}(\mathcal{S})$ the set of network input and output tapes and the set of I/O input and output tapes, respectively.

Note that for every $\mathcal{S}$ we have that $\mathsf{start} \in \mathcal{T}(\mathcal{S})$ implies $\mathsf{start} \in \mathcal{T}_{in}(\mathcal{S})$ and $\mathsf{decision} \in \mathcal{T}(\mathcal{S})$ implies $\mathsf{decision} \in \mathcal{T}_{out}(\mathcal{S})$.

Given two systems $\mathcal{P}$ and $\mathcal{Q}$, by

$$\mathcal{P} \,|\, \mathcal{Q}$$

we denote the parallel composition $\mathcal{P}' \,||\, \mathcal{Q}'$ where $\mathcal{P}'$ and $\mathcal{Q}'$ are obtained from $\mathcal{P}$ and $\mathcal{Q}$ by renaming the internal tapes of $\mathcal{P}$ and $\mathcal{Q}$, respectively, such that $\mathcal{T}(\mathcal{P}') \cap \mathcal{T}_{int}(\mathcal{Q}') = \emptyset$ and $\mathcal{T}_{int}(\mathcal{P}') \cap \mathcal{T}(\mathcal{Q}') = \emptyset$. The intuition is that $\mathcal{P}$ and $\mathcal{Q}$ are different systems (e.g., a protocol and its environment) which communicate via their external tapes; they should not interfere on their internal tapes.

Two systems $\mathcal{P}$ and $\mathcal{Q}$ are *compatible* iff $\mathcal{T}^{net}_{in}(\mathcal{P}) = \mathcal{T}^{net}_{in}(\mathcal{Q})$, $\mathcal{T}^{net}_{out}(\mathcal{P}) = \mathcal{T}^{net}_{out}(\mathcal{Q})$, $\mathcal{T}^{io}_{in}(\mathcal{P}) = \mathcal{T}^{io}_{in}(\mathcal{Q})$, and $\mathcal{T}^{io}_{out}(\mathcal{P}) = \mathcal{T}^{io}_{out}(\mathcal{Q})$, i.e., $\mathcal{P}$ and $\mathcal{Q}$ coincide on their external tapes for every type of external tapes.

The systems $\mathcal{P}$ and $\mathcal{Q}$ are *I/O-compatible* if they have the same set of I/O tapes and disjoint sets of network tapes, i.e., $\mathcal{T}^{net}_{ext}(\mathcal{P}) \cap \mathcal{T}^{net}_{ext}(\mathcal{Q}) = \emptyset$, $\mathcal{T}^{io}_{in}(\mathcal{P}) = \mathcal{T}^{io}_{in}(\mathcal{Q})$, and $\mathcal{T}^{io}_{out}(\mathcal{P}) = \mathcal{T}^{io}_{out}(\mathcal{Q})$.

A system $\mathcal{Q}$ is *connectible* for $\mathcal{P}$ if each common external tape of $\mathcal{P}$ and $\mathcal{Q}$ has the same type in both (network or I/O) and complementary directions (input or output), i.e., for all $c \in \mathcal{T}_{ext}(\mathcal{P}) \cap \mathcal{T}_{ext}(Q)$, we have that $c \in \mathcal{T}^{net}_{ext}(\mathcal{P}) \cap \mathcal{T}^{net}_{ext}(\mathcal{Q})$ or $c \in \mathcal{T}^{io}_{ext}(\mathcal{P}) \cap \mathcal{T}^{io}_{ext}(\mathcal{Q})$, and $c \in \mathcal{T}_{in}(\mathcal{P}) \cap \mathcal{T}_{out}(\mathcal{Q})$ or $c \in \mathcal{T}_{out}(\mathcal{P}) \cap \mathcal{T}_{in}(\mathcal{Q})$. Note that this connectability relation is symmetric. Given a set $\mathbf{B}$ of systems, we denote by $\mathrm{Con}_{\mathbf{B}}(\mathcal{Q})$ the set of all systems $\mathcal{P}$ in $\mathbf{B}$ such that $\mathcal{P}$ is connectible for $\mathcal{Q}$.

A system $\mathcal{A}$ is *adversarially connectible* for $\mathcal{P}$ if $\mathcal{A}$ is connectible for $\mathcal{P}$ and the set of external tapes of $\mathcal{A}$ is disjoint from the set of I/O tapes of $\mathcal{P}$. Thus, an adversary can only connect on the network tapes of a protocol. Given a set $\mathbf{B}$ of systems, we denote by $\mathrm{Adv}_{\mathbf{B}}(\mathcal{P})$ the set of all systems $\mathcal{A}$ in $\mathbf{B}$ such that $\mathcal{A}$ is adversarially connectible for $\mathcal{P}$. With $\mathrm{Sim}^{\mathcal{P}}_{\mathbf{B}}(\mathcal{F})$ we denote the set of all systems $\mathcal{S}$ in $\mathbf{B}$ such that $\mathcal{S}$ is adversarially connectible for $\mathcal{F}$ and $\mathcal{S} \,|\, \mathcal{F}$ is compatible with $\mathcal{P}$.

We call $\mathcal{E}$ an *environmental (environmentally connectible) system* for $\mathcal{P}$ if $\mathcal{E}$ is connectible for $\mathcal{P}$ and $\mathcal{T}_{ext}(\mathcal{E}) \cap \mathcal{T}^{net}_{ext}(\mathcal{P}) = \emptyset$. In other words, an environmental system only connects on the I/O tapes of $\mathcal{P}$. Given a set $\mathbf{B}$ of systems, we denote by $\mathrm{Env}_{\mathbf{B}}(\mathcal{P})$ the set of all systems $\mathcal{E}$ in $\mathbf{B}$ such that $\mathcal{E}$ is environmentally connectible for $\mathcal{P}$.

Definition 3. *Two almost bounded systems $\mathcal{P}$ and $\mathcal{Q}$ are called* equivalent *or* indistinguishable *($\mathcal{P} \equiv \mathcal{Q}$) iff the function*

$$f(1^\eta, a) = |\mathit{Prob}[\mathcal{P}(1^\eta, a) \rightsquigarrow 1] - \mathit{Prob}[\mathcal{Q}(1^\eta, a) \rightsquigarrow 1]|$$

is negligible (in the sense defined at the end of Section 1).

We will later consider what we call a *dummy ITM* $\mathcal{D}$ which simply forwards messages between entities: The dummy ITM has for all of its input tapes a corresponding output tape and all input tapes are enriching. The concrete set of input and output tapes that $\mathcal{D}$ has depends on the entities between which $\mathcal{D}$ is put. The dummy ITM accepts all messages on input tapes in mode CheckAddress and in mode Compute it simply copies a message received on an input tape to the corresponding output tape. Note that this is possible since all input tapes are enriching. We also emphasize that, except for the set of input and output tapes, $\mathcal{D}$ does not depend on the entities between which it is put.

More precisely, let $\mathcal{T}_{in}$ and $\mathcal{T}_{out}$ be disjoint finite sets of tapes. Moreover, let $\mathcal{T}'_{in} = \{c' \mid c \in \mathcal{T}_{in}\}$ and $\mathcal{T}'_{out} = \{c' \mid c \in \mathcal{T}_{out}\}$ where c' is a new copy of c, i.e., a new tape with a new name.

We define

$$\mathcal{D} = \mathcal{D}(\mathcal{T}_{in}, \mathcal{T}_{out})$$

to be an ITM with input tapes $\mathcal{T}_{out} \cup \mathcal{T}'_{in}$ and output tapes $\mathcal{T}_{in} \cup \mathcal{T}'_{out}$. Every input tape of $\mathcal{D}$ is declared to be enriching. In mode CheckAddress, $\mathcal{D}$ always

accepts. In mode Compute, $\mathcal{D}$ copies every message received on $c \in \mathcal{T}_{out}$ onto $c' \in \mathcal{T}'_{out}$ and every message received on $c' \in \mathcal{T}'_{in}$ onto $c \in \mathcal{T}_{in}$.

By

$$\mathcal{D}^{net} = \mathcal{D}^{net}(\mathcal{T}_{in}, \mathcal{T}_{out})$$

we denote the version of $\mathcal{D}$ where all input and output tapes are considered network tapes.

By

$$\mathcal{D}^{io} = \mathcal{D}^{io}(\mathcal{T}_{in}, \mathcal{T}_{out})$$

we denote the version of $\mathcal{D}$ where all tapes c' are declared to be I/O tapes and all tapes c are declared to be network tapes.

3 Properties of Systems

In this section, we summarize some useful properties of systems.

The following lemma, which easily follows from the definition of systems, says that consistently changing the names of tapes or their type (network or I/O) in a system does not change the behavior of the system.

Lemma 1. *Let $\mathcal{S}_1, \ldots, \mathcal{S}_k$ be systems such that $\mathcal{S}_i$ is connectible for $\mathcal{S}_{i+1} \mid \cdots \mid \mathcal{S}_k$ for every i. Then,*

$$\mathcal{S}_1 \mid \cdots \mid \mathcal{S}_k \equiv \mathcal{S}'_1 \mid \cdots \mid \mathcal{S}'_k$$

where $\mathcal{S}'_i$ is derived from $\mathcal{S}_i$ by consistently (w.r.t. the other $\mathcal{S}'_j$) renaming external tapes (start and decision may not be renamed) and possibly declaring some network tapes to be I/O tapes and vice versa.

The next lemma says that the dummy ITM can be plugged between two systems without changing the behavior of the overall system. In particular, using this dummy ITM the FORWARDER property mentioned in the introduction can be satisfied.

Lemma 2. *Let $\mathcal{P}$ and $\mathcal{Q}$ be two systems such that $\mathcal{P}$ is connectible for $\mathcal{Q}$ and $\mathcal{P} \mid \mathcal{Q}$ is (almost) bounded. Let $\mathcal{T}_{ext} = \mathcal{T}_{ext}(\mathcal{P}) \cap \mathcal{T}_{ext}(\mathcal{Q})$, $\mathcal{D} = \mathcal{D}(\mathcal{T}_{ext} \cap \mathcal{T}_{in}(\mathcal{P}), \mathcal{T}_{ext} \cap \mathcal{T}_{out}(\mathcal{P}))$, and $\mathcal{Q}'$ be obtained from $\mathcal{Q}$ by renaming all tapes c in $\mathcal{T}_{ext}$ by c'. Then, we have that the system $\mathcal{P} \mid \mathcal{D} \mid \mathcal{Q}$ is (almost) bounded and*

$$\mathcal{P} \mid \mathcal{Q} \equiv \mathcal{P} \mid \mathcal{D} \mid \mathcal{Q}'.$$

We now show that every well-formed system within a more complex system can be replaced by a single ITM. This is the core of the joint state theorem as stated in [9, 7]. In what follows, we say that an input tape c is enriching in a system $\mathcal{Q}$ if there is an ITM M in $\mathcal{Q}$ such that c is an enriching input tape of M.

Lemma 3. *Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be well-formed systems such that $\mathcal{Q}_1$ is connectible for $\mathcal{Q}_2$ and $\mathcal{Q}_1 \mid \mathcal{Q}_2$ is (almost) bounded. Then, there exists an ITM M compatible with $\mathcal{Q}_2$ such that a tape c of M is enriching iff c is enriching in $\mathcal{Q}_2$ and*

$$\mathcal{Q}_1 \mid \mathcal{Q}_2 \equiv \mathcal{Q}_1 \mid M.$$

Moreover, the construction of M only depends on $\mathcal{Q}_2$ and in mode CheckAddress *M accepts every message.*

PROOF. Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be given as stated in the lemma. For the time being, we assume that $\mathcal{Q}_2$ does not contain a master ITM.

To show that there exists an ITM M as required, we first prove that there exists a polynomial p such that for every η and a, and at every time in a run of $(\mathcal{Q}_1 \mid \mathcal{Q}_2)(\eta, a)$ the following is true: The number of copies of ITMs of $\mathcal{Q}_2$ and the size of the configurations of these copies (i.e., the length of the contents of the tapes of these copies) is $\leq p(\eta + l)$ where l is the length of the input that has been written on enriching input tapes of the copies of ITMs of $\mathcal{Q}_2$ so far. The proof is similar to the one of Theorem 1.

Let $M_1, \ldots, M_n$ be the ITMs occurring in $\mathcal{Q}_2$. Since $\mathcal{Q}_2$ is well-formed, we know that $G_{\mathcal{Q}_2}$ is acyclic. Hence, as in the proof of Theorem 1, we can conclude that there exists a total ordering $<$ on $M_1, \ldots, M_n$ which is consistent with $G_{\mathcal{Q}_2}$, i.e., if there is an edge from M_i to M_j in $G_{\mathcal{Q}_2}$, then $M_i < M_j$. W.l.o.g. we may assume that $M_1 < M_2 < \cdots < M_n$. By definition of $G_{\mathcal{Q}_2}$, (a copy of) M_i can only be invoked via an enriching input tape by (a copy of) M_j for $j < i$ or some external ITM, i.e., an ITM of $\mathcal{Q}_1$. In particular, only an ITM M_j, $j < i$, or an external ITM can generate a copy of M_i.

Consequently, M_1 can only be invoked by an external ITM. Hence, the number of copies of M_1 at the given time of the run, is bounded by l, with l defined as above. By definition of ITMs, the output produced by M_1 so far is bounded by a polynomial in $\eta + l$. In particular, the number of copies of $M_2, \ldots, M_n$ generate by M_1 is bounded by a polynomial in $\eta + l$ as well. It follows that the number of copies of M_2 is bounded by a polynomial in $\eta + l$ and the input given to these copies of M_2 via enriching input tapes is also bounded by a polynomial in $\eta + l$. Consequently, the output produced by copies of M_2 is polynomially bounded in $\eta + l$. Iterating this argument for the remaining ITMs, we obtain that the number of copies of ITMs occurring in $\mathcal{Q}_2$ and the input given to these copies on enriching input tapes can polynomially be bounded in $\eta + l$. (Note that n is a constant that does not depend on η or l.) By the definition of ITMs, this implies that the size of the configurations is bounded by a polynomial in $\eta + l$.

It follows that M can store all configurations of copies of ITMs of $\mathcal{Q}_2$ on its work tapes (without exceeding a polynomial bound). In mode CheckAddress, we define M to accept all incoming messages. In mode Compute, M first, as specified in Appendix A, simulates the ITMs in the configurations stored in mode CheckAddress to see whether one of these ITMs accepts the incoming message or whether a new copy of an ITM needs to be generated internally. If one ITM, say M', accepts (possibly the newly generated copy, if any), then M simulates M' in mode Compute with the given input and the corresponding configuration. Note that by definition of ITMs, both the simulation of the ITMs in mode CheckAddress and the simulation of the chosen ITM in mode Compute takes only polynomial time in the security parameter, the size of the configurations stored, and the given input, and hence, these simulations can be carried out by M.

The reasoning in case one of the M_i is a master ITM is similar. As in Theorem 1, we use that, since $\mathcal{Q}_2$ is well-formed, the master ITM is not in the scope of a bang, and hence, every run of $\mathcal{Q}_1 \mid \mathcal{Q}_2(\eta, a)$ will contain at most one copy of the master ITM. □

The following lemma shows how the parallel composition of systems can be combined into one system with only consuming external tapes. This is used for moving entities (such as adversarial systems) into an environmental system.

Lemma 4. *Let $\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3$ be systems such that $\mathsf{start} \notin \mathcal{T}(\mathcal{Q}_3)$, $\mathcal{Q}_2$ is connectible for $\mathcal{Q}_3$, $\mathcal{Q}_1$ is connectible for $\mathcal{Q}_2 \mid \mathcal{Q}_3$, and $\mathcal{Q}_1 \mid \mathcal{Q}_2 \mid \mathcal{Q}_3$ is well-formed. Then there exists a system $\mathcal{Q}$ which satisfies the following conditions:*

1. *$\mathcal{Q}$ is compatible with $\mathcal{Q}_1 \mid \mathcal{Q}_2$.*
2. *All tapes in $\mathcal{T}_{in}(\mathcal{Q})$ are consuming, except for start, which may be enriching (if it occurs in $\mathcal{Q}$).*
3. *$\mathcal{Q}_1 \mid \mathcal{Q}_2 \mid \mathcal{Q}_3 \equiv \mathcal{Q} \mid \mathcal{Q}_3$.*

Proof. Let $\mathcal{S} = \mathcal{Q}_1 \mid \mathcal{Q}_2 \mid \mathcal{Q}_3$. If all tapes in $\mathcal{T}_{ext}(\mathcal{Q}_1 \mid \mathcal{Q}_2) \setminus \{\mathsf{start}\}$ are consuming, then we set $\mathcal{Q} = \mathcal{Q}_1 \mid \mathcal{Q}_2$. Obviously, $\mathcal{Q}$ satisfies all three conditions.

Otherwise, we define a single ITM $\mathcal{Q} = M$ which simulates $\mathcal{Q}_1 \mid \mathcal{Q}_2$. The input and output tapes of $\mathcal{Q}$ are the tapes in $\mathcal{T}_{in}(\mathcal{Q}_1 \mid \mathcal{Q}_2)$ and $\mathcal{T}_{out}(\mathcal{Q}_1 \mid \mathcal{Q}_2)$, respectively.

Before we specify how $\mathcal{Q}$ works, observe that by Lemma 3 we can assume that $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are single ITMs. By Theorem 1, we know that $\mathcal{S}$ is bounded, and hence, by Lemma 6, there exists a polynomial p such that the overall number of transitions taken by ITMs in any run of $\mathcal{S}(1^\eta, a)$ is bounded by $p(\eta, |a|)$.

We now define $\mathcal{Q}$ to simulate the system $\mathcal{Q}_1 \mid \mathcal{Q}_2$ as follows: If invoked in mode $\mathsf{CheckAddress}$, $\mathcal{Q}$ will simulate $\mathcal{Q}_1$ or $\mathcal{Q}_2$ in mode $\mathsf{CheckAddress}$ depending on whether a message was sent to $\mathcal{Q}_1$ or $\mathcal{Q}_2$. In mode $\mathsf{Compute}$, $\mathcal{Q}$ will simulate $\mathcal{Q}_1 \mid \mathcal{Q}_2$ where, however, not more than $p(\eta, |a|)$ transitions of ITMs in $\mathcal{Q}_1 \mid \mathcal{Q}_2$ are simulated. In case this bound is reached, the simulation stops and from that point on $\mathcal{Q}$ ignores all incoming messages, i.e., when invoked in mode $\mathsf{Compute}$ it does not produce any output. Note that in no run of $\mathcal{S}$ the bound will be reached.

It is now easy to see that $\mathcal{Q}$ satisfies the required conditions. □

We will denote $\mathcal{Q}$ as constructed in the proof of the above lemma by $[\mathcal{Q}_1 \mid \mathcal{Q}_2]_{\mathcal{Q}_3}$.

The next lemma will allow us to "open" $[\mathcal{Q}_1 \mid \mathcal{Q}_2]_{\mathcal{Q}_3}$, i.e., replace $[\mathcal{Q}_1 \mid \mathcal{Q}_2]_{\mathcal{Q}_3}$ by $\mathcal{Q}_1 \mid \mathcal{Q}_2$, in a context different from $\mathcal{Q}_3$.

Lemma 5. *Let $\mathcal{Q}_1$ and $\mathcal{Q}_2$ be two systems which do not contain a master ITM, are well-formed and compatible, and satisfy the following condition: $\mathcal{E} \mid \mathcal{Q}_1 \equiv \mathcal{E} \mid \mathcal{Q}_2$ for every $\mathcal{E} \in Con_{\mathbf{E}}(\mathcal{Q}_1)$. (Note that $\mathcal{E} \mid \mathcal{Q}_1$ and $\mathcal{E} \mid \mathcal{Q}_2$ are well-formed.) Then for every system $\mathcal{E}_1$ connectible for $\mathcal{Q}_1$ and every system $\mathcal{E}_2$ connectible for $\mathcal{E}_1 \mid \mathcal{Q}_1$ such that $\mathcal{E}_2 \mid \mathcal{E}_1 \mid \mathcal{Q}_1$ is well-formed, we have that $[\mathcal{E}_2 \mid \mathcal{E}_1]_{\mathcal{Q}_1} \mid \mathcal{Q}_2 \equiv \mathcal{E}_2 \mid \mathcal{E}_1 \mid \mathcal{Q}_2$ and $\mathcal{E}_2 \mid \mathcal{E}_1 \mid \mathcal{Q}_2$ is almost bounded.*

PROOF. First, recall that by definition, $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1}$ exactly simulates all transitions taken by ITMs in $\mathcal{E}_2 \,|\, \mathcal{E}_1$ up to a certain polynomial bound, where the polynomial is in the security parameter plus the length of the input on start. By construction of $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1}$ when running $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1} \,|\, \mathcal{Q}_1$, this bound is never reached. It follows that the probably that this bound is reached when running the system $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1} \,|\, \mathcal{Q}_2$ is negligible. Otherwise, one can easily construct $\mathcal{E}' \in \mathrm{Con}_{\mathbf{E}}(\mathcal{Q}_1)$ such that $\mathcal{E}' \,|\, \mathcal{Q}_1 \not\equiv \mathcal{E}' \,|\, \mathcal{Q}_2$, in contradiction to 2: $\mathcal{E}'$ simulates $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1}$ and outputs 1 iff the bound is not reached.

It follows that with overwhelming probability $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1}$ exactly simulates $\mathcal{E}_2 \,|\, \mathcal{E}_1$ in the system $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1} \,|\, \mathcal{Q}_2$. Thus, we obtain $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1} \,|\, \mathcal{Q}_2 \equiv \mathcal{E}_2 \,|\, \mathcal{E}_1 \,|\, \mathcal{Q}_2$, and since $[\mathcal{E}_2 \,|\, \mathcal{E}_1]_{\mathcal{Q}_1} \,|\, \mathcal{Q}_2$ is well-formed, and hence, bounded (Theorem 1), it follows that $\mathcal{E}_2 \,|\, \mathcal{E}_1 \,|\, \mathcal{Q}_2$ is almost bounded. □

4 Notions of Simulation-Based Security

In this section, we define the notions of simulation-based security mentioned in the introduction.

We first need to define protocol, adversarial, and environmental systems to specify the corresponding classes of entities. Here we define what we call IO-enriching protocol systems (or simply protocol systems) and IO-network-enriching adversarial systems (or simply adversarial systems). In this and the following two sections, we will study simulation-based security w.r.t. these classes of protocol and adversarial systems. In Section 7, different classes of protocol and adversarial systems will be considered. The definition of the environmental systems will stay the same in both settings.

An *(IO-enriching) protocol system* $\mathcal{P}$ is a well-formed system such that i) no tape in $\mathcal{P}$ is named start or decision, ii) all network tapes of $\mathcal{P}$ are consuming (I/O-tapes may be enriching), and iii) for every ITM M occurring in $\mathcal{P}$ such that M is not in the scope of a bang, we require that M accepts every incoming message in mode CheckAddress. We denote the set of protocol systems by $\mathbf{P}$. The motivation behind condition iii) is that if M does not occur in the scope of a bang, then in every run of $\mathcal{P}$ (in some environment) there will be at most one copy of M. Hence, there is no reason to address different copies of M, and therefore, in mode CheckAddress, M should accept every incoming message. This condition will be used in the proof of the composition theorem (Theorem 4 and Corollary 1).

An *(IO-network-enriching) adversarial system* $\mathcal{A}$ is a well-formed system such that no tape in $\mathcal{A}$ is named start or decision. We denote the set of adversarial systems by $\mathbf{A}$ or by $\mathbf{S}$. Note that we allow all external tapes of $\mathcal{A}$ to be enriching.

An *environmental system* $\mathcal{E}$ is a well-formed system such that all external tapes are consuming, except for start which may be enriching. We denote the set of environmental systems by $\mathbf{E}$. Note that $\mathcal{E}$ may contain start and decision. In particular, $\mathcal{E}$ may contain a master ITM (while protocol and adversarial systems may not). This choice is justified by results shown in [10] and corresponds to the choice made in other models (see, e.g., [5, 7]).

The security notions can now be defined in a concise and simple way. Note that from the definition of the different entities (in particular, the restrictions regarding what tapes may be enriching), it follows easily that all systems in the following definition, except for $\mathcal{E} \mid \mathcal{A} \mid \mathcal{S} \mid \mathcal{F}$, are well-formed, and hence, bounded.

Definition 4. *Let $\mathcal{P}$ and $\mathcal{F}$ be I/O-compatible protocol systems, the real and ideal protocol, respectively.*

Strong Simulatability (SS): $\mathcal{P} \leq^{SS} \mathcal{F}$ *iff* $\exists \mathcal{S} \in Sim^{\mathcal{P}}_{\mathbf{S}}(\mathcal{F})\ \forall \mathcal{E} \in Con_{\mathbf{E}}(\mathcal{P})$: $\mathcal{E} \mid \mathcal{P} \equiv \mathcal{E} \mid \mathcal{S} \mid \mathcal{F}$.

Black-box Simulatability (BB): $\mathcal{P} \leq^{BB} \mathcal{F}$ *iff* $\exists \mathcal{S} \in Sim^{\mathcal{P}}_{\mathbf{S}}(\mathcal{F})\ \forall \mathcal{A} \in Adv_{\mathbf{A}}(\mathcal{P})\ \forall \mathcal{E} \in Env_{\mathbf{E}}(\mathcal{A} \mid \mathcal{P})$: $\mathcal{E} \mid \mathcal{A} \mid \mathcal{P} \equiv \mathcal{E} \mid \mathcal{A} \mid \mathcal{S} \mid \mathcal{F}$ *and* $\mathcal{E} \mid \mathcal{A} \mid \mathcal{S} \mid \mathcal{F}$ *is almost bounded.*

Universal Simulatability/Composability (UC): $\mathcal{P} \leq^{UC} \mathcal{F}$ *iff* $\forall \mathcal{A} \in Adv_{\mathbf{A}}(\mathcal{P})\ \exists \mathcal{I} \in Sim^{\mathcal{A} \mid \mathcal{P}}_{\mathbf{S}}(\mathcal{F})\ \forall \mathcal{E} \in Env_{\mathbf{E}}(\mathcal{A} \mid \mathcal{P})$: $\mathcal{E} \mid \mathcal{A} \mid \mathcal{P} \equiv \mathcal{E} \mid \mathcal{I} \mid \mathcal{F}$.

Dummy Version of UC (UCdummy): $\mathcal{P} \leq^{UCdum} \mathcal{F}$ *iff* $\exists \mathcal{I} \in Sim^{\mathcal{D}^{io} \mid \mathcal{P}}_{\mathbf{S}}(\mathcal{F})\ \forall \mathcal{E} \in Env_{\mathbf{E}}(\mathcal{D}^{io} \mid \mathcal{P})$: $\mathcal{E} \mid \mathcal{D}^{io} \mid \mathcal{P} \equiv \mathcal{E} \mid \mathcal{I} \mid \mathcal{F}$ *where* $\mathcal{D}^{io} = \mathcal{D}^{io}(\mathcal{T}_{in}(\mathcal{P}), \mathcal{T}_{out}(\mathcal{P}))$.

Reactive Simulatability (RS): $\mathcal{P} \leq^{RS} \mathcal{F}$ *iff* $\forall \mathcal{A} \in Adv_{\mathbf{A}}(\mathcal{P})\ \forall \mathcal{E} \in Env_{\mathbf{E}}(\mathcal{A} \mid \mathcal{P})\ \exists \mathcal{I} \in Sim^{\mathcal{A} \mid \mathcal{P}}_{\mathbf{S}}(\mathcal{F})$: $\mathcal{E} \mid \mathcal{A} \mid \mathcal{P} \equiv \mathcal{E} \mid \mathcal{I} \mid \mathcal{F}$.

Using the property of dummy ITMs (Lemma 2), it is easy to see that all security notions introduced above are reflexive (modulo renaming of tapes), i.e., every protocol can be simulated by itself. Also, unlike in previous models, the above security notions do not exhibit the unintuitive properties anymore mentioned in the introduction: a real protocol in fact realizes almost identical ideal protocols. (Recall the example from the introduction where an ideal protocol coincides with the ideal protocol except that on the network interface the ideal protocol outputs the bit-wise complement of the messages the real protocol outputs.)

5 Relationships Between Notions of Simulation-Based Security

We study the relationships between the different security notions. In a nutshell, we have two classes of unconditionally equivalent notions: i) strong, black-box, and dummy universal simulatability, and ii) universal and reactive simulatability. All notions are equivalent if the ideal protocol is what we call generous.

To define generous protocols, we need the following notion: Given a security parameter η and a polynomial p, we say that a non-negative integer n is *polynomially at least as big as* a non-negative integer i w.r.t. η and p if $p(\eta + n) \geq i$.

Now, roughly speaking, an (ideal) protocol is generous if the length of the output it writes on network tapes is polynomially at least as big as the length of the input it receives on I/O tapes. In other words, a generous protocol gives at least as much computation power to a simulator as it receives on its I/O interface. If this property is not satisfied for a given ideal protocol, it is often possible to have the ideal protocol output dummy messages without changing the desired security properties of the protocol (see, e.g., the functionality for

signatures in [6]). If the ideal protocols are formulated in a "non-interactive way", i.e., the simulator hardly interacts with the functionality (see, e.g., the new formulation of signatures and encryption in [7]), then in order to make these ideal protocols generous one could, for example, modify the ideal protocol in such a way that it initially expects to receive the overall length of messages it is supposed to handle on the I/O interface (per party) and have the ideal protocol forward this information to the simulator in an initial phase. Such an ideal protocol would still be more flexible than an ideal protocol with an a priori bound on the number and length of messages it can handle.

Formally, generous protocols are defined as follows:

Definition 5. *We call a protocol system* $\mathcal{F}$ generous *if there exists a polynomial* p *such that for every* $\mathcal{E} \in Con_{\mathbf{E}}(\mathcal{F})$, η, a, *and in every run of* $(\mathcal{E} \,|\, \mathcal{F})(\eta, a)$, *whenever* $\mathcal{F}$ *sends a message on an external tape, then the lengths of the output written so far by* $\mathcal{F}$ *on network tapes is polynomially at least as big as the length of the input received so far by* $\mathcal{F}$ *on enriching I/O tapes w.r.t.* η *and* p.

The following theorem summarizes the relationships between the security notions.

Theorem 2. *Let* $\mathcal{P}$ *and* $\mathcal{F}$ *be I/O compatible protocol systems. We have that:*

1. $\mathcal{P} \leq^{SS} \mathcal{F}$ *iff* $\mathcal{P} \leq^{BB} \mathcal{F}$ *iff* $\mathcal{P} \leq^{UCdum} \mathcal{F}$.
2. $\mathcal{P} \leq^{UC} \mathcal{F}$ *iff* $\mathcal{P} \leq^{RS} \mathcal{F}$.
3. *If* $\mathcal{F}$ *is generous, then:* $\mathcal{P} \leq^{SS} \mathcal{F}$ *iff* $\mathcal{P} \leq^{BB} \mathcal{F}$ *iff* $\mathcal{P} \leq^{UCdum} \mathcal{F}$ *iff* $\mathcal{P} \leq^{UC} \mathcal{F}$ *iff* $\mathcal{P} \leq^{RS} \mathcal{F}$.

Most of the above equivalences can be proved by equational reasoning using the equational principles established in Section 3. A detailed proof of the above theorem can be found in Appendix B. We note that for the equivalence of universal and reactive simulatability we use that the environment gets auxiliary input, i.e., is non-uniform. As shown in [15], the two notions are not equivalent in the uniform case; this is also true if, in case of reactive simulatability, the auxiliary input provided to the environment is chosen before the ideal adversary.

6 Composition Theorems

We first state a composition theorem for composing a constant number of protocols and present the proof, which is based on the equational principles established in Section 5. We then extend this theorem to an unbounded number of copies of protocols.

Theorem 3. *Let* $\mathcal{P}_1, \ldots, \mathcal{P}_k, \mathcal{F}_1, \ldots, \mathcal{F}_k$ *be protocol systems such that* $\mathcal{P}_1 \,|\, \cdots \,|\, \mathcal{P}_k$ *and* $\mathcal{F}_1 \,|\, \cdots \,|\, \mathcal{F}_k$ *are well-formed and for every* j *the following conditions are satisfied:*

1. $\mathcal{P}_j$ *is environmentally connectible for* $\mathcal{P}_{j+1} \,|\, \cdots \,|\, \mathcal{P}_k$.
2. $\mathcal{F}_j$ *is environmentally connectible for* $\mathcal{F}_{j+1} \,|\, \cdots \,|\, \mathcal{F}_k$.

3. $\mathcal{P}_j$ *and* $\mathcal{F}_j$ *are* I/O*-compatible.*
4. $\mathcal{P}_j \leq^{SS} \mathcal{F}_j$.

Then, $\mathcal{P}_1 \,|\, \cdots \,|\, \mathcal{P}_k \leq^{SS} \mathcal{F}_1 \,|\, \cdots \,|\, \mathcal{F}_k$.

Proof. We prove the theorem for $k = 2$. For $k > 2$ the statement follows by induction on k.

Let $\mathcal{E} \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P}_1 \,|\, \mathcal{P}_2)$. Since $\mathcal{P}_1 \leq^{SS} \mathcal{F}_1$ and $\mathcal{P}_2 \leq^{SS} \mathcal{F}_2$ we have

1. $\exists\, \mathcal{S}_1 \in \mathrm{Sim}_{\mathbf{S}}^{\mathcal{P}_1}(\mathcal{F}_1)\ \forall\, \mathcal{E}' \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P}_1)$: $\mathcal{E}' \,|\, \mathcal{P}_1 \equiv \mathcal{E}' \,|\, \mathcal{S}_1 \,|\, \mathcal{F}_1$, and
2. $\exists\, \mathcal{S}_2 \in \mathrm{Sim}_{\mathbf{S}}^{\mathcal{P}_2}(\mathcal{F}_2)\ \forall\, \mathcal{E}' \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P}_2)$: $\mathcal{E}' \,|\, \mathcal{P}_2 \equiv \mathcal{E}' \,|\, \mathcal{S}_2 \,|\, \mathcal{F}_2$.

Define $\mathcal{S} = \mathcal{S}_1 \,|\, \mathcal{S}_2$. Because the set of network tapes of $\mathcal{S}_1$ and $\mathcal{S}_2$ are disjoint, it easily follows that $\mathcal{S}$ is well-formed; more precisely, $\mathcal{S} \in \mathrm{Sim}_{\mathbf{S}}^{\mathcal{P}_1 \,|\, \mathcal{P}_2}(\mathcal{F}_1 \,|\, \mathcal{F}_2)$. Now, we obtain

$$
\begin{aligned}
\mathcal{E} \,|\, \mathcal{P}_2 \,|\, \mathcal{P}_1 &\equiv [\mathcal{E} \,|\, \mathcal{P}_2]_{\mathcal{P}_1} \,|\, \mathcal{P}_1 && \text{(Lemma 4)} \\
&\equiv [\mathcal{E} \,|\, \mathcal{P}_2]_{\mathcal{P}_1} \,|\, \mathcal{S}_1 \,|\, \mathcal{F}_1 && \text{(1.)} \\
&\equiv \mathcal{E} \,|\, \mathcal{P}_2 \,|\, \mathcal{S}_1 \,|\, \mathcal{F}_1 && \text{(1., Lemma 5)} \\
&\equiv \mathcal{E} \,|\, \mathcal{S}_1 \,|\, \mathcal{F}_1 \,|\, \mathcal{P}_2 \\
&\equiv [\mathcal{E} \,|\, \mathcal{S}_1 \,|\, \mathcal{F}_1]_{\mathcal{P}_2} \,|\, \mathcal{P}_2 && \text{(Lemma 4)} \\
&\equiv [\mathcal{E} \,|\, \mathcal{S}_1 \,|\, \mathcal{F}_1]_{\mathcal{P}_2} \,|\, \mathcal{S}_2 \,|\, \mathcal{F}_2 && \text{(2.)} \\
&\equiv \mathcal{E} \,|\, \mathcal{S}_1 \,|\, \mathcal{F}_1 \,|\, \mathcal{S}_2 \,|\, \mathcal{F}_2 && \text{(2., Lemma 5)} \\
&\equiv \mathcal{E} \,|\, \mathcal{S}_1 \,|\, \mathcal{S}_2 \,|\, \mathcal{F}_1 \,|\, \mathcal{F}_2 \\
&\equiv \mathcal{E} \,|\, \mathcal{S} \,|\, \mathcal{F}_1 \,|\, \mathcal{F}_2 && \text{(Definition of } \mathcal{S}\text{)}
\end{aligned}
$$

□

Next we present a general composition theorem for composing a polynomial number of copies of protocols where the polynomial is determined by the environment. To address the different copies of a protocol, we use the mode CheckAddress of ITMs combined with session identifiers (SIDs)

More precisely, we turn a system $\mathcal{Q}$ into its session version $\underline{\mathcal{Q}}$, which allows us to address different copies of (ITMs occurring in) $\mathcal{Q}$ by a particular SID. We first define the session version of a single ITM.

The *session version* $\underline{M}$ of an ITM M simulates M except that all messages received have to be prefixed by a particular SID (i.e., in mode CheckAddress the ITM $\underline{M}$ will reject all messages not prefixed by the particular SID) and all messages sent out are prefixed by this SID. The SID $\underline{M}$ will use is the one with which $\underline{M}$ is first activated (hence, in the first activation, $\underline{M}$ will accept the incoming message in mode CheckAddress and then store the SID). More precisely, $\underline{M}$ behaves as follows in mode CheckAddress and Compute, respectively:

– When activated in mode CheckAddress, $\underline{M}$ does the following: If $\underline{M}$ has never been activated before, it accepts an incoming message m' only if the following is satisfied: i) m' is of the form (s, m) where s is interpreted as a SID, and ii) the simulated M accepts m in mode CheckAddress. (In mode Compute, s will be stored by $\underline{M}$.) If $\underline{M}$ was activated before, then $\underline{M}$ will accept an

incoming message m' only if the following is satisfied: i) m' is of the form (s, m) where s is the SID stored in the first activation (in mode Compute), and ii) m is accepted by the simulated M in mode CheckAddress.
- When activated in mode Compute, $\underline{M}$ does the following: If $\underline{M}$ has never been activated before (in mode Compute), then by the definition of $\underline{M}$ in mode CheckAddress it follows that the incoming message is of the form (s, m). Now, $\underline{M}$ first stores s and then simulates M on input m in mode Compute. If M produces output, say m', then $\underline{M}$ sends the output (s, m'), i.e., prefixes m' with s. If $\underline{M}$ was activated before (in mode Compute), then by definition of $\underline{M}$ in mode CheckAddress it follows that the incoming message is of the form (s, m) where s is the SID stored in the first activation. Now, as before, $\underline{M}$ simulates M on input m in mode Compute and prefixes the output produced (if any) with s.

Now, the *session version* $\underline{\mathcal{Q}}$ of a system $\mathcal{Q}$ is obtained from $\mathcal{Q}$ by replacing every ITM occurring in $\mathcal{Q}$ by its session version.

The following theorem, which is proved in Appendix C, says that if a real protocol securely realizes an ideal protocol, then an unbounded number of copies of the real protocol securely realize an unbounded number of copies of ideal protocol.

Theorem 4. *Let $\mathcal{P}, \mathcal{F}$ be protocol systems such that $\mathcal{P}$ and $\mathcal{F}$ are I/O-compatible and $\mathcal{P} \leq^{SS} \mathcal{F}$. Then, $!\underline{\mathcal{P}} \leq^{SS} !\underline{\mathcal{F}}$.*

We remark that in the above composition theorem, the session versions and the SIDs are simply used as a means to address certain (ITMs belonging to) copies of protocols. A protocol itself is not and does not need to be aware of the SID used to address ITMs belonging to it, and the specific addressing mechanism used.

As an immediate consequence of Theorem 3 and 4 we obtain the following corollary.

Corollary 1. *Let $\mathcal{P}_1, \mathcal{P}_2, \mathcal{F}_1, \mathcal{F}_2$ be protocol systems such that the systems $\mathcal{P}_1 \mid !\underline{\mathcal{P}_2}$ and $\mathcal{F}_1 \mid !\underline{\mathcal{F}_2}$ are well-formed and the following conditions are satisfied for $j \in \{1, 2\}$:*

1. *$\mathcal{P}_1$ is environmentally connectible for $!\underline{\mathcal{P}_2}$.*
2. *$\mathcal{F}_1$ is environmentally connectible for $!\underline{\mathcal{F}_2}$.*
3. *$\mathcal{P}_j$ and $\mathcal{F}_j$ are I/O compatible.*
4. *$\mathcal{P}_j \leq^{SS} \mathcal{F}_j$.*

Then, $\mathcal{P}_1 \mid !\underline{\mathcal{P}_2} \leq^{SS} \mathcal{F}_1 \mid !\underline{\mathcal{F}_2}$. If $\mathcal{P}_1$ and $\mathcal{F}_1$ coincide up to the names of the network tapes (to ensure that $\mathcal{P}_1$ and $\mathcal{F}_1$ are I/O compatible), we do not need to require 4. for $j = 1$.

In this corollary, for the case that $\mathcal{P}_1$ and $\mathcal{F}_1$ coincide up to the names of the network tapes, we use that $\mathcal{P}_1 \leq^{SS} \mathcal{F}_1$: one can choose the simulator to be the dummy ITM. In this setting, the corollary says that if (an unbounded

number of copies of) an ideal protocol $\mathcal{F}_2$ is used as a component in a more complex system $\mathcal{P}_1$ ($\mathcal{F}_1$), then it can be replaced by its realization $\mathcal{P}_2$. Clearly, by iteratively applying Theorem 3 and 4, one can construct much more complex systems than those described in the above corollary.

Using the equivalences between the security notions stated in Section 5, the above composition theorems immediately carry over to the other security notions considered in this paper. (Note that if the ideal protocols $\mathcal{F}$, $\mathcal{F}_1, \ldots, \mathcal{F}_k$ are generous, then so are $!\underline{\mathcal{F}}$ and $\mathcal{F}_1 \,|\, \cdots \,|\, \underline{\mathcal{F}}_k$.)

7 IO-Network-Enriching Protocol Systems

In this section, we briefly discuss how our general computational model can be applied to a different class of protocol systems, called IO-network-enriching protocol systems.

In IO-network-enriching protocol systems not only I/O tapes but also network tapes may be enriching. The class of IO-network-enriching protocol systems is quite similar in terms of expressivity to the class of polynomially shaped weakly polynomial collections defined in [14].

The security notions universal and reactive simulatability for IO-network-enriching protocol systems can be defined just as in the case of IO-enriching protocol systems (see Definition 4). However, to ensure that the systems $\mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{P}$ and $\mathcal{E} \,|\, \mathcal{I} \,|\, \mathcal{P}$ are well-formed, we need to restrict the class of adversarial systems (the definition of environmental systems remains unchanged). Note that with the current definition of (IO-network-enriching) adversarial systems the systems $\mathcal{A} \,|\, \mathcal{P}$ and $\mathcal{I} \,|\, \mathcal{F}$ might not be well-formed anymore if $\mathcal{P}$ and $\mathcal{F}$ may be IO-network-enriching protocol systems: $\mathcal{A}$ ($\mathcal{I}$) may connect to $\mathcal{P}$ ($\mathcal{F}$) by enriching network tapes, and vice versa. One therefore has to restrict adversarial systems to be IO-enriching, i.e., network tapes have to be consuming.

As in the case of IO-enriching protocols, it is not hard to show that the notions universal and reactive simulatability as defined above are equivalent (where, as in the case of IO-enriching protocol systems, we use that the environment is non-uniform). Also, a composition theorem similar to Theorem 3 can be proved.

Comparing IO-Network-Enriching and IO-Enriching Protocol Systems. The obvious advantage of IO-network-enriching protocol systems compared to IO-enriching protocol systems is that the runtime of such systems may depend on the length of the input received on network tapes *and* I/O tapes (rather than only on I/O tapes). Hence, IO-network-enriching protocol systems can forward arbitrarily long messages from the adversary. However, this can be mimiced by IO-enriching protocol systems since additional resources for forwarding messages coming from network tapes can be supplied by the environment. Hence, the additional feature of IO-network-enriching protocol systems does not seem to be very essential. In fact, IO-enriching protocol systems and the security notions defined for them in this paper appear to be the more favorable and useful setting for several reasons:

As demonstrated in this paper, IO-enriching protocol systems allow for natural and simple definitions of all five security notions: strong, black-box, dummy universal, universal, and reactive simulatability. It remains to be investigated whether for the former three notions—which are often prefered over universal and reactive simulatability as they typically greatly simply proofs—equally natural definitions exist also for IO-network-enriching protocol systems. We note that in the model by Hofheinz et al. [14] dummy universal simulatability can be formulated for a class of protocols similar to IO-network-enriching protocols. However, the definitions are much more complex than those presented here for IO-enriching protocol systems. (Strong and black-box simulatability have not been defined in their setting.)

The notion of universal simulatability for IO-network-enriching protocols as defined here is problematic if the ideal adversary is invoked often by the ideal protocol (e.g., to supply ciphertexts) since the ideal adversary could run out of resources. (Recall that the tapes with which the ideal adversary connects to the ideal protocol are consuming.) Hence, certain natural ideal protocols are not realizable. In the setting for IO-enriching protocols, where all tapes of the ideal adversary may be enriching, such problems do not occur.

8 Conclusion

We have proposed an expressive general computational model for systems containing an unbounded number of inexhaustible ITMs and involving a generic addressing mechanism for copies of ITMs. This model extends and simplifies certain aspects of previous models. Based on this model, we demonstrated that several security notions for different classes of protocol systems can be formulated in a simple and uniform way and that, unlike in previous models, these security notions exhibit intuitive properties. We also proved general composition theorems. Many of the proofs could be carried out by mere equational reasoning based on a few equational principles on systems of ITMs.

Almost all models for simulation-based security (including our model) are sequential in the sense that at any time in a run at most one machine is active (an exception is the model in [18]). While, in order to concentrate on cryptographic issues, this is a good abstraction of distributed systems, the computation in real distributed systems is concurrent, i.e., many machines can be active at the same time. It would be interesting to see in how far the model presented here, including the security notions considered, could be extended to a real concurrent model.

Acknowledgments. We would like to thank Ran Canetti for many interesting discussions on models for simulation-based security. We also thank Michael Backes, Anupam Datta, and John Mitchell.

References

1. M. Backes and D. Hofheinz. How to Break and Repair a Universally Composable Signature Functionality. In *ISC 2004*, volume 3225 of *Lecture Notes in Computer Science*, pages 61–72. Springer, 2004.
2. M. Backes, B. Pfitzmann, and M. Waidner. A composable cryptographic library with nested operations. In *CCS 2003*, pages 220–230. ACM, 2003.
3. M. Backes, B. Pfitzmann, and M. Waidner. A General Composition Theorem for Secure Reactive Systems. In *TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 336–354. Springer, 2004.
4. M. Backes, B. Pfitzmann, and M. Waidner. Secure Asynchronous Reactive Systems. Technical Report 082, Cryptology ePrint Archive, 2004.
5. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *FOCS 2001*, pages 136–145. IEEE Computer Society, 2001.
6. R. Canetti. Universally Composable Signature, Certification, and Authentication. In *CSFW 2004*, pages 219–233. IEEE Computer Society, 2004.
7. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Technical report, Cryptology ePrint Archive, December 2005. Online available at `http://eprint.iacr.org/2000/067.ps`.
8. R. Canetti and H. Krawczyk. Universally Composable Notions of Key Exchange and Secure Channels. In *EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 337–351. Springer, 2002.
9. R. Canetti and T. Rabin. Universal Composition with Joint State. In *CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 265–281. Springer, 2003.
10. A. Datta, R. Küsters, J. Mitchell, and A. Ramanathan. On the Relationships Between Notions of Simulation-Based Security. In *TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 476–494. Springer-Verlag, 2005.
11. A. Datta, R. Küsters, J. Mitchell, A. Ramanathan, and V. Shmatikov. Unifying Equivalence-Based Definitions of Protocol Security. In *WITS 2004*, 2004.
12. O. Goldreich. *Foundations of Cryptography*, volume 1. Cambridge Press, 2001.
13. C. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.
14. D. Hofheinz, J. Müller-Quade, and D. Unruh. Polynomial Runtime in Simulatability Definitions. In *CSFW 2005*, pages 156–169. IEEE Computer Society, 2005.
15. D. Hofheinz and D. Unruh. Comparing two notions of simulatability. In *TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 86–103. Springer-Verlag, 2005.
16. R. Küsters. Simulation-Based Security with Inexhaustible Interactive Turing Machines. In *Proceedings of the 19th IEEE Computer Security Foundations Workshop (CSFW-19 2006)*. IEEE Computer Society, 2006. To appear.
17. R. Milner. *A Calculus of Communicating Syustems*. Springer-Verlag, 1980.
18. J. Mitchell, A. Ramanathan, A. Scedrov, and V. Teague. A probabilistic polynomial-time calculus for analysis of cryptographic protocols (preliminary report). In *17th Annual Conference on the Mathematical Foundations of Programming Semantics*, 2001.
19. B. Pfitzmann and M. Waidner. A Model for Asynchronous Reactive Systems and its Application to Secure Message Transmission. In *IEEE Symposium on Security and Privacy*, pages 184–201. IEEE Computer Society Press, 2001.

A Defining Runs of Systems

In this section, we present a more formal definition of runs of systems than the one presented in Section 2.3 and provide proofs.

In what follows, let $\mathcal{S}(1^\eta, a)$ be a system with security parameter η and external input a. We often do not distinguish between an ITM M and its current configuration. We write

$$M(\mathsf{Compute}, \eta, c, m) \rightarrow^\pi M'$$

to say that when running the ITM M starting from its current configuration in mode $\mathsf{Compute}$ with η written on the security parameter tape, m written on input tape c, and the empty bit string written on all other input tapes, on all output tapes, and the address decision tape, we obtain with probability π a configuration M' after the computation is finished (where in M' a message might be written on one of the output tapes).

The (global) configuration of $\mathcal{S}(\eta, a)$ in a run is a tuple of the form (A, P) where, as explained in Section 2.3, A is a sequence of configurations (the sequence of previously activated machines) and P is a system (the passive machines). The initial configuration is (A_0, P_0) where A_0 is the empty sequence and $P_0 = \mathcal{S}$.

Given a configuration (A, P), we now describe how the system evolves when a message m which was output on an output tape c is read by one of the ITMs in the system. We will write $(A, P) \rightarrow^\pi_{(c,m)} (A', P')$ to say that with probability $\pi > 0$ we obtain (A', P') as a successor configuration of (A, P) after m was read on c (by some ITM). We call this event a *communication step with given input* (c, m) and say that (A', P') is a $\rightarrow_{(c,m)}$*-successor* of (A, P).

We have $(A, P) \rightarrow^\pi_{(c,m)} (A', P')$ if one of the following conditions is satisfied where we assume that $A = M_1, \ldots, M_n$.

1. There exists $i \in \{1, \ldots, n\}$ such that $M_i(\mathsf{CheckAddress}, \eta, c, m) = \mathsf{accept}$. If i is minimal with this property and M_i' is a configuration with $M_i(\mathsf{Compute}, \eta, c, m) \rightarrow^\pi M_i'$, then A' is obtained from A by replacing the content of every input and output tape of a configuration in A by the empty bit string and then replacing M_i by M_i' where the content of the input tapes of M_i' are replaced by the empty bit string. Moreover, $P' = P$. (Note that one of the output tapes of M_i' may contain a non-empty bit string and all other output tapes, including those of other ITMs, are empty.)
2. There does not exist $1 \leq i \leq n$ such that $M_i(\mathsf{CheckAddress}, \eta, c, m) = \mathsf{accept}$, but there occurs an ITM M in P (which we identify with its initial configuration) such that $c \in \mathcal{T}_{in}(M)$, c is *enriching* for M, and $M(\mathsf{CheckAddress}, \eta, c, m) = \mathsf{accept}$. If M' is a configuration such that $M(\mathsf{Compute}, \eta, c, m) \rightarrow^\pi M'$, then, A' is obtained from A by replacing the content of every input and output tape of a configuration of A by the empty bit string and appending M' at the end of A where the content of the input tapes of M' is also deleted. (Note that one of the output tapes of M' may contain a non-empty bit string and all other output tapes, including those of other ITMs, are empty.) If M

is in the scope of a bang in P, then $P' = P$, otherwise P' is obtained from P by removing M from P.
3. If neither Condition 1. nor 2. is satisfied, then $\pi = 1$, $A' = A$, and $P' = P$.

Note that in 2., if some M occurs in P with $c \in \mathcal{T}_{in}(M)$, then M is uniquely determined. This is so because by definition of systems we assume that the set of names of input tapes of different occurrences of ITMs in a system are disjoint.

Let (A, P) be a configuration where all output tapes of the configurations occurring in A are empty, except for at most one output tape. (This will be the case after every communication step.) We write $(A, P) \to^{\pi} (A', P')$ if one of the following conditions is satisfied:

1. All output tapes of the configurations in A are empty, $(A, P) \to^{\pi}_{(\mathsf{start},\varepsilon)} (A', P')$ where ε denotes the empty bit string.
2. There is a configuration in A where one of the output tapes, say c, is non-empty, say its content is m, $c \neq \mathsf{decision}$, and $(A, P) \to^{\pi}_{(\mathsf{start},\varepsilon)} (A', P')$.

We refer to $(A, P) \to^{\pi} (A', P')$ as a *communication step* and to (A', P') as a $\to$-successor of (A, P). Informally speaking, Condition 1. means that if no output was produced, then a master ITM is triggered. Condition 2. describes the situation where in the previous communication step output was produced by an ITM and this output is now fed into another ITM. However, a run stops if the output produced was written on an output tape named $\mathsf{decision}$.

A *(complete) run* ρ of a system $\mathcal{S}$ given the security parameter η and external input a (*a run of* $\mathcal{S}(\eta, a)$, for short) is a sequence of configurations $(A_0, P_0), \ldots, (A_k, P_k)$ such that the following conditions are satisfied:

1. (A_0, P_0) is an initial configuration.
2. $(A_0, P_0) \to^{\pi_1}_{(\mathsf{start},a)} (A_1, P_1)$ for some $\pi_1 \in (0, 1]$.
3. $(A_i, P_i) \to^{\pi_{i+1}} (A_{i+1}, P_{i+1})$ for every $1 \leq i \leq k-1$ and some $\pi_{i+1} \in (0, 1]$.
4. (A_k, P_k) does not have a $\to$-successor or all output tapes of configurations in A_{k-1} and A_k are empty, and k is minimal with this property, i.e., there does not exist $k' < k$ such that $(A_{k'}, P_{k'})$ satisfies this property.

Condition 4. defines when a run stops: Either if there is no successor configuration (because output was written on $\mathsf{decision}$) or a master ITM was triggered in the last communication step but did not produce output.

We call $\pi_1 \cdots \pi_k$ the *probability* of ρ, k the *length of* ρ, and say that ρ *outputs* or *returns* m if on an output tape named $\mathsf{decision}$ of some configuration in A_k the message m is written.

Definition 6. *(restated from Definition 2) Let p be a polynomial p and ρ be a run of $\mathcal{S}(1^\eta, a)$. Then, ρ is* p-bounded *if the length of all outputs written on output tapes during the run is $\leq p(\eta + |a|)$.*

A system $\mathcal{S}$ is p-bounded *if for all security parameters η and external inputs a all runs of $\mathcal{S}(1^\eta, a)$ are p-bounded.*

A system $\mathcal{S}$ is (polynomially) bounded *if there exists a polynomial p such that $\mathcal{S}$ is p-bounded.*

For p-bounded systems we obtain:

Lemma 6. *If $\mathcal{S}$ is p-bounded, then there exists a polynomial q such that for every η, a, and run ρ of $\mathcal{S}(\eta, a)$ we have:*

1. *The overall length of inputs given to ITMs in ρ is $\leq q(\eta + |a|)$.*
2. *The length of ρ is $\leq q(\eta + |a|)$.*
3. *The number of ITMs invoked in ρ (both in mode CheckAddress and Compute) and the number of active machines is $\leq q(\eta + |a|)$.*
4. *The length of the contents of work tapes of ITMs in all configurations occurring in ρ is $\leq q(\eta + |a|)$.*
5. *The overall number of transitions taken by ITMs in ρ is $\leq q(\eta + |a|)$.*

Proof. It suffices to find polynomials for every single statement. The polynomial q can then be chosen as the sum of these polynomials.

Statement 1.: Follows immediately with $q = p$.

Statement 2.: To see this statement, we set $q(\eta + |a|) = 2 \cdot p(\eta + |a|) + 1$. Now, it suffices to observe that by definition of runs in at least every other communication step, except for the last two communication step, output must be produced.

Statement 3.: By 2. and the fact that in every communication step at most one configuration is added to the set of activated machines, it follows that the number of activated machines is bounded by some polynomial in $\eta + |a|$. Also, in every communication step the activated machines are invoked at most twice (once in mode CheckAddress and once in mode Compute). The passive machines (whose number is constant), are also invoked at most twice. It follows that the number of invocations of ITMs is bounded by a polynomial in $\eta + |a|$.

Statement 4.: Immediately follows from the definition of computations of ITMs in mode Compute.

Statement 5.: Since by 3. only a polynomial number of ITMs are invoked it suffices to bound the number of transitions taken by an ITM in one activation. This immediately follows from the definition of computations of ITMs in mode CheckAddress and Compute. □

An immediate consequence of the above lemma is:

Proposition 2. *(restated from Proposition 1) Every bounded system can be simulated by a single ITM.*

Definition 7. *(restated from Section 2.3) A system $\mathcal{S}$ is* almost p-bounded *if the probability*

$$f(1^\eta, a) = \mathsf{Prob}[\textit{run of } \mathcal{S}(1^\eta, a) \textit{ is not } p\textit{-bounded}]$$

is negligible.

We call $\mathcal{S}$ almost bounded *if there exists a polynomial p such that $\mathcal{S}$ is almost p-bounded.*

As already mentioned in Section 2.3, for almost p-bounded systems Proposition 2 also holds, except that a simulated run may deviate from a run in the original system with negligible probability.

Recall that in Section 2.3, we have provided examples which illustrate that not all systems are (almost) bounded. While these systems were not well-formed, for well-formed systems we can prove:

Theorem 5. *(restated from Theorem 1) Well-formed systems are bounded.*

Proof. Assume that $\mathcal{S}$ is well-formed and let $G_{\mathcal{S}}$ be the graph associated with $\mathcal{S}$ (see Section 2.2). By assumption, $G_{\mathcal{S}}$ is acyclic and the master ITM (if any) is not in the scope of a bang. We need to show that there exists a polynomial p such that for every η and a, and for every run of $\mathcal{S}(\eta, a)$, the length of the output written on output tapes is bounded by $p(\eta + |a|)$.

If $\mathcal{S}$ does not contain a master ITM, then no ITM in $\mathcal{S}$ will be activated. Hence, no output is produced, which implies that $\mathcal{S}$ is bounded. So, let us assume that $\mathcal{S}$ contains a master ITM.

Let $M_1, \ldots, M_n$ be the ITMs occurring in $\mathcal{S}$. Since $G_{\mathcal{S}}$ is acyclic there is a total ordering $<$ on the M_i's consistent with $G_{\mathcal{S}}$, i.e., if there is an edge from M_i to M_j in $G_{\mathcal{S}}$, then $M_i < M_j$. W.l.o.g. we may assume that $M_1 < M_2 < \cdots < M_n$. By definition of $G_{\mathcal{S}}$, (a copy of) M_i can only be invoked via an enriching input tape by (a copy of) M_j for $j < i$. In particular, only an ITM M_j, $j < i$, can generate a copy of M_i.

There exists exactly one $l \in \{1, \ldots, n\}$ such that M_l is a master ITM, i.e., has start as input tape. It is easy to see by induction on j that the ITMs M_j for $j < l$ are never invoked: If $j = 1 < l$, M_1 could only be invoked via start, i.e., if M_1 were a master ITM. However, since $1 < l$, this is not the case. Now, M_j for $j < l$ can only be invoked by the ITMs $M_1, \ldots, M_{j-1}$. However, by induction, these ITMs are never invoked so M_j cannot be invoked.

Consequently, the first (w.r.t. $<$) ITM to be invoked is the master ITM M_l. It is invoked via start. Since, by assumption, M_l is not in the scope of a bang, in every run of $\mathcal{S}(\eta, a)$, there will only be one copy of M_l. The only input on an enriching tape that M_l obtains is the input a on tape start. By definition of ITMs, this means that in every run of $\mathcal{S}(\eta, a)$, the length of the output produced by M_l is bounded by a polynomial in $\eta + |a|$. In particular, M_l can only generate a polynomial number of copies of ITMs $M_{l+1}, \ldots, M_n$ and the input given to these copies on enriching input tapes by M_l is also bounded by a polynomial.

This implies that in every run of $\mathcal{S}(\eta, a)$ there are only a polynomial number of copies of M_{l+1} and that the input to these copies on enriching input tapes is bounded by a polynomial in $\eta + |a|$. By definition of ITMs, it then follows that the length of the output produced by M_{l+1} is bounded by a polynomial in $\eta + |a|$. In particular, copies of M_{l+1} can only generate a polynomial number of copies of ITMs $M_{l+2}, \ldots, M_n$.

The same argument can iteratively be applied to the remaining ITMs M_{l+2}, $\ldots, M_n$. Since n is a constant (which does not depend on η and $|a|$), it follows that $\mathcal{S}$ is bounded. □

B Proof of Theorem 2

To prove Theorem 2, we first show that strong simulatability is equivalent to dummy universal and black-box simulatability (Appendix B.1 and B.2). We then prove equivalence of universal and reactive simulatability (Appendix B.3). Finally, for the case that the ideal protocol is generous, we establish equivalence between universal and strong simulatability. Combining these equivalences immediately implies Theorem 2.

In what follows, let $\mathcal{P}$ and $\mathcal{F}$ be I/O-compatible protocol systems.

B.1 Equivalence of Strong and Dummy Universal Simulatability

We prove that $\mathcal{P} \leq^{SS} \mathcal{F}$ iff $\mathcal{P} \leq^{UCdum} \mathcal{F}$, and start by showing that strong simulatability implies dummy universal simulatability:

1. Assume that $\mathcal{P} \leq^{SS} \mathcal{F}$.
2. By definition of SS, we obtain: $\exists\, \mathcal{S}{\in}\mathrm{Sim}^{\mathcal{P}}_{\mathbf{S}}(\mathcal{F})\ \forall\, \mathcal{E}{\in}\mathrm{Con}_{\mathbf{E}}(\mathcal{P})$:
$$\mathcal{E} \,|\, \mathcal{P} \equiv \mathcal{E} \,|\, \mathcal{S} \,|\, \mathcal{F}.$$
3. With $\mathcal{S}$ as in 2. and $\mathcal{D}^{io}$ as in Definition 4, we have for all $\mathcal{E}{\in}\mathrm{Env}_{\mathbf{E}}(\mathcal{D}^{io} \,|\, \mathcal{P})$:
$$\begin{aligned} \mathcal{E} \,|\, \mathcal{D}^{io} \,|\, \mathcal{P} &\equiv \mathcal{E}' \,|\, \mathcal{P} && \text{(Lemma 2 and 1)} \\ &\equiv \mathcal{E}' \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(2.)} \\ &\equiv \mathcal{E} \,|\, \mathcal{S}' \,|\, \mathcal{F} && \text{(Lemma 1)} \end{aligned}$$
where $\mathcal{E}'$ is obtained from $\mathcal{E}$ by renaming the external tapes c' of $\mathcal{E}$ connecting to $\mathcal{D}^{io}$ to c (see the definition of $\mathcal{D}^{io}$) and declaring them to be network tapes, and $\mathcal{S}'$ is obtained from $\mathcal{S}$ by renaming the tapes c connecting to $\mathcal{E}'$ to c' and declaring these tapes to be I/O tapes.

We now show that dummy universal simulatability implies strong simulatability.

1. Assume that $\mathcal{P} \leq^{UCdum} \mathcal{F}$.
2. By definition of UCdummy, we obtain: $\exists\, \mathcal{I}{\in}\mathrm{Sim}^{\mathcal{D}^{io} \,|\, \mathcal{P}}_{\mathbf{S}}(\mathcal{F})\ \forall\, \mathcal{E}{\in}\mathrm{Env}_{\mathbf{E}}(\mathcal{D}^{io} \,|\, \mathcal{P})$:
$$\mathcal{E} \,|\, \mathcal{D}^{io} \,|\, \mathcal{P} \equiv \mathcal{E} \,|\, \mathcal{I} \,|\, \mathcal{F}$$
where $\mathcal{D}^{io} = \mathcal{D}^{io}(\mathcal{T}_{in}(\mathcal{P}), \mathcal{T}_{out}(\mathcal{P}))$.
3. With $\mathcal{S} = \mathcal{I}$ as in 2., we have for all $\mathcal{E}{\in}\mathrm{Con}_{\mathbf{E}}(\mathcal{P})$:
$$\begin{aligned} \mathcal{E} \,|\, \mathcal{P} &\equiv \mathcal{E}' \,|\, \mathcal{D}^{io} \,|\, \mathcal{P} && \text{(Lemma 2 and 1)} \\ &\equiv \mathcal{E}' \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(2.)} \\ &\equiv \mathcal{E} \,|\, \mathcal{S}' \,|\, \mathcal{F} && \text{(Lemma 1)} \end{aligned}$$
where $\mathcal{E}'$ is obtained from $\mathcal{E}$ by renaming the external tapes c of $\mathcal{E}$ connecting to the network tapes of $\mathcal{P}$ to c' and declaring them to be I/O tapes, and $\mathcal{S}'$ is obtained from $\mathcal{S}$ by renaming the tapes c' connecting to $\mathcal{E}'$ to c and declaring them to be network tapes. □

B.2 Equivalence of Strong and Black-Box Simulatability

We prove that $\mathcal{P} \leq^{SS} \mathcal{F}$ iff $\mathcal{P} \leq^{BB} \mathcal{F}$, and start by showing that black-box simulatability implies strong simulatability:

1. Assume that $\mathcal{P} \leq^{BB} \mathcal{F}$.
2. By definition of BB we have: $\exists\, \mathcal{S} \in \mathrm{Sim}_{\mathbf{S}}^{\mathcal{P}}(\mathcal{F})\ \forall\, \mathcal{A} \in \mathrm{Adv}_{\mathbf{A}}(\mathcal{P})\ \forall\, \mathcal{E} \in \mathrm{Env}_{\mathbf{E}}(\mathcal{A} \,|\, \mathcal{P})$:

$$\mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{P} \equiv \mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{S} \,|\, \mathcal{F}$$

   and $\mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{S} \,|\, \mathcal{F}$ is almost bounded.
3. With $\mathcal{S}$ as in 2., we have for all $\mathcal{E} \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P})$:

$$\begin{aligned} \mathcal{E} \,|\, \mathcal{P} &\equiv \mathcal{E}' \,|\, \mathcal{D}^{io} \,|\, \mathcal{P} && \text{(Lemma 2 and 1)} \\ &\equiv \mathcal{E}' \,|\, \mathcal{D}^{io} \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(2.)} \\ &\equiv \mathcal{E} \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(Lemma 2)} \end{aligned}$$

   where $\mathcal{D}^{io} = \mathcal{D}^{io}(\mathcal{T}_{in}^{net}(\mathcal{P}), \mathcal{T}_{out}^{net}(\mathcal{P}))$ and $\mathcal{E}'$ is obtained from $\mathcal{E}$ by renaming the external tapes named c of $\mathcal{E}$ connecting to the network tapes of $\mathcal{P}$ to c' and declaring them to be I/O tapes. Note that by 2. the system $\mathcal{E}' \,|\, \mathcal{D}^{io} \,|\, \mathcal{S} \,|\, \mathcal{F}$ is almost bounded.

We now show the implication in the other direction.

1. Assume that $\mathcal{P} \leq^{SS} \mathcal{F}$.
2. By definition of SS, we obtain: $\exists\, \mathcal{S} \in \mathrm{Sim}_{\mathbf{S}}^{\mathcal{P}}(\mathcal{F})\ \forall\, \mathcal{E} \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P})$:

$$\mathcal{E} \,|\, \mathcal{P} \equiv \mathcal{E} \,|\, \mathcal{S} \,|\, \mathcal{F}.$$

3. With $\mathcal{S}$ as in 2., we have for all $\forall\, \mathcal{A} \in \mathrm{Adv}_{\mathbf{A}}(\mathcal{P})\ \forall\, \mathcal{E} \in \mathrm{Env}_{\mathbf{E}}(\mathcal{A} \,|\, \mathcal{P})$:

$$\begin{aligned} \mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{P} &\equiv [\mathcal{E} \,|\, \mathcal{A}]_{\mathcal{P}} \,|\, \mathcal{P} && \text{(Lemma 4)} \\ &\equiv [\mathcal{E} \,|\, \mathcal{A}]_{\mathcal{P}} \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(2.)} \\ &\equiv \mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(2. and Lemma 5)} \end{aligned}$$

   and $\mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{S} \,|\, \mathcal{F}$ is almost bounded (Lemma 5). □

B.3 Equivalence of Universal and Reactive Simulatability

We prove that $\mathcal{P} \leq^{UC} \mathcal{F}$ iff $\mathcal{P} \leq^{RS} \mathcal{F}$. The implication from left to right is trivial. The argument in the other direction is analogous to the one presented in [7]. Let us present the proof sketch:

Assume that $\mathcal{P} \leq^{RS} \mathcal{F}$. This implies that $\forall\, \mathcal{A} \in \mathrm{Adv}_{\mathbf{A}}(\mathcal{P})\ \forall\, \mathcal{E} \in \mathrm{Env}_{\mathbf{E}}(\mathcal{A} \,|\, \mathcal{P})$ $\exists\, \mathcal{I} \in \mathrm{Sim}_{\mathbf{S}}^{\mathcal{A} \,|\, \mathcal{P}}(\mathcal{F})$:

$$\mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{P} \equiv \mathcal{E} \,|\, \mathcal{I} \,|\, \mathcal{F}.$$

We choose $\mathcal{E}$ to be a "universal" Turing machine (more precisely, a universal ITM) which takes as external input (i.e., input on start) a tuple of the form $(a, e, 1^t)$ where e is an encoding of some ITM (representing an environmental

system $\mathcal{E}'$), a is interpreted as an external input to $\mathcal{E}'$, and t is interpreted as a runtime. (By Lemma 3, we may assume that e encodes a single ITM with only consuming input tapes and which accepts every message in mode CheckAddress.) The universal ITM $\mathcal{E}$ simulates $\mathcal{E}'$ with external input a up to t steps. Now, when ranging over all tuples $(a, e, 1^t)$—of polynomially bounded length in the security parameter—$\mathcal{E}$ simulates all environmental systems $\mathcal{E}'$. Hence, $\mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{P} \equiv \mathcal{E} \,|\, \mathcal{I} \,|\, \mathcal{F}$ implies $\mathcal{E}' \,|\, \mathcal{A} \,|\, \mathcal{P} \equiv \mathcal{E}' \,|\, \mathcal{I} \,|\, \mathcal{F}$ for every $\mathcal{E}' \in \mathrm{Env}_{\mathbf{E}}(\mathcal{A} \,|\, \mathcal{P})$. Thus, $\mathcal{P} \leq^{UC} \mathcal{F}$ follows.

B.4 Equivalence of Universal and Strong Simulatability

Assume that $\mathcal{F}$ is generous. We show that $\mathcal{P} \leq^{SS} \mathcal{F}$ iff $\mathcal{P} \leq^{UC} \mathcal{F}$, and start with the direction from right to left.

Obviously, $\mathcal{P} \leq^{UC} \mathcal{F}$ implies $\mathcal{P} \leq^{UCdum} \mathcal{F}$. Since strong simulatability and dummy universal simulatability are equivalent, we obtain that $\mathcal{P} \leq^{UC} \mathcal{F}$ implies $\mathcal{P} \leq^{SS} \mathcal{F}$. Note that for this direction the assumption that $\mathcal{F}$ is generous is not needed.

We now show that $\mathcal{P} \leq^{SS} \mathcal{F}$ implies $\mathcal{P} \leq^{UC} \mathcal{F}$:

1. Assume that $\mathcal{P} \leq^{SS} \mathcal{F}$.
2. By definition of SS, we obtain: $\exists\, \mathcal{S} \in \mathrm{Sim}^{\mathcal{P}}_{\mathbf{S}}(\mathcal{F})\ \forall\, \mathcal{E} \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P})$:

$$\mathcal{E} \,|\, \mathcal{P} \equiv \mathcal{E} \,|\, \mathcal{S} \,|\, \mathcal{F}.$$

3. With $\mathcal{S}$ as in 2., we have $\forall\, \mathcal{A} \in \mathrm{Adv}_{\mathbf{A}}(\mathcal{P})\ \forall\, \mathcal{E} \in \mathrm{Env}_{\mathbf{E}}(\mathcal{A} \,|\, \mathcal{P})$:

$$\begin{aligned} \mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{P} &\equiv [\mathcal{E} \,|\, \mathcal{A}]_{\mathcal{P}} \,|\, \mathcal{P} && \text{(Lemma 4)} \\ &\equiv [\mathcal{E} \,|\, \mathcal{A}]_{\mathcal{P}} \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(2.)} \\ &\equiv \mathcal{E} \,|\, \mathcal{A} \,|\, \mathcal{S} \,|\, \mathcal{F} && \text{(2. and Lemma 5)} \\ &\equiv \mathcal{E} \,|\, \mathcal{I} \,|\, \mathcal{F} && \text{(*)} \end{aligned}$$

   We need to define $\mathcal{I}$, which will only depend on $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{F}$, and prove (*), which then shows that $\mathcal{P} \leq^{UC} \mathcal{F}$.

Before defining $\mathcal{I}$, we make some useful observations:

(a) By Lemma 3, $\mathcal{P}$ can be simulated by a single ITM, and hence, by definition of ITMs there exists a polynomial in the security parameter η and the length of the input received on enriching input tapes of $\mathcal{P}$ which bounds the length of the output produced. This polynomial is independent of the context $\mathcal{E}$ in which $\mathcal{P}$ runs.

(b) The probability that at some point in a run of $\mathcal{E} \,|\, \mathcal{S} \,|\, \mathcal{F}$ the length of the output written by $\mathcal{S}$ on one of the output network tapes to $\mathcal{E}$ exceeds the bound mentioned in (a) is negligible, in η and the external input a given to $\mathcal{E}$. (Otherwise, one can easily construct $\mathcal{E}'$ such that $\mathcal{E}' \,|\, \mathcal{P} \not\equiv \mathcal{E}' \,|\, \mathcal{S} \,|\, \mathcal{F}$, in contradiction to 2.: $\mathcal{E}'$ simply simulates $\mathcal{E}$ and outputs 1 iff the length of the input received on the network tapes is bounded as expected.)

(c) Since $\mathcal{F}$ is generous, in every run of the system $\mathcal{E} \mid \mathcal{A} \mid \mathcal{S} \mid \mathcal{F}$ whenever $\mathcal{S}$ (i.e., one of the ITMs that belong to $\mathcal{S}$) is activated the length of the input $\mathcal{S}$ received from $\mathcal{F}$ is polynomially at least as big as the length of the input $\mathcal{F}$ received on its enriching I/O tapes so far, where the polynomial only depends on $\mathcal{F}$.

(d) By Lemma 3, we may replace $\mathcal{A}$ and $\mathcal{S}$ by single ITMs $M_{\mathcal{A}}$ and $M_{\mathcal{S}}$, respectively.

We define $\mathcal{I}$ as follows: The input and output tapes of $\mathcal{I}$ are defined in such a way that $\mathcal{I}$ is compatible with the system $M_{\mathcal{A}} \mid M_{\mathcal{S}}$. The machine $\mathcal{I}$ simulates $M_{\mathcal{A}} \mid M_{\mathcal{S}}$, and the external tapes between $M_{\mathcal{A}}$ and $M_{\mathcal{S}}$ are work tapes in $\mathcal{I}$.

To see that $\mathcal{I}$ can in fact simulate $M_{\mathcal{A}} \mid M_{\mathcal{S}}$, we prove that the overall length of the bit strings written on the tapes between $M_{\mathcal{A}}$ and $M_{\mathcal{S}}$ can polynomially be bounded in η plus the length of the input to $\mathcal{I}$ on enriching input tapes: This follows from the fact that by (b) the output sent by $M_{\mathcal{S}}$ to $M_{\mathcal{A}}$ is (with overwhelming probability) polynomially bounded in η plus the length l of the input $\mathcal{F}$ received from $\mathcal{E}$ so far. By (c), l is bounded by a polynomial in the security parameter and the length l' of the input $M_{\mathcal{S}}$ received from $\mathcal{F}$ so far. Hence, the output sent by $M_{\mathcal{S}}$ to $M_{\mathcal{A}}$ is (with overwhelming probability) polynomially bounded in η plus the length of the input $M_{\mathcal{S}}$ received from $\mathcal{F}$ so far. But then, the output from $M_{\mathcal{A}}$ to $M_{\mathcal{S}}$ is polynomially bounded in η plus the length of the input given to $M_{\mathcal{A}}$ by $\mathcal{E}$ and l'. Note that if the length of the input from $M_{\mathcal{S}}$ to $M_{\mathcal{A}}$ exceeds the polynomial bound in $\eta + l'$, then $\mathcal{I}$ can simply stop the simulation as this happens only with negligible probability.

From the above it follows that $M_{\mathcal{A}} \mid M_{\mathcal{S}}$ can in fact be simulated in polynomial time in the security parameter and the length of the input given to $\mathcal{I}$ on enriching input tapes. The system $\mathcal{I}$ exactly mimics $M_{\mathcal{A}} \mid M_{\mathcal{S}}$, except when the overall length of messages sent from $M_{\mathcal{S}}$ to $M_{\mathcal{A}}$ exceeds a certain bound. But since this happens only with negligible probability, the simulation is faithful with overwhelming probability.

Now, (*) easily follows. □

C Proof of Theorem 4

For every system $\mathcal{Q}$, we define

$$P_{\mathcal{Q}}(\eta, a) = \mathsf{Prob}[\mathcal{Q}(\eta, a) \rightsquigarrow 1].$$

Let $\mathcal{P}, \mathcal{F}$ be protocol systems such that $\mathcal{P}$ and $\mathcal{F}$ are I/O-compatible and $\mathcal{P} \leq^{SS} \mathcal{F}$. We need to show that $!\underline{\mathcal{P}} \leq^{SS} !\underline{\mathcal{F}}$.

Since $\mathcal{P} \leq^{SS} \mathcal{F}$, there exists $\mathcal{S} \in \mathrm{Sim}^{\mathcal{P}}_{\mathbf{S}}(\mathcal{F})$ such that for all $\mathcal{E} \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P})$ we have that

$$\mathcal{E} \mid \mathcal{P} \equiv \mathcal{E} \mid \mathcal{S} \mid \mathcal{F}. \tag{1}$$

By Lemma 3, we may assume that $\mathcal{S}$ is a single ITM which in mode CheckAddress accepts all messages. We denote by $\underline{\mathcal{S}}$ the session version of $\mathcal{S}$.

Obviously, $!\underline{\mathcal{S}} \in \mathrm{Sim}_{\mathbf{S}}^{!\underline{\mathcal{P}}}(!\underline{\mathcal{F}})$. We show that

$$\mathcal{E} \,|\, !\,\underline{\mathcal{P}} \equiv \mathcal{E} \,|\, !\,\underline{\mathcal{S}} \,|\, !\,\underline{\mathcal{F}}. \tag{2}$$

for every $\mathcal{E} \in \mathrm{Con}_{\mathbf{E}}(!\,\underline{\mathcal{P}})$, which then concludes the proof of Theorem 4.

To show (2), let $\mathcal{E} \in \mathrm{Con}_{\mathbf{E}}(!\,\underline{\mathcal{P}})$ and let p_1 and p_2 be polynomials. We need to prove that there exists η_0 such that for all $\eta > \eta_0$ and all bit strings $a \in \bigcup_{\eta' \leq p_2(\eta)} \{0,1\}^{\eta'}$ we have that

$$d(\eta, a) := |P_{\mathcal{E} \,|\, !\,\underline{\mathcal{P}}}(1^\eta, a) - P_{\mathcal{E} \,|\, !\,\underline{\mathcal{S}} \,|\, !\,\underline{\mathcal{F}}}(1^\eta, a)| \leq \frac{1}{p_1(\eta)}. \tag{3}$$

The proof proceeds by a hybrid argument.

In what follows, let $\underline{\mathcal{P}}'$ be the version of $\underline{\mathcal{P}}$ obtained from $\underline{\mathcal{P}}$ by renaming every tape c occurring in $\underline{\mathcal{P}}$ to c'. Analogously, let $\underline{\mathcal{P}}''$ be obtained from $\underline{\mathcal{P}}$ by renaming every tape c occurring in $\underline{\mathcal{P}}$ to c''. Similarly for $\underline{\mathcal{F}}'$, $\underline{\mathcal{F}}''$, $\underline{\mathcal{S}}'$, and $\underline{\mathcal{S}}''$.

We now define a system $\mathcal{E}'$ which basically simulates $\mathcal{E}$ and which will run in the system $\mathcal{E}' \,|\, !\,\underline{\mathcal{P}}'' \,|\, !\,\underline{\mathcal{S}}' \,|\, !\,\underline{\mathcal{F}}' \,|\, \mathcal{P}$ and $\mathcal{E}' \,|\, !\,\underline{\mathcal{P}}'' \,|\, !\,\underline{\mathcal{S}}' \,|\, !\,\underline{\mathcal{F}}' \,|\, \mathcal{S} \,|\, \mathcal{F}$, respectively. More precisely, $\mathcal{E}'$ is given a parameter i on its external input. The first i copies of the protocol invoked by $\mathcal{E}$ will be copies of $\underline{\mathcal{S}}' \,|\, \underline{\mathcal{F}}'$, the i+1st copy will be $\mathcal{P}$ and $\mathcal{S} \,|\, \mathcal{F}$, respectively, and the remaining copies will be copies of $\underline{\mathcal{P}}''$.

Formally, $\mathcal{E}'$ is obtained from $\mathcal{E}$ as follows. By Lemma 3, we may assume that $\mathcal{E}$ is a single ITM which in mode CheckAddress always accepts. The ITM $\mathcal{E}'$ will also always accept in mode CheckAddress. The behavior of $\mathcal{E}'$ in mode Compute is specified next.

First, we need to make sure that $\mathcal{E}'$ has the appropriate tapes to connect to the different entities. The system $\mathcal{E}$ may already have tapes to connect to the external tapes of $\mathcal{P}$ and $\mathcal{S} \,|\, \mathcal{F}$. For each such tape c, we add to $\mathcal{E}'$ a tape c' and c'' to connect to the external tapes of $\underline{\mathcal{S}}' \,|\, \underline{\mathcal{F}}'$ and $\underline{\mathcal{P}}''$, respectively.

Next, we need to specify how $\mathcal{E}'$ redirects protocol invocations of $\mathcal{E}$ in the way described above: $\mathcal{E}'$ interprets its external input, i.e., the input received on tape start, as a tuple of the form (i, a) where i encodes a non-negative integer. $\mathcal{E}'$ then stores i and keeps a list L of SIDs, which initially is empty, and the length l of the list, which initially is 0. From now on, $\mathcal{E}'$ simulates $\mathcal{E}$ with external input a. In particular, if $\mathcal{E}$ produces output, then so does $\mathcal{E}'$, and if $\mathcal{E}'$ receives input, then $\mathcal{E}$ is simulated with this input. However, as explained next, the behavior of $\mathcal{E}'$ deviates from that of $\mathcal{E}$ when it comes to sending and receiving messages to the different copies of protocols.

1. If $\mathcal{E}$ produces output m on some external tape c of $\underline{\mathcal{P}}$ (and hence, $\underline{\mathcal{S}} \,|\, \underline{\mathcal{F}}$) which is not prefixed by a SID, then $\mathcal{E}'$ does not produce output.
2. If $\mathcal{E}$ produces output m on some external tape c of $\underline{\mathcal{P}}$ (and hence, $\underline{\mathcal{S}} \,|\, \underline{\mathcal{F}}$) prefixed with s, then $\mathcal{E}'$ checks whether s occurs in L. Let j be the position where s occurs in L (the positions in L start from position 1); let $j = 0$ if s does not occur in L.
   (a) If $0 < j \leq i$, then $\mathcal{E}'$ writes m on tape c'.

(b) If $j = i{+}1$, then $\mathcal{E}'$ outputs m' on c where m' is a message such that $m = (s, m')$, i.e., s is removed from m.
(c) If $j > i{+}1$, then $\mathcal{E}'$ writes m on tape c''.
(d) If $j = 0$, then s is appended at the end of L and l is increased by 1. Next, depending on whether $0 < l \leq i$, $l = i{+}1$, or $l > i{+}1$, $\mathcal{E}'$ proceeds as in a), b), or c), respectively.

3. If $\mathcal{E}'$ receives input on tape c'' where c'' is an external tape of $\underline{\mathcal{P}}''$ corresponding to an external tape c of $\underline{\mathcal{P}}$, then $\mathcal{E}'$ behaves as $\mathcal{E}$ in case input was received on tape c.
4. If $\mathcal{E}'$ receives input on tape c' where c' is an external tape of $\underline{\mathcal{S}}' \,|\, \underline{\mathcal{F}}'$ corresponding to an external tape c of $\underline{\mathcal{S}} \,|\, \underline{\mathcal{F}}$, then $\mathcal{E}'$ behaves as $\mathcal{E}$ in case input was received on tape c.
5. If $\mathcal{E}'$ receives input m on tape c where c is an external tape of $\mathcal{P}$ (and hence, $\mathcal{S} \,|\, \mathcal{F}$), then $\mathcal{E}'$ behaves as $\mathcal{E}$ in case input $(L[i{+}1], m)$ was received on tape c where $L[i{+}1]$ denotes the $i{+}1$st entry of L. By construction, this entry exists in L since $\mathcal{E}$ must have invoked the $i{+}1$st copy.

We also consider a simplified version $\widehat{\mathcal{E}}'$ of $\mathcal{E}'$ which is intended to run in the system $\widehat{\mathcal{E}}' \,|\, !\underline{\mathcal{P}}'' \,|\, !\underline{\mathcal{S}}' \,|\, !\underline{\mathcal{F}}'$. Instead of invoking $\mathcal{P}$ $(\mathcal{S} \,|\, \mathcal{F})$, $\widehat{\mathcal{E}}'$ invokes $\underline{\mathcal{P}}''$ as its $i{+}1$st protocol copy. This is achieved by modifying $\mathcal{E}'$ in the obvious way.

Since, as mentioned above, we may assume that $\mathcal{E}$ is an ITM with start as its only enriching input tape, we know that there exists a polynomial q such that the length of the output produced by $\mathcal{E}$ in any context is bounded by $q(\eta + |a|)$ where η is the security parameter and a the external input given to $\mathcal{E}$ on start. In particular, the number of copies of protocols that can be invoked by $\mathcal{E}$ is also bounded by $q(\eta + |a|)$ since to invoke a new ITM non-empty output must be sent to the ITM.

In what follows, let $\widehat{\mathcal{E}}'' = \widehat{\mathcal{E}}' \,|\, !\underline{\mathcal{P}}'' \,|\, !\underline{\mathcal{S}}' \,|\, !\underline{\mathcal{F}}'$. By construction, we obtain:

$$P_{\mathcal{E} \,|\, !\underline{\mathcal{P}}}(1^\eta, a) = P_{\widehat{\mathcal{E}}''}(1^\eta, (0, a))$$
$$P_{\mathcal{E} \,|\, !\underline{\mathcal{S}} \,|\, !\underline{\mathcal{F}}}(1^\eta, a) = P_{\widehat{\mathcal{E}}''}(1^\eta, (q(\eta + |a|), a))$$

for every η and a. Define

$$d_i(\eta, a) = |P_{\widehat{\mathcal{E}}''}(1^\eta, (i, a)) - P_{\widehat{\mathcal{E}}''}(1^\eta, (i+1, a))|$$

for every η, bit string a, and $i < q(\eta + |a|)$.

By the triangle inequality, we have that

$$d(\eta, a) = |P_{\widehat{\mathcal{E}}''}(1^\eta, (0, a)) - P_{\widehat{\mathcal{E}}''}(1^\eta, (q(\eta), a))| \leq \sum_{i=0}^{q(\eta+|a|)-1} d_i(\eta, a).$$

We now show that for every polynomial p there exists η_0 such that

$$d_i(\eta, a) \leq \frac{1}{p(\eta)} \tag{4}$$

for every $\eta > \eta_0$, $a \in \bigcup_{\eta' \leq p_2(\eta)} \{0,1\}^{\eta'}$, and $0 \leq i < q(\eta + |a|)$. From this, with $p(\eta) = p_1(\eta) \cdot q(\eta + p_2(\eta))$, we immediately obtain that $d(\eta, a) \leq \frac{1}{p_1(\eta)}$ for every $\eta > \eta_0$ and $a \in \bigcup_{\eta' \leq p_2(\eta)} \{0,1\}^{\eta'}$, which shows (3), and hence, concludes the proof of Theorem 4.

To prove (4), let $\mathcal{E}'' = \mathcal{E}' \,|\, !\underline{\mathcal{P}''} \,|\, !\underline{\mathcal{S}'} \,|\, !\underline{\mathcal{F}'}$. By construction, we obtain that

$$P_{\widehat{\mathcal{E}''}}(1^\eta, (i, a)) = P_{\mathcal{E}'' \,|\, \mathcal{P}}(1^\eta, (i, a)), \tag{5}$$

$$P_{\widehat{\mathcal{E}''}}(1^\eta, (i{+}1, a)) = P_{\mathcal{E}'' \,|\, \mathcal{S} \,|\, \mathcal{F}}(1^\eta, (i, a)). \tag{6}$$

For (5) we use that $\mathcal{P}$ is a protocol system. In particular, we use property iii) of protocol systems. If this property were not satisfied, i.e., $\mathcal{P}$ contains an ITM M which is not in the scope of a bang but which could reject a message in mode CheckAddress, the following could happen. In a run of $(\mathcal{E}'' \,|\, \mathcal{P})(1^\eta, (i, a))$ a message is sent to M, but it is rejected by M (in mode CheckAddress). Then, since M is not in the scope of a bang, no new copy of M will be generated. Conversely, if in a run of $\widehat{\mathcal{E}''}(1^\eta, (i, a))$ a message is sent to a copy of the session version $\underline{M}$ of M prefixed with the $i{+}1$st SID generated by $\mathcal{E}$ and the simulated M in $\underline{M}$ would reject the message, then it could happen that a new copy of $\underline{M}$ is generated (since $\underline{M}$ is in the scope of a bang in $\widehat{\mathcal{E}''}$) which then would not have a corresponding entity in a run of the system $\mathcal{E}'' \,|\, \mathcal{P}(1^\eta, (i, a))$. In short, by property iii) of protocol systems it is guaranteed that for ITMs that do not occur in the scope of a bang in $\mathcal{P}$ only at most one copy is generated per SID in the run of $\widehat{\mathcal{E}''}$.

Analogously, for (6), since $\mathcal{F}$ is a protocol system and $\mathcal{S}$ is (w.l.o.g.) an ITM that accepts all messages in mode CheckAddress, we have that in every run of $\widehat{\mathcal{E}''}$ at most one copy is generated per SID for ITMs in $\mathcal{F}$ that do not occur in the scope of a bang and for $\mathcal{S}$.

It is easy to verify that $\mathcal{E}'' \in \mathrm{Con}_{\mathbf{E}}(\mathcal{P})$. By (1), we know that $\mathcal{E}'' \,|\, \mathcal{P} \equiv \mathcal{E}'' \,|\, \mathcal{S} \,|\, \mathcal{F}$. Consequently, for every polynomial $p(\eta)$ and $p'(\eta) = p_2(\eta) + q(\eta + p_2(\eta))$ there exists η_0 such that

$$\begin{aligned} d_i(\eta, a) &= |P_{\mathcal{E}'' \,|\, \mathcal{P}}(1^\eta, (i, a) - P_{\mathcal{E}'' \,|\, \mathcal{S} \,|\, \mathcal{F}}(1^\eta, (i, a)| \\ &\leq \frac{1}{p(\eta)} \end{aligned}$$

for every $\eta > \eta_0$, $0 \leq i < q(\eta + |a|)$, and $(i, a) \in \bigcup_{\eta' \leq p'(\eta)} \{0,1\}^{\eta'}$. Now, (4) follows.