On the Notion of Statistical Security in Simulatability Definitions

Dennis Hofheinz and Dominique Unruh

IAKS, Arbeitsgruppe Systemsicherheit, Prof. Dr. Th. Beth,
Fakultät für Informatik, Universität Karlsruhe, Am Fasanengarten 5,
76 131 Karlsruhe, Germany

Abstract. We investigate the definition of statistical security (i.e., security against unbounded adversaries) in the framework of reactive simulatability. This framework allows to formulate and analyze multi-party protocols modularly by providing a composition theorem for protocols. However, we show that the notion of statistical security, as defined by Backes, Pfitzmann and Waidner for the reactive simulatability framework, does not allow for secure composition of protocols. This in particular invalidates the proof of the composition theorem. We give evidence that the reason for the non-composability of statistical security is no artifact of the framework itself, but of the particular formulation of statistical security. Therefore, we give a modified notion of statistical security in the reactive simulatability framework. We prove that this notion allows for secure composition of protocols.
As to the best of our knowledge, no formal definition of statistical security has been fixed for Canetti's universal composability framework, we believe that our observations and results can also help to avoid potential pitfalls there.
Keywords: Reactive simulatability, universal composability, statistical security, protocol composition.

1 Introduction

Recently, simulatability-based characterizations of security for cryptographic protocols received a lot of attention. In particular, several modelings of multi-party computation (e.g., [PW00,Can00,PW01,Can01,BPW04]) have been presented which allow for secure composition of protocols. All these models share the idea of simulatability: a protocol is considered secure only relative to another protocol. That is, a protocol π is as secure as another protocol τ (usually an idealization of the respective protocol task), if every attack on π can be simulated by an attack on τ.

A little more formally, this means that for every adversary A_π attacking π, there is an adversary A_τ (sometimes referred to as the simulator) that attacks τ, such that from an outside view, both attacks and protocols "look the same." There are different interpretations of what "looking the same" means concretely. In any case, a designated entity called the "honest user" and denoted H is employed to check for differences between protocol π (together with adversary A_π) and protocol τ (with A_τ). Therefore, H may interact with protocol participants and even with the adversary.

One might now choose H in dependence of A_τ; alternatively, the simulator A_τ may be allowed to depend on the respective distinguisher H. For more discussion on relations between the two induced security notions, cf. [HU05].

Orthogonal to this, one can demand perfect indistinguishability of π and τ, i.e., that every distinguisher H has identical views when running with π, resp. τ. Alternatively, one may demand that these views are only statistically close, or that they are only computationally indistinguishable.[1]

For the reactive simulatability framework due to Backes, Pfitzmann and Waidner (see, e.g., [PW01,BPW04]), formal definitions of these requirements have been given. For all possible combinations of requirements, the induced security definition was shown to behave well under composition of protocols. That is, it was proved in [PW01] that once a protocol π is as secure as another protocol τ, it can be substituted for τ in a larger protocol without losing security (in the sense that the protocol which uses π is as secure as the one which uses τ).

1.1 Our Results

Here we show that the notion of statistical security given in [PW01,BPW04] does not allow for secure composition of protocols (in the above sense). In particular, this disproves the composition theorem of [PW01] for statistical security. However, a change in the definition of statistical security fixes the problem, so that the original proof idea applies. We show this by reproving the composition theorem for the statistical case.

We motivate the change in the definition of statistical security and point out other problems (apart from the composability issue) of the old definition. As to the best of our knowledge, no formal definition of statistical security has been fixed for Canetti's model of universal composability [Can01], we believe that our observations and results can also help to avoid potential pitfalls there.

1.2 Organization

After recalling the mathematical preliminaries in Section 2 (note that an overview over the reactive simulatability framework is given in Appendix A), we explain in Section 3 why the original definition of statistical security does not compose; to this end, we give a counterexample. In Section 4, we give a modified criterion for statistical security and prove that this criterion allows for secure composition of protocols. Section 5 concludes this work.

2 Mathematical Preliminaries

First, we recall the notion of statistical distance.

Definition 1. *Let X and Y be Ω-valued random variables. Then the* statistical distance *$\Delta_{\mathsf{stat}}(X,Y)$ of X and Y is*

$$\Delta_{\mathsf{stat}}(X,Y) = \sup_{\substack{M\subseteq\Omega \\ M \text{ measurable}}} |\mathsf{Pr}[X \in M] - \mathsf{Pr}[Y \in M]|.$$

[1] In the latter case, which captures computational security, generally only polynomially bounded adversaries and honest users are considered.

Note that if Ω is countable or finite, we can write this as

$$\Delta_{\mathsf{stat}}(X,Y) = \frac{1}{2}\sum_{z\in\Omega} |\mathsf{Pr}[X=z] - \mathsf{Pr}[Y=z]|.$$

For re-proving the composition theorem for our notion of statistical security, we will need the following technical lemma:

Lemma 2. *Let X and Y be Ω-valued random variables.*

(i) For any function $f : \Omega \to \Omega'$, we have

$$\Delta_{\mathsf{stat}}(f(X), f(Y)) \le \Delta_{\mathsf{stat}}(X,Y).$$

(ii) If X and Y are sequences of random variables, so that $X = (X_1, X_2, \dots)$, and $Y = (Y_1, Y_2, \dots)$ with $X_i, Y_i \in T$ and $\Omega = T^{\mathbb{N}}$ for some set T, then

$$\Delta_{\mathsf{stat}}(X,Y) = \sup_t \ \Delta_{\mathsf{stat}}(X_{1\dots t}, Y_{1\dots t})$$

where $X_{1\dots t} := (X_1, X_2, \dots, X_t)$ is the prefix of X of length t, and $Y_{1\dots t}$ is defined analogously.

The proof of (i) is straightforward from Definition 1, and (ii) is show in Appendix B.

As in [PW01,BPW04], we use the notion "class of small functions" for capturing what it means that the statistical distance of two user-views gets "small" eventually (i.e., for large security parameters). Formally, we call a set *SMALL* of functions $\mathbb{N} \to \mathbb{R}_{\ge 0}$ a *class of small functions* if it is closed under addition, and contains with a function g every function g' with $g' \le g$.

Typically used classes of small functions are the set

$$\mathit{NEGL} := \{f : \mathbb{N} \to \mathbb{R}_{\ge 0} \ \mid \ \forall c \in \mathbb{N}\ \exists k_c \in \mathbb{N}\ \forall k > k_c :\ f(k) < k^{-c}\}$$

of negligible functions, or the set

$$\mathit{EXPSMALL} := \{f : \mathbb{N} \to \mathbb{R}_{\ge 0} \ \mid \ \exists c \in \mathbb{N},\ k_0 \in \mathbb{N}\ \forall k > k_0 :\ f(k) < 2^{-k} \cdot k^{-c}\}$$

of exponentially small functions.

3 A Counterexample to Composition

In the present section, we present a simple counterexample to the composition theorem of [PW01,Bac02]. A reader unfamiliar with the reactive simulatability framework might want to read the short summary of that framework in Appendix A first.

Let M_0 be a machine with ports $\mathsf{in}?$, $\mathsf{out}!$ and $\mathsf{out}^{\triangleleft}!$ (cf. Figure 1), i.e., the machine has an incoming connection $\mathsf{in}!$, and an outgoing connection $\mathsf{out}!$ on which it can enforce

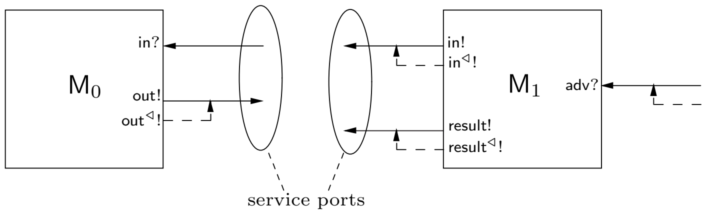

Fig. 1. Machines used in the counterexample. The honest user may only connect to the service ports.

immediate delivery by using the clock port[2] $\mathsf{out}^{\triangleleft}!$. The program of M_0 is really trivial, M_0 simply ignores any inputs, and never generates outputs.

Now consider the machine M_0' that has the same ports as M_0 and the following program:

– Upon the 2^k-th input on $\mathsf{in}?$, output `alive` on $\mathsf{out}!$ (and sent a 1 on $\mathsf{out}^{\triangleleft}!$ to ensure the immediate delivery of that message). Here k denotes the security parameter.

Using these machines we can now define two protocols $\hat{M}_0$ and $\hat{M}_0'$:[3] $\hat{M}_0 := \{\mathsf{M}_0\}$ consists only of the machine M_0, and all ports of M_0 may be accessed by the honest user, i.e., there are no special connections intended for the communication between protocol and adversary.

Formally, in the modeling of [PW01], the protocol $\hat{M}_0$ is represented by the structure $(\hat{M}_0, S)$ with $S := \{\mathsf{in}^{\leftrightarrow}?, \mathsf{in}^{\triangleleft}?, \mathsf{out}^{\leftrightarrow}!\}$.

The protocol $\hat{M}_0'$ is defined analogously to $\hat{M}_0$, but consists of M_0' instead of M_0.

Recapitulating, we now have two protocols $\hat{M}_0$ and $\hat{M}_0'$, the first of which never reacts to inputs, while the second gives an answer after 2^k inputs.

Now consider the definition of statistical security:

Definition 3 (Statistical security [PW01,Bac02,BPW04]). *Let $(\hat{M}_1, S)$ and $(\hat{M}_2, S)$ be structures with identical service ports S. We say that $(\hat{M}_1, S)$ is* statistically as secure *as $(\hat{M}_2, S)$ for a class SMALL of small functions, written $(\hat{M}_1, S) \geq_{\mathsf{sec}}^{SMALL} (\hat{M}_2, S)$, if the following holds:*

For every configuration $conf_1 = (\hat{M}_1, S, \mathsf{H}, \mathsf{A}_1) \in \mathsf{Conf}^{\hat{M}_2}(\hat{M}_1, S)$, there is a configuration $conf_2 = (\hat{M}_2, S, \mathsf{H}, \mathsf{A}_2) \in \mathsf{Conf}(\hat{M}_2, S)$ such that it holds for every polynomial l, that

$$\Delta_{\mathsf{stat}}(view_{conf_1,l}(\mathsf{H}), view_{conf_2,l}(\mathsf{H})), \tag{1}$$

as a function in the security parameter k, is contained in SMALL.

Here by $view_{conf_i,l}(\mathsf{H})$ we denote the prefix consisting of the first $l(k)$ components of $view_{conf_i}(\mathsf{H})$.

[2] The existence of clock ports is required by the modeling of [PW01]. However, in our counterexample they play only a very subordinate role. The reader can ignore them and simply assume that any message is immediately delivered to the recipient.

[3] Both are in fact "one-party protocols". It would be possible to make two-party protocols out of these. However one-party protocols are sufficient for the purpose of creating a counterexample, so for the sake of simplicity we formulate the example using these.

In other words, we demand that for every real adversary A_1 and user H, there is a simulator A_2, such that the statistical difference of polynomial prefixes of H's views in real and ideal model is small in the security parameter. Note that H, as well as the adversaries A_1, A_2 are allowed to be unbounded.

Since the honest user's view is restricted to a polynomial number of messages sent and received from the protocol, he will not be able to distinguish the two protocols, so we get the following lemma (some care must however be taken for the case where the adversary connects to some of M_0's ports, see the proof):

Lemma 4. *We have $(\hat{M}_0, S) \geq_{\mathsf{sec}}^{NEGL} (\hat{M}_0', S)$, i.e., the protocol $\hat{M}_0$ is statistically as secure as $\hat{M}_0'$.*

Proof. To show the lemma, we have to show that for any honest user H and any adversary A_1, s.t. $\mathsf{conf}_1 := (\hat{M}_0, S, \mathsf{H}, \mathsf{A}_1) \in \mathsf{Conf}^{\hat{M}_0'}(\hat{M}_0, S)$ (which essentially means that H and A_1 only connect to ports they are allowed to connect to), there exists a simulator A_2, s.t. the following holds. First, $\mathsf{conf}_2 := (\hat{M}_0', S, \mathsf{H}, \mathsf{A}_2) \in \mathsf{Conf}(\hat{M}_0', S)$ (i.e., A_2 connects only to ports it may connect to), and, second, polynomial prefixes of H's view in runs with A_1 (together with protocol $\hat{M}_0$) and A_2 (together with protocol $\hat{M}_0'$) are statistically close.

We now distinguish the following cases (since H is only allowed to have ports $\mathsf{out}?$, $\mathsf{in}!$, $\mathsf{in}^{\triangleleft}!$ and ports going to the adversary):

1. H has port $\mathsf{in}!$ or $\mathsf{in}^{\triangleleft}!$ (and ports to the adversary).
2. H has neither $\mathsf{in}!$ nor $\mathsf{in}^{\triangleleft}!$.

We first examine case 1. The machine M_0 is only activated when a message to M_0 is sent via $\mathsf{in}!$ and is scheduled via $\mathsf{in}^{\triangleleft}!$. So any activation of M_0 implies a prior activation of H (and thus an entry in H's view). Since M_0 behaves identically to M_0' for the first $2^k - 1$ activations, we can replace M_0 by M_0' without changing the first $2^k - 1$ entries of H's view. Formally

$$view_{\{\mathsf{H},\mathsf{A}_1,\mathsf{M}_0\},2^k-1}(\mathsf{H}) = view_{\{\mathsf{H},\mathsf{A}_1,\mathsf{M}_0'\},2^k-1}(\mathsf{H}),$$

and so for any polynomial l,

$$\Delta_{\mathsf{stat}}\big(view_{\{\mathsf{H},\mathsf{A}_1,\mathsf{M}_0\},l}(\mathsf{H}),\ view_{\{\mathsf{H},\mathsf{A}_1,\mathsf{M}_0'\},l}(\mathsf{H})\big)$$

vanishes for sufficiently large k and therefore is in particular negligible. So setting $\mathsf{A}_2 := \mathsf{A}_1$, we have found a simulator.

Now let us consider case 2. Here the adversary A_1 has ports $\mathsf{in}!$ and $\mathsf{in}^{\triangleleft}!$. Let now A_2 be identical to A_1 with the exception, that any output on $\mathsf{in}!$ and $\mathsf{in}^{\triangleleft}!$ is suppressed. Since M_0 ignores these outputs anyway, this does not change the view of H. In the resulting network, no message is ever sent to the machine M_0, so we can replace M_0 by M_0' without changing H's view. I.e.,

$$view_{\{\mathsf{M}_0,\mathsf{H},\mathsf{A}_1\}}(\mathsf{H}) = view_{\{\mathsf{M}_0,\mathsf{H},\mathsf{A}_2\}}(\mathsf{H}) = view_{\{\mathsf{M}_0',\mathsf{H},\mathsf{A}_1\}}(\mathsf{H})$$

which shows that A_2 is a simulator for A_1 in Case 2. □

To disprove the composition theorem, we now construct a protocol $\hat{M}_1$ that uses $\hat{M}_0$, and show that in that context, $\hat{M}_0$ may not be replaced by $\hat{M}'_0$ without loss of security.

The machine M_1 is a machine with ports $\{\mathsf{in}!, \mathsf{in}^{\triangleleft}!, \mathsf{adv}?, \mathsf{result}!, \mathsf{result}^{\triangleleft}!\}$, i.e., the machine has two outgoing connections in! and result! that it schedules itself, as well as an incoming connection adv? (cf. Figure 1). (The seeming misnomer in! for an outgoing connection stems from the fact that this port will later be connected to the in-port in? of M_0.)

The machine M_1 has the following program:

- Upon the i-th message (denoted m here) via adv?, where $i \le 2^k$, send the message m on in! (and deliver it by sending 1 on $\mathsf{in}^{\triangleleft}!$.
- Upon the i-th message via adv?, where $i > 2^k$, send a message done on result! (and deliver it by sending 1 on $\mathsf{result}^{\triangleleft}!$).

The protocol $\hat{M}_1$ is then defined to contain only the machine M_1. The honest user is allowed access to the in! and result! connection of M_1, but the connection adv! is only visible to the adversary.

Formally, the protocol is defined to be the structure $(\hat{M}_1, S_1)$ with $\hat{M}_1 = \{\mathsf{M}_1\}$, and $S_1 = \{\mathsf{in}^{\leftrightarrow}!, \mathsf{result}^{\leftrightarrow}!\}$.

We can now examine the composition of the protocols $\hat{M}_1$ and $\hat{M}_0$. This composition (as depicted in Figure 2) yields a protocol $\hat{M}_{10} = \{\mathsf{M}_1, \mathsf{M}_0\}$. The honest user may connect to out! and result!, but not to adv?.

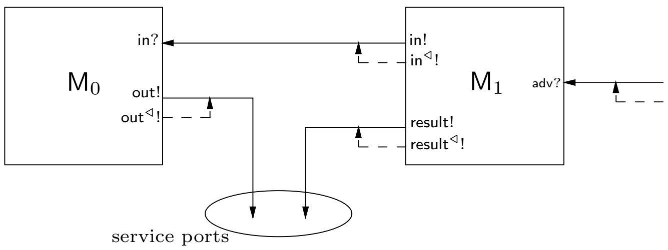

Fig. 2. The composed protocol $\hat{M}_{10}$. The honest user may only connect to the service ports.

Similarly, we have the composition $\hat{M}'_{10}$ of $\hat{M}_1$ and $\hat{M}'_0$.

We can now show the

Lemma 5. *It is* $(\hat{M}_{10}, S) \not\geq_{\mathsf{sec}}^{NEGL} (\hat{M}'_{10}, S)$*, i.e., the protocol* $\hat{M}_{10}$ *is not statistically as secure as* $\hat{M}'_{10}$.

Proof. The protocol $\hat{M}_{10}$ behaves as follows: The first 2^k input messages from the adversary (on adv?) are forwarded from M_1 to M_0, where they are ignored. Every further input on adv? results in a message done sent to the honest user.

Consider the following adversary A_1: It has the ports $\mathsf{adv}!, \mathsf{adv}^{\triangleleft}!, \mathsf{clk}^{\triangleleft}?$.[4] In each of its first 2^k+1 activations, A_1 sends a message `ping` on $\mathsf{adv}!$ to M_1. After the 2^k+1-th activation, A_1 halts. The honest user H is defined to have the ports $\mathsf{result}?$, $\mathsf{out}?$. The honest user simply reads all incoming messages (which implies that these messages are added to its view).

Then in a run of the protocol $\hat{M}_{10}$ with A_1 and H, the first 2^k messages from A_1 will be transmitted via M_1 to M_0 and then ignored, while the (2^k+1)-st message will trigger a message `done` from M_1 to H.

So, when running with $\hat{M}_{10}$ and A_1, the view of H consists only of one incoming message `done` on $\mathsf{result}?$.

Now consider the protocol $\hat{M}'_{10}$: The first 2^k inputs from the adversary (on $\mathsf{adv}?$) are forwarded from M_1 to M'_0. Upon the 2^k-th of these, M'_0 will send `alive` via $\mathsf{out}!$ to H. Only upon the (2^k+1)-st message via $\mathsf{adv}?$, a message is sent via $\mathsf{result}!$ to the honest user.

Therefore, for any simulator A_2, if the view of H (running with $\hat{M}'_{10}$ and A_2) contains a message `done` on $\mathsf{result}?$, it also contains a message `alive` on $\mathsf{out}?$ at some earlier point of its view. Thus, no simulator can mimic the view of H when running with $\hat{M}_{10}$ and A_1.

This shows that $\hat{M}_{10}$ is not statistically as secure as $\hat{M}'_{10}$. $\square$

Lemmas 4 and 5 are easily adapted to the case of universal statistical security (universal security means that the simulator only depends on the adversary, not on the honest user). Furthermore, the used class *NEGL* of small functions can be substituted by, e.g., the class *EXPSMALL*.

The composition theorem states that if $\hat{M}_0$ is statistically as secure as $\hat{M}'_0$, then the composed protocol $\hat{M}_{10}$ is statistically as secure as the composed protocol $\hat{M}'_{10}$. Thus we get from Lemmas 4 and 5 the

Corollary 6. *The composition theorem of [PW01,Bac02] does not hold for statistical security.*

To see why the proof of [PW01] of the composition theorem fails in this case, see the comments in our proof of Theorem 8.

3.1 Further difficulties

In this section we sketch some further problems arising from Definition 3 to show why we believe that not the composition theorem but the details of Definition 3 should be fixed.

In [Bac02, Section 3.2], a variant of the security notion was introduced, which essentially consists of restricting the set of possible honest users to such machines which connect to *all* service ports (in the normal definition of security, the honest user connects to a subset of the service ports). It was then shown that this modified notion of security is equivalent to the old one. Again, using a counterexample very similar to that of the preceding section, one can show that this does not hold with respect to statistical security.

[4] The master clock port $\mathsf{clk}^{\triangleleft}?$ is a special port marking the so-called master scheduler. This machine is always activated when no message is to be delivered.

We very roughly sketch the counterexample: Let M_i be a machine with ports $\mathsf{in}!$, $\mathsf{out}!$, $\mathsf{out}^{\triangleleft}!$. At the 2^k-th activation of M_i, it outputs i on port $\mathsf{out}!$ (and triggers immediate delivery by writing on $\mathsf{out}^{\triangleleft}!$). Then let $\hat{M}_i := \{\mathsf{M}_i\}$ be the protocol consisting only of M_i and where all ports are service ports, i.e., the honest user can (and—in the modified definition—*must*) connect to $\mathsf{in}?$ and $\mathsf{out}!$. Now, in the modified definition, $\hat{M}_1$ is statistically as secure as $\hat{M}_2$, since the honest user will not see a different reaction from M_1 than from M_2 within a polynomial prefix of its view. However, when allowing honest users which connect only to a subset of the service ports, the following attack is possible: the honest user connects only to $\mathsf{out}!$, while the real adversary activates M_1 at least 2^k times through $\mathsf{in}?$. Then, in its first activation, the honest user gets the message "1" from M_1, which cannot happen with simulator machine M_2 (which can only send message "2"). So $\hat{M}_1$ is not statistically as secure as $\hat{M}_1$ with respect to the old notion (Definition 3).

In [Bac02, Section 3.1], another lemma states that restricting honest users so that they may only have one connection to and one connection from the adversary does not change the security notion. Though we could not construct a counterexample, the proof of that statement does not go through in the statistical case using Definition 3.[5]

These two examples together with the invalidity of the composition theorem should give enough motivation for changing the definition of statistical security. Such a changed definition will be presented in the next section.

4 The Modified Notion

To address these problems of the definition of statistical security from [PW01,BPW04], we present a new one. Technically, we vary Definition 3 only slightly: instead of requiring that only polynomial prefixes of the honest user H's view in real and ideal executions have small statistical distance, we require that the statistical distance between the whole of H's views is small.[6]. As will be discussed, this coincides with the requirement that the statistical distance between all families of finite prefixes of H-views is small.

But even though the definitional change is only minor, its implications are major. First, we show below that this modified notion allows for composition. Second, complications which the original notion caused in proofs (see above) do not arise with our modified definition.

Third, the intuitive security promise of the new notion is noticably stronger: with statistical security in the sense of Definition 3, only polynomial prefixes of user-views are considered. A protocol may be statistically secure in that sense even if it loses every intuitive security property after, say, 2^k input messages.[7] As shown in the proof of Lemma 5, such

[5] In the proof of [Bac02, Section 3.1, Theorem 3.1] it is used that (using the notation of that proof) from $view_{\mathsf{conf}_{\mathsf{A\_H},1}}(\mathsf{H}_{\mathsf{A\_H}}) \approx view_{\mathsf{conf}_{\mathsf{A\_H},2}}(\mathsf{H}_{\mathsf{A\_H}})$ it follows $view_{\mathsf{conf}_{\mathsf{A\_H},1}}(\mathsf{H}) \approx view_{\mathsf{conf}_{\mathsf{A\_H},2}}(\mathsf{H})$ where H is a submachine of $\mathsf{H}_{\mathsf{A\_H}}$. However, as detailed in the proof of Theorem 8 in the next section, such a conclusion is not valid if $\approx$ means statistical indistinguishability of polynomial prefixes.

[6] Note that this is a statistical distance between two random variables with non-countable domain.

[7] Of course, when constructing such a protocol, care has to be taken for the cases in which the adversary A connects to service ports—formally, this is allowed.

protocols can break down under composition with a larger protocol that only sparsely communicates with the honest user.

In contrast to this, the new notion requires that H's complete views in real and ideal runs are statistically close. This in particular excludes protocols that are secure only for, say, 2^k invocations. Rather, a protocol must remain secure after an arbitrary number of input messages to achieve statistical security in the new sense.

For example, consider a protocol which allows an arbitrary number of invocations, and in each single invocation gets insecure with probability 2^{-k}. Such a protocol may be secure w.r.t. the old notion, but is certainly insecure w.r.t. the new notion. To see this, note that for the new notion, even prefixes which cover, say, 2^{2k} protocol continuous protocol executions initiated by a suitable honest user are considered. With overwhelming probability, the protocol gets insecure in at least one of these 2^{2k} executions.

This property of our new notion captures a natural requirement for secure composition with larger, unbounded protocols. See below for alternatives to our formulation that only deal with polynomial prefixes of user-views, but impose restrictions on protocols which are allowed for composition.

Now we turn to the actual definition of our modified notion of statistical security, which we call "strict statistical security."

Definition 7 (Strict statistical security). *Let $(\hat{M}_1, S)$ and $(\hat{M}_2, S)$ be structures with identical service port sets S. We say that $(\hat{M}_1, S)$ is* strictly statistically as secure as *$(\hat{M}_2, S)$ for a class SMALL of small functions, written $(\hat{M}_1, S) \geq_{\mathsf{sec}}^{\mathsf{s},SMALL} (\hat{M}_2, S)$, if the following holds:*

For every configuration $conf_1 = (\hat{M}_1, S, \mathsf{H}, \mathsf{A}_1) \in \mathsf{Conf}^{\hat{M}_2}(\hat{M}_1, S)$, there is a configuration $conf_2 = (\hat{M}_2, S, \mathsf{H}, \mathsf{A}_2) \in \mathsf{Conf}(\hat{M}_2, S)$ such that

$$\Delta_{\mathsf{stat}}(view_{conf_1}(\mathsf{H}), view_{conf_2}(\mathsf{H})), \tag{2}$$

as a function in the security parameter k, is contained in SMALL.

In other words, we demand that for every real adversary A_1 and user H, there is a simulator A_2, such that the statistical difference of H's views in real and ideal model is small in the security parameter. Note that on the technical side, the only difference between Definitions 3 and 7 is that Definition 3 considers only polynomial prefixes of user-views, whereas Definition 7 considers the user-views as a whole.

Universal and black-box flavors of this security definition are derived as usual (e.g., for the universal case, we demand that A_2 does not depend on H). Similarly, this notion can be lifted to systems, i.e., sets of protocols which capture several different corruption situations.

We remark that requiring the term in (2) to be in *SMALL* is equivalent to requiring that the statistical distance of the $\ell(k)$-step prefixes $view_{conf_1,\ell(k)}(\mathsf{H})$ and $view_{conf_1,\ell(k)}(\mathsf{H})$ lies in *SMALL* for *all* functions $\ell : \mathbb{N} \to \mathbb{N}$. (This is straightforward from Lemma 2(ii).) This observation may be of practical interest when conducting proofs, since the latter requirement may be easier to show.

In view of this remark, strictly statistical security obviously implies statistical security as in Definition 3. However, the converse does not hold. Consider the protocols $\hat{M}_0$ and $\hat{M}'_0$ from Section 3. For these, we have shown that $\hat{M}_0$ is statistically as secure as $\hat{M}'_0$ (w.r.t. Definition 3), but this does not hold w.r.t. strict statistical security.

A corresponding attack would simply consist of activating the machine M_0 (resp. M'_0) 2^k times and waiting for an `alive` output on port `out`?. With our notion of security, such an output is considered for distinction of $\hat{M}_0$ and $\hat{M}'_0$, since the whole of H's view is regarded (not only polynomial prefixes, as with Definition 3).

Of course, it is crucial to validate that the composition theorem holds for the new notion. In fact, we only need to re-check the original proof from [PW01] for this notion. Note that it suffices to prove the composition theorem for structures, as it can then be lifted to systems.

In the formulation and the proof of the theorem, we will make use of the composition operator "||" for structures defined in [PW01] (cf. also Appendix A; informally, "||" simply combines two protocols so that they may use each other).

Theorem 8. *Let structures $(\hat{M}_0, S_0)$ and $(\hat{M}'_0, S_0)$ with the same set of service ports S_0 be given. Let furthermore $(\hat{M}_1, S_1)$ be a structure that is composable with both $(\hat{M}_0, S_0)$ and $(\hat{M}'_0, S_0)$. Let SMALL be a class of small functions. Then $(\hat{M}_0, S_0) \geq_{\mathsf{sec}}^{\mathsf{s},SMALL} (\hat{M}'_0, S_0)$ implies $(\hat{M}_0, S_0)||(\hat{M}_1, S_1) \geq_{\mathsf{sec}}^{\mathsf{s},SMALL} (\hat{M}'_0, S_0)||(\hat{M}_1, S_1)$.*

Proof. The proof is the same as in [PW01]; here, we focus on the places where the original definition of statistical security causes problems.

We write $(\hat{M}^{\#}, S) := (\hat{M}_0, S_0)||(\hat{M}_1, S_1)$ and $(\hat{M}^*, S) := (\hat{M}'_0, S_0)||(\hat{M}_1, S_1)$. We have to show that for every suitable configuration $conf^{\#} = (\hat{M}^{\#}, S, \mathsf{H}, \mathsf{A}^{\#})$ of $(\hat{M}^{\#}, S)$ (i.e., for every honest user H and adversary $\mathsf{A}^{\#}$), there is a corresponding configuration $conf^*$ of $(\hat{M}^*, S)$ (i.e., a simulator A^*), such that the statistical distance

$$\Delta_{\mathsf{stat}}(view_{conf^{\#}}(\mathsf{H}), view_{conf^*}(\mathsf{H})) \tag{3}$$

is, as a function in k, contained in $SMALL$.

The basic proof idea is depicted in Figure 3.

So let a configuration $conf^{\#} = (\hat{M}^{\#}, S, \mathsf{H}, \mathsf{A}^{\#}) \in \mathsf{Conf}^{\hat{M}^*}(\hat{M}^{\#}, S)$ be given. Combining the machines in $\hat{M}_1$ and the honest user H yields a new honest user H_0. Together with the adversary $\mathsf{A}_0 := \mathsf{A}^{\#}$, this forms a new configuration $conf_0 = (\hat{M}_0, S_0, \mathsf{H}_0, \mathsf{A}_0) \in \mathsf{Conf}^{\hat{M}'_0}(\hat{M}_0, S_0)$.

By the combination lemma (Lemma 2.1 in [PW01]), the views of H in these configurations are identical.[8] Formally, we have

$$view_{conf^{\#}}(\mathsf{H}) = view_{conf_0}(\mathsf{H}). \tag{4}$$

[8] Note however, that in $conf_0$, H is only a submachine of the honest user H_0.

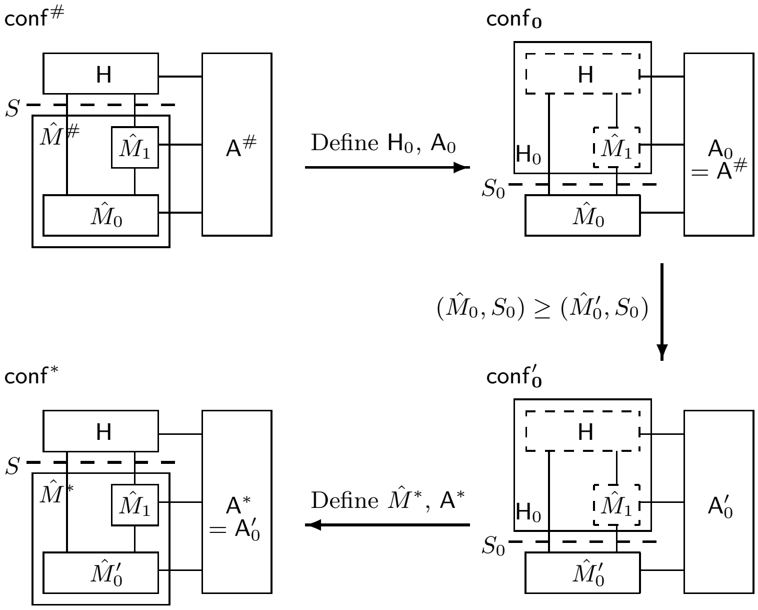

Fig. 3. Proof sketch for the composition theorem reproduced from [PW01]

Since $(\hat{M}_0, S_0) \geq_{\mathsf{sec}}^{\mathsf{s},\mathit{SMALL}} (\hat{M}_0', S_0)$ by assumption, there exists an adversary A_0' and a configuration $\mathit{conf}_0' = (\hat{M}_0', S_0, \mathsf{H}_0, \mathsf{A}_0') \in \mathsf{Conf}(\hat{M}_0', S_0)$ such that

$$\Delta_{\mathsf{stat}}(\mathit{view}_{\mathit{conf}_0}(\mathsf{H}_0), \mathit{view}_{\mathit{conf}_0'}(\mathsf{H}_0)) \tag{5}$$

is, as a function in k, contained in SMALL.

The crucial point here is that $(\hat{M}_0, S_0) \geq_{\mathsf{sec}}^{\mathsf{s},\mathit{SMALL}} (\hat{M}_0', S_0)$ only guarantees that H_0's views are close (meaning that the statistical difference lies in SMALL). For going further, we have to deduce that also H's views are close.

Fortunately, since H is a submachine of H_0, and thus $\mathit{view}(\mathsf{H})$ is a function of $\mathit{view}(\mathsf{H}_0)$, this is directly implied by Lemma 2 (i). Thus,

$$\Delta_{\mathsf{stat}}(\mathit{view}_{\mathit{conf}_0}(\mathsf{H}), \mathit{view}_{\mathit{conf}_0'}(\mathsf{H})) \tag{6}$$

is, as a function in k, contained in SMALL.

However, this implication is in general not valid when dealing with statistical security in the sense of Definition 3: there, $(\hat{M}_0, S_0) \geq_{\mathsf{sec}}^{\mathit{SMALL}} (\hat{M}_0', S_0)$ means that only polynomial prefixes of H_0-views are close. But not every polynomial H-prefix may be contained in a polynomial H_0-prefix, e.g., when H's activations are "sparse" w.r.t. H_0-activations. (The example from Section 3 shows a simple situation in which this is the case.)

So here we cannot conclude that all polynomial H-prefixes are statistically close. This explains why the original proof from [PW01] cannot go through for the original notion of statistical security.

To carry on with the proof for strict statistical security, we decompose the honest user H_0 into H and the machines from $\hat{M}_1$. The resulting configuration $conf^*$ thus consists of user H, adversary $\mathsf{A}^* := \mathsf{A}'_0$ and the composed structure $(\hat{M}^*, S) = (\hat{M}'_0, S_0)||(\hat{M}_1, S_1)$. Again, the combination lemma ensures that this decomposition does not affect the views of any machine; in particular, we have

$$view_{conf^*}(\mathsf{H}) = view_{conf'_0}(\mathsf{H}). \tag{7}$$

Combining (4,6,7), we find that (3) is in fact a small function in k, which finishes the proof. □

This shows that our notion behaves well under composition. Inspection of the proofs in Sections 3.1 and 3.2 of [Bac02] shows furthermore that the problems depicted in Section 3.1 of this work (which arise with the original definition of statistical security) vanish with our definition of strictly statistical security.

However, the approach for modifying statistical security that we chose for Definition 7 is certainly not the only one imaginable. In particular, one may be interested in a composable definition that only considers polynomial prefixes of user-views (as did the original definition). This might be appreciable in situations in which a protocol is guaranteed to be used in larger protocols only a polynomial number of times.

In fact, if one restricts to protocols $\hat{M}_1$ that are polynomial-time, then the composition theorem of [PW01,BPW04] holds for the original statistical security definition.[9] As explained in the proof of Theorem 8, the situation gets problematic only when the larger protocol $\hat{M}_1$ is not polynomial-time (and thus, in the notation of that proof, H's view might be "sparse" in H_0's view).

Alternatively, one could think of restricting to users, adversaries and protocol machines which halt after a polynomial number of activations (but need not be computationally bounded in each single activation). With such a restriction, only users with polynomial-sized views are considered, and thus, statistical security in the sense of Definition 3 is then equivalent to that of Definition 7.

5 Conclusion

We have shown that the original notion of statistical security for multi-party protocols from [PW01,BPW04] does not compose. Furthermore, we have depicted problems in proofs which this notion causes.

As a possible solution, we have introduced an alternative definition of statistical security which we have then proved to behave well under composition. The mentioned problems in proofs do not appear with our notion.

[9] Note however, that the proof problems mentioned in Section 3.1 remain with the original notion of statistical security even restricting to strictly polynomial protocols.

Acknowledgements

We thank Jörn Müller-Quade for valuable discussions. This work was partially funded by the EC project ProSecCo under IST-2001-39227.

A Reactive Simulatability

Here we review the notion of reactive simulatability. This introduction only very roughly sketches the definitions, and the reader is encouraged to read [BPW04] for more detailed information and formal definitions.

Reactive simulatability is a definition of security which defines a protocol $\hat{M}_1$ (the *real protocol*) to be *as secure as* another protocol $\hat{M}_2$ (the *ideal protocol*, the *trusted host*), if for any adversary A_1 (also called the *real adversary*), and any *honest user* H, there is a *simulator* A_2 (also called the *ideal adversary*), s.t. the view of H is indistinguishable in the following two scenarios:

- The honest user H runs together with the real adversary A_1 and the real protocol $\hat{M}_1$
- The honest user H runs together with the simulator A_2 and the ideal protocol $\hat{M}_2$.

Note that there is a security parameter k common to all machines, so that the notion of indistinguishability makes sense.

This definition allows to specify some trusted host—which is defined to be a secure implementation of some cryptographic task—as the ideal protocol, and then to consider the question, whether a real protocol is as secure as the trusted host (and thus also a secure implementation of that task). In order to understand the above definitions in more

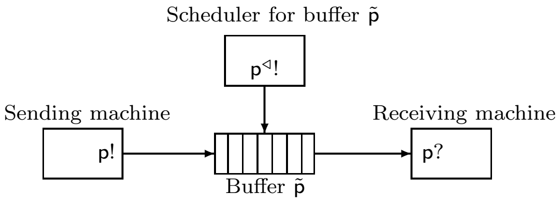

Fig. 4. A connection

detail, we have to specify what is meant by machines "running together". Consider a set of machines (called a *collection*). Each machine has so-called *simple in-ports* (written $\mathsf{p}?$), *simple out-ports* (written $\mathsf{p}!$), and *clock out-ports* (written $\mathsf{p}^{\triangleleft}!$). Ports with the same name (p in our example) are considered to belong together and are associated with a *buffer* $\tilde{\mathsf{p}}$. These are then interconnected as in Figure 4 (note that some or all ports may originate from the same machine). Now when a collection runs, the following happens: at every point in time, exactly one machine is activated. It may now read its simple in-ports (representing incoming network connections), do some work, and then write output to its simple out-ports. After such an activation the contents of the simple out-ports $\mathsf{p}!$ are appended to the queue of messages stored in the associated buffer $\tilde{\mathsf{p}}$. However, since now all messages are stored in buffers and will not be delivered by themselves, machines additionally have after each activation the possibility to write a number $n \geq 1$ to at most one clock out-port $\mathsf{p}^{\triangleleft}!$. Then the n-th undelivered message of buffer $\tilde{\mathsf{p}}$ will be written to the simple in-port $\mathsf{p}?$ and deleted from the buffer's queue. The machine that has the simple in-port $\mathsf{p}?$ will be activated next. So the clock out-ports control the scheduling. Usually, a connection is clocked by (i.e., the corresponding clock out-port is part of) the sender, or by the adversary. Since the most important use of a clock out-port is to write a 1 onto it ("deliver the oldest message in the buffer"), we say a machine schedules a connection or a message when a machine writes a 1 onto the clock port of that connection.

At the start of a run, or when no machine is activated at some point, a designated machine called the *master scheduler* is activated. For this, the master scheduler has a special port, called the *master clock port* $\mathsf{clk}^{\triangleleft}?$.

Note that not all collections can be executed, only so-called *closed* collections, where all connections have their simple in-, simple out-, and clock out-port. If a collection is not closed, we call the ports having no counterpart *free ports*.

In order to understand how this idea of networks relates to the above sketch of reactive simulatability, one has to get an idea of what is meant by a protocol. A protocol is represented by a so-called *structure* $(\hat{M}, S)$, consisting of a collection $\hat{M}$ of the protocol participants (parties, trusted hosts, etc.), and a subset of the free ports of $\hat{M}$, the so-called *service ports* S.[10] The service ports represent the protocol's interface (the connections to the protocol's users). The honest user can then only connect to the service ports (and to the adversary), all other free ports of the protocol are intended for the communication with the adversary (they may e.g. represent side channels, possibilities of attack, etc.). Since usually a protocol does not explicitly communicate with an adversary, such free non-service ports are more commonly found with trusted hosts, explicitly modelling their imperfections.

With this information, we can review the above "definition" of security. Namely, the honest user H, the adversary, and the simulator are nothing else but machines, and the protocols are structures. The view of H is then the restriction of the run (the transcripts of all states and in-/output of all machines during the protocols execution, also called trace) to the ports and states of H.

The definition, as presented so far, still has one drawback. We have not introduced the concept of a corruption. This can be accommodated by defining so-called systems. A *system* is a set of structures, where to each "corruption situation" (set of machines, which are corrupted) corresponds one structure. That is, when a machine is corrupted, it is not present anymore in the corresponding structure, and the adversary takes its place. For a trusted host, the corresponding system usually consists of structures for each corruption situation, too, where those connections of the trusted host that are associated with a corrupted party, are under the control of the adversary.

[10] The exact definition of *service ports* is a little complicated, since it gives the ports of the buffers the honest user can connect to, not the ports of the protocol machines. On an intuitive level however, one can image that the service port indicate the protocol parties' ports the honest user can use.

We can now refine the definition of security as follows: A *real system* Sys_1 is as secure as an *ideal system* Sys_2, if every structure in Sys_1 is as secure as the corresponding structure in Sys_2.

A major advantage of a security definition by simulatability is the possibility of *composition*. The notion of composition can be sketched as follows: If we have one structure or system A (usually a protocol) implementing some other structure or system B (usually some primitive), and we have some protocol X^B (having B as a sub-protocol, i.e. using the primitive), then by replacing B by A in X^B, we get a protocol X^A which is as secure as X^B. This allows to design protocols modularly: first we design a protocol X^B, and then we find an implementation for B.

Since formally, it is not important which protocol is the outer and which the inner one in the composition, we write the composition of structures in a more symmetric fashion: $(\hat{M}_1, S)\|(\hat{M}_0, S)$ denotes the composition of structures $(\hat{M}_1, S)$ and $(\hat{M}_0, S)$ (instead of writing the cumbersome $(\hat{M}_1, S)^{(\hat{M}_0,S)}$).

A.1 Glossary

In this section we explain the technical terms used in this paper. Longer and formal definitions can be found in [BPW04].

$[\hat{C}]$: Completion of the collection $\hat{C}$. Results from adding all missing buffers to $\hat{C}$. **buffer**: Stores message sent from a simple out- to a simple in-port. Needs an input from a clock port to deliver. **clock out-port $\mathsf{p}^{\triangleleft}!$**: A port used to schedule connection p. **closed collection**: A collection is closed if all ports have all their necessary counterparts. **collection**: A set of machines. **combination**: The combination of a set of machines is a new machine simulating the other machines. A set of machines can be replaced by its combination without changing the view of any machine. **composition**: Replacing sub-protocols by other sub-protocols. **computational security**: When in the security definition, honest user and adversary are restricted to machines running in polynomial time, and the views are computationally indistinguishable. **configuration**: A structure together with an honest user and an adversary. **$\mathsf{Conf}(\hat{M}_2, S)$**: Set of ideal configurations that are possible for structure $(\hat{M}_2, S)$. **$\mathsf{Conf}^{\hat{M}_2}(\hat{M}_1, S)$**: Set of real configurations possible for structure $(\hat{M}_1, S)$ when comparing it with ideal protocol $\hat{M}_2$. ***EXPSMALL***: The set of exponentially small functions. **free ports**: The free ports of a collection are those missing their counterpart. **honest user**: Represents the setting in which the protocol runs. Also called environment. **master clock port $\mathsf{clk}^{\triangleleft}?$**: A special port by which the master scheduler is activated. **master scheduler**: The machine that gets activated when no machine would get activated. ***NEGL***: The set of negligible functions (asymptotically smaller than the inverse of any polynomial). **perfect security**: When in the security definition, the real and ideal run have to be identical, not only indistinguishable. Further the machines are completely unrestricted. **$\mathsf{ports}(M)$**: The set of all ports a machine or collection M has. **run**: The transcript of everything that happens while a collection is run. Formally a random variable over sequences. $run_{\mathsf{conf},k,l}$ is the random variable of the run when running the configuration conf upon security parameter k, restricted to its first l elements. If k is omitted, a family of random variables is meant. If l is omitted, we mean the full run. **service ports**: The ports of a structure to which the honest user may connect. They represent the interface of the protocol. As service ports are most often ports of a buffer, they are sometimes specified through the set S^c of their complementary ports; S^c consists of all ports which directly connect to a service port. **simple in-port p?**: A port of a machine, where it can receive messages from other machines. **simple out-port p!**: As simple in-port, but for sending. **statistical security**: When in the security definition the statistical distance of polynomial prefixes of the views have a statistical distance which lies in a set of small functions *SMALL* (in the security parameter k). Usually $SMALL = NEGL$. Further the machines are completely unrestricted. (See also Definition 3.) **structure**: A collection together with a set of service ports, represents a protocol. **trace**: Synonym for *run*. **view**: A subsequence of the run. The $view(M)$ of some collection or machine M consists of the run restricted to the ports and states of M. Possible indices are as with runs.

B Proof of Lemma 2 (ii)

Let $\mu(M) := \mathsf{Pr}[X \in M] - \mathsf{Pr}[Y \in M]$ for measurable $A \subseteq \Omega$. Then μ is a finite signed measure and therefore there is a Hahn decomposition for μ [Hal74, § 29, Th. A], i.e., a partition of Ω into disjoint measurable sets Ω^+ and Ω^-, s.t. $\mu(A) \geq 0$ for all measurable $A \subseteq \Omega^+$, and $\mu(A) \leq 0$ for all measurable $A \subseteq \Omega^-$. Therefore $\mu(\Omega^+)$ is maximal.

Let A be a measurable subset of T^n. Then $A' := \{\alpha\omega : \alpha \in A, \omega \in T^{\mathbb{N}}\}$, i.e., A' is the set of all sequences beginning with words in A. Then the set

$$\mathcal{A}_0 := \{A' : n \in \mathbb{N},\ A \subseteq T^n \text{ measurable}\}$$

is an algebra and generates the σ-algebra $\mathcal{A}$ of measurable subsets of $T^{\mathbb{N}} = \Omega$.

Let now $\tilde{\mu}(A) := \mu(A \cap \Omega^+) - \mu(A \cap \Omega^-)$ for measurable $A \subseteq \Omega$. Since $\tilde{\mu}$ is a finite measure, there are $C_n \in \mathcal{A}_0$ that approximate Ω^+ [Hal74, § 13, Th. D] in the sense that

$$\lim_{n\to\infty} \tilde{\mu}\big((\Omega^+ \setminus C_n) \cup (C_n \setminus \Omega^+)\big) = 0.$$

However

$$\begin{aligned}
&\tilde{\mu}\big((\Omega^+ \setminus C_n) \cup (C_n \setminus \Omega^+)\big)\\
&\quad = \mu\Big(\big((\Omega^+ \setminus C_n) \cup (C_n \setminus \Omega^+)\big) \cap \Omega^+\Big) - \mu\Big(\big((\Omega^+ \setminus C_n) \cup (C_n \setminus \Omega^+)\big) \cap \Omega^-\Big)\\
&\quad = \mu(\Omega^+ \setminus C_n) - \mu(C_n \cap \Omega^-).
\end{aligned}$$

Since $\mu(\Omega^+ \setminus C_n) \geq 0$ and $\mu(C_n \cap \Omega^-) \leq 0$ and the difference goes to 0, it is

$$\lim_{n\to\infty} \mu(\Omega^+ \setminus C_n) = \lim_{n\to\infty} \mu(C_n \cap \Omega^-) = 0$$

and thus

$$\lim_{n\to\infty} \mu(C_n) = \lim_{n\to\infty} \mu(\Omega^+) - \mu(\Omega^+ \setminus C_n) + \mu(C_n \cap \Omega^-) = \mu(\Omega^+).$$

Since $C_n \in \mathcal{A}_0$, there is a $t_n \in \mathbb{N}$, s.t. $C_n = A'$ for some $A \subseteq T^{t_n}$, and then

$$\begin{aligned}
\mu(C_n) = \mu(A') &= \mathsf{Pr}\left[X \in A'\right] - \mathsf{Pr}\left[Y \in A'\right] \\
&= \mathsf{Pr}\left[X_{1\ldots t_n} \in A\right] - \mathsf{Pr}\left[Y_{1\ldots t_n} \in A\right] \\
&\leq \sup_{\substack{A \subseteq T^{t_n} \\ A \text{ measurable}}} \left|\mathsf{Pr}\left[X_{1\ldots t_n} \in A\right] - \mathsf{Pr}\left[Y_{1\ldots t_n} \in A\right]\right| \\
&= \Delta_{\mathsf{stat}}(X_{1\ldots t_n}, Y_{1\ldots t_n})
\end{aligned}$$

and therefore (recall that $\mu(\Omega^+)$ was maximal)

$$\begin{aligned}
\Delta_{\mathsf{stat}}(X, Y) = \mu(\Omega^+) &= \lim_{n\to\infty} \mu(C_n) \\
&\leq \lim_{n\to\infty} \Delta_{\mathsf{stat}}(X_{1\ldots t_n}, Y_{1\ldots t_n}) \\
&\leq \sup_{t\in\mathbb{N}} \Delta_{\mathsf{stat}}(X_{1\ldots t}, Y_{1\ldots t})
\end{aligned}$$

Since further $\Delta_{\mathsf{stat}}(X, Y) \geq \sup_{t\in\mathbb{N}} \Delta_{\mathsf{stat}}(X_{1\ldots t}, Y_{1\ldots t})$ by Lemma 2 (i), we have

$$\Delta_{\mathsf{stat}}(X, Y) = \sup_{t\in\mathbb{N}} \Delta_{\mathsf{stat}}(X_{1\ldots t}, Y_{1\ldots t})$$

which concludes the proof. □

References

[Bac02] Michael Backes. *Cryptographically Sound Analysis of Security Protocols*. PhD thesis, Universität des Saarlandes, 2002.

[BPW04] Michael Backes, Birgit Pfitzmann, and Michael Waidner. Secure asynchronous reactive systems. IACR ePrint Archive, March 2004.

[Can00] Ran Canetti. Security and composition of multi-party cryptographic protocols. *Journal of Cryptology*, 3(1):143–202, 2000.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42th Annual Symposium on Foundations of Computer Science, Proceedings of FOCS 2001*, pages 136–145. IEEE Computer Society, 2001.

[Hal74] Paul R. Halmos. *Measure Theory*. Springer-Verlag, New York, 1974.

[HU05] Dennis Hofheinz and Dominique Unruh. Comparing two notions of simulatability. In *Theory of Cryptography, Proceedings of TCC 2005*, Lecture Notes in Computer Science. Springer-Verlag, 2005. To be published.

[PW00] Birgit Pfitzmann and Michael Waidner. Composition and integrity preservation of secure reactive systems. In *7th ACM Conference on Computer and Communications Security, Proceedings of CCS 2000*, pages 245–254. ACM Press, 2000.

[PW01] Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *IEEE Symposium on Security and Privacy, Proceedings of SSP 2001*, pages 184–200. IEEE Computer Society, 2001.