Int. J. Inf. Secur.
DOI 10.1007/s10207-007-0050-x

REGULAR CONTRIBUTION

Estimating the maximum information leakage

Alessandro Aldini · Alessandra Di Pierro

Received: 31 July 2006

Abstract Preventing improper information leaks is a greatest challenge of the modern society. In this paper, we present a technique for measuring the ability of several families of adversaries to set up a covert channel. Our approach relies on a noninterference based formulation of security which can be naturally expressed by semantic models of the program execution. In our analysis the most powerful adversary is measured via a notion of approximate process equivalence. Even if finding the most powerful adversary is in general impractical, we show that this requires only a finite number of checks for a particular family of adversaries which are related to a probabilistic information flow property.

Keywords Covert channels · Approximate noninterference · Probabilistic models · Process algebra · Bisimulation semantics

This work is a full and extended version of "Noninterference and the Most Powerful Probabilistic Adversary", presented at the 6th International Workshop on Issues in the Theory of Security (WITS 2006). This work has been supported by the IST project SENSORIA funded by EU.

A. Aldini
Istituto STI, University of Urbino "Carlo Bo", Urbino, Italy
e-mail: aldini@sti.uniurb.it

A. Di Pierro
Dipartimento di Informatica,
University of Verona, Verona, Italy

A. Di Pierro (✉)
Dipartimento di Informatica, University di Pisa,
Largo Bruno Pontecorro, 3, Pisa 56127, Italy
e-mail: dipierro@di.unipi.it; dipierro@sci.univr.it

1 Introduction

Ideally, a completely secure computer system should have no covert channels, though in practice it is virtually impossible to guarantee that information flows only along authorized paths and to eliminate all those states in which information leaks to unauthorized receivers (see, e.g., [3,4,15,31,35,37]). Hence, in a realistic scenario security cannot be an absolute requirement and approximate versions of security properties would be more appropriate. Moreover, in a quantitative setting the estimation of the approximation can be used as a measure for the security of the system, as it essentially expresses the difference between the real system and an idealized perfectly secure system or, in other words, a quantitative estimation of the information flow. In terms of confidentiality, this corresponds to the amount of system's information leakage.

In this paper, we address the problem of providing a quantitative estimate of the confidentiality of a system by measuring its information leakage; this corresponds to the amount of information that is illegally revealed to an unauthorized user because of the interference of an adversary.

Formally, we base our treatment on an action-labeled transition system model which includes both probabilistic choice and nondeterminism; the latter is due to the possible interactions of the system with the environment. More precisely, our model is a mixture of the generative and reactive models of probabilities [36]. This model has been successfully used in [4] as a base for the analysis of probabilistic noninterference. The problem of measuring the confinement of a system has been addressed in [15,16] in the context of purely generative systems. The approach in [15,16] is based on the idea that checking noninterference is actually checking the indistinguishability (in terms of a given process equivalence) of the involved processes [30]; then the quantifiable

amount of interference is defined via a notion of approximate confinement, which allows for some leakage of information corresponding to a statistical measure of the power of the adversary.

In [5] we adapt the approach in [15,16] to the mixed generative and reactive model of [4]. In this setting the environment is viewed as a hostile element, and an adversary is identified with a scheduler that guides the interactions of the system with the environment. Such interactions affect both the probabilistic behavior of the system and the way in which the nondeterminism is resolved, thus causing possible covert channels. The way in which the adversary can govern such interactions determines its expressive power.

Based on this, we provide a classification of the possible adversaries interacting with the system in three different classes. For each class we give a formal definition of the interference caused by an adversary in that class, and analyze the complexity of evaluating the most powerful representative of the class. This corresponds to the adversary for which the probability that its interference is revealed by an external observer is maximal. As this requires the checking of possibly infinitely many adversary strategies, such an evaluation might in general be impractical as its cost grows factorially with the number of the states of the analyzed system. This was already noted in [6], where the approach in [5] is applied for analyzing a probabilistic protocol in order to obtain a numerical estimate of its security against probabilistic covert channels. It is therefore important to single out conditions under which this cost can be reduced and the calculation can be done effectively. An important result of our study is the introduction of a method that for one class, namely the one we call *history-dependent* adversaries, allows us to find the most powerful adversary by checking only a finite number of adversary strategies.

We then consider a probabilistic process algebra conforming the generative-reactive model, in which we can characterize the adversaries in terms of system's behaviors expressed via a transition system semantics. This probabilistic process algebra and its operational semantics were introduced in [4] in order to give a semantical characterization (based on a probabilistic process equivalence inspired by the weak bisimulation [11]) of some security properties; these are essentially probabilistic versions of the information flow properties introduced in [17] in a possibilistic setting.

In such a process algebraic framework we establish a correspondence between the probabilistic noninterference based properties of [4] and the classes of adversaries introduced in this paper. In particular, we show that the two classes that we call of *simple* and *interactive* adversaries are related to the probabilistic version of noninterference and nondeducibility on compositions, respectively. The third class, namely the class of history-dependent adversaries, has no counterpart in the existing classification of probabilistic noninterference properties. Rather, it bears a strong analogy with an information flow property based on the notion of nondeducibility on strategies which was introduced in [23] in the setting of a nondeterministic state machine model of computation. Based on this analogy, we introduce a probabilistic version of this property that we call probabilistic nondeducibility on strategies, PNDS, and we relate it to the class of history-dependent adversaries, by showing that these capture the same information leakage revealed by PNDS. The important conclusion is that if a system does not satisfy PNDS, then the observable attack that maximizes the information flow is computable with our method. As an application of this result we report on a case study in which we show how to analyze a probabilistic non-repudiation protocol and efficiently estimate its maximum information leak.

1.1 Related work

The idea of quantifying information flow has been used in [9] in the setting of cryptographic protocols. This work is the closest to ours in spirit as it addresses the computational aspects of estimating the information leakage. This estimate is obtained by measuring the distance between the different behaviors of the high user that result in different views of the low user (i.e., probability distributions) and then maximizing the resulting measurements for different behaviors of the high user. Contrary to our process algebraic approach, the model considered in [9] is automata based and consists in probabilistic state-transition machines which interact to each other asynchronously with a distributed scheduling.

In [39], it is shown how closely information leakage is related to the anonymity degree of systems; this is estimated in terms of covert channel capacity and the quantitative analysis is conducted within the framework of information theory.

The problem of estimating the anonymity of systems is also investigated in [14], where a notion of weak probabilistic anonymity is proposed, with the aim of modeling situations in which some amount of probabilistic information may be revealed due to either the interference of an attacker or the imperfection of the anonymity protocol itself. As in our framework, a formal model combining both nondeterministic and probabilistic choice is employed, which is essentially based on the probabilistic automata of [34]. The authors define a notion of α-anonymity, where α expresses the tolerated distance between the system views under the interactions with different users. If re-formulated in terms of noninterference, α-anonymity is nothing else than the notion of approximate confinement at the base of our work. However, the authors do not discuss a general approach for determining the minimum value of α that allows a given system to satisfy the α-anonymity property.

A quantitative notion of indistinguishability based on a probabilistic bisimulation semantics is also exploited in [28],

where the objective is to estimate the robustness of cryptographic protocols in the case of computational adversaries and imperfect cryptography.

Finally, in [13] the security of cryptographic protocols is estimated within the probabilistic I/O automata framework of Lynch et al. [34]. This model includes a combination of nondeterminism and probabilistic choices. With respect to our framework, the adversary is modeled as a scheduler that exhibits resource-bounded cryptographic analysis capabilities.

1.2 Outline

The rest of the paper is organized as follows. First, we describe the action-labeled transition system that is based on the generative and reactive models of probabilities of [36] (Sect. 2). We then explain how the noninterference approach is interpreted in this model and we introduce three classes of probabilistic adversaries which differ from each other for their expressive power (Sect. 3). The efficiency of such adversaries is formally stated through an approximate equivalence relation. In particular, we show how to measure the difference between the system at hand and a perfectly secure system (Sect. 4). The objective is then to establish conditions under which it is practical to estimate the interference of the most powerful adversary, i.e., the maximum probability of revealing an information leakage for each class of adversaries (Sect. 5). The material presented in these sections is an extended and revised version of [7]. We then consider a process algebraic notion of probabilistic noninterference based on the generative-reactive model. In this setting we relate the expressive power of the three classes of adversaries to the kind of attacks revealed by corresponding noninterference based properties (Sect. 6). We finally present a case study that demonstrates the relation between the expressive power of the adversary and the complexity of finding the most powerful adversary. We conclude the paper by reporting on some perspectives for future work (Sect. 8).

2 Generative-reactive transition systems

The formal model at the base of our analysis is called generative-reactive transition system (GRTS) [4,12]. A GRTS is a labeled transition system including both probabilistic and nondeterministic behaviors in a way inspired by the automata model in [33] and [38]. Each transition in a GRTS is labeled with an action name and a probability.

The actions model output and input events that allow the system to interact with the environment. Formally, we call *AType* the set of visible action types, ranged over by $a, b, \ldots$. We also use the special type τ to express an unobservable event internally executed by the system. Then the set of action names is defined as $Act = \{a_* \mid a \in AType\} \cup \{a \mid a \in AType \cup \{\tau\}\}$, where a_* denotes an input action of type a and a denotes either an output action of type a or an invisible action τ. *Act* is ranged over by $\pi, \pi', \ldots$.

The execution probability of the actions is interpreted as follows. On the one hand, the output actions and the internal actions τ are governed by a generative model of probabilities: the system autonomously decides, on the basis of a probability distribution, which output/internal action has to be performed [36]. On the other hand, the input actions follow a reactive model of probabilities: the choice of the input action type is nondeterministically left to the environment. Then, the choice of the particular input action of the chosen type, say a, is performed on the basis of a probability distribution associated with the input actions of type a offered by the system [36].

Therefore, GRTSs express both probabilistic behaviors guided by probability distributions and nondeterministic behaviors. Technically speaking, transitions leaving a state are grouped into several bundles [33]. We have a single generative bundle composed of all the transitions labeled with an output or an invisible action, and several reactive bundles, each one referring to a different action type a and consisting of all the transitions labeled with a_*. A bundle of transitions expresses a probabilistic choice guided by a probability distribution. The choice among bundles is nondeterministic.

Definition 1 A generative-reactive transition system is a tuple (S, Act, T), where S is a set of states, *Act* is a set of actions, and $T \subseteq S \times Act \times]0, 1] \times S$ is a transition relation such that:

1. $\forall s \in S, a_* \in Act. \sum\{\!|\, p \mid \exists t \in S. (s, a_*, p, t) \in T \,|\!\} \in \{0, 1\}$
2. $\forall s \in S. \sum\{\!|\, p \mid \exists a \in Act, t \in S. (s, a, p, t) \in T \,|\!\} \in \{0, 1\}$.

A rooted GRTS is a tuple (S, Act, T, s_0), such that $s_0 \in S$ is the initial state of the GRTS (S, Act, T).

The two requirements of Definition 1 say that for each bundle leaving a state—which can be either a reactive bundle of a given action type (see req. 1) or the unique generative bundle (see req. 2)—the sum of the probabilities of the transitions composing the bundle is equal to 1. In the following we restrict ourselves to finite state, finitely branching GRTSs.

Example 1 The initial state of the GRTS of Fig. 1 is made of a generative bundle which enables two transitions (labeled with the output actions i and j, respectively), and a reactive bundle of type h enabling a single transition. The choice between the bundles is nondeterministic. Instead, within each bundle, the choice is probabilistic. In particular, if an output is executed, then the choice between the two possible output

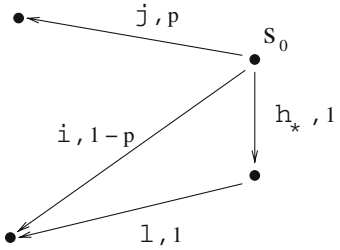

Fig. 1 A generative-reactive transition system

actions is guided by a probability distribution that assigns to j (resp. i) the execution probability p (resp. $1-p$). On the other hand, if the system executes an input of type h, then the unique input action h_* is executed with probability 1. In the following we omit the probability from the transition label whenever this is equal to 1.

In essence, the nondeterminism in a GRTS derives from the input actions that can be viewed as *incomplete* actions whose execution depends on the environment behavior. Hence, once the environment resolves such a nondeterminism (e.g., through a probabilistic choice, as exemplified in the next section), we obtain a fully specified system, i.e., a fully probabilistic transition system that does not include reactive bundles. Therefore, from such a transition system a well-defined discrete time Markov Chain [24] can be trivially derived which in turn can be analyzed in order to get performance measures of the system.

3 GRTS and noninterference

In this section, we show how probabilistic noninterference can be expressed in the formal setting of GRTSs. As it is common in the security setting, we assume that action types are classified into low level and high level. Syntactically, we denote by L the set of low-level action types, ranged over by $i, j, l, \ldots$, and with H the set of high-level action types, ranged over by $h, k, \ldots$. L and H are disjoint and form a covering of *AType*. Accordingly, we denote by Act_L (resp. Act_H) the set of low-level (resp. high-level) actions.

A low-level observer (Low, for short) can only see low-level actions. Therefore, the interactions between the system and the high-level environment represent unobservable events from the viewpoint of Low. According to the standard notion of noninterference [20], the high-level environment (which we simply refer to as the adversary) interferes with Low if the execution of the high-level actions has an observable impact on the execution of the low-level actions.

Example 2 Consider again the GRTS of Fig. 1, where either the system executes one of two possible low-level outputs, j and i with probabilities p and $1-p$ respectively, or the adversary interferes through a high-level input of type h (and in such a case the unique action h_* enabled by the system is executed with probability 1). If the adversary does not interact, the system behavior observable by Low results in the GRTS of Fig. 2a. By contrast, the adversary may interact with the system thus solving somehow the nondeterministic choice between the output events and the input. For instance, the nondeterminism can be probabilistically solved through a choice guided by a parameter q chosen by the adversary, thus resulting in the GRTS of Fig. 2b. Such a GRTS models what Low can see. In particular, the transition labeled with an input action in Fig. 1 has been turned into a transition labeled with an unobservable action τ in Fig. 2b. This is because the interaction between the system and the adversary becomes—from the viewpoint of Low—an invisible event performed by the system. Obviously, the choice of parameter q determines which probabilistic adversary actually interacts with the system.

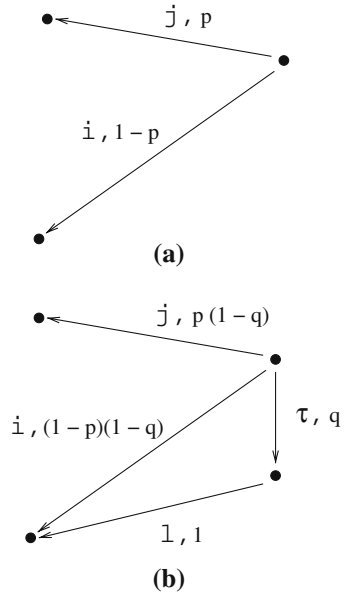

Fig. 2 Two evolutions of the GRTS of Fig. 1 which depend on the environment behavior

In the case the chosen model of communication is asynchronous, it is possible to abstract away from the high-level outputs, which represent events that are completely independent of the high-level environment. Thus, from the viewpoint of Low they cannot have any visible effect. However, if we consider a synchronous model of communication then the success of any kind of communication depends on the environment behavior. In other words, both high-level outputs and high-level inputs are under the control of the high-level environment, which may exploit them to set up a covert channel [1,21,29]. In this respect, we assume the synchronous

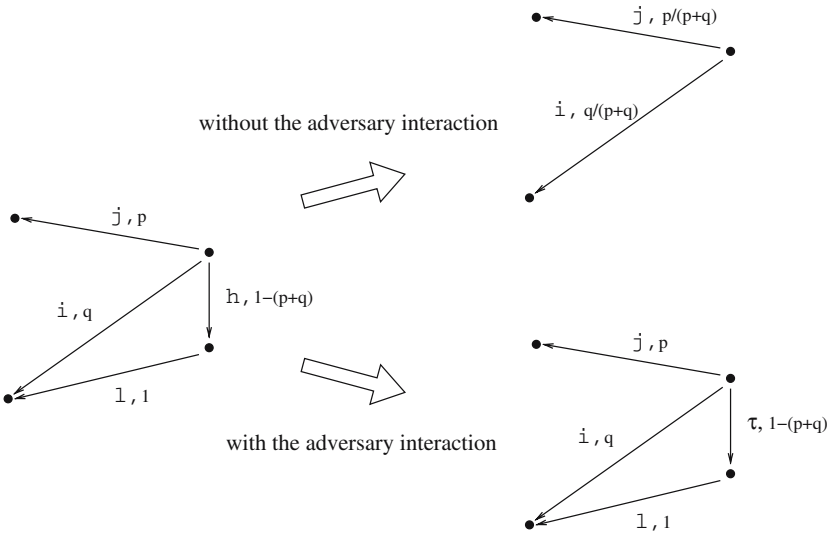

Fig. 3 A GRTS with two possible evolutions which depend on the adversary behavior

model of communication, whose effect is illustrated in the following example.

Example 3 In Fig. 3 it is shown a variant of the GRTS of Fig. 1 that replaces the high-level input by a high-level output of the same type. Hence, the initial state enables the generative bundle only, which is made of three transitions whose probabilities sum up to 1. If the adversary blocks the high-level output then the probabilities of the two remaining outputs must be normalized in order to fulfill the requirements of Definition 1. On the other hand, if the adversary interacts, then the output of type h is simply turned into an action τ, because its execution cannot be directly observed by Low.

As shown in the examples above, the interference of an adversary is revealed by comparing the semantics of different low-level views of the system, namely the view of the system in isolation (no adversary interactions) and the view of the system in the presence of the adversary. Hence, the effectiveness of the adversary strategy is established by the observations of Low with or without the adversary. In particular, Low reveals the adversary's interference if the two system views exhibit either different observations or the same observations with different probability distributions. On the other hand, the efficiency of the adversary strategy is estimated in terms of the probability for Low of observing the interference of such an adversary. In the following, we define several such strategies, which differ from each other for the observational power of the adversary. For the sake of simplicity, we consider systems that are fully specified from the viewpoint of Low, i.e., the corresponding GRTSs do not include reactive bundles of low-level type. Thus the nondeterminism is limited to the interactions with the high-level environment. The more general case in which the system can accept low-level inputs will be discussed in Sect. 8.

3.1 Defining the absence of the adversary

In the simplest scenario, an adversary A does not interact with the system in any way. We formalize this situation in terms of GRTS by defining the system $\mathcal{S}\backslash A$, expressing what Low can see when the high-level interface of the system $\mathcal{S}$ is not active because of the absence of adversaries.

Definition 2 Let $\mathcal{S} = (S, Act, T, s_0)$ be a GRTS. Then $\mathcal{S}\backslash A = (S', Act, T', s_0)$, where $S' \subseteq S$ and T' are obtained as follows:
$S' := \emptyset$; $T' := \emptyset$;
$No\_Adv(s_0)$;
where function $No\_Adv(s)$ is defined as follows:
$No\_Adv(s)$:
  $S' := S' \cup \{s\}$;
  for each $(s, a, q, t) \in T$
    if $a \notin Act_H$ then $T' := T' \cup \{(s, a, q/p(s), t)\}$;
  for each $(s, \_, \_, t) \in T'$
    if $t \notin S'$ then $No\_Adv(t)$;
where
$p(s) = \sum\{\!|\, p \mid \exists s' \in S, \exists a \in Act_L \cup \{\tau\}.\, (s, a, p, s') \in T \,|\!\}$.

In $\mathcal{S}\backslash A$ all the transitions labeled with high-level actions are simply removed from $\mathcal{S}$. Such a restriction may impose a normalization of the generative bundle of every state, as also shown in Example 3. The view of the system defined by $\mathcal{S}\backslash A$ expresses, at the GRTS level, the semantics of a process algebraic restriction operator applied to the high-level actions of the system (see e.g., [4]).

3.2 Families of probabilistic adversaries

Having identified an adversary with the high-level environment, we identify accordingly a probabilistic adversary A

that interacts with a system $\mathcal{S}$ with a probabilistic scheduler which guides the execution of the inputs/outputs constituting the interface of $\mathcal{S}$ with the high-level environment. We can define several different scheduling policies and correspondingly several classes of adversaries with different expressive power. For each class, we show how to express the semantics of the low-level view of $\mathcal{S}$ under the interference of an adversary A in that class, which we formally denote by $\mathcal{S} \mid A$.

3.2.1 Simple adversaries

We start with considering adversaries that, for every high-level type, choose a priori the behavior of the corresponding inputs/outputs and do not change it at run-time. Hence, in a sense, this class of schedulers is history-independent and we will refer to it as the class $\mathcal{A}_S$ of simple adversaries.

Definition 3 A simple adversary A is defined by a pair (A_g, A_r), where $A_g \subseteq H$ is the set of types of the high-level output actions that A can accept, and $A_r \subseteq H \times]0, 1[$ is a set of pairs of the form (h, p_h) such that p_h expresses the probability distribution associated with the reactive bundle of type h.

Definition 4 Let $\mathcal{S} = (S, Act, T, s_0)$ be a GRTS and $A = (A_g, A_r)$ be a simple adversary. Then $\mathcal{S} \mid A = (S', Act, T', s_0)$, where $S' \subseteq S$ and T' are obtained as follows:
$S' := \emptyset$; $T' := \emptyset$;
$S\_Adv(s_0)$;
where function $S\_Adv(s)$ is defined as follows:
$S\_Adv(s)$:
  $S' = S' \cup \{s\}$;
  $Gen(s, A_g)$;
  for each reactive bundle of type $h \in H$ enabled at s
    if $(h, p_h) \in A_r$ then $React(s, h, p_h)$;
  for each $(s, \_, \_, t) \in T'$
    if $t \notin S'$ then $S\_Adv(t)$;

$Gen(s, I)$:
  for each $(s, a, q, t) \in T$
    if $a \notin Act_H$ then $T' := T' \cup \{(s, a, q/p(s, I), t)\}$;
    if $a \in I$ then $T' := T' \cup \{(s, \tau, q/p(s, I), t)\}$;
where

$$p(s, I) = \sum\{\!|\, p \mid \exists s' \in S, \exists a \in Act_L \cup I \cup \{\tau\} . (s, a, p, s') \in T \,|\!\}.$$

$React(s, h, p_h)$:
  if the generative bundle is non-empty at $s \in S'$ then
    for each $(s, a, q, t) \in T' q := q \cdot (1 - p_h)$;
    for each $(s, h_*, q, t) \in T$,
    $T' := T' \cup \{(s, \tau, q \cdot p_h, t)\}$;
  else
    for each $(s, h_*, q, t) \in T$,
    $T' := T' \cup \{(s, \tau, q, t)\}$;

Intuitively, in each state of the GRTS all the high-level actions that can be executed because of the adversary strategy are turned into unobservable actions, because they cannot be observed by Low. All the other high-level actions are simply removed. A twofold normalization of the generative bundle may occur. The former occurs in function *Gen* and is due to the restriction of the high-level output actions not in A_g, while the latter is due to the relabeling of the high-level input actions. In particular, here and in the rest of the paper we assume that the reactive bundles with type in A_r are considered by following the alphabetic order of the high-level type names. Then, for each reactive bundle h enabled by the adversary, in function *React* the following operations are performed. If the generative bundle is non-empty, parameter p_h is used to redistribute the probabilities of the actions τ obtained by hiding the input actions h_*—whose overall probability must be equal to p_h—and the probabilities associated with the pre-existing transitions of the generative bundle, whose overall probability must be equal to $1 - p_h$. By so doing, the requirements of Definition 1 are preserved.

Example 4 The GRTSs of Fig. 2a and b express what Low can see when observing the GRTS of Fig. 1 under the interference of a simple adversary such that $(h, \_) \notin A_r$ and $(h, q) \in A_r$, respectively. Similarly, in the case of Fig. 3, it is possible to observe the behavior of a system under the interference of a simple adversary such that either $h \notin A_g$ or $h \in A_g$.

3.2.2 Interactive adversaries

The main limitation of the simple adversaries is that they cannot take into consideration the current state of the system when deciding their strategy. By relaxing this constraint, we obtain a more powerful class of schedulers which can decide the behavior of the high-level inputs/outputs on the basis of the high-level interface that is currently enabled by the system. We refer to it as the class $\mathcal{A}_I$ of interacting adversaries.

Definition 5 An interactive adversary A is defined by a pair (A_g, A_r), where $A_g : \mathcal{P}(H) \to \mathcal{P}(H)$ is such that $A_g(G)$ is the set of types of the high-level output actions that A can accept when G is the set of types of the high-level output actions currently enabled, and $A_r : \mathcal{P}(H) \to \mathcal{P}(H \times]0, 1[)$ is such that $A_r(R)$ is a set of pairs of the form (h, p_h) such that p_h expresses the probability distribution associated with the reactive bundle of type h when R is the set of types of the reactive bundles currently enabled.

Given $s \in S$ we denote by H_s the set of types of the high-level actions labeling the transitions of the generative bundle enabled at s, and by H_{*s} the set of types of the high-level reactive bundles enabled at s.

Definition 6 Let $\mathcal{S} = (S, Act, T, s_0)$ be a GRTS and $A = (A_g, A_r)$ be an interactive adversary. Then $\mathcal{S} \mid A = (S', Act, T', s_0)$, where $S' \subseteq S$ and T' are obtained as follows:
$S' := \emptyset$; $T' := \emptyset$;
$I\_Adv(s_0)$;
where function $I\_Adv(s)$ is defined as follows:
$I\_Adv(s)$:
  $S' = S' \cup \{s\}$;
  $Gen(s, A_g(H_s))$;
  for each reactive bundle of type $h \in H$ enabled at s
   if $(h, p_h) \in A_r(H_{*s})$ then $React(s, h, p_h)$;
  for each $(s, \_, \_, t) \in T'$
   if $t \notin S'$ then $I\_Adv(t)$;

The novelty with respect to the algorithm of Definition 4 is that the choice of the high-level behavior is not fixed a priori, but it can change depending on the high-level interface enabled by the system at the current state.

Proposition 1 $\mathcal{A}_S \subset \mathcal{A}_I$.

Proof It is an immediate consequence of Definition 3 and of Definition 5. □

3.2.3 History-dependent adversaries

We now consider another extension of the simple adversaries that allows the high-level behavior to be governed by the previous history, which is described by a trace of events $Tr \in Act^*$. Hence, we call this class the class of history-dependent adversaries and we refer to it as $\mathcal{A}_{HD}$.

Definition 7 A history-dependent adversary A is defined by a pair (A_g, A_r), where $A_g : Act^* \to \mathcal{P}(H)$ is such that $A_g(Tr)$ is the set of types of the high-level output actions that A can accept when the executed trace is Tr, and $A_r : Act^* \to \mathcal{P}(H \times]0, 1[)$ is such that $A_r(Tr)$ is a set of pairs of the form (h, p_h), where p_h expresses the probability distribution associated with the reactive bundle of type h when the executed trace is Tr.

At each execution step the previous history, which is modeled by a trace Tr, affects the adversary strategy that governs the high-level behavior. Therefore, with respect to Definition 5, the choice of the high-level inputs and outputs that can be executed depends on the previous history rather than the current state. As a consequence, each state of $\mathcal{S}$ may result in several different states depending on which trace has been executed to reach that state. Hence, each state of $\mathcal{S} \mid A$ is actually described by a pair (s, Tr), with s a state of $\mathcal{S}$ and Tr an execution trace. In the following definition, we denote by ϵ the empty trace.

Definition 8 Let $\mathcal{S} = (S, Act, T, s_0)$ be a GRTS and $A = (A_g, A_r)$ be a history-dependent adversary. Then $\mathcal{S} \mid A = (S', Act, T', (s_0, \epsilon))$, where $S' \subseteq \mathcal{P}(S \times Act^*)$ and $T' \subseteq S' \times Act \times]0, 1] \times S'$ are obtained as follows:
$S' := \emptyset$; $T' := \emptyset$;
$H\_Adv((s_0, \epsilon))$;
where function $H\_Adv((s, Tr))$ is defined as follows:
$H\_Adv((s, Tr))$:
  $S' = S' \cup \{(s, Tr)\}$;
  for each $(s, a, q, t) \in T$
   if $a \notin Act_H$ then $T' := T' \cup$
     $\{((s, Tr), a, q/p(s, A_g(Tr)), (t, Tr.a))\}$;
   if $a \in A_g(Tr)$ then $T' := T' \cup$
     $\{((s, Tr), \tau, q/p(s, A_g(Tr)), (t, Tr.a))\}$;
  for each reactive bundle of type $h \in H$ enabled at s
   if $(h, p_h) \in A_r(Tr)$ then
    if the generative bundle is non-empty at
     (s, Tr) then
     for each $((s, Tr), a, q, \_) \in T' q := q \cdot (1 - p_h)$;
     for each $(s, h_*, q, t) \in T$,
      $T' := T' \cup \{((s, Tr), \tau, q \cdot p_h, (t, Tr.h_*))\}$;
    else
     for each $(s, h_*, q, t) \in T$,
      $T' := T' \cup \{((s, Tr), \tau, q, (t, Tr.h_*))\}$;
  for each$((s, Tr), _, _, (t, Tr.\pi)) \in T'$
   if $(t, Tr.\pi) \notin S'$ then $H\_Adv((t, Tr.\pi))$;

Proposition 2 $\mathcal{A}_S \subset \mathcal{A}_{HD}$.

Proof It is an immediate consequence of Definition 3 and of Definition 7. □

4 Measuring noninterference

We evaluate the interference of an adversary A with respect to a system $\mathcal{S} = (S, Act, T, s_0)$ and the low-level observer by comparing the visible behavior of $\mathcal{S}$ in the absence of A, given by $\mathcal{S} \backslash A$, and the visible behavior of $\mathcal{S}$ in the presence of A, given by $\mathcal{S} \mid A$. The comparison between the two system views is performed on the basis of a behavioral equivalence that abstracts from unobservable actions, similar to the weak probabilistic bisimulation of [11].

As done in [4] we consider a probabilistic variant of the weak probabilistic bisimulation, denoted by $\approx_{\mathrm{PB}}$, which replaces the classical weak transitions of the Milner's weak bisimulation by the probability of reaching classes of equivalent states. For this purpose we employ a function $Prob$ such that $Prob(s, \pi, s')$ denotes the aggregate probability of going from s to s' by executing an action π. Sometimes we use the abbreviation $Prob(s, s')$ to represent the aggregate probability of going from s to s' via sequences of any number and type of actions. Similarly, $Prob(s, \pi, C)$ denotes the aggregate probability of going from s to a state in the equivalence class C by executing an action π, and $Prob(s, \tau^* a, C)$ expresses the aggregate probability of going from s to a state

in the equivalence class C via sequences of any number of τ actions followed by an action a.

Lemma 1 *The value of $Prob(s, \tau^* a, C)$ is the minimal non-negative solution to the equation system:*

$$\begin{cases} 1 & \text{if } a = \tau \wedge s \in C \\ \sum_{s' \in S} Prob(s, \tau, s') \cdot Prob(s', \tau^*, C) & \text{if } a = \tau \wedge s \notin C \\ \sum_{s' \in S} Prob(s, \tau, s') \cdot Prob(s', \tau^* a, C) + Prob(s, a, C) & \text{if } a \neq \tau \end{cases}$$

As shown in [4], this system has a least solution.

Definition 9 An equivalence relation $R \subseteq S \times S$ is a weak probabilistic bisimulation if and only if, whenever $(s, s') \in R$, then for all C in the quotient set $S_{/R}$:

1. $Prob(s, \tau^* a, C) = Prob(s', \tau^* a, C)\ \forall a \in Act$
2. $Prob(s, a_*, C) = Prob(s', a_*, C)\ \forall a_* \in Act$.

Two states $s, s' \in S$ are weakly probabilistically bisimilar, denoted $s \approx_{PB} s'$, if there exists a weak probabilistic bisimulation including the pair (s, s').

On the basis of $\approx_{PB}$, we say that Low cannot detect the interference of the adversary A whenever there exists a weak probabilistic bisimulation including the pair of initial states of $\mathcal{S}\backslash A$ and $\mathcal{S} \mid A$. In this case we write $\mathcal{S}\backslash A \approx_{PB} \mathcal{S} \mid A$. Since $\mathcal{S}\backslash A$ and $\mathcal{S} \mid A$ are fully generative — in the former all the reactive actions are removed while in the latter they are removed or turned into actions τ — the noninterference check is performed by verifying only condition 1 of Definition 9.

Example 5 Consider the GRTS of Fig. 1. The interference of a simple adversary A is revealed to Low by the execution of the output action l. In fact, the GRTS of Fig. 2a, modeling $\mathcal{S}\backslash A$, cannot execute the output action l, while in the GRTS of Fig. 2b, modeling $\mathcal{S} \mid A$, such an action occurs with probability q, i.e., $\mathcal{S}\backslash A \not\approx_{PB} \mathcal{S} \mid A$.

Whenever the outcome of the comparison based on $\approx_{PB}$ is negative, it is important and useful to measure the difference between the observable behaviors of $\mathcal{S}\backslash A$ and $\mathcal{S} \mid A$ in order to get an estimate of the security of the system against the adversary A (or equivalently of the power of A). This can be done by replacing $\approx_{PB}$ with an approximate equivalence relation R that allows states with slightly different weak transition probabilities to belong to the same class [6,5]. In practice, if R is not a weak probabilistic bisimulation, there exists at least a pair $(s, s') \in R$ that does not satisfy condition 1 of Definition 9. Such a pair of states effectively expresses the interference caused by A only in the case $s \in \mathcal{S}\backslash A$ and $s' \in \mathcal{S} \mid A$. In fact, if $(s, s') \in R$ and both s and s' belong to the same system view—either $\mathcal{S}\backslash A$ or $\mathcal{S} \mid A$—then the comparison between them does not actually reveal any interference, as it locally refers to a single system view.

Formally, we can express the quantitative estimation of the adversary interference as follows:

$$d_A^R(s, s', a, C) = \mid Prob(s, \tau^* a, C) - Prob(s', \tau^* a, C) \mid$$

where $s \in \mathcal{S}\backslash A$, $s' \in \mathcal{S} \mid A$, $(s, s') \in R$, $a \in Act$, and $C \in S_{/R}$. Whenever such a difference is equal to zero for every pair of states in R and for all actions and classes, then R turns out to be a weak probabilistic bisimulation, and Low cannot distinguish the way in which the system and the adversary communicate.

The difference $d_A^R(s, s', a, C)$ between the two weak transition probabilities expresses the local distance between s and s' with respect to a and C. We say that the distance above is local because, as stated by the notion of bisimulation, the comparison between s and s' does not take into account the way in which such states have been reached from the initial ones. On the other hand, a measure of the interference of the adversary A should not only consider the local distance between s and s', but also the probability of reaching such states. For instance, the local distance between the initial state of $\mathcal{S}\backslash A$, which we call s_0, and the initial state of $\mathcal{S} \mid A$, which we call s_0^A, does not have the same impact on the Low perception of the difference between $\mathcal{S}\backslash A$ and $\mathcal{S} \mid A$ as the local distance between two states that can be reached from the initial ones with negligible probability. We deal with this problem by considering a difference weighted by the probability of reaching s and s':

$$\bar{d}_A^R(s, s', a, C) = Prob(s_0, s) \cdot Prob(s_0^A, s') \cdot \mid Prob(s, \tau^* a, C) - Prob(s', \tau^* a, C) \mid$$

where $s \in \mathcal{S}\backslash A$, $s' \in \mathcal{S} \mid A$, $(s, s') \in R$, $a \in Act$, and $C \in S_{/R}$. The difference $\bar{d}_A^R(s, s', a, C)$ between the two weak transition probabilities expresses the weighted local distance between s and s' with respect to a and C, which is a measure less than (or equal to) the local distance $d_A^R(s, s', a, C)$. In particular, the difference between the weighted local distance and the local distance is inversely proportional to the probability of being in the considered states. The approach based on the weighted local distance is conservative with respect to the notion of approximate weak probabilistic bisimulation. More precisely, two states that are indistinguishable according to $\approx_{PB}$ are expected to have distance equal to zero in the approximate version of $\approx_{PB}$.

The pair of states in R which maximally differ expresses the degree of approximation of R with respect to $\approx_{PB}$. Formally, the most effective interference caused by A under the relation R is given by:

$$\delta_A^R = \max\{\!|\, \bar{d}_A^R(s, s', a, C) \mid s \in \mathcal{S}\backslash A, s' \in \mathcal{S} \mid A, (s, s') \in R, a \in Act, C \in S_{/R} \,|\!\}.$$

In essence, we take the pair of states belonging to different system views and to the same class which maximize the probability of revealing to Low the strategy of A.

Among all the possible adversaries, we are interested in determining the adversary with the strategy that maximizes the difference between R and $\approx_{\mathrm{PB}}$ or, equivalently, the adversary that is most easily revealed by Low. Formally,

$$\varepsilon_R = \sup_A \delta_A^R$$

is the maximum distance between the two system views that any adversary may cause when the considered relation is R. In other words, ε_R represents the maximum information leakage with respect to R that can be set up from the high level to the low level.

Among all the possible relations, we recall that we must evaluate the closest approximation of $\approx_{\mathrm{PB}}$. For this purpose, it is necessary to impose a requirement on the choice of R, which should include the pair (s_0, s_0^A). Indeed, according to the notion of approximate weak probabilistic bisimulation, if $(s_0, s_0^A) \notin R$ the observable difference between $\mathcal{S}\backslash A$ and $\mathcal{S}|A$ is erroneously undetected. For instance, in the case s_0 and s_0^A are not in the same class it is easy to provide a partitioning $S_{/R}$ such that if $s \in \mathcal{S}\backslash A$ and $s' \in \mathcal{S} \mid A$ then $(s, s') \notin R$. This implies that the distance between the two system views is null, even if intuitively an external observer can immediately notice the difference between them by observing what happens at the initial states. In the following, we call $\mathcal{R}$ the set of relations satisfying the condition on the initial states. Then, it holds that

$$\bar{\varepsilon} = \inf_{R \in \mathcal{R}} \varepsilon_R$$

represents the maximum adversary interference for the closest approximation of $\approx_{\mathrm{PB}}$. Thus, the corresponding adversary is the most powerful adversary that maximizes the information leakage from the high level to the low level.

Example 6 Consider the family of simple adversaries and the scenario of Fig. 3. Two different adversary strategies may be applied. The first one blocks the output action h, but in such a case $\mathcal{S}\backslash A \approx_{\mathrm{PB}} \mathcal{S} \mid A$. The second one enables the output action h, thus enabling the observation $\tau.l$, which is not offered by $\mathcal{S}\backslash A$. In such a case, it holds that $\bar{\varepsilon} = 1 - (p + q)$, which corresponds to the probability of observing the distinguishing behavior.

The analysis seems to be more complicated in the case of Fig. 1, where infinitely many adversaries may interact with the system, one for each possible value assigned to parameter q that solves the nondeterminism due to the high-level input (see Fig. 2b). However, it can be proved that the most powerful simple adversary A assigns to q the limiting value 1. In fact, for each $R \in \mathcal{R}$, it holds that $\delta_A^R = 1$, from which the result follows.

4.1 Relation to discrete time markov chain analysis

As emphasized in the previous section, the GRTSs modeling the two system views to be compared are fully specified, i.e., they do not include reactive bundles. Therefore, as already observed in Sect. 2, such GRTSs are well-defined action-labeled Discrete Time Markov Chains [12] (DTMCs). In this section, we compare our approach to the estimation of the adversary interference with the numerical analysis of DTMCs that is usually conducted for performance evaluation purposes.

Definition 10 A finite action-labeled DTMC is a tuple $\mathcal{M} = (S, Act, T, s_0)$ where S is a finite set of states, $s_0 \in S$ is the initial state, Act is a non-empty set of activities, and $T \subseteq S \times Act \times]0, 1] \times S$ is a finite transition relation such that for all $s \in S$ it holds that $\sum\{\!\{ p \mid \exists a \in Act, t \in S.\, (s, a, p, t) \in T \}\!\} \in \{0, 1\}$.

As far as the analysis of action-labeled DTMCs is concerned, a typical approach to performance measure specification relies on reward structures [22]. This requires associating real numbers with states (rate rewards) and transitions (instantaneous rewards) of the DTMC, respectively. A rate reward y_s expresses the rate at which a gain (or a loss, if the number is negative) is accumulated while sojourning in state s. By contrast, an instantaneous reward $b_{s,\pi,p,s'}$ specifies the instantaneous gain (or loss) implied by the execution of the transition $(s, \pi, p, s') \in T$.

The instant-of-time value of a performance measure expresses the gain (loss) received at a particular instant of time [32]. If the system reaches a steady behavior at a limiting execution time, in the stationary case the instant-of-time performance measure expresses the long run gain (loss) per unit of time.

Formally, the instant-of-time value of a performance measure specified through a reward structure is computed through the following equation:

$$\sum_{s \in S} y_s \cdot \pi(s) + \sum_{(s,\pi,p,s') \in T} b_{(s,\pi,p,s')} \cdot \phi(s, \pi, p, s'), \tag{1}$$

where $\pi(s)$ is the probability of being in s at the considered instant of time and $\phi(s, \pi, p, s')$ is the frequency of the transition (s, π, p, s') at the considered instant of time, which is given by $\pi(s) \cdot p$.

For instance, two typical performance measures are system throughput and resource utilization. On one hand, in order to measure the throughput of the system with respect to the action a, i.e., the number of occurrences of a observed per unit of time, it is sufficient to set $b_{(\_,a,\_,\_)} = 1$ and set to 0 any other reward. On the other hand, in order to compute the percentage of time during which the system is in a certain state s (modeling e.g., the usage of a certain resource), we need to set $y_s = 1$ and set to 0 any other reward.

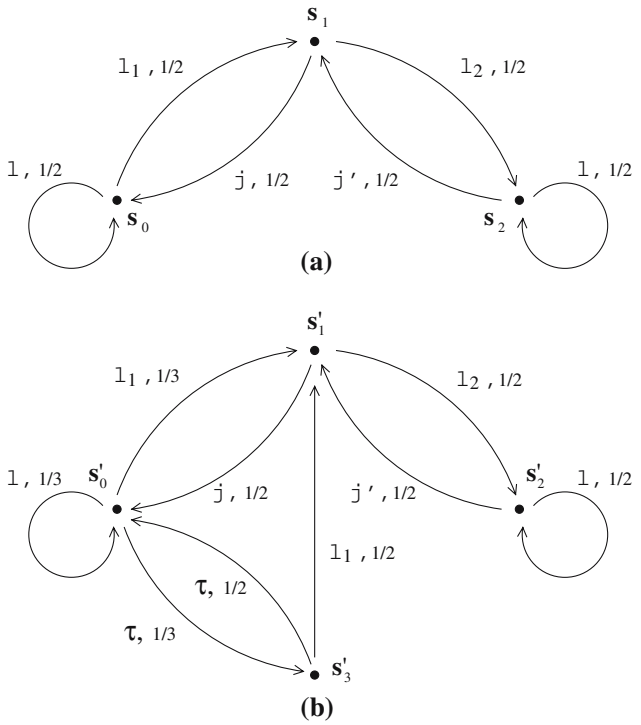

Fig. 4 Two examples of discrete-time Markov chains

Example 7 For the DTMC of Fig. 4a the steady state probability vector is (1/3, 1/3, 1/3), meaning that at a limiting execution time we have the same probability 1/3 of being in one of the three states s_0, s_1, or s_2. Then, the value of the stationary performance measure expressing the throughput of the system in terms of occurrences of the transition labeled with action j' is given by $b_{s_2,j',1/2,s_1} \cdot \phi(s_2, j', 1/2, s_1) = 1 \cdot \pi(s_2) \cdot 1/2 = 1/6$.

Similarly, consider the same performance measure for the DTMC of Fig. 4b, whose steady-state probability vector is (3/10, 3/10, 3/10, 1/10). In the long run we have the same probability 3/10 of being in one state among s'_0, s'_1, or s'_2, while the probability of being in s'_3 is 1/10. Then, the throughput of action j' is given by $b_{s'_2,j',1/2,s'_1} \cdot \phi(s'_2, j', 1/2, s'_1) = 1 \cdot \pi(s'_2) \cdot 1/2 = 3/20$.

By following the performance evaluation approach, the distance between two different systems can be estimated in terms of the difference between the values of the same performance metrics. In particular, if the chosen metric is the system throughput with respect to action a, the distance between two states s and s' is:

$$| \pi(s) \cdot p - \pi(s') \cdot p' | \tag{2}$$

if the probability of observing a at s is p and the probability of observing a at s' is p'.

Example 8 Suppose that we are interested in estimating the difference between the DTMCs of Fig. 4. If the comparison parameter is the throughput of action j', then the distance between such systems is given by $| 1/6 - 3/20 | = 1/60$.

Similarly, if the comparison is conducted on the basis of the throughput of the actions l_1, then we have $| 1/3 \cdot 1/2 - (3/10 \cdot 1/3 + 1/10 \cdot 1/2) | = 1/60$.

The measure estimated according to Eq. (2) is a global distance, because it is obtained by comparing two values that depend on the steady state probabilities of the states under consideration. We now compare this performance-related notion of distance with the weighted local distance presented in the previous section. In the stationary case, the probability $Prob(s_0, s)$ of reaching state s from the initial state corresponds to the stationary probability $\pi(s)$. Therefore, in the long run the weighted local distance between s and s' with respect to a and C can be formulated as follows:

$$\pi(s) \cdot \pi(s') \cdot | p - p' | \tag{3}$$

in the case $Prob(s, \tau^* a, C) = p$ and $Prob(s', \tau^* a, C) = p'$. As we have seen, with respect to Eq. (2) the measure estimated according to Eq. (3) expresses a local view of the distance between s and s', in accordance with the bisimulation-based approach. In fact, it is obtained by comparing two values that depend on the local probabilistic behavior of the system. The result of this local comparison is then weighted according to the global information that is represented by the steady state probabilities of the states under consideration. As seen in the previous section, this notion of distance is conservative with respect to the approximate bisimulation. In particular, two bisimilar states continue to have distance equal to zero when relaxing the bisimulation relation. This would not be the case with the performance-related notion of distance.

Example 9 Suppose that the two DTMCs of Fig. 4 represent two different system views to be compared. The former expresses the behavior of the system in the absence of adversary interferences, while the latter shows what happens when the adversary does interfere. In particular, the transitions labeled with τ actions model the interactions between the system and the adversary.

First of all, we have to construct a relation R including the pair (s_0, s'_0) that approximates $\approx_{PB}$ as close as possible. To this aim, in order to minimize the distance between s_0 and s'_0, corresponding states in the two DTMCs that can be reached by s_0 and s'_0, respectively, must be related by R, i.e., $(s_1, s'_1) \in R$. Similarly, it also holds that $(s_2, s'_2) \in R$. Finally, by looking at the ingoing/outgoing transitions, we have that the remaining state s'_3 can be put either in the same class of the initial states, or in a novel class. As can be easily verified by considering the execution probability of such transitions, it turns out that the relation we must consider induces the partition $S_{/R} = \{\{s_0, s'_0, s'_3\}, \{s_1, s'_1\}, \{s_2, s'_2\}\}$.

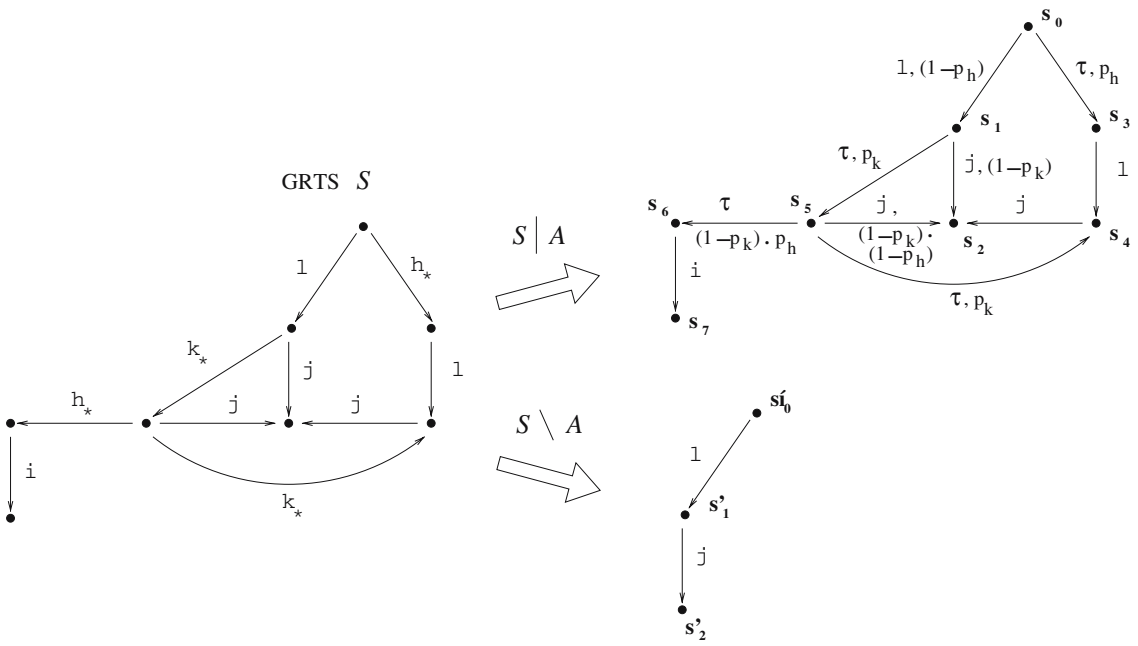

Fig. 5 A GRTS S, its interaction with a simple adversary A, and its behavior in the absence of adversaries

Now, if we consider s_2 and s_2' with respect to action j' and class $C = \{s_1, s_1'\}$, we have that the weighted local distance between s_2 and s_2' is equal to zero, because the two states are locally indistinguishable from the viewpoint of Low. Instead, in the performance evaluation setting we obtained a global distance equal to $1/60$. As another example, if we consider the difference between s_0 and s_0' in the long run and with respect to action l_1 and class C, we have $Prob(s_0, \tau^* l_1, C) = 1/2$, $Prob(s_0', \tau^* l_1, C) = 3/5$, and a weighted local distance equal to $1/3 \cdot 3/10 \cdot (3/5 - 1/2) = 1/100$. Note that in the performance evaluation setting we obtained the global distance $1/60$.

5 Evaluating the most powerful adversary

The objective of an adversary is to apply the strategy that maximizes the probability of distinguishing, from the viewpoint of Low, the behavior of the system without high-level interferences from the behavior of the system with high-level interferences. As we have shown, the nondeterministic choice between a high-level input and any other event can be solved by a probabilistic adversary through an infinite number of different strategies. It is therefore important to investigate conditions which allow us to analyze only a finite number of such strategies and yet guarantee a correct evaluation of the power of the probabilistic adversary. In the following, we show for each class whether the most powerful adversary can be determined or not by considering a finite subset of such strategies.

5.1 Simple adversaries

When defining a simple adversary, infinite sets of the form $\{(h_1, p_1), \ldots, (h_n, p_n)\}$ can describe the probabilistic behavior of the high-level input interface represented by the reactive bundles of type $h_1, \ldots, h_n$. One may ask whether the most powerful simple adversary can be found by considering a finite subset of such sets, like e.g., the sets containing only the limiting probability values 0 and 1, as suggested by the example of Fig. 1. However, the following examples show that in general it is not possible to predict the set of probability values that determine the most powerful adversary.

Example 10 Consider the GRTS S of Fig. 5, where the adversary controls the probability distribution of the inputs of type h and k. Hence, a simple adversary is described by two probability values p_h and p_k that are assigned to the reactive bundles of type h and k, respectively. Whenever the adversary is absent, see $S \backslash A$, the system executes the trace $l.j$. Instead, the interference of a simple adversary A, see $S \mid A$, may enable the action i, which allows Low to distinguish the two system views. Hence, $S \backslash A$ and $S \mid A$ cannot be weakly bisimilar. In order to approximate $\approx_{PB}$ and to relate $S \backslash A$ and $S \mid A$ we follow the same argumentation provided in Example 9. First, we impose $(s_0', s_0), (s_1', s_1), (s_2', s_2) \in R$. Then, from the analysis of the ingoing/outgoing transitions of s_3, s_4, s_5 it follows $(s_3, s_0) \in R$, $(s_1, s_4) \in R$, and $(s_1, s_5) \in R$. Finally, we consider state s_6, which is the one enabling the distinguishing action and, therefore, is the unique representative of its class. Thus, we obtain:

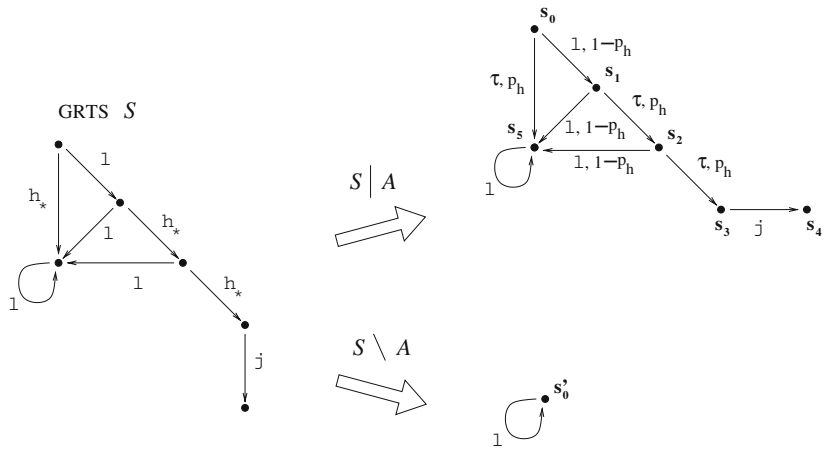

Fig. 6 A GRTS S, its interaction with a simple adversary A, and its behavior in the absence of adversaries

$R = \{\{s'_0, s_0, s_3\}, \{s'_1, s_1, s_4, s_5\}, \{s'_2, s_2\}, \{s_6\}, \{s_7\}\}$

for which the maximum weighted local distance is given by $\bar{d}^R_A(s'_1, s_1, j, \{s'_2, s_2\}) = (1-p_h)\cdot p_k \cdot p_h \cdot (1-p_k)$. This value, which is also obtained by calculating, e.g., $\bar{d}^R_A(s'_1, s_5, j, \{s'_2, s_2\})$ or else $\bar{d}^R_A(s'_1, s_5, i, \{s_7\})$, intuitively represents the probability of observing the distinguishing action i. Now it is easy to see that each simple adversary described by the limiting probability values 0 and 1 prevents the system from executing action i. This is because the adversary should first disable h_* and force k_*, while thereupon should follow the opposite policy. However, a simple adversary cannot change its strategy at run time. Formally, the adversary that maximizes ε_R sets $p_h = p_k = \frac{1}{2}$, for which we obtain the maximum value $1/16$.

As another example, consider the GRTS S of Fig. 6. There, the execution of the output j, which is the unique action that can be observed only whenever a simple adversary A interacts with the system, depends on which states enable the input of type h, which is an event controlled by the adversary, compare $S\backslash A$ and $S \mid A$. Formally, in order to approximate $\approx_{\mathrm{PB}}$ it is sufficient to observe that (s_0, s'_0) must be in R, while s_1, s_2, s_5 belong to the same class if we ignore the distinguishing state s_3, which is the one enabling the distinguishing action. Thus, we obtain:

$R = \{\{s'_0, s_0, s_1, s_2, s_5\}, \{s_3\}, \{s_4\}\}.$

Let p_h be the probability associated with the input of type h by the simple adversary. Then, the maximum weighted local distance is given by $\bar{d}^R_A(s'_0, s_1, j, \{s'_4\}) = (1 - p_h) \cdot p_h \cdot p_h$, which is maximized by taking $p_h = 2/3$, for which the maximum probability of observing the interference is $\bar{\varepsilon} = 4/27$.

In general, the most powerful simple adversary is determined as follows. For each $R \in \mathcal{R}$, we maximize — for all $a \in Act$, $C \in S_{/R}$ and $(s, s') \in R$ such that s and s' belong to different system views — the function expressing the weighted local distance between s and s'. This corresponds to solving a constraint programming problem with as many variables as the number of different high-level actions, each one being associated with a different probability value (like in the examples above where e.g., the high-level input action of type h is associated with the probability value p_h). We point out that the number of relations in $\mathcal{R}$ exponentially depends, according to the Bell formula [27], on the number of states of the two system views to be compared. This makes the problem of finding the most powerful adversary intractable as its complexity is hyper-exponential.

In all the examples above, where the difference between the two system views is expressed by a distinguishing action that can be performed by one view only, the closest approximation of the weak probabilistic bisimulation turns out to be computed as follows. The state enabling the distinguishing action is the unique representative of its class, while corresponding states of the two views are lumped in the same class. The related partitioning is simply obtained by ignoring the state enabling the distinguishing action and by lumping all the other states according to the weak bisimulation semantics. In such a way, the differences between the two system views are limited to the analysis of the distinguishing behavior.

5.2 Interactive adversaries

Despite the enhanced expressiveness of the strategies that characterize the interactive adversaries, the analysis of the most powerful interactive adversary suffers from the same problems as in the case of the simple adversaries.

Example 11 Consider again the system S of Fig. 5. In order to maximize the execution probability of the action i, an interactive adversary can work as follows. First, such an adversary associates the empty set with set $\{h\}$, which expresses the high-level input interface of the system at the initial state, thus forcing the execution of the action l. Then, she/he associates

$\{(k, 1)\}$ with set $\{k\}$ and $\{(h, 1)\}$ with set $\{h, k\}$ (note that 1 represents a limiting value), thus reaching the distinguishing behavior, which is observed by Low with probability tending to 1.

Unfortunately, in general we cannot restrict ourselves to considering the limiting probability values in order to determine the most powerful interactive adversary.

Example 12 Consider again the system S of Fig. 6. In order to maximize the probability of executing the output j, the first occurrence of h_* should be disregarded. Instead, the following occurrences of h_* should be forced. However, an interactive adversary cannot follow such a strategy, as each state where she/he can interfere has the same high-level input interface, which is given by $\{h\}$. Hence, it turns out that the most powerful interactive adversary associates $\{(h, \frac{2}{3})\}$ with set $\{h\}$, exactly as done by the most powerful simple adversary.

In general, with respect to the simple adversaries the situation is even worse with the interactive adversaries, because in such a case a given high-level action may be associated with different probability values depending on the number of different states in which it is enabled. Hence, the number of variables of each constraint programming problem is greater with respect to the case of the simple adversaries.

5.3 History-dependent adversaries

Based on the definition of history-dependent adversary given in Sect. 3.2.3, a system described by a finite-state GRTS $\mathcal{S}$ can result in an infinite-state GRTS $\mathcal{S} \mid A$ when interacting with an adversary A with unbounded time resources. In such a case the infinite-state system $\mathcal{S} \mid A$ can be reduced to a finite-state system by limiting the time resources of A.

In the computational complexity approach it is common to check the system against attackers with polynomially bounded resources with respect to a security parameter ν representing e.g., the key length. In our setting the time resources of a history-dependent adversary are bounded by limiting the number of interactions occurring between such an adversary and the system. Such a limitation does not underestimate the amount of information leakage if the system is safe with respect to the adversary, i.e., if, as the number of visited states and the length of the observable traces increase, the probability of augmenting the information flow approaches zero.

Definition 11 Let $\mathcal{S} \mid A = (S \times Act^*, Act, T, (s_0, \epsilon))$ be an infinite-state GRTS. We say that system $\mathcal{S}$ is safe with respect to the history-dependent adversary A and the security parameter ν if and only if whenever

$$\sigma = ((s_0, \epsilon), \pi_1, p_1, (s_1, \pi_1)), ((s_1, \pi_1), \pi_2, p_2, (s_2, \pi_1\pi_2)), \ldots$$

is an infinite sequence such that:

1. $\forall i \in \mathbf{N}.\, ((s_{i-1}, \gamma), \pi_i, p_i, (s_i, \gamma\pi_i)) \in T$,
2. there does not exist $((s, \gamma), \pi, p, (s', \gamma\pi)) \in \sigma$ such that $\forall \gamma'.\, \gamma \leq \gamma'$:

$$A_g(\gamma') = A_g(\gamma) \wedge A_r(\gamma') = A_r(\gamma)$$

3. $\forall i \in \mathbf{N}\, \exists j, k \in \mathbf{N}.\, i < j < k, \exists h \in Act_H, \exists l \in Act_L$, such that:

$$((s_{j-1}, \gamma), h, p_j, (s_j, \gamma h)) \in \sigma \;\wedge\; ((s_{k-1}, \gamma), l, p_j, (s_k, \gamma l)) \in \sigma$$

then it holds that

$$\forall k \in \mathbf{N}\, \exists \mu \in \mathbf{N}\, \forall (s, \gamma) \in \mathcal{S} \mid A.\; |\gamma| > \mu \wedge (\_, \_, \_, (s, \gamma)) \in \sigma \Rightarrow Prob((s_0, \epsilon), (s, \gamma)) < \nu^{-k}$$

Intuitively, condition 1 says that σ is an infinite trace in $\mathcal{S} \mid A$, condition 2 requires that A is actually an adversary with unbounded time resources, that is the adversary strategy can change at any time, and condition 3 requires that such an adversary is always active, in the sense that at any moment its behavior in the next future could still be detected by Low. Then, the notion of safety says that if any prefix γ of the infinite trace σ is long enough, then the probability of reaching (s, γ) is negligible. Assuming active adversaries—see condition 3—is necessary in order to exclude traces reaching states like, e.g., s_5 in the GRTS of Fig. 6b, for which the probability of any trace does not obviously approach zero, but the adversary cannot affect the low-level view of the system anymore.

If the system is safe with respect to A and ν then it is correct to block the interfering activity of A when the length of the trace is long enough and further interferences of A are negligible, according to a tolerance threshold $\epsilon(\nu)$ that quantifies what negligible means from the viewpoint of the low-level observer.

Proposition 3 *Let $\mathcal{S} = (S, Act, T, s_0)$ be a GRTS safe with respect to A and ν, $\epsilon(\nu)$ be the tolerance threshold, and R be a relation approximating $\approx_{\mathrm{PB}}$. Moreover, let $\delta_A^R(n)$ be the maximum observable interference caused by A with respect to R whenever every high-level interaction in $(s, \gamma) \in \mathcal{S} \mid A$ is prevented if $|\gamma| > n$. Then, there exists μ such that $|\, \delta_A^R - \delta_A^R(\mu)\, | < \epsilon(\nu)$.*

Proof By Definition 11, for all infinite traces modeling active, unbounded time interferences of A it is possible to choose $k \in \mathbf{N}$ and $\mu \in \mathbf{N}$ such that if $|\gamma| > \mu$ then $Prob((s_0, \epsilon), (s, \gamma)) < \nu^{-k} < \epsilon(\nu)$. Hence, all the states that do not belong to a prefix with length less than μ are actually reachable with negligible probability. Now, it is sufficient to group all (and only) such states in a new class $C_{\epsilon(\nu)}$. By virtue of the definition of δ_A^R, the weighted local distance

(with respect to $C_{\epsilon(v)}$) between any state in $\mathcal{S}\backslash A$, which cannot reach any state in $C_{\epsilon(v)}$, and any state in $\mathcal{S} \mid A$ is less than $\epsilon(v)$, from which the result immediately follows. □

This result confirms that considering a prefix of the history generated by $\mathcal{S} \mid A$, i.e., an adversary A with bounded resources, does not significantly alter the measurement of the interference caused by A.

Now, as for the case of the simple and interactive adversaries, we consider the problem of determining the most powerful history-dependent adversary, which can be found by checking a finite set of strategies.

Theorem 1 *Let $\mathcal{S} = (S, Act, T, s_0)$ be a GRTS such that $(\_, \tau, \_, \_) \notin T$, and let $A \in \mathcal{A}_{HD}$ be the most powerful history-dependent adversary for $\mathcal{S}$ such that $\mathcal{S}$ is safe with respect to $\mathcal{A}$ and v.*

Then, A is defined by a pair (A_g, A_r) such that for each execution trace Tr either $A_r(Tr) = \emptyset$, or $A_g(Tr) = \emptyset$ and $\forall h \in H$ such that $(h, p_h) \in A_r(Tr)$ it holds that p_h is the limiting value 1.

Proof Along the proof, we use the following notation:

- $I_s^\pi = \{t \mid (s, \pi, \_, t) \in T\}$ is the set of states that are reachable from s by executing an action π.
- $p_{s,s'}^\pi$ is the probability of going from s to s' by executing an action π.
- $X_{a,C}(s, t, p_{s,t}^\pi) = p_{s,t}^\pi \cdot Prob(t, \tau^* a, C)$.
- $p(s, A_g(Tr)) = \sum\{\!\!\{\, p \mid \exists a \in Act_L \cup A_g(Tr) . (s, a, p, \_) \in T \,\}\!\!\}$.

By hypothesis, there exists $R \in \mathcal{R}$ such that $\delta_A^R = \inf_{R' \in \mathcal{R}} \sup_{A'} \delta_{A'}^{R'}$. In particular, δ_A^R is the maximum distance $|Prob(s', \tau^* a, C) - Prob(s'', \tau^* a, C)| \cdot Prob(s_0^A, s') \cdot Prob(s_0, s'')$, where $a \in Act, C \in S_{/R}, s' \in \mathcal{S} \mid A, s'' \in \mathcal{S}\backslash A$, $(s', s'') \in R$, and s_0^A, s_0 are the initial states of $\mathcal{S} \mid A, \mathcal{S}\backslash A$, respectively.

We observe that $Prob(s'', \tau^* a, C)$ and $Prob(s_0, s'')$ are fixed values that do not depend on the adversary strategy. Therefore, in order to maximize the distance, the adversary can follow two different strategies. In the former the adversary aims at maximizing both $Prob(s_0^A, s')$ and $Prob(s', \tau^* a, C)$, while in the latter the adversary aims at maximizing $Prob(s_0^A, s')$ and minimizing $Prob(s', \tau^* a, C)$. By virtue of Definition 7, the strategy determining $Prob(s_0^A, s')$ is completely independent of the strategy followed to derive $Prob(s', \tau^* a, C)$. Hence, in the following we show how to maximize $Prob(s', \tau^* a, C)$, by observing that we can symmetrically argue in order to minimize it and we can follow the same argumentation in order to maximize $Prob(s_0^A, s')$.

We can assume that $s' = (s, Tr')$ is obtained from $s \in S$ by applying the algorithm of Definition 8. We now show how to maximize $Prob(s', \tau^* a, C)$ by induction on the length of the trace Tr that starts at s'.

Base step: $Tr = \varepsilon$, i.e., the current state is indeed $s' = (s, Tr')$. The strategy followed by A at s' is as follows. First, we assume $(h, \_) \notin A_r(Tr')$ for each high-level type h such that the reactive bundle of type h is empty at s. Then, the probability value associated with each other reactive bundle is calculated on the basis of the structure of the bundles enabled at s:

1. If the generative bundle of s is empty and one reactive bundle, of type h, is enabled at s then:

$$Prob(s', \tau^* a, C) = \sum_{t \in I_s^{h_*}} X_{a,C}(s, t, p_{s,t}^{h_*})$$

   Given that the probability value associated with the reactive bundle of type h is not used, we can assume $(h, 1) \in A_r(Tr)$.
2. If the generative bundle of s is non-empty and no reactive bundle is enabled at s, then

$$Prob(s', \tau^* a, C) = \frac{1}{p(s, A_g(Tr))} \cdot Prob(s, a, C) + \frac{1}{p(s, A_g(Tr))} \cdot \sum_{k \in A_g(Tr)} \sum_{t \in I_s^k} X_{a,C}\left(s, t, p_{s,t}^k\right)$$

   As a consequence, we set $A_r(Tr) = \emptyset$. Moreover, $A_g(Tr) \subseteq H_s$ can be chosen by A in order to maximize $Prob(s', \tau^* a, C)$. Indeed, by Definition 7 the strategy of A at the current state is completely independent of the strategy of A in the next future.
3. If the generative bundle of s is empty and $\{h_1, \ldots, h_n\}$, with $n > 1$, is the (alphabetically ordered) sequence of types of the non-empty reactive bundles enabled at s, then

$$\begin{aligned} &Prob(s', \tau^* a, C) \\ &= \prod_{1 < j \le n} (1 - p_j) \cdot \sum_{t \in I_s^{h_{1*}}} X_{a,C}\left(s, t, p_{s,t}^{h_{1*}}\right) \\ &+ \sum_{1 < i < n} p_i \cdot \prod_{i < j \le n} (1 - p_j) \cdot \sum_{t \in I_s^{h_{i*}}} X_{a,C}\left(s, t, p_{s,t}^{h_{i*}}\right) \\ &+ p_n \cdot \sum_{t \in I_s^{h_{n*}}} X_{a,C}\left(s, t, p_{s,t}^{h_{n*}}\right) \end{aligned}$$

By hypothesis, there exists

$$X = \max \left\{ \sum_{t \in I_s^{h_{i*}}} X_{a,C}(s, t, p_{s,t}^{h_{i*}}) \mid 1 \leq i \leq n \right\}.$$

Let $X = \sum_{t \in I_s^{h_{j*}}} X_{a,C}(s, t, p_{s,t}^{h_{j*}})$, with $j \in \{1, \ldots, n\}$. Since the current strategy of A does not affect the future behavior of A, the maximum value for the formula above is obtained by taking $A_r(Tr) = \{(h_j, 1)\}$.

4. If the generative bundle of s is non-empty and $\{h_1, \ldots, h_n\}$, with $n \geq 1$, is the (alphabetically ordered) sequence of types of the non-empty reactive bundles enabled at s, then

$$\begin{aligned} &Prob(s', \tau^* a, C) \\ &= \prod_{1 \leq i \leq n} (1 - p_i) \cdot \frac{1}{p(s, A_g(Tr))} \cdot Prob(s, a, C) \\ &\quad + \prod_{1 \leq i \leq n} (1 - p_i) \cdot \frac{1}{p(s, A_g(Tr))} \\ &\quad \cdot \sum_{k \in A_g(Tr)} \sum_{t \in I_s^k} X_{a,C}(s, t, p_{s,t}^k) \\ &\quad + \sum_{1 \leq i < n} p_i \cdot \prod_{i < j \leq n} (1 - p_j) \\ &\quad \cdot \sum_{t \in I_s^{h_{i*}}} X_{a,C}(s, t, p_{s,t}^{h_{i*}}) \\ &\quad + p_n \cdot \sum_{t \in I_s^{h_{n*}}} X_{a,C}(s, t, p_{s,t}^{h_{n*}}) \end{aligned}$$

Calculate X similarly as done before. With respect to the previous case, now it is worth considering the generative bundle, which may enable high-level output actions. Hence, if there exists $G \subseteq H_s$ such that

$$\begin{aligned} &\frac{1}{p(s, G)} \cdot Prob(s, a, C) + \frac{1}{p(s, G)} \\ &\quad \cdot \sum_{k \in G} \sum_{t \in I_s^k} X_{a,C}(s, t, p_{s,t}^k) > X \end{aligned}$$

then the adversary sets $A_g(Tr) = G$ and $A_r(Tr) = \emptyset$. Otherwise, assume h_j to be the type of the reactive bundle related to X. Then, the adversary sets $A_g(Tr) = \emptyset$ and $A_r(Tr) = \{(h_j, 1)\}$.

Inductive step: the length of Tr is $n - 1$ with $n > 1$, i.e., the current state is not $s' = (s, Tr')$. Instead, the current state is of the form $(\_, Tr'')$ where $Tr'' = Tr'Tr$ and Tr is a sequence of high-level actions executed to reach the current state. Note that several different states can be reached from s by executing the trace Tr. Hence, assume that $\{s_1, \ldots, s_m\} \subseteq S$ is the set of states such that (s_j, Tr''), with $j \in \{1, \ldots, m\}$, is reachable from $s' = (s, Tr')$.

We now define the strategy of the n-th move of the adversary A, by assuming that, by induction hypothesis, the result holds for each $i < n$. For this purpose, note that for each (s_j, Tr'') the adversary must follow the same strategy.

Let $Prob(s', Tr, (s_j, Tr''))$ be the aggregate probability of reaching (s_j, Tr'') from s' through the trace Tr.

Then, it holds that the strategy followed by A at each (s_j, Tr'') maximizes the following expression:

$$\sum_{j=1}^{m} Prob(s', Tr, (s_j, Tr'')) \cdot Prob(s_j, \tau^* a, C) \tag{4}$$

Assume $\{h_1, \ldots, h_n\}$ to be the (alphabetically ordered) sequence of types of the union of the high-level reactive bundles enabled at $s_1, \ldots, s_m$, with $\{p_1, \ldots, p_n\}$ the related probability distributions chosen by the adversary. According to what we have shown in the base case, each $Prob(s_j, \tau^* a, C)$ is computed as the summation of a number of sub-terms, each one possibly multiplied by some variable p_i and/or $(1 - p_i)$ determined by the adversary.

Therefore, we now simplify Eq. 4 by moving towards the outermost position of the equation all the variables p_i. By commutativity of multiplication and distributivity of multiplication with respect to summation, we obtain a summation of some of the sub-terms forming the following equation:

$$\begin{aligned} &\sum_{i=1}^{n} p_i \cdot X'_{p_i} + \sum_{Sub \subseteq \{1, \ldots, n\}} \prod_{i \in Sub} (1 - p_i) \cdot X'_{Sub} \\ &\quad + \sum_{i=1}^{n-1} \sum_{Sub \subseteq \{i+1, \ldots, n\}} p_i \cdot \prod_{j \in Sub} (1 - p_j) \cdot X'_{p_i, Sub} + X'_{\emptyset} \end{aligned} \tag{5}$$

where X'_{p_i} is the sub-term that is to be multiplied by p_i, X'_{Sub} is the sub-term that is to be multiplied by $\prod_{i \in Sub}(1 - p_i)$, $X'_{p_i, Sub}$ is the sub-term that is to be multiplied by $p_i \cdot \prod_{j \in Sub}(1 - p_j)$, and $X'_{\emptyset}$ is the sub-term that is not multiplied by any variable determined by the adversary. Note that Eq. (5) really equates Eq. (4) by assuming $X'_* = 0$ for each sub-term X'_* that does not occur in Eq. (4).

Let $k \in \{1, \ldots, n\}$ be the maximum value such that $p_k \cdot X'_{p_k}$ is a non-null sub-term of Eq. (5), and $k' \in \{1, \ldots, n\}$ be the number of variables of set $\{p_1, \ldots, p_n\}$ occurring in non-null sub-terms of Eq. (5). We now show by induction on k' that Eq. (5) is maximized whenever $\forall j : p_j \in \{0, 1\}$.

Base case: $k' = 1$. By Definition 8, Eq. (5) consists of the sub-terms $p_k \cdot X'_{p_k}$, $(1 - p_k) \cdot X'_{1 - p_k}$ and $X'_{\emptyset}$. The related expression is easily maximized by taking either $p_k = 1$ or $p_k = 0$.

Inductive step: $k' > 1$. By Definition 8, Eq. (5) includes a unique sub-term that is multiplied by p_k, that is $p_k \cdot X'_{p_k}$.

Hence, we can rewrite Eq. (5) as follows:

$$p_k \cdot X'_{p_k} + \sum_{Sub \subseteq \{1,\ldots,k-1\} \cup \{k\}} \prod_{i \in Sub} (1 - p_i) \cdot X'_{Sub} + \sum_{i=1}^{k-1} \sum_{Sub \subseteq \{i+1,\ldots,k-1\} \cup \{k\}} p_i \cdot \prod_{j \in Sub} (1 - p_j) \cdot X'_{p_i, Sub} + X_{\overline{1-p_k}} \quad (6)$$

where $X_{\overline{1-p_k}}$ is the summation over all the non-null sub-terms of Eq. (5) which are not multiplied neither by p_k nor by $(1 - p_k)$. Now assume $p_k = 0$, so that Eq. (5) reduces to a summation of sub-terms over a set of $k' - 1$ variables of the set $\{p_1, \ldots, p_n\}$. By induction hypothesis, such a summation is maximized by taking either $p_j = 1$ or $p_j = 0$ for each $j \neq k$. If the maximum value we obtain is greater than $X'_{p_k} + X_{\overline{1-p_k}}$, then we assume $p_k = 0$ to maximize Eq. (5), otherwise we assume $p_k = 1$, from which we derive the result and, as an immediate consequence, the proof of the theorem. □

Example 13 Consider again the example of Fig. 5. A history-dependent adversary can block the action h_* at the initial state, force the execution of the action k_* after the trace l, and force the execution of the action h_* after the trace $l.k_*$. In this way, the distinguishing state is reached with probability tending to 1. Similarly, in the case of Fig. 6, a history-dependent adversary can first block the action h_* and, after one step, force its execution twice, thus reaching the distinguishing behavior with probability tending to 1.

In general, in order to determine the most powerful adversary, it is sufficient to check a finite number of strategies without solving a set of constraint programming problems; this finite number exponentially depends on the amount of different high-level actions that the system enables. Theorem 1 applies to systems that do not execute internal invisible actions. We now discuss the need for such a requirement through the following example.

Example 14 Consider the GRTS of Fig. 7, which includes two nondeterministic choices between a high-level input action and an internal system activity. The interference of the adversary is revealed by the execution of the output i. If the observational power of the adversary does not reveal the events internally executed by the system, then the initial state and the state reachable by executing the action τ are indistinguishable from the viewpoint of the adversary. That means the adversary cannot change strategy when the system moves from the former to the latter. As a consequence, the most powerful history-dependent adversary solves the nondeterministic choices between k_* and τ by assuming a probability distribution governed by parameter 1/2. On the other hand, if we assume that the adversary can reveal the internal moves performed by the system—that is the traces ε and τ are different from the viewpoint of the adversary—then Theorem 1 holds without restrictions on the form of the GRTS. Indeed, under such a hypothesis, the adversary can change strategy after each visible and invisible event performed by the system. Hence, in our example, the most powerful history-dependent adversary assumes $A_r(\varepsilon) = \emptyset$ and $A_r(\tau) = \{(k, 1)\}$.

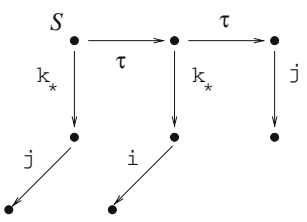

Fig. 7 A GRTS with internal unobservable activities

6 A process algebraic view of adversaries

In this section, we rephrase the theory and application of approximate noninterference in the setting of a process algebra whose semantics is given by the GRTS model introduced before. The objective is to relate the three classes of adversaries to probabilistic noninterference properties defined in the framework of such a process algebra.

6.1 Probabilistic process algebra

The probabilistic process algebra we consider relies on the GRTS model. Hence, the actions describing the behavior of concurrent systems are classified as in Sect. 2. Probabilistic information is encoded by enriching the algebraic operators with probability values. In the following we briefly recall the syntax and the operational semantics of such a language [4,12].

The set $\mathcal{L}$ of process terms is generated by the syntax:

$$P ::= \underline{0} \mid \pi.P \mid P +^p P \mid P \parallel_S^p P \mid P /_a^p \mid P \backslash L \mid C,$$

where $S, L \subseteq AType$, $a \in AType$, and $p \in]0, 1[$. We denote by $\mathcal{G}$ the set of finite state, guarded terms of $\mathcal{L}$, which we call processes [25]. Similarly as done before, we restrict ourselves to the set $\mathcal{G}_L$ of processes that are fully specified from the viewpoint of Low, i.e., no low-level input action is enabled. Thus the nondeterminism is limited to the interactions with the high-level environment. Essentially, the algebraic operators implement the mixed generative and reactive model of probabilities described in Sect. 2:

– $\underline{0}$ represents the null, deadlocked process (we usually omit $\underline{0}$ when it is clear from the context).

- $\pi.P$ performs the action π with probability 1 and then behaves like P.
- $P +^p Q$ is a CCS-like alternative choice operator, where every probabilistic choice between P and Q is governed by parameter p. In particular, in a probabilistic choice either P moves with probability p or Q moves with probability $1 - p$.

Example 15 The GRTS underlying process $(j +^p i) +^q h_*.l$ is that of Fig. 1. The innermost choice operator models the probabilistic choice between the two output actions j and i, which is governed by parameter p of the operator. On the other hand, the outermost choice operator does not express any probabilistic choice. Indeed, the choice between the input action and the two output actions is nondeterministic, so that parameter q is not used.

Similarly, take process $(j +^{1/2} i) +^{2/3} h.l$, whose corresponding GRTS is depicted in Fig. 3 under the hypothesis $p = q = 1 - (p + q) = 1/3$. As imposed by the outermost choice operator, the probability of executing the high-level output h is equal to $1 - 2/3 = 1/3$, while the probability of executing one between j and i is equal to $2/3$. More precisely, the probability of executing the output j (resp. i) is equal to $2/3 \cdot 1/2 = 1/3$, as imposed by the combination of the two choice operators.

- $P \parallel_S^p Q$ is a CSP-like parallel composition operator. P and Q asynchronously and independently execute all the actions not in S, while they are constrained to synchronously execute actions of type in S if they are of the same type, which becomes the type of the resulting action. A synchronization is possible only between an output action and an input action, which results in an output action, or between two input actions, which results in an input action. The probability p is the parameter of a probabilistic scheduler that, in each system state where P and Q enable actions of the same bundle, probabilistically decides which action of that bundle must be scheduled, i.e., one of P with probability p or one of Q with probability $1 - p$. Because of the synchronization policy, some actions locally enabled by P and Q could be prevented when composing in parallel P and Q. Thus, a normalization of the generative bundle enabled by $P \parallel_S^p Q$ is needed to fulfill the second requirement of Definition 1.

Example 16 Consider process $((j +^p i) +^q h_*.l) \parallel_{\{h\}}^{p'} \underline{0}$, whose semantics is the GRTS of Fig. 2a. The synchronization policy blocks the input of type h, because the right-hand process does not enable an input/output action of the same type. Since no probabilistic choice occurs between the left-hand process and the right-hand process, parameter p' is not used, similarly as seen in the case of the choice operator.

Now, take process $(j +^{1/2} i) +^{2/3} h.l \parallel_{\{h\}}^{p'} \underline{0}$, which corresponds to the GRTS without adversary interference depicted in Fig. 3 under the hypothesis $p = q = 1 - (p + q) = 1/3$. Again, the synchronization policy blocks the action of type h. Hence, the probability $1/3$ associated with the blocked event is redistributed between the other two available events, as illustrated in Fig. 3.

In order to illustrate an example of synchronous communication, take again the GRTS of Fig. 2a, which may be generated by process $(j +^{p'} i_*) \parallel_{\{i,j\}}^{p} (i +^{p''} j_*)$. The local output j offered by the left-hand process is executable by the overall system thanks to the input of the same type offered by the right-hand process. We can argue symmetrically in the case of output i. Their execution probability is governed by parameter p of the operator exactly as seen for the choice operator, while parameters p' and p'' are not used for analogous reasons.

- The hiding operator $P/_a^p$ turns output and input actions of type a into actions τ, by normalizing, if necessary, the associated probabilities.

Example 17 Take process $((j +^p i) +^{p'} h_*.l)/_h^q$, whose semantics is given by the GRTS of Fig. 2b. Hiding the input action of type h means turning an action belonging to a reactive bundle into an internal action belonging to the generative bundle. In order to correctly define the resulting generative bundle, the probabilities associated with the input actions of type h and the probabilities associated with the previously enabled generative actions are normalized according to parameter q of the hiding operator, as illustrated in Fig. 2b.

Similarly, process $((j +^{1/2} i) +^{2/3} h.l)/_h^q$ corresponds to the GRTS with adversary interference depicted in Fig. 3 under the hypothesis $p = q = 1 - (p + q) = 1/3$. Here, hiding the output action of type h does not alter the structure of the generative bundle, so that any normalization is not needed and parameter q of the operator is not used.

- The restriction operator $P \backslash L$ prevents from execution all the actions with type in L. Similarly as seen in the case of the parallel composition operator, a normalization of the execution probability of the remaining actions of the generative bundle may be necessary to fulfill the second requirement of Definition 1. We do not detail such an operator because it can be expressed in terms of parallel composition. In fact, $P \backslash L$ and $P \parallel_L^p \underline{0}$ have the same semantics independently of the value of p.
- Constant C is used to specify recursive systems. In particular, we assume a set of constants defining equations of the form $C \triangleq P$ to be given.

We now provide a brief formal presentation of the semantics of the probabilistic process algebra. Formally, the

Table 1 Operational semantics (part I)

$$\pi.P \xrightarrow{\pi,1} P$$

$$\frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P +^p Q \xrightarrow{a_*,p\cdot q} P'} \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}\!\!\!\!\!\!/}{P +^p Q \xrightarrow{a_*,q} P'}$$

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{GAct}}{P +^p Q \xrightarrow{a,p\cdot q} P'} \qquad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{GAct}\!\!\!\!\!\!/}{P +^p Q \xrightarrow{a,q} P'}$$

$$\frac{P \xrightarrow{a_*,q} P' \quad P \xrightarrow{GAct}}{P/_a^p \xrightarrow{\tau,p\cdot q} P'/_a^p} \qquad \frac{P \xrightarrow{a_*,q} P' \quad P \xrightarrow{GAct}\!\!\!\!\!\!/}{P/_a^p \xrightarrow{\tau,q} P'/_a^p}$$

$$\frac{P \xrightarrow{b_*,q} P'}{P/_a^p \xrightarrow{b_*,q} P'/_a^p}\ a \neq b$$

$$\frac{P \xrightarrow{b,q} P' \quad P \xrightarrow{a_*}}{P/_a^p \xrightarrow{b,(1-p)\cdot q} P'/_a^p}\ a \neq b \qquad \frac{P \xrightarrow{a,q} P' \quad P \xrightarrow{a_*}}{P/_a^p \xrightarrow{\tau,(1-p)\cdot q} P'/_a^p}$$

$$\frac{P \xrightarrow{b,q} P' \quad P \xrightarrow{a_*}\!\!\!\!\!\!/}{P/_a^p \xrightarrow{b,q} P'/_a^p}\ a \neq b \qquad \frac{P \xrightarrow{a,q} P' \quad P \xrightarrow{a_*}\!\!\!\!\!\!/}{P/_a^p \xrightarrow{\tau,q} P'/_a^p}$$

$$\frac{P \xrightarrow{\pi,q} P'}{C \xrightarrow{\pi,q} P'}\ \text{if } C \triangleq P$$

Table 2 Operational semantics (part II)

$$\frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P \parallel_S^p Q \xrightarrow{a_*,p\cdot q} P' \parallel_S^p Q}\ a \notin S$$

$$\frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}\!\!\!\!\!\!/}{P \parallel_S^p Q \xrightarrow{a_*,q} P' \parallel_S^p Q}\ a \notin S$$

$$\frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*,q'} Q'}{P \parallel_S^p Q \xrightarrow{a_*,q\cdot q'} P' \parallel_S^p Q'}\ a \in S$$

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{G_{S,P}}}{P \parallel_S^p Q \xrightarrow{a,p\cdot q/\nu_P(G_{S,Q})} P' \parallel_S^p Q}\ a \notin S$$

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{G_{S,P}}\!\!\!\!\!\!/}{P \parallel_S^p Q \xrightarrow{a,q/\nu_P(G_{S,Q})} P' \parallel_S^p Q}\ a \notin S$$

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{a_*,q'} Q' \quad Q \xrightarrow{G_{S,P}}}{P \parallel_S^p Q \xrightarrow{a,p\cdot q'\cdot q/\nu_P(G_{S,Q})} P' \parallel_S^p Q'}\ a \in S$$

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{a_*,q'} Q' \quad Q \xrightarrow{G_{S,P}}\!\!\!\!\!\!/}{P \parallel_S^p Q \xrightarrow{a,q'\cdot q/\nu_P(G_{S,Q})} P' \parallel_S^p Q'}\ a \in S$$

semantics of a process P is given by a rooted GRTS whose transition relation is the least multi-set satisfying the operational rules reported in Tables 1 and 2, and whose initial state is P.

As far as the notation is concerned, we denote by $RAct$ and $GAct$ the sets of input actions and of output and internal actions, respectively. Then, we use the abbreviations $P \xrightarrow{\pi}$ to stand for $\exists p,\ P' : P \xrightarrow{\pi,\ p} P'$, denoting that P can execute action π with probability p and then behave as P', and $P \xrightarrow{G}$, with $G \subseteq GAct$, to stand for $\exists a \in G : P \xrightarrow{a}$, meaning that P can execute a generative action belonging to set G.

As far as $P +^p Q$ and $P \parallel_S^p Q$ are concerned, in addition to the reported rules, which refer to the local moves of the left-hand process P, we also consider the symmetric rules taking into account the local moves of the right-hand process Q. Such symmetric rules are obtained by exchanging the roles of terms P and Q and by replacing p with $1 - p$ in the label of the derived transitions.

The semantic rules of Table 2 for parallel composition show that the execution probability of each generative transition of P executed by $P \parallel_S^p Q$ is subject to the normalization factor $\nu_P(G_{S,Q})$, which is necessary because, as we have seen, some actions of the generative bundle of P may be blocked by the context $\_ \parallel_S^p Q$. In particular, the set $G_{S,Q}$ contains all the types of the actions that are executable in the context $\_ \parallel_S^p Q$, i.e., the action types not belonging to the synchronization set S, which are not subject to any constraint, and the action types belonging to S for which an input action of Q can be performed, thus allowing for a synchronization. Formally, $G_{S,Q} = \{a \in AType \cup \{\tau\} \mid a \notin S \vee (a \in S \wedge Q \xrightarrow{a_*})\}$. Then, function $\nu_P(A)$ computes the sum of the probabilities of all the generative transitions of P with type in A. Formally, $\nu_P : \mathcal{P}(AType \cup \{\tau\}) \longrightarrow]0, 1]$ and $\nu_P(A) = \sum\{\!|\ p \mid \exists a,\ P' : P \xrightarrow{a,\ p} P' \wedge a \in A\ |\!\}$.

Obviously, we can argue symmetrically for the generative transitions of Q, which are subject to the normalization factor $\nu_Q(G_{S,P})$ when executed in the context $P \parallel_S^p \_$.

6.2 Adversaries vs noninterference properties

In this section, we study the relation between the three families of adversaries of Sect. 3.2 and some probabilistic noninterference-based property defined in the setting of our probabilistic process algebra. In particular, we will show that every property we consider expresses the behavior of a specific class of adversaries.

First, we consider the probabilistic extension of noninterference, which is called bisimulation-based strong probabilistic noninterference (BSPNI, see [4]). In order to detect the high-level interference for a system model P, the BSPNI

property compares the low-level behavior of P under two scenarios differing in the high-level behaviors only. In the former, P is isolated from the high-level environment, so that all the high-level interactions are prevented, while in the latter P can interact with the high-level environment through any action of the high-level interface of P.

In the following we assume that $\{h_1^P, \ldots, h_{n_P}^P\} \subseteq H$ is the set of high-level action types that syntactically occur in the action prefix operators within P.

Definition 12 $P \in$ BSPNI if and only if

$$P\backslash H \approx_{PB} P/^{p_1}_{h_1^P} \cdots /^{p_{n_P}}_{h_{n_P}^P} \qquad \forall p_1 \cdots p_{n_P} \in]0, 1[.$$

On one hand, $P\backslash H$ models the absence of any adversary interference. On the other hand, the vector of values $p_1 \cdots p_{n_P}$ in the hiding operators expresses the strategy of an adversary interacting with the system. Because of the universal quantification, the BSPNI property considers an infinite set of such strategies, which we call $\mathcal{A}_{\text{BSPNI}}$. Intuitively, such strategies, which are independent of the behavior of the system itself and of the passage of time, correspond to a subset of the simple adversaries.

Proposition 4 $\mathcal{A}_{\text{BSPNI}} \subset \mathcal{A}_S$.

Proof Definition 3 subsumes, at the GRTS level, the semantics of the probabilistic hiding operator when $A_g = H$ and $\forall h \in H : (h, p_h) \in A_r$, with $p_h \in]0, 1[$. Hence, the result immediately follows from Definition 12. □

Whenever the system satisfies BSPNI it holds that every strategy in $\mathcal{A}_{\text{BSPNI}}$ does not succeed in revealing the adversary interference to the low-level observer. On the other hand, if the system does not satisfy BSPNI, then there exists at least one adversary strategy that violates the bisimulation based check of Definition 12 and, therefore, sets up a covert channel from high-level to low-level. In our setting, it is possible not only to find one such strategy, but also to estimate the covert channel in terms of the maximum probability of observing it, which is given by the quantity $\bar{\varepsilon}$. More precisely, the maximum information leakage can be estimated by measuring how far is from $\approx_{PB}$ the closest approximation of $\approx_{PB}$ that relates $P\backslash H$ and $P/^{p_1}_{h_1^P} \cdots /^{p_{n_P}}_{h_{n_P}^P}$ whenever $p_1, \ldots, p_{n_P}$ model the most powerful adversary.

Unfortunately, solving the problem of finding the most powerful simple adversary in $\mathcal{A}_{\text{BSPNI}}$ is impractical. In particular, the examples and the observations shown in Sect. 5 make it clear the complexity of this problem, which is hyper-exponential.

Then, we consider a stronger property, that is the probabilistic variant of bisimulation-based nondeducibility on compositions (PBNDC, see [4]), which says that the probabilistic low-level behavior of P must be invariant with respect to the execution of P in parallel with any high-level process $\Pi \in \mathcal{G}_H$, where $\mathcal{G}_H \subseteq \mathcal{G}$ is the set of processes that enable high-level actions only.

Definition 13 $P \in$ PBNDC if and only if

$$P\backslash H \approx_{PB} ((P \parallel^p_K \Pi)/^{p_1}_{h_1^K} \ldots /^{p_{n_K}}_{h_{n_K}^K})\backslash H$$

$$\forall K = \{h_1^K, \ldots, h_{n_K}^K\} \subseteq H, \Pi \in \mathcal{G}_H, p, p_1 \ldots p_{n_K} \in]0, 1[.$$

As shown in [4], PBNDC implies BSPNI. In essence, the high-level processes Π of the PBNDC definition represent adversary strategies among which we also have the simple adversaries. Similarly as seen in the case of the interactive adversaries, a high-level interference is caused by the interaction between the system and an active adversary Π that takes into consideration the current high-level interface of the system. However, the high-level processes Π and the interactive adversaries differ from each other because of some technicalities. In fact, the strategy of an interactive adversary A strictly depends on the current high-level interface of the system, while the strategy of a high-level process Π strictly depends on the sequence of synchronizations between P and Π, because Π moves through such communications only. Thus, if the system interacts with the high-level environment without altering its high-level interface, then A cannot change strategy while Π can. By contrast, if the system interacts with the low-level environment and, in doing so, its high-level interface changes, then A can change strategy while Π cannot.

In spite of such differences, both families of adversaries suffer from the same limitations, which make it impossible to foresee the behavior of the most powerful adversary whenever the PBNDC property is not satisfied. In practice, finding the process Π that maximizes the observable difference between $P\backslash H$ and the low-level view of $(P \parallel^p_K \Pi)$ is as difficult as finding the most powerful interactive adversary.

Example 18 The system of Fig. 6 is not PBNDC-secure. In particular, finding the high-level process Π that maximizes the covert channel revealed by PBNDC leads to an argumentation similar to that of Example 12. More precisely, Π cannot adopt different strategies for the two first occurrences of the input of type h, exactly as seen for the interactive adversaries.

Based on the considerations above, our objective is to determine whether a reasonable security property can be defined that models the expressive power of the class of history-dependent adversaries. This would allow us to exploit the result of Theorem 1 in order to efficiently estimate the maximum covert channel that can be set up in the case of property violation (or, equivalently, evaluate the most powerful adversary). As far as we know such a property has not been defined yet in the literature, at least in the probabilistic case. Indeed, if we abstract away from the probabilistic

information, it turns out that the notion of history-dependent adversary applies to the labeled transition system model just as the classical definition of nondeducibility on strategies (NDS, see [23]) applies to the nondeterministic state machine model.

For the definition of a probabilistic security property inspired by NDS, we need an operator that implements at the process-algebraic level the idea of history-dependent adversary. This has been done in [1] in the nondeterministic case. Here, we propose a probabilistic extension of such an operator inspired by Definition 7.

In the following, we assume that each state reachable from P under the interference of the history-dependent adversary A is described by a pair (P', γ), where P' is a process reachable from P and γ expresses the sequence of actions executed along the path from P to P'. For the initial state, the associated configuration is (P, ε). Note that for each process P' reachable from P we can have several different states of the form (P', γ) depending on the number of traces that may be executed to reach P'.

We now introduce a probabilistic strategy operator $(P, \gamma)|A$, whose semantics is defined by the operational rules of Table 3. Intuitively, $(P, \gamma)|A$ describes the interactions between P and the environment on the basis of the strategy decided by the adversary A, which in turn depends on the past history modeled by the trace γ. Formally, both invisible and low-level actions are enabled (see the first two rules), while the high-level actions are blocked except for those enabled by A, which are turned into τ actions (see the other two rules). A normalization of the probability distribution of the transitions of the generative bundle is needed exactly as shown in the previous sections. In particular, function ν is used to redistribute the probabilities of the generative transitions according to the restriction of some high-level output actions (see function g) and the hiding of some high-level input actions (see function r). Similarly, function ν_h is used to normalize the probabilities of the τ actions obtained by hiding high-level input actions of type h. The operator $<$ (applied to action types like in $h < h_i$) must be interpreted as a lexicographic comparison operator.

Table 3 Operational semantics of the strategy operator

$$\frac{P \xrightarrow{a,p} P'}{(P,\gamma)|A \xrightarrow{a,p\cdot\nu(P,\gamma,A)} (P',\gamma a)|A} \; a \in Act_L \cup \{\tau\}$$

$$\frac{P \xrightarrow{a_*,p} P'}{(P,\gamma)|A \xrightarrow{a_*,p} (P',\gamma a_*)|A} \; a \in Act_L$$

$$\frac{P \xrightarrow{h,p} P'}{(P,\gamma)|A \xrightarrow{\tau,p\cdot\nu(P,\gamma,A)} (P',\gamma h)|A} \; h \in A_g(\gamma)$$

$$\frac{P \xrightarrow{h_*,p} P'}{(P,\gamma)|A \xrightarrow{\tau,p\cdot\nu_h(P,\gamma,A)} (P',\gamma h_*)|A} \; (h,p_h) \in A_r(\gamma)$$

where:

$$\nu(P,\gamma,A) = (1/g(P,A_g(\gamma))) \cdot r(P,A_r(\gamma))$$

$$g(P,A_g(\gamma)) = \sum\{\!|\, q \mid \exists Q \in \mathcal{G}, a \in Act_L \cup A_g(\gamma) \cup \{\tau\}. P \xrightarrow{a,q} Q \,|\!\}$$

$$r(P,A_r(\gamma)) = \prod\{\!|\, (1-p_i) \mid (h_i,p_i) \in A_r(\gamma) \wedge P \xrightarrow{h_{i*}} \,|\!\}$$
s. t. $r(P,A_r(\gamma)) = 1$ if the multi-set is empty

$$\nu_h(P,\gamma,A) = q \cdot \prod\{\!|\, (1-p_i) \mid (h_i,p_i) \in A_r(\gamma) \wedge h < h_i \wedge P \xrightarrow{h_{i*}} \,|\!\}$$
s. t. $\nu_h(P,\gamma,A) = q$ if the multi-set is empty

$$q = \begin{cases} p_h & \text{if } g(P,A_g(\gamma)) > 0 \vee \exists (h_i,p_i) \in A_r(\gamma). \\ & h_i < h \wedge P \xrightarrow{h_{i*}} \\ 1 & \text{otherwise} \end{cases}$$

Definition 14 $P \in \text{PNDS}$ if and only if

$$P\backslash H \approx_{PB} (P,\varepsilon)|A \quad \forall A \in \mathcal{A}_{HD}.$$

This definition states that P is PNDS-secure if its execution is invariant, from the viewpoint of Low, with respect to any strategy decided by a history-dependent adversary. If we call $\mathcal{A}_{\text{PNDS}}$ the set of adversary strategies described by the PNDS property, then it is immediate to derive the following result.

Proposition 5 $\mathcal{A}_{\text{PNDS}} = \mathcal{A}_{HD}$.

Proof The result follows from the definition of history-dependent adversary and from the semantics of the probabilistic strategy operator. □

Proposition 6 PNDS $\subset$ BSPNI.

Proof The result follows from the previous proposition, from Propositions 2 and 4. □

The PNDS property can be viewed as a probabilistic variant of NDS defined in a process-algebraic setting. The main strength of this property with respect to the other properties introduced in [4] is that in the case of property violation, it is possible to efficiently estimate the covert channel that is set up by the most powerful adversary. This can be done, thanks to Theorem 1, by studying the behavior of a finite set of history-dependent adversaries.

7 A case study

We show the importance of the result established in the previous section by means of a case study considered in [8,6],

i.e., a probabilistic non-repudiation protocol. In particular, in [6] we show that the protocol is not BSPNI-secure and we calculate the adversary that maximizes the information leakage by means of some ad hoc mathematical argumentation (as done in Example 10).

Repudiation consists of the denial by one of the entities involved in a message exchange protocol of having participated in all or part of the protocol itself: *non-repudiation of origin* is intended to prevent the *originator* of a message from denying having sent the message, and *non-repudiation of receipt* is intended to prevent the *recipient* of a message from denying having received the message. Especially in e-commerce, non-repudiation is needed to protect a transaction against any attempt to repudiate either the payment for the service or the delivery of the service.

The protocol we analyze offers a non-repudiation service without resorting to a trusted third party [26]. In particular, it offers a fair exchange of a message, sent by the originator O, which offers a service, for an acknowledgment, sent by the recipient R, which is expected to confirm the received service. The protocol is ε-*fair*, i.e., at each step of the protocol run, either both parties receive their expected information, or the probability that a cheating party gains any valuable information, while the other party gains nothing, is less than ε.

We now describe the protocol by presenting its formal model in the probabilistic process algebra of Sect. 6. In Fig. 8 we show the GRTS model of the protocol. For the sake of simplicity, we abstract from the cryptosystem used by the two authenticated parties and we concentrate on the packet exchange between them. We also abstract from the channel and the transmission delays, by assuming that a message which is delayed (not sent) by a participant is not delivered to the other participant. The algebraic specification describes the behavior of the originator as follows:

$$Or \triangleq receive\_request_*.send\_msg.receive\_ack_*.Or'$$
$$Or' \triangleq send\_msg.Or'' +^p send\_msg.Or'''$$
$$Or'' \triangleq receive\_ack_*.\underline{0} + receive\_stop_*.unfair.\underline{0}$$
$$Or''' \triangleq receive\_ack_*.Or' + receive\_stop_*.\underline{0}$$

The recipient R starts the protocol by sending a signed, timestamped request for a service to the originator O (input action $receive\_request_*$), which in turn sends the first signed, timestamped message containing the requested information M encrypted with a fresh key k (output action $send\_msg$). Upon receiving the first message from O, R sends a corresponding signed, timestamped acknowledgment message (input action $receive\_ack_*$). This initial handshake, modeled by process Or, precedes the execution of the probabilistic part of the protocol.

Then, at each protocol step O probabilistically decides whether to continue the protocol (with probability $1 - p$), by sending to R a signed, timestamped message containing a key different from k, or to terminate the protocol (with probability p), by sending a signed, timestamped message containing the key k needed to obtain the plain text M. Process Or' models such a probabilistic behavior of O.

Since the result of the probabilistic choice is not revealed by O, R cannot guess when the protocol terminates and, as a consequence, when it will receive the final message. Hence, a honest recipient transmits, for each received message, the corresponding ack message. Obviously, key to success of the protocol is the immediacy in sending back the ack. In particular, a cheating recipient may delay the transmission of the ack in order to check the validity of the received key and then block the transmission of the ack once the correct key has been received. If such an unfair behavior is detected, then O prematurely stops the protocol. To this aim, O fixes a deadline for the reception of each ack, after which, if the ack is not received, the protocol is stopped.

From the algebraic specification standpoint, the reception of the ack is modeled by the input $receive\_ack_*$, while the expiration of the timeout is abstracted through the input $receive\_stop_*$. The nondeterminism between these two events derives from the fact that the related choice depends on the environment behavior rather than an internal decision of the originator.

Process O''' models the waiting for an ack before the execution of another protocol step, while process O'' models the final step of the protocol. Upon the reception of the ack related to the last message containing k, O correctly terminates the protocol in a fair way. On the other hand, if the protocol

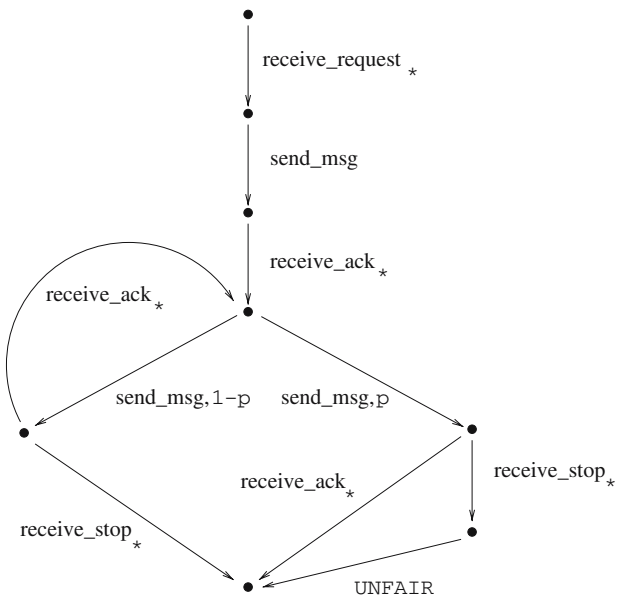

Fig. 8 GRTS model of the non-repudiation protocol

terminates before the transmission of the message that reveals k, then neither O nor R obtain any valuable information. The unique unfair case occurs whenever O sends the final message without obtaining the corresponding ack; this makes the protocol terminate in an unfair way from the viewpoint of the originator. This is signaled by executing the action of type *unfair*.

In general, given that the non-repudiation of origin is guaranteed by the message containing the encrypted value of M and the message containing the related key, while the non-repudiation of receipt is given by the last ack sent by R, the protocol guarantees non-repudiation of origin with probability 1 and non-repudiation of recipient with a probability less than 1. Hence, our goal is to estimate the probability of violating the non-repudiation of recipient.

Formally, the distinction between high-level actions and low-level actions follows an approach described in [19] for the analysis of network security and cryptographic protocols. More precisely, since the recipient is possibly an unfair adversary, we consider all the communications between the involved parties as high-level actions. Hence, the only low-level action is *unfair*, which reveals to a low-level observer the violation of the fairness condition.

In [6], it is shown that the verification of BSPNI reveals the potential unfair behavior of R. Then, a quantitative estimate of the efficiency of such an information leakage has been evaluated through the approximate noninterference approach. In particular, the two system views that derive from the definition of BSPNI are, respectively, the $\underline{0}$ process and the following version of the GRTS of Fig. 8: each high-level action is turned into a τ action, while the nondeterminism that derives from the choice between the high-level input actions $receive\_ack_*$ and $receive\_stop_*$ is probabilistically solved by a simple adversary as an internal choice guided by a given parameter q. Such a GRTS contains eight states, which can be lumped in classes to form a partitioning of the GRTS in 4,140 different ways. If we abstract from the sequence modeling the preliminary handshake, we can simply deal with five states that induce 52 equivalence relations. Among them, the closest approximation R of $\approx_{PB}$ is determined as follows. The unique state enabling the distinguishing action is the unique representative of its class, while all the other states, which enable the unobservable trace leading to the $\underline{0}$ process, belong to the same class of the initial state. Therefore, computing ε_R is a hard problem that corresponds to analyze an infinite number of simple adversaries, each one determining the value of parameter q, as shown in [6].

The security analysis of the repudiation protocol can be efficiently performed by considering PNDS. First, by virtue of Proposition 6 it holds that the protocol is not PNDS-secure. Second, according to the result of Theorem 1, it holds that the most powerful history-dependent adversary that maximizes the probability of violating non-repudiation employs limiting probabilities (tending to 1) to solve the nondeterminism due to the high-level inputs. More precisely, at each step of the protocol, only two strategies must be considered, i.e., the one blocking the protocol (by enabling the action $receive\_stop_*$ only) and the one sending a regular acknowledgment (by enabling the action $receive\_ack_*$ only). In particular, the adversary A_1 that follows the former strategy at the first step is easily described by the following strategy:

- $A_r(\varepsilon) = \{(receive\_request, 1)\}$
- $A_g(receive\_request_*) = (send\_msg)$
- $A_r(receive\_request_*.send\_msg) = \{(receive\_ack, 1)\}$
- $A_g(receive\_request_*.send\_msg.receive\_ack_*) = (send\_msg)$
- $A_r(receive\_request_*.send\_msg.receive\_ack_*.send\_msg) = \{(receive\_stop, 1)\}$

for which it holds that $\delta^R_{A_1} = p$, i.e., the information leakage occurs with probability p, which is a parameter under the control of O. The other possible adversary follows the latter strategy and allows the protocol to be continued, which is an event occurring with probability $1-p$. In general, at the $i-$th step the adversary A_i following the former strategy is such that $\delta^R_{A_i} = (1-p)^i \cdot p$, while the adversary A that always follows the latter strategy does not enable the unfair behavior, i.e., $\delta^R_A = 0$. Therefore, it holds that $\bar{\varepsilon} = \delta^R_{A_1} = p$. It is worth considering that this result is obtained without considering an exponential number of constraint programming problems with variable q, as it suffices to consider the two unique adversaries that assume the limiting behaviors $q = 0$ and $q = 1$, respectively.

As far as a comparison with the performance evaluation approach discussed in Sect. 4.1 is concerned, the following remarks are in order. In the example above, the unique action observable by Low is put into the protocol specification in order to reveal the information leakage that can be set up by the adversary. All the other actions are unobservable as are under the control of the adversary. As a consequence, only the system view modeling the interference of the adversary enables an invisible trace leading to a state enabling the unique low-level action. This is a common feature of most of protocols described through the approach defined in [19]. Based on such assumptions, it holds that Eqs. (2) and (3) of Sect. 4.1, which express the distance between the two different system views according to the performance evaluation and our approximate bisimulation approach, respectively, yield the same result. Indeed, if s' denotes the unique state of process $\underline{0}$ representing the view of the system in the absence of the adversary, then it follows that $\pi(s') = 1$ and $p' = 0$, from which the proof immediately follows.

8 Conclusion

We have shown how to effectively compute an estimate of the maximum information leakage of an insecure system. In our approach the system confidentiality is expressed via a notion of approximate noninterference; this is based on process indistinguishability with respect to a weak probabilistic bisimulation semantics. In practice, the lack of information leakage is expressed by a successful weak probabilistic bisimulation based check. Whenever such a check fails, approximate relations relax the conditions imposed by the weak probabilistic bisimulation, in such a way that the level of approximation represents an estimate of the amount of information leakage.

We have characterized the system information leakage via the behavior and the expressive power of classes of adversaries that are described by noninterference-based security properties. In the case of property violation, the most powerful adversary among those described by that property expresses the maximum information leakage that can be revealed by the property itself.

According to such a characterization, we have seen that estimating the maximum information leakage is in general impractical. In fact, the complexity of finding the most powerful adversary is hyper-exponential and depends on the verification of a possibly infinite number of adversary strategies. However, for a particular class of adversaries the calculation can be performed efficiently as it requires only a finite number of checks that exponentially depends on the number of activities controlled by the adversary. This class of adversaries corresponds to a novel probabilistic security property that turns out to be a probabilistic variant of the nondeducibility on strategies of [23].

In order to simplify the presentation of our approach to the analysis of approximate noninterference, we have considered systems that are fully specified from the viewpoint of Low. The more complex scenario in which the system can accept inputs from Low can be dealt with by extending in a natural way the approach adopted in this paper. The idea consists in parameterizing the formulation of the noninterference property with respect to a specific low-level user, who resolves all the nondeterminism due to the possible interactions of the low-level input interface of the system with the environment. By so doing, the result of the noninterference check expresses a measure of the maximum information leakage observed by that particular low-level observer.

We have presented an application of our approach to the analysis of a probabilistic non-repudiation protocol. This example also shows the relation of our approach with the performance evaluation approach that, e.g., has been employed to assess the security of a complex system, the NRL pump, in [2]. As a future work we intend to implement our approach in order to examine to what extent we can apply it to complex frameworks, like e.g., the weak probabilistic anonymity of [14] and the NRL pump itself. We also aim to investigate the applicability of our approach to cryptographic protocols and, in particular, its relationship with the reactive simulatability model of [9,10].

Finally, we plan to extend the notion of history dependent adversary so as to include timing issues; this would allow us to deal with another important kind of covert channels, namely the so-called covert timing channels, in which one process signals information to another process by modulating its own use of system resources (e.g., CPU time) in such a way that this manipulation affects the real response time observed by the second process.

References

1. Aldini, A.: Classification of security properties in a linda-like process algebra. Sci. Comput. Program. **63**, 16–38 (2006)
2. Aldini, A., Bernardo, M.: An integrated view of security analysis and performance evaluation: Trading qos with covert channel bandwidth. In: SAFECOMP'04: Proceedings of the 23rd International Conference on Computer Safety, Reliability and Security, Lecture Notes in Computer Science, vol. 3219, pp. 283–296. Springer (2004)
3. Aldini, A., Bravetti, M., Di Pierro, A., Gorrieri, R., Hankin, C., Wiklicky, H.: Two formal approaches for approximating noninterference properties. In: Focardi and Gorrieri [18], pp. 1–43
4. Aldini, A., Bravetti, M., Gorrieri, R.: A process algebraic approach for the analysis of probabilistic non-inetrference. J. Comput. Security **12**(1), 191–245 (2004)
5. Aldini, A., Di Pierro, A.: A quantitative approach to noninterference for probabilistic systems. In: Bravetti M., Gorrieri G. (eds.) Proceedings of the MEFISTO Project 2003, Formal Methods for Security and Time, Electronic Notes in Theoretical Computer Science, vol. 99, pp. 155–182. Elsevier (2004)
6. Aldini, A., Di Pierro, A.: On quantitative analysis of probabilistic protocols. In: Cerone, A., Pierro, A.D. (eds.) QAPL 2004, 2nd Workshop on Quantitative Aspects of Programming Languages, Electronic Notes in Theoretical Computer Science, vol. 112, pp. 131–148. Elsevier (2005)
7. Aldini, A., Di Pierro, A.: Noninterference and the most powerful probabilistic adversary. In: WITS 2006, 6th International Workshop on Issues in the Theory of Security (2006)
8. Aldini, A., Gorrieri, R.: Security analysis of a probabilistic non-repudiation protocol. In: PAPM-PROBMIV 2002, 2nd Joint International Workshop on Process Algebra and Performance Modelling, Probabilistic Methods in Verification, Lecture Notes in Computer Science, vol. 2399, pp. 17–36. Springer (2002)
9. Backes, M.: Quantifying probabilistic information flow in computational reactive systems. In: De Capitani di Vimercati, S., Syverson, P.F., Gollmann, D. (eds.) ESORICS '05: Proceedings of the 10th European Symposium on Research in Computer Security, Lecture Notes in Computer Science, vol. 3679, pp. 336–354. Springer (2005)
10. Backes, M., Pfitzmann, B., Waidner, M.: A composable yptographic library with nested operations. In: CCS'03: Proceedings of the 10th ACM conference on Computer and Communications Security, pp. 220–230. ACM Press, New York, (2003). doi:10.1145/948109.948140
11. Baier, C., Hermanns, H.: Weak bisimulation for fully probabilistic processes. In: Proceedings of the 9th International Conference

on Computer Aided Verification, Lecture Notes in Computer Science, vol. 1254, pp. 119–130. Springer (1997)
12. Bravetti, M., Aldini, A.: Disete time generative-reactive probabilistic processes with different advancing speeds. Theor. Comput. Sci. **290**, 355–406 (2003)
13. Canetti, R., Cheung, L., Kaynar, D., Liskov, M., Lynch, N., Pereira, O., Segala, R.: Using probabilistic I/O automata to improve the analysis of cryptographic protocols. ERCIM News **63**, 40–41 (2005)
14. Deng, Y., Palamidessi, C., Pang, J.: Weak probabilistic anonymity. In: SecCo 2005, 2nd International Workshop on Security Issues in Coordination Models, Languages, and Systems, Electronic Notes in Theoretical Computer Science (2005)
15. Di Pierro, A., Hankin, C., Wiklicky, H.: Approximate noninterference. J. Comput. Security **12**(1), 37–81 (2004)
16. Di Pierro, A., Hankin, C., Wiklicky, H.: Measuring the confinement of probabilistic systems. Theoretical Computer Science **340**(1), 3–56 (2005)
17. Focardi, R., Gorrieri, R.: A classification of security properties. J. Comput. Security **3**, 5–33 (1995)
18. Focardi, R., Gorrieri, R. (eds.): Lecture Notes in Computer Science, vol. 2946. Springer (2004)
19. Focardi, R., Gorrieri, R., Martinelli, F.: Classification of security properties (part ii: Network security). In: Focardi and Gorrieri [18], pp. 139–185
20. Goguen, J., Meseguer, J.: Security policy and security models. In: IEEE Symp. on Security and Privacy (SSP'82), pp. 11–20 (1982)
21. Guttman, J., Nadel, M.E.: What needs securing. In: 1st IEEE Computer Security Foundations Workshop (CSFW-1 1988), pp. 34–57. New Hampshire, USA (1988)
22. Howard, R.: Dynamic Probabilistic Systems. Wiley, New York (1971)
23. Wittbold, J.T., Johnson, D.J.: Information flow in nondeterministic systems. In: Proceedings of the IEEE Symposium on Security and Privacy (SSP'90), pp. 144–161. IEEE Computer Society (1990)
24. Kleinrock, L.: Queueing Systems. Wiley, New York (1975)
25. Milner, R.: Communication and Concurrency. Prentice Hall, New Jersey (1989)
26. Markowitch, O., Roggeman, Y.R.: Probabilistic non-repudiation without trusted third party. In: Conference on Security in Communication Networks (SCN'99) (1999)
27. Erdös, P., Graham, R.G.: Old and new problems and results in combinatorial number theory. Monographies de L'Enseignement Mathématique 28 (1980)
28. Ramanathan, A., Mitchell, J.C., Scedrov, A., Teague, V.: Probabilistic bisimulation and equivalence for security analysis of network protocols. In: FoSSaCS 2004, Foundations of Software Science and Computation Structures, Lecture Notes in Computer Science, vol. 2987, pp. 468–483. Springer (2004)
29. Roscoe, A.: Csp and determinism in security modelling. In: Proceedings of the IEEE Symposium on Security and Privacy (SSP'95), pp. 114–127. IEEE Computer Society (1995)
30. Ryan, P., Schneider, S.: Process algebra and non-interference. J. Comput. Security **9**(1/2), 75–103 (2001). Special Issue on CSFW-12
31. Sabelfeld, A., Sands, D.: Probabilistic noninterference for multi-threaded programs. In: Proceedings of the 13th IEEE Computer Security Foundations Workshop, pp. 200–214 (2000)
32. Sanders, W.H., Meyer, J.F.: A unified approach for specifying measures of performance, dependability, and performability. Dependable Comput. Fault-Tolerant Syst. Dependable Comput. Critical Appl. **4**, 215–237 (1991)
33. Segala, R.: Modeling and verification of randomized distributed real-time systems. Ph.D. thesis, MIT, Boston (MA) (1995)
34. Segala, R., Lynch, N.: Probabilistic simulations for probabilistic processes. Nordic J. Comput. **2**, 250–273 (1995)
35. Smith, G.: Probabilistic noninterference through weak probabilistic bisimulation. In: Proceedings of the 16th IEEE Computer Security Foundations Workshop (CSFW-16 2003), pp. 3–13. IEEE Computer Society, Pacific Grove, (2003)
36. van Glabbeek, R., Smolka, S., Steffen, B.: Reactive, generative and stratified models of probabilistic processes. Inf. Comput. **121**, 59–80 (1995)
37. Volpano, D., Smith, G.: Probabilistic noninterference in a concurrent language. In: Proceedings of the 11th IEEE Computer Security Foundations Workshop (CSFW '98), pp. 34–43. IEEE, Washington (1998)
38. Wu, S., Smolka, S., Stark, E.: Composition and behaviors of probabilistic i/o automata. Theor. Comput. Sci. **176**, 1–38 (1997)
39. Zhu, Y., Bettati, R.: Anonymity vs. information leakage in anonymity systems. In: ICDCS 2005, 25th IEEE International Conference on Distributed Computing Systems, pp. 514–524. IEEE Computer Society (2005)

Author biographies

Alessandro Aldini is an Assistant Professor in Computer Science at the Information Science and Technology Institute of the University of Urbino, Italy. He received the Laurea (with honors) and the Ph.D. degrees in Computer Science from the University of Bologna, in 1998 and 2002, respectively. His current research interests include theory of concurrency, formal description techniques and tools for concurrent and distributed computing systems, security, and performance evaluation.

Alessandra Di Pierro is a lecturer in Computer Science at the University of Verona (Italy). Her research interests include the development of probabilistic semantics-based techniques for the static analysis of programs and computer systems. She has recently been investigating the role of probabilistic semantics in capturing quantitative notions of security, and is co-organizer of a series of annual workshops on this subject. She is co-author of over 50 publications in international journals and conference proceedings.