Noninterference and the Most Powerful Probabilistic Adversary

Alessandro Aldini[1]* and Alessandra Di Pierro[2]

[1] Istituto STI, University of Urbino "Carlo Bo", Italy
[2] Dipartimento di Informatica, University of Pisa, Italy

Abstract. Probabilistic noninterference extends the classical possibilistic notion introduced by Goguen and Meseguer in order to capture the information leakage caused by adversaries that set up probabilistic covert channels. In this setting we investigate how to evaluate the observational power of an adversary to the purpose of establishing the maximal security degree of a given system. We introduce three classes of probabilistic adversaries, which represent the different observational power of an adversary, and then we establish properties for each such classes which state the complexity of effectively computing the most powerful adversary.

1 Introduction

Noninterference is widely studied in the security community as a property which formally specifies the absence of illegal information flow. In the recent literature various probabilistic variants of this notion have been proposed which allow for the specification and analysis of probabilistic and approximate security properties (see, e.g., [17, 15, 11, 16, 2, 9]). In particular, the approximate approach aims at estimating the degree to which a system can be considered to be secure [10]. This depends ultimately on the power of an *adversary* entity whose actual specification depends on the particular application or case study. Independently of its specification, the most powerful adversary effectively represents the maximal amount of information which an insecure system may leak; it is therefore important to be able to determine and evaluate such an adversary. In [4, 5] the effectiveness of the adversary strategy is evaluated in the process algebraic setting of [2] by measuring the maximal distance between two non-bisimilar (thus distinguishable) processes, the distance being defined in terms of transition probabilities. This approach has been applied in [3] for the analysis of a probabilistic non-repudiation protocol leading to a quantitative estimation of the probability of a security violation. As observed already in that paper, the cost of such an estimation grows factorially with the number of the states of the analyzed system. Thus, it is important to single out conditions under which this cost can be reduced and the calculation can be done effectively.

By following a noninterference-based approach inspired by [2], in this paper we take up the problem mentioned above by studying three classes of adversaries.

* This work has been supported by the IST project SENSORIA funded by EU.

For each class we give a formal definition of the interference of an adversary in that class with a given system and we show how the most powerful adversary can be evaluated, which determines the maximum information leakage. An important result of our study is that for one class that we call *history-dependent* adversaries, the most powerful adversary can be found by checking a finite number of adversary strategies.

2 Setting the Context

The system model we consider is based on a variant of probabilistic labeled transition systems, called generative-reactive transition systems (GRTS) [7]. In a GRTS, each transition is labeled with an action and a probability.

Actions model output and input events which 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. Then the set of actions is defined as $Act = \{a_* \,|\, a \in AType\} \cup \{a \,|\, a \in AType \cup \{\tau\}\}$, where a_* denotes an input action of type a and a denotes either an output action of type a or an action τ. Act is ranged over by $\pi, \pi', \ldots$.

The execution of the output actions is governed by a generative model of probabilities [8]. That means the system autonomously decides, on the basis of a probability distribution, which output action will be executed. The same holds for the unobservable actions, because an action τ is an event that is autonomously executed by the system. On the other hand, we assume the input actions to follow a reactive model of probabilities [8]. That means the choice of the input action type is non-deterministically left to the environment. Then, the choice of the particular input action of the chosen type, say a, is performed by the system on the basis of a probability distribution associated with the input actions of type a.

Therefore, GRTSs express both probabilistic behaviors guided by probability distributions governed by the system and nondeterministic behaviors due to the possible interactions with the environment. Technically speaking, transitions leaving a state are grouped into several bundles. We have a single generative bundle composed of all the transitions labeled with an output/invisible action, and several reactive bundles, each one referring to a different action type a and composed of all the transitions labeled with a_*. A bundle of transitions expresses a probabilistic choice. The choice among bundles is performed nondeterministically.

Definition 1. *A Generative-Reactive Transition System is a triple* (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. $\forall s \in S, \forall a_* \in Act. \sum \{\!| \, p \,|\, \exists t \in S : (s, a_*, p, t) \in T \,|\!\} \in \{0, 1\}$
2. $\forall s \in S. \sum \{\!| \, p \,|\, \exists a \in Act, t \in S : (s, a, p, t) \in T \,|\!\} \in \{0, 1\}$

A rooted GRTS is a quadruple (S, Act, T, s_0)*, with* (S, Act, T) *a GRTS and* $s_0 \in S$ *the initial state.*

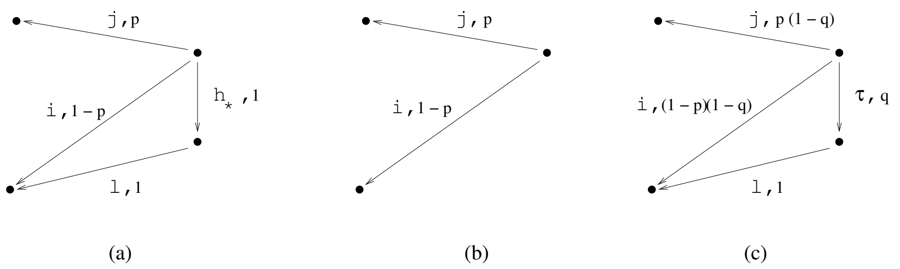

Fig. 1. Example of GRTS (a) and its low-level view for two different adversary strategies (b and c).

The two requirements of Def. 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, if there are any, 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(a) is made of the 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, as it depends on the environment behavior. Instead, within each bundle, the choice is probabilistic. In particular, if an output is executed, then the choice between the two possible output 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 enabled by the environment, then the input action h_* is executed with probability 1 (we usually omit the probability from the transition label when equal to 1).

3 GRTS and Noninterference

In this section we show how to express and analyze information flow in the GRTS model. As usual 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 see low-level actions only. Therefore, the interactions between the system and the high-level environment represent unobservable events from the viewpoint of Low. Based on a standard notion of noninterference [12], the high-level environment (which we simply refer to as the adversary) interferes with Low if the execution of the high-level actions has

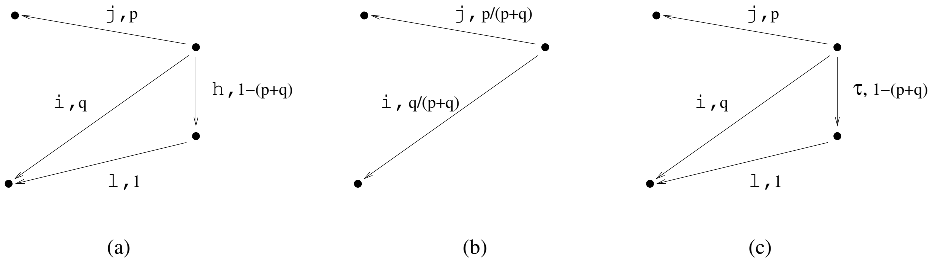

Fig. 2. Example of GRTS (a) and its low-level view for two different adversary strategies (b and c).

an observable impact on the execution of the low-level actions. We now explain through an example what Low can see depending on the interactions between the adversary and the system.

Example 2. Consider again the GRTS of Fig. 1(a), where either the system executes one of two possible low-level outputs, or the adversary interferes through a high-level input. If the adversary does not interact, the system behavior observable by Low results in the GRTS of Fig. 1(b). By contrast, the adversary may interact with the system thus solving somehow the nondeterministic choice between the output events and the input. In our setting the nondeterminism is probabilistically solved through a choice governed by a parameter q chosen by the adversary, thus resulting in the GRTS of Fig. 1(c). Such a GRTS models what Low can see. In particular, note that the transition labeled with an input action in Fig. 1(a) has been turned into a transition labeled with an unobservable action τ in Fig. 1(c). 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 particular adversary actually interacts with the system.

If the communication model is asynchronous, it is possible to abstract from the high-level outputs, which represent events that are not under the control 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, both high-level outputs and high-level inputs may have an impact on the low-level behavior of the system [13, 14], as shown in the following example.

Example 3. In Fig. 2(a) it is shown a variant of the GRTS of Fig. 1(a) which replaces the high-level input by a high-level output of the same type. Hence, the initial state is made of the generative bundle enabling three transitions, whose probabilities sum up to 1. If the adversary blocks the high-level action then only the two low-level outputs can be executed; thus their probabilities must be normalized in order to fulfil the requirements of Def. 1, as shown in Fig. 2(b). Instead, 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 intuitively shown in the examples above, the efficiency of the adversary interference on the low-level view of the system is revealed when Low compares the system behaviors under different adversary strategies.

In the following we define several such strategies by considering 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. In order to consider the case in which the system can accept low-level inputs, we should take into account each possible way in which Low can interact with such inputs.

3.1 Defining the Absence of the Adversary

In the simplest scenario, the adversary A does not interact with the system in any way. We now define $\mathcal{S}\backslash A$, which formally expresses what Low can see when the high-level interface of the system $\mathcal{S}$ is completely blocked because of the adversary absence.

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

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 Fig. 2(b). 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. [2]).

3.2 Families of Probabilistic Adversaries

A non-trivial adversary A interacting with a system $\mathcal{S}$ is a probabilistic scheduler which guides the execution of the inputs/outputs modeling 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 $\mathcal{S}/A$.

Simple Adversaries We start with considering adversaries that, for every high-level type, choose a priori the behavior of the corresponding inputs/outputs without changing it at run-time. Hence, in a sense, such schedulers are history-independent.

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 $A = (A_g, A_r)$ a simple adversary. Then $\mathcal{S}/A = (S', Act, T', s_0)$, 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 Def. 1 are preserved.

The algorithm above subsumes, at the GRTS level, the semantics of the probabilistic hiding operator of [2] when $A_g = H$ and for each $h \in H$ there exists $p_h \in]0, 1[$ such that $(h, p_h) \in A_r$. As an example, the GRTSs of Fig. 1(c) and of Fig. 2(c) express the interference of simple adversaries.

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. To overcome this constraint, we define the family of the interacting adversaries, 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.

Definition 5. *An interactive adversary A is defined by a pair (A_g, A_r), where $A_g : \mathcal{P}(H) \rightarrow \mathcal{P}(H)$ 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 $A_r : \mathcal{P}(H) \rightarrow \mathcal{P}(H \times]0, 1[)$ 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.*

As a notation, 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 $A = (A_g, A_r)$ an interactive adversary. Then $\mathcal{S}/A = (S', Act, T', s_0)$, 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 Def. 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.

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 say that such adversaries are history-dependent.

Definition 7. *A history-dependent adversary A is defined by a pair (A_g, A_r), where $A_g : Act^* \rightarrow \mathcal{P}(H)$ 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 $A_r : Act^* \rightarrow \mathcal{P}(H \times]0,1[)$ 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 Def. 5, the choice of the high-level inputs/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 $\mathcal{S}/A$ 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 $A = (A_g, A_r)$ a history-dependent adversary. Then $\mathcal{S}/A = (S', Act, T', (s_0, \varepsilon))$, where $S' \subseteq \mathcal{P}(S \times Act^*)$ and $T' \subseteq S' \times Act \times]0,1] \times S'$ are obtained as follows:*

```
$S' := \emptyset$; $T' := \emptyset$;
$H\_Adv((s_0, \varepsilon))$;
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 $H\_Adv((t, Tr.\pi))$;
```

By fixing a polynomial m^k as an upper bound to the length of the execution traces produced by $\mathcal{S}$ when interacting with the high-level environment, we can restrict ourselves to the set of polynomial-time adversaries. In this case, the algorithm that computes $\mathcal{S}/A$ is stopped after at most m^k execution steps and the resulting GRTS is a finite state transition system.

4 Noninterference as Indistinguishability

The interference of an adversary A with respect to a system $\mathcal{S}$ is evaluated by Low by comparing the observable behavior of $\mathcal{S}$ in the absence of A, given by $\mathcal{S}\backslash A$, and the observable behavior of $\mathcal{S}$ in the presence of A, given by $\mathcal{S}/A$. In this respect, the comparison between the two system views is performed on the basis of a behavioral equivalence that abstracts from unobservable actions, like e.g. the weak probabilistic bisimulation of [6].

As in [2] we consider here a probabilistic variant of such a relation, termed $\approx_{\mathrm{PB}}$, which replaces the classical weak transitions of the Milner's weak bisimulation by the probability of reaching classes of equivalent states. To this purpose we use a function *Prob* such that $\mathit{Prob}(s,\pi,C)$ denotes the aggregate probability of going from s to a state in the equivalence class C by executing an action π, and $\mathit{Prob}(s,\tau^*a,C)$ 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 $\mathit{Prob}(s,\tau^*a,C)$ is the minimal non-negative solution to the equation system:*

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

As shown in [2], 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, when $(s,s')\in R$, then for all C in the quotient set $S_{/R}$:*

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

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

On the basis of $\approx_{\mathrm{PB}}$, 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 $\mathcal{S}/A$. In this case we write $\mathcal{S}\backslash A\approx_{\mathrm{PB}}\mathcal{S}/A$. Since $\mathcal{S}\backslash A$ and $\mathcal{S}/A$ 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 Def. 9.

Example 4. Consider the GRTS of Fig. 1(a). 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. 1(b), modeling $\mathcal{S}\backslash A$, cannot execute the output action l, while in the GRTS of Fig. 1(c), modeling $\mathcal{S}/A$, such an action occurs with probability q, i.e. $\mathcal{S}\backslash A\not\approx_{\mathrm{PB}}\mathcal{S}/A$.

Whenever the outcome of the comparison based on $\approx_{\mathrm{PB}}$ is negative, it is important and useful to measure the difference between the observable behaviors of $\mathcal{S}\backslash A$ and $\mathcal{S}/A$. In this way we can obtain an estimate of the security of the system against the adversary A or equivalently of the power of A. This can be done by relaxing the equivalence relation above [3, 4]. In essence, given that $\mathcal{S}\backslash A$ and $\mathcal{S}/A$ are not weakly probabilistically bisimilar, we look for an equivalence relation R, including the pair of initial states of $\mathcal{S}\backslash A$ and $\mathcal{S}/A$ (which we call s_0 and s_0^A, respectively), that approximates $\mathcal{S}\backslash A \approx_{\mathrm{PB}} \mathcal{S}/A$. Given $\mathcal{R}_{\mathcal{S}}$ the set of equivalence relations that satisfy the condition above, since $R \in \mathcal{R}_{\mathcal{S}}$ is not a bisimulation, it holds that there exists at least a pair $(s, s') \in R$ such that:

$$Prob(s, \tau^* a, C) \neq Prob(s', \tau^* a, C)$$

for some $a \in Act$ and equivalence class C in $S_{/R}$.

The difference between these two weak transition probabilities expresses a quantitative estimation of the interference caused by A if $s \in \mathcal{S}\backslash A$ and $s' \in \mathcal{S}/A$. Indeed, if $(s, s') \in R$ and both s and s' belong to the same system view – either $\mathcal{S}\backslash A$ or $\mathcal{S}/A$ – then their distance does not actually reveal any interference, as it locally refers to a single system view. Moreover, when estimating the interference of the adversary, we should not only calculate the distance between s and s', but also take into account the probability of reaching such states. This is guaranteed by considering weak transition probabilities weighted by the probability of reaching s and s'. By virtue of these considerations, given $R \in \mathcal{R}_{\mathcal{S}}$ and $Prob(s_1, s_2)$ the aggregate probability of going from s_1 to s_2 via sequences of any number and type of actions, we compute the maximum interference of A for R as follows:

$$\begin{aligned}\delta_A^R = \max\{&|Prob(s, \tau^* a, C) - Prob(s', \tau^* a, C)| \cdot Prob(s_0, s) \cdot Prob(s_0^A, s') \mid \\ &s \in \mathcal{S}\backslash A, s' \in \mathcal{S}/A, (s, s') \in R, a \in Act, C \in S_{/R}\}\end{aligned}$$

In practice, we take the pair of states belonging to different system views and to the same equivalence class for which the weighted distance is maximum.

Given an enumeration of all the relations in $\mathcal{R}_{\mathcal{S}}$ and of the (possibly infinite) adversaries, we define a matrix where the columns range over the relations, the rows range over the adversaries, and the element with coordinates (i, j) is $\delta_{A_i}^{R_j}$. By so doing, the maximum element of a column j, call it ϵ_{R_j}, represents the most powerful adversary for the relation R_j. Formally, $\epsilon_{R_j} = \sup_A \delta_A^{R_j}$ is the maximum distance between the two system views that any adversary may cause when the considered relation is R_j. Since we are interested in the relation that is the closest approximation of $\approx_{\mathrm{PB}}$, the maximum adversary interference is given by $\inf_{R \in \mathcal{R}_{\mathcal{S}}} \epsilon_R$. Thus, the corresponding adversary is the most powerful adversary that maximizes the probability of revealing its presence to Low for the closest approximation of the weak probabilistic bisimulation.

Example 5. Consider the examples of Fig. 1 and Fig. 2 and the family of simple adversaries.

In the case of Fig. 2(a), there exist two possible adversaries. The first one blocks the output action h, but in such a case $\mathcal{S}\backslash A \approx_{\mathrm{PB}} \mathcal{S}/A$. The second one enables the output action h, i.e. Low observes the interference with probability $1-(p+q)$.

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 q. However, it can be proved that the most powerful simple adversary A assigns to q the limiting value 1. In fact, for each $R \in \mathcal{R}_{\mathcal{S}}$, it holds that $\delta_A^R = 1$, from which the result follows.

4.1 Evaluating the Most Powerful Adversary

Independently of the class under consideration, the main goal of the adversary is to maximize 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 seen, the choice between a high-level input and any other event can be solved by the 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 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.

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 modeled 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, we now show through some examples that it is not possible to restrict a priori the set of probability values in order to determine the most powerful simple adversary.

Example 6. Consider the GRTS of Fig. 3(a), where the adversary controls the probability distribution of the inputs of type h and k and can reveal its presence to Low only if the output action i is executed. Note that each simple adversary described by the limiting probability values 0 and 1 prevents the system from executing such an action. 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. Given p_h and p_k the probabilities that the adversary assigns to the reactive bundles of type h and k, respectively, it can be shown that the maximum interference for the closest approximation of $\approx_{\mathrm{PB}}$ is given by function $(1-p_h) \cdot p_k \cdot p_h \cdot (1-p_k)$, which intuitively represents the probability of executing the distinguishing action. The maximum value of this function is $\frac{1}{16}$ obtained when $p_h = p_k = \frac{1}{2}$.

As another example, consider the GRTS depicted in Fig. 3(b). There, the execution of the output action j, which is the unique action that allows Low to distinguish the behavior of the adversary, depends on which states enable the

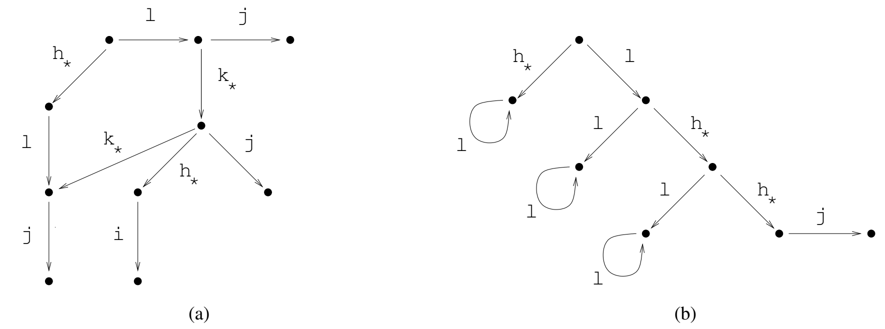

Fig. 3. Examples of GRTS.

input of type h, which is an event controlled by the adversary. Let p_h be the probability associated with the input of type h by a simple adversary. By analyzing all possible relations, it turns out that the maximum adversary interference is described by function $(1 - p_h) \cdot p_h \cdot p_h$, which is maximized by taking $p_h = \frac{2}{3}$, for which the probability of observing the interference is $\frac{4}{27}$.

In general, in order to determine the most powerful simple adversary, for each $R \in \mathcal{R}_\mathcal{S}$ we have to consider each pair of states belonging to different views and to the same equivalence class and, for every $a \in Act$ and $C \in S_{/R}$, we must solve a constraint programming problem with as many variables as the number of probability values that govern the probability distribution of the high-level input actions. The complexity of finding the most powerful adversary is therefore hyper-exponential.

Interactive adversaries The following examples show that in general the class of interactive adversaries suffers from the same problems elucidated in the case of simple adversaries.

Example 7. Consider the GRTS of Fig. 3(a). In order to maximize the execution probability of the output action i, an interactive adversary A can work as follows. First, A 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 output l. Then, A 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.

Despite of the simple example above, unfortunately we cannot restrict ourselves to considering the limiting probability values in order to determine the most powerful interactive adversary.

Example 8. Consider again the GRTS of Fig. 3(b). 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.

History-dependent Adversaries Differently from the previous cases, the most powerful polynomial-time history-dependent adversary 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 be the most powerful polynomial-time history-dependent adversary for $\mathcal{S}$ such that $\mathcal{S}\backslash A \not\approx_{\mathrm{PB}} \mathcal{S}/A$.*

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.

Example 9. Consider again the example of Fig. 3(a). 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 such a way, the distinguishing state is reached with probability tending to 1. Similarly, in the case of Fig. 3(b), 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; this exponentially depends on the number of different high-level actions that the system enables.

Thm. 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 10. Consider the GRTS of Fig. 4(a), 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 $\frac{1}{2}$. 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 Thm. 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 $A_r(\tau) = \{(k, 1)\}$.

We conclude with an example that is a simple abstraction of the case study analyzed in [3], i.e. a probabilistic non-repudiation protocol.

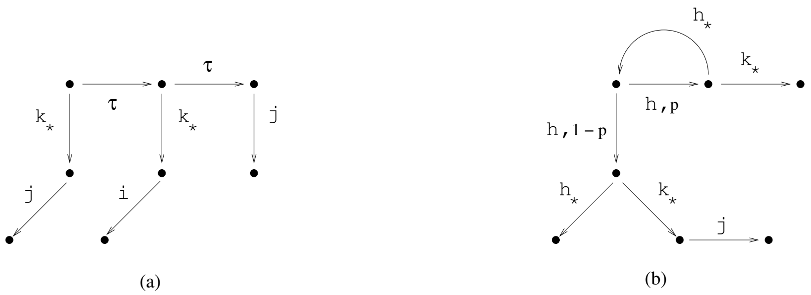

Fig. 4. Examples of GRTS.

Example 11. Consider the GRTS of Fig. 4(b). In the absence of high-level interactions Low cannot observe anything. Instead, in the presence of adversary interferences an output of type j may occur. As mathematically shown in [3], the most powerful adversary disables the input h_* and enables the output h and the input k_*. This behavior equates the strategy of a history-dependent adversary A such that $A_g(\varepsilon) = \{h\}$, $A_r(\varepsilon) = \emptyset$, and $A_g(h) = \emptyset$, $A_r(h) = \{(k, 1)\}$. A is easily found by applying Thm. 1, while in [3] the same result is obtained by analyzing 52 different relations each of which requires the solution of a constraint programming problem for each pair of considered states.

5 Conclusions and Future Work

We have studied three classes of adversaries in the context of a GRTS-based definition of noninterference. Adversaries possess a different observational power depending on the class they belong to. For example, while simple adversaries can only interfere on the basis of a fixed strategy, an interactive one is able to change its strategy depending on the current state of the system, and a history-dependent adversary can even remember the history of events which has determined the current state.

We have established the properties of each class at the basis of the evaluation of the most powerful of all adversaries in that class. It turns out that only for the class of history-dependent adversaries the most powerful adversary can be determined by checking a finite number of possible strategies.

As a future work we intend to examine the relation between the families of adversaries considered in this paper and the expressive power of probabilistic noninterference based properties. While it is quite clear the relation between the probabilistic extension of noninterference [2] and the class of simple adversaries, as well as the relation between probabilistic nondeducibility on compositions [2] and the class of interactive adversaries, the history-dependent adversaries seem to be related to a new probabilistic property that has not been considered in the literature.

Under certain assumptions, we have shown how to efficiently determine the maximum probability of revealing an information flow for an insecure system. If the system under the interference of the most powerful adversary can be made into a Markov chain with stationary distribution, the next immediate step consists of determining the maximum rate of the information flow or, equivalently, the bandwidth of the related covert channel, as shown in [1].

References

1. A. Aldini, M. Bernardo. *"An Integrated View of Security Analysis and Performance Evaluation: Trading QoS with Covert Channel Bandwidth"*, Proc. Int. Conf. on Computer Safety, Reliability and Security, LNCS 3219:283–296, 2004.
2. A. Aldini, M. Bravetti, R. Gorrieri, *"A Process-algebraic Approach for the Analysis of Probabilistic Non-interference"*, International Journal of Computer Security 12(2):191–245, 2004.
3. A. Aldini, A. Di Pierro, *"On Quantitative Analysis of Probabilistic Protocols"*, Proc. of Quantitative Aspects of Programming Languages, ENTCS 112:131–148, 2004.
4. A. Aldini, A. Di Pierro, *"A Quantitative Approach to Noninterference for Probabilistic Systems"*, Selected Papers from MEFISTO project *"Metodi formali per la sicurezza e il tempo"*, ENTCS 99:155–182, 2004.
5. A. Aldini, M. Bravetti, A. Di Pierro, R. Gorrieri, C. Hankin, H. Wiklicky, *"Two Formal Approaches for Approximating Noninterference Properties"*, *Foundations of Security Analysis and Design II – Tutorial Lectures*, LNCS 2946:1–43, 2004.
6. C. Baier, H. Hermanns, *"Weak Bisimulation for Fully Probabilistic Processes"*, Proc. Int. Conf. on Computer Aided Verification, LNCS 1254:119–130, 1997.
7. M. Bravetti, A. Aldini, *"Discrete Time Generative-Reactive Probabilistic Processes with Different Advancing Speeds"*, Theoretical Computer Science 290(1):355–406, 2003.
8. R.J. van Glabbeek, S.A. Smolka, B. Steffen, *"Reactive, Generative and Stratified Models of Probabilistic Processes"*, Information and Computation 121:59-80, 1995.
9. A. Di Pierro, C. Hankin, H. Wiklicky, *"Approximate Non-Interference"*, Journal of Computer Security 12(1):37–81, 2004.
10. A. Di Pierro, C. Hankin, H. Wiklicky, *"Measuring the Confinement of Probabilistic Systems"*, Theoretical Computer Science 340(1):3–56, 2005.
11. A. Di Pierro, C. Hankin, and H. Wiklicky, *"Probabilistic Confinement in a Declarative Framework"*, Declarative Programming, Selected Papers from AGP2000, ENTCS 48:1–23, 2001.
12. J. A. Goguen, J. Meseguer, *"Security Policy and Security Models"*, Proc. IEEE Symp. on Security and Privacy, pp. 11–20, 1982.
13. J. Guttman, M. Nadel, *"What Needs Securing?"*, Proc. Computer Security Foundation Workshop, pp. 34–57, 1988.
14. A. W. Roscoe, *"CSP and Determinism in Security Modelling"*, Proc. IEEE Symp. on Security and Privacy, pp. 114–127, 1995.
15. A. Sabelfeld, D. Sands, *"Probabilistic Noninterference for Multi-threaded Programs"*, Proc. IEEE Computer Security Foundations Workshop, pp. 200–214, 2000.
16. G. Smith, *"Probabilistic Noninterference through Weak Probabilistic Bisimulation"*, Proc. IEEE Computer Security Foundations Workshop, pp. 3–13, 2003.
17. D. Volpano, G. Smith, *"Probabilistic Noninterference in a Concurrent Language"*, Proc. IEEE Computer Security Foundations Workshop, pp. 34–43, 1998.