On Quantitative Analysis of Probabilistic Protocols

Alessandro Aldini [1]

Istituto STI, Università Carlo Bo, Urbino, Italy

Alessandra Di Pierro [2]

Dipartimento di Informatica, Università di Pisa, Italy

Abstract

We advocate the use of approximate noninterference for the security analysis of probabilistic protocols. Our approach relies on a formalisation of the protocol in the setting of a probabilistic process algebra and a notion of process similarity based on weak probabilistic bisimulation. We illustrate this approach by presenting the analysis of a probabilistic nonrepudiation protocol which allows us to quantitatively estimate its fairness degree.

Key words: Approximate Noninterference, Case Study, Process Equivalence, Probabilistic Process Algebra

1 Introduction

Security services such as authentication, confidentiality, non-repudiation, etc. are nowadays crucial to many applications given the growing importance of open networks. Thus the study of security protocols that guarantee such services is equally crucial to systems developers and has recently gained a primary importance in the research activities in computer science. The recent literature has shown that the consideration of probabilistic elements in such a study is essential for a more realistic formalisation and analysis of the security problem. Most notably, various approaches [16,17,9,4,13] have been proposed which rely on probabilistic variations of noninterference [12].

In this paper we investigate possible applications of the noninterference model of security to the analysis of probabilistic protocols. In particular, we show how an *approximate* formulation of a probabilistic noninterference

[1] Email: aldini@sti.uniurb.it

[2] Email: dipierro@di.unipi.it

This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

property [3,4] can be used to conduct a *quantitative* analysis of protocols which aim at ensuring that property. To this aim we follow an approach introduced in [1] which is based on the approximate noninterference model of [8]. The basic idea behind such an approach is to express the security requirement of the application at hand as an equivalence problem between two appropriately defined systems (see, e.g., [10,4]). Then the analysis consists essentially of an equivalence check and an evaluation of the similarity degree between the two systems in case of non-equivalence.

Formally, the approach we propose is defined in the framework of a probabilistic calculus [4] and a weak probabilistic bisimulation equivalence semantics [5]. In this setting, the noninterference based security property we consider is a probabilistic extension of the Strong Nondeterministic Noninterference property [10], which we simply call Probabilistic Noninterference [3] (*PNI*). Intuitively, *PNI* compares, from an external observer standpoint, the view of the system in the absence of adversary interferences and the view of the system when the adversary interacts with the system. If the equivalence check is satisfied, then an external observer that sees the result of the protocol run cannot infer whether or not the behaviour of the system has been altered by the adversary. Otherwise, we estimate the effectiveness of the adversary strategy (i.e., we provide a quantitative estimation of its capability of revealing its presence to the external observer) by means of a technique introduced in [1] and corresponding to measuring the maximal "distance" between two non-equivalent processes. This distance is defined in terms of transition probabilities and gives an estimate of the behavioural difference of the two processes. The meaning we attribute to the resulting measure is related to the number of tests an external observer needs to perform in order to infer whether or not the behaviour of the system has been altered by the adversary. A justification for this statistical interpretation was given in [3] following the model introduced in [8].

As a case study, we apply this technique to the analysis of a probabilistic nonrepudiation protocol [14], which has been previously modeled and analysed in [2] through the same process algebraic setting considered in this paper. The novel analysis methodology we propose allows us to calculate a number which gives an estimate of the fairness of the protocol, i.e. of its security degree. We also suggest possible variations of the basic approach which may lead to finer analyses from which a more precise upper bound can be deduced for the security degree of the protocol. With respect to [2], we formally provide a measure of the effectiveness of the most powerful adversary that is able to violate the security property of interest.

In the following we formally introduce the process algebraic framework and the approximate noninterference approach to security (Section 2), by describing the syntax and the semantics of the probabilistic calculus, the weak probabilistic bisimulation equivalence, the *PNI* property rephrased in such a setting, and an approximate version of *PNI*. Then, we illustrate the case

study (Section 3), by describing the probabilistic nonrepudiation protocol [14] together with the process algebraic model of an implementation of such a protocol, on which we quantitatively estimate the security property of interest. We finally outline directions for further work in Section 4.

2 Noninterference and Probabilistic Adversary

We base our approach on a notion of approximate noninterference [8] in the setting of a calculus used in [4] to define a probabilistic extension of nondeterministic noninterference [10]. In this section, we briefly describe such a calculus, and present definition of probabilistic noninterference parameterised by a class $\mathcal{A}$ of probabilistic adversaries [3,1] as well as a quantitative approach to the evaluation of the maximal interference caused by $\mathcal{A}$ [1].

2.1 Probabilistic Process Algebra

The probabilistic calculus we consider derives from a simple nondeterministic process algebra where actions are syntactically divided into input actions and output actions. Formally, for each visible action type a, we distinguish the output action a and the input action a_*. Process terms synchronously communicate with the environment through their inputs and outputs, and perform internal computations through unobservable actions, termed τ actions.

Probabilities are introduced by adding probabilistic information to the algebraic operators. The probabilistic model we adopt [7] is a mixture of the generative and reactive approaches of [11]. In particular, we assume the internal and output actions behaving as *generative* actions, i.e. the system autonomously decides, on the basis of a probability distribution, which internal/output action will be executed and how to behave after such an event. On the other hand, we assume the input actions behaving as *reactive* actions, i.e. the system reacts internally to the action type, say a, chosen by the environment. Then, the choice of the reactive action of type a to be executed is performed on the basis of a probability distribution associated with the reactive actions of type a the system can perform. In practice, we see the input actions as underspecified, since their execution is guided by the environment behaviour. The mixed generative-reactive model allows for a representation of both probabilistic behaviours guided by probability distributions decided by the system and nondeterministic behaviours due to the possible interactions of the system with the environment (see, e.g., [6] for more details).

The syntax of the probabilistic process calculus is as follows:

$$P ::= \underline{0} \mid \pi.P \mid P +^p P \mid P \parallel^p_S P \mid P\backslash L \mid P/^p_a \mid A.$$

We use $\underline{0}$ to represent the terminated process (we usually omit it). Action π is drawn from set Act and can be an internal action τ, an output action a, or an input action a_*, where a belongs to the set of visible action types $AType$. $\pi.P$ performs the action π with probability 1 and then behaves like P.

The alternative choice operator $P +^p Q$, with $p \in (0,1)$, performs a mixed probabilistic/nondeterministic choice among the actions of P and Q. More precisely, $P +^p Q$ executes a generative (reactive of type a) action of P with probability p and a generative (reactive of type a) action of Q with probability $1-p$. If one process P or Q cannot execute generative (reactive of type a) actions, $P +^p Q$ chooses a generative (reactive of type a) action of the other process with probability 1. The choice among generative and reactive actions and among reactive actions of different types is purely nondeterministic. Hence, the parameter that probabilistically guides the choices comes into play if and only if a probabilistic choice is really to be performed.

The parallel composition operator $P \|^p_S Q$, with $p \in (0,1)$ and $S \subseteq \mathit{AType}$, asynchronously performs all the actions of P and Q that do not belong to the synchronisation set S. Instead, all the actions belonging to S are constrained to synchronise. In particular, a synchronisation between two actions can occur if either they are both input actions of the same type a (and the result is an input action of type a), or one of them is an output action of type a and the other one is an input action of type a (and the result is an output action of type a). The probabilistic choice mechanism among the actions of P and Q is the same as that described for the choice operator. We just point out that the execution of some actions of P may be prevented in $P \|^p_S Q$ because of the synchronisation rule. Such a restriction imposes a careful calculation of the probability distribution of the actions of $P \|^p_S Q$ that follow the generative model of probabilities [11]. In order to obtain a probability distribution, we normalise the probabilities of executing the generative actions of P executable by $P \|^p_S Q$ (similarly for Q). Such an approach is commonly applied when restricting actions in the generative model of probabilities [11].

The restriction operator $P\backslash L$ prevents the execution of the actions of type in $L \subseteq \mathit{AType}$. We omit further discussions about it as $P\backslash L$ can be expressed in terms of the parallel operator. Indeed, it is easy to see that $P\backslash L$ corresponds to process $P \|^p_L \underline{0}$, for any choice of parameter p.

The hiding operator $P/^p_a$ turns actions of type a into actions τ. In particular, we recall that the reactive actions a_* of P are governed by their own probability distribution, while the actions τ of P are governed by the probability distribution associated with the generative actions enabled by P. Therefore, when turning actions a_* into actions τ we must pay attention to the computation of the probability distribution of the generative actions enabled by $P/^p_a$. To this aim, we use parameter p to express the probability that generative actions τ obtained by hiding reactive actions a_* of P are executed with respect to the generative actions previously enabled by P. In practice, when turning reactive actions a_* into actions τ, parameter p probabilistically resolves the nondeterminism among the reactive actions of type a and the generative actions. As an example, consider process $a_* +^q b$ (note that the choice is nondeterministic, i.e. q is not meaningful), and hide the action a_*. The semantics of process $(a_* +^q b)/^p_a$ is the probabilistic choice $\tau +^p b$, guided by parameter p,

between τ, obtained by hiding a_*, and b. Parameter p of the hiding operator is not used when hiding generative actions, because in such a case no nondeterminism must be resolved. Intuitively, we turn reactive actions into generative internal actions in order to obtain closed (fully generative) systems from open systems (i.e. systems enabling reactive choices). To this purpose, the nondeterministic choices due to the potential interactions with the environment are probabilistically resolved by employing parameter p of the hiding operator.

Constants A are used to specify recursive systems. In general, when defining a process term, we assume a set of constants defining equations of the form $A \stackrel{def}{=} P$ (with P a guarded term [15]) to be given.

In the rest of the paper, we denote by $\mathcal{G}$ the set of finite state, guarded, and closed terms [15], called processes, generated by the syntax above. Moreover, we assume $p = \frac{1}{2}$ in the case parameter p of a probabilistic operator is omitted.

Now, we briefly introduce the semantics of the calculus [4]. To this purpose, we introduce the following notation: $RAct$ and $GAct$ denote the sets of input actions and of output and internal actions, respectively; we use the abbreviations $P \stackrel{\pi}{\longrightarrow}$ to stand for $\exists p \in]0,1], P' : P \stackrel{\pi,p}{\longrightarrow} P'$, denoting that P can execute action π with probability p and then behave as P', and $P \stackrel{G}{\longrightarrow}$, with $G \subseteq GAct$, to stand for $\exists a \in G : P \stackrel{a}{\longrightarrow}$, that means P can execute a generative action $a \in G$.

The operational semantics of the probabilistic process algebra is given by the labeled transition system $(\mathcal{G}, Act, T)$, whose states are process terms and the transition relation T is the least multiset satisfying the operational rules reported in Table 1 and in Table 2. As far as the rules for $P +^p Q$ and $P \|^p_S Q$ 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 in the premises and by replacing p with $1 - p$ in the label of the derived transitions.

The semantics rules reflect the informal presentation of the syntax of the operators. Here, we describe in detail the restriction mechanism adopted by the parallel operator, which, as we have seen, uses a normalisation factor in order to obtain a probability distribution to be associated with the generative actions enabled by $P \|^p_S Q$. To this purpose, we employ the following notation: Set $G_{S,Q} = \{a \in AType \cup \{\tau\} \mid a \notin S \vee (a \in S \wedge Q \stackrel{a_*}{\longrightarrow})\}$ contains the action types not belonging to set S and the action types belonging to S for which an input action of Q can be performed. In practice, $G_{S,Q}$ determines which actions can be executed by a process in the context $\_ \|^p_S Q$. Function $\nu_P(G_{S,Q}) : \mathcal{P}(AType \cup \{\tau\}) \longrightarrow]0,1]$ computes the sum of the probabilities of the generative transitions of P with type in $G_{S,Q}$. Hence, the value $\nu_P(G_{S,Q})$ is used to normalise the probabilities of the generative transitions of P that are enabled by $P \|^p_S Q$.

To conclude the presentation of the semantics of the calculus, we introduce

Table 1
Operational semantics (part I)

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

$$\frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P +^p Q \xrightarrow{a_*,p\cdot q} P'} \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \not\xrightarrow{a_*}}{P +^p Q \xrightarrow{a_*,q} P'}$$

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{GAct}}{P +^p Q \xrightarrow{a,p\cdot q} P'} \qquad \frac{P \xrightarrow{a,q} P' \quad Q \not\xrightarrow{GAct}}{P +^p Q \xrightarrow{a,q} P'}$$

$$\frac{P \xrightarrow{a_*,q} P' \quad P \xrightarrow{GAct}}{P/^p_a \xrightarrow{\tau,p\cdot q} P'/^p_a} \qquad \frac{P \xrightarrow{a_*,q} P' \quad P \not\xrightarrow{GAct}}{P/^p_a \xrightarrow{\tau,q} P'/^p_a}$$

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

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

$$\frac{P \xrightarrow{b,q} P' \quad P \not\xrightarrow{a_*}}{P/^p_a \xrightarrow{b,q} P'/^p_a} \; a \neq b \qquad \frac{P \xrightarrow{a,q} P' \quad P \not\xrightarrow{a_*}}{P/^p_a \xrightarrow{\tau,q} P'/^p_a}$$

$$\frac{P \xrightarrow{\pi,q} P'}{A \xrightarrow{\pi,q} P'} \text{ if } A \stackrel{def}{=} P$$

a notion of process equivalence based on which we compare the observable behaviours of different systems. To this aim we need an equivalence relation that takes into account the observational power of an external observer, i.e. is able to abstract away from unobservable internal details. In particular, we consider a probabilistic variant of the weak bisimulation semantics (borrowed from [5], where fully probabilistic processes are considered). Such a relation, termed $\approx_{\rm PB}$, extends the weak bisimulation ($\approx_{\rm B}$) of [15] by replacing the classical weak transitions of $\approx_{\rm B}$ by the probability of reaching classes of equivalent states. More precisely, we use a function *Prob*, such that $Prob(P, \pi, C)$ denotes the aggregate probability of going from P to a term in the class (of equivalent terms) C by executing an action π, and $Prob(P, \tau^* a, C)$ expresses the aggregate probability of going from P to a term in the equivalence class C via sequences of the form $\tau^* a$ (if $a \neq \tau$) or τ^* (if $a = \tau$).

Table 2
Operational semantics (part II)

$$\frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P \parallel^p_S Q \xrightarrow{a_*,p\cdot q} P' \parallel^p_S Q} \; a \notin S \qquad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*} \not\longrightarrow}{P \parallel^p_S Q \xrightarrow{a_*,q} P' \parallel^p_S Q} \; a \notin S$$

$$\frac{P \xrightarrow{a_*,q} P' \; Q \xrightarrow{a_*,q'} Q'}{P \parallel^p_S Q \xrightarrow{a_*,q\cdot q'} P' \parallel^p_S Q'} \; a \in S$$

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

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{G_{S,P}} \not\longrightarrow}{P \parallel^p_S Q \xrightarrow{a,q/\nu_P(G_{S,Q})} P' \parallel^p_S Q} \; a \notin S$$

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

$$\frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{a_*,q'} Q' \quad Q \xrightarrow{G_{S,P}} \not\longrightarrow}{P \parallel^p_S Q \xrightarrow{a,q'\cdot q/\nu_P(G_{S,Q})} P' \parallel^p_S Q'} \; a \in S$$

Lemma 2.1 *The value of $Prob(P, \tau^* a, C)$ is the minimal non-negative solution to the equation system*

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

As shown in [4], the equation system above has a least solution. We now are ready to define the weak probabilistic bisimulation equivalence.

Definition 2.2 An equivalence relation $R \subseteq \mathcal{G} \times \mathcal{G}$ is a weak probabilistic bisimulation if and only if, whenever $(P, Q) \in R$, then for all $C \in \mathcal{G}/R$:

- $Prob(P, \tau^* a, C) = Prob(Q, \tau^* a, C)\ \forall a \in GAct$
- $Prob(P, a_*, C) = Prob(Q, a_*, C)\ \forall a_* \in RAct$.

Two terms $P, Q \in \mathcal{G}$ are weakly probabilistically bisimulation equivalent, denoted $P \approx_{\mathrm{PB}} Q$, if there exists a weak probabilistic bisimulation R including the pair (P, Q).

2.2 Approximate Probabilistic Noninterference

A high-level user (High, for short) interferes with a low-level user (Low, for short) if what High can do is reflected on what Low can observe [12]. High can perform high-level activities only and observe all the interactions between the system and the environment. Low can perform low-level activities only and is not allowed to directly observe the occurrence of high-level events. In our setting, what Low can see is not only the logical low-level behaviour of the system, but also the probability distribution of each low-level activity. Despite of the absence of a direct communication channel from High to Low, High may succeed in altering the low-level view of the system, thus passing information to Low, by interacting with the high-level interface of the system. In the following we describe a formalisation of noninterference, where High is considered to be an adversary that tries to maximise the information leakage from High to Low.

Roughly, the noninterference approach can be described as follows. First, we derive two models from the considered system, corresponding to two different low-level views of the system, and then we verify the $\approx_{\rm PB}$ based equivalence between such derived models. The choice of the models to be compared depends on the definition of the security property. Here, we consider a probabilistic extension [4] of the Strong Nondeterministic Noninterference property of [10], which we simply call Probabilistic Noninterference [3] (*PNI*). Such a property compares the low-level view of the system without high-level interferences and the low-level view of the system in the presence of high-level interactions. If such models turn out to be equivalent, then a low-level observer cannot deduce the behaviour of the high-level user by interacting with the low-level interface of the system.

Formally, we divide actions into high-level actions and low-level actions, denoted *High* and *Low*, respectively, depending on the nature of the activities they represent. *High* and *Low* are disjoint and form a covering of *AType*. Given a process P, we denote with $\bar{h}^P = h_1^P, \ldots, h_n^P$ the sequence (in alphabetic order) of types of the high-level actions that syntactically occur in the action prefix operators within P. Then, the application of the security check to P is as follows. The view of P without high-level operations is modeled by $P \backslash High$. The low-level view of P in the presence of high-level interactions is expressed by the family of processes $P/^{p_1}_{h_1^P} \ldots /^{p_n}_{h_n^P}$, $p_1, \ldots, p_n \in (0,1)$, where $\bar{p} = p_1, \ldots, p_n$ is the sequence of parameters modelling the probability distribution (chosen by High) of the hidden high-level input actions enabled by the system. We use the abbreviation $P/^{\bar{p}}_{\bar{h}^P}$ to stand for $P/^{p_1}_{h_1^P} \ldots /^{p_n}_{h_n^P}$. Finally, the *PNI* property can be formalised as follows.

Definition 2.3 $P \in PNI \Leftrightarrow P \backslash High \approx_{PB} P/^{p_1}_{h_1^P} \ldots /^{p_n}_{h_n^P} \; \forall p_1, \ldots, p_n \in (0,1)$.

Parameters $p_1, \ldots, p_n$ express the probabilistic adversary that interacts with the system and tries to maximise the information leakage from High to Low.

The universal quantification over all possible sequences $p_1, \ldots, p_n \in (0,1)$ means that the equivalence check must hold for an infinite number of adversaries. In particular, as also shown in [3], the class $\mathcal{A}$ of adversaries expressed by *PNI* contains active and memoryless high-level users. On the one hand, they are active as the probabilistic low-level behaviour of the system can be altered when the reactive high-level actions are hidden. On the other hand, they are memoryless as they cannot alter their strategy depending on the previous history. Indeed, the probability distribution of the hidden high-level inputs, modeled by parameters $p_1, \ldots, p_n$, is chosen a priori and does not change during the system execution. If the condition of Definition 2.3 holds, then the system does not leak any information from an adversary in $\mathcal{A}$ to Low.

We now show how to calculate a quantitative estimate of the maximal amount of information leakage caused by $\mathcal{A}$ in the case the equivalence check is not satisfied. In the following we restrict ourselves to systems that are fully specified from the viewpoint of Low. Hence, we assume that the only reactive actions enabled by the system are high-level actions [1].

The probability of observing an information flow from High to Low can be estimated by relaxing the behavioural equivalence relation expressed by $\approx_{\mathrm{PB}}$. As we have seen, an information leakage occurs in P whenever, for a given sequence $\bar{p}$ chosen by High, for each equivalence relation $R \subseteq \mathcal{G} \times \mathcal{G}$ including the pair $(P\backslash High, P/_{\bar{h}^P}^{\bar{p}})$, there exist $C \in \mathcal{G}/R$, $a \in GAct$, and a pair $(P', P'') \in R$, such that $Prob(P', \tau^* a, C) \neq Prob(P'', \tau^* a, C)$. The difference between these two probabilities can be used to give an estimate of the amount of information leakage. More precisely, for every equivalence relation R including the pair $(P\backslash High, P/_{\bar{h}^P}^{\bar{p}})$, we consider the pair of states (of a class in $\mathcal{G}/R$) where the weak transition probabilities are maximally different and calculate the difference. We can then define a measure of the security of P as the minimal of these differences over all equivalence relations.

Formally, we define the quantity $\delta_{\bar{p}}^R(P)$ (or simply $\delta_{\bar{p}}^R$ when it is clear from the context), which expresses the behavioural distance between $P\backslash High$ and $P/_{\bar{h}^P}^{\bar{p}}$ with respect to a relation $R \subseteq \mathcal{G} \times \mathcal{G}$ including the pair $(P\backslash High, P/_{\bar{h}^P}^{\bar{p}})$ and a sequence of parameters $\bar{p} = p_1, \ldots, p_n$ governing the interactions between the high-level input actions of P and High. By using this quantity we then define a measure $\varepsilon_{\bar{p}}$ for the security degree of a system P against the adversary modeled by $\bar{p}$.

Definition 2.4 Let P be a process, $R \subseteq \mathcal{G} \times \mathcal{G}$ an equivalence relation including the pair $(P\backslash High, P/_{\bar{h}^P}^{\bar{p}})$, and $\bar{p} = p_1, \ldots, p_n$ a sequence of parameters such that $p_i \in (0,1), 1 \leq i \leq n$. We define

$$\delta_{\bar{p}}^R = sup_{\substack{(P', P'') \in R, \\ a \in GAct,\ C \in \mathcal{G}/R}} \mid Prob(P', \tau^* a, C) - Prob(P'', \tau^* a, C) \mid$$

and then

$$\varepsilon_{\bar{p}} = inf_R\ \delta_{\bar{p}}^R.$$

Chosen a relation R, $\delta^R_{\bar{p}}$ expresses the maximal difference between the low-level view of the system without high-level actions and the one modelling the interactions of the system with the high-level user probabilistically modeled by sequence $\bar{p}$. Then, the relation that is the best approximation of a weak probabilistic bisimulation is obtained by computing the minimum ($\varepsilon_{\bar{p}}$) over all the possible $\delta^R_{\bar{p}}$. That means, $\varepsilon_{\bar{p}}$ expresses how similar are the two low-level views of the system to be compared [1]. Note that this quantity depends on parameters $p_1, \ldots, p_n$ forming the sequence $\bar{p}$, which models an adversary of the family $\mathcal{A}$ defined by PNI. The measure $\varepsilon_{\bar{p}}$ can also be interpreted as the *effectiveness* of such an adversary [8]. In fact, it determines *how easy* it is for a low-level user to obtain some information, in terms of the number of tests (system executions) Low needs to perform in order to distinguish with success the behaviours with and without the adversary interferences. The maximal $\varepsilon_{\bar{p}}$ obtained by varying $\bar{p}$, i.e. by changing the adversary strategy, determines the effectiveness of the most powerful adversary in $\mathcal{A}$. In the following, we will show that the problem of finding such an adversary corresponds to solving a (non-linear) optimisation problem with as many variables as the number of parameters contained in the sequence $\bar{p}$ [1].

3 A Case Study: Probabilistic Nonrepudiation

As an example of application of the methodology surveyed above, in this section we present a case study: a probabilistic nonrepudiation protocol [14]. Here, we show how the maximal information leakage, expressed in terms of the maximum probability of violating a fairness property during system execution, can be estimated.

3.1 An Overview of the Protocol

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: *nonrepudiation of origin* is intended to prevent the *originator* of a message from denying having sent the message, and *nonrepudiation of receipt* is intended to prevent the *recipient* of a message from denying having received the message. Especially in e-commerce, nonrepudiation is needed to protect a transaction against any attempt to repudiate either the payment for the service or the delivery of the service. Here, we consider a protocol that offers a nonrepudiation service, guaranteed with a certain probability, without resorting to a trusted third party [14]. Such a protocol 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 probabilistic 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.

In the following we suppose that the protocol is preceded by a secure authentication phase, during which the involved parties exchange their public keys of a public key cryptosystem. Moreover, we denote by $Sign_E(M)$ the message M encrypted with the private key of the entity E, by k a secret key chosen by O to encrypt a message M with a symmetric encryption algorithm, and by t a timestamp which each message is enriched with. Finally, $R \rightarrow O$: *Msg* expresses a message *Msg* sent by R and received by O.

We now describe an implementation of the protocol illustrated in [14]. The recipient R starts the protocol by sending a signed, timestamped request for a service to the originator O, which in turn sends the first signed, timestamped message containing M encrypted with k. Upon receiving the first message from O, R sends a related signed, timestamped acknowledgment message containing $\text{ack}_1 = (1, R, O, t)$:

$$\begin{array}{ll} 1.\ R \rightarrow O & : Sign_R(\text{request}, R, O, t) \\ 2.\ O \rightarrow R & : Sign_O(\{M\}_k, O, R, t) \\ 3.\ R \rightarrow O & : Sign_R(\text{ack}_1). \end{array}$$

Then, at each protocol step i, O probabilistically decides whether to continue the protocol (with probability $1 - p$), by sending a key k' different from k, or to terminate the protocol (with probability p), by sending the key k needed to obtain the plaintext M. On the other hand, for each received message, R transmits the related ack message $\text{ack}_i = (i, R, O, t)$:

$$\begin{array}{ll} 2i. & O \rightarrow R\ : [p]Sign_O(k, O, R, t)\ +\ [1-p]Sign_O(k', O, R, t) \\ 2i+1.\ R \rightarrow O & : Sign_R(\text{ack}_i). \end{array}$$

Since R does not know the result of the probabilistic choice, it cannot determine when the protocol will end and, as a consequence, when it will receive the final message. Upon the reception of the ack related to the last message containing k, O correctly terminates the protocol. Note that an *end of protocol* message sent by O is not mandatory. Indeed, after not receiving further messages, R is aware of the protocol state and is able to compute M.

As far as the security guarantees are concerned, each message conveys a timestamp, which is used to determine the freshness of the message and to protect the parties against replication attacks. The nonrepudiation of origin is guaranteed by the messages $Sign_O(\{M\}_k, O, R, t)$ and $Sign_O(k, O, R, t)$, and the nonrepudiation of receipt is given by the last message $Sign_R(\text{ack}_n)$. If the protocol terminates after the delivery of $Sign_R(\text{ack}_n)$, both parties obtain their expected information and the protocol is fair. If the protocol terminates before the transmission of $Sign_O(k, O, R, t)$, then neither O nor R obtain any valuable information, so that the fairness is preserved. However, at each step R could try to verify whether M can be obtained by employing the last key received from O and, once the correct key k is received, violate the fairness of the protocol by blocking the transmission of the last ack. Hence, key to success of the protocol is the immediacy in sending back the ack. Under such

a condition, if the transmission of an ack is delayed by R, then O can detect this unfair strategy and prematurely stop the protocol. To this aim, the choice of the encryption algorithm must be in such a way that the decryption of the ciphertext takes more time than the transmission of an ack. Hence, O decides a deadline for the reception of each ack, after which, if the ack is not received, the protocol is stopped. Finally, we observe that such a protocol is exposed to the attack of a malicious recipient that tries to randomly guess the number of protocol steps and block the final ack. In fact, as we will see, this is the kind of attack that we formally analyse in the next sections.

3.2 Modelling the Protocol

The protocol described in Section 3.1 guarantees nonrepudiation of origin with probability 1 and nonrepudiation of recipient with a probability less than 1. Hence, our goal is to estimate the probability of violating the nonrepudiation of recipient. In order to determine the effectiveness of the most powerful adversary strategy against the originator, we model and analyse the behaviour of the originator and we consider the recipient as a potential adversary.

In our model, we abstract from the cryptosystem used within the protocol and we simply describe the packet exchange between the two involved parties. 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 specification of the originator is as follows:

$$
\begin{aligned}
O &\stackrel{def}{=} \mathit{receive\_request}_*.\mathit{snd\_msg}.\mathit{receive\_ack}_*.O' \\
O' &\stackrel{def}{=} \mathit{snd\_msg}.O'' +^p \mathit{snd\_msg}.O''' \\
O'' &\stackrel{def}{=} \mathit{receive\_ack}_*.\underline{0} + \mathit{receive\_stop}_*.\mathit{unfair}.\underline{0} \\
O''' &\stackrel{def}{=} \mathit{receive\_ack}_*.O' + \mathit{receive\_stop}_*.\underline{0}
\end{aligned}
$$

At the first round of the protocol (term O), the originator is ready to accept an incoming request, send the first message containing $\{M\}_k$, and then receive the related ack message. Afterward, at the beginning of each new step (term O'), the originator probabilistically decides whether to send with probability p the last message containing k, thus reaching term O'', or to send with probability $1-p$ a garbage message, thus reaching term O'''. In term O''', the originator waits for the ack before starting another step. The expiration of the timeout is abstracted through the reception of a message of type *receive_stop*, that means the protocol is stopped. Note that in this case the protocol execution is fair, because the final message containing k has not been sent yet. In term O'', the originator waits for the last ack. Upon receiving such an ack, the protocol correctly terminates in a fair way. Otherwise, the protocol terminates in an unfair way from the viewpoint of the originator. This is signaled by executing the action of type *unfair*.

3.3 Measuring the Security of the Protocol

As far as the security analysis is concerned, we make the following assumptions. Since the recipient is a potential 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 Figure 1 we show the labelled transition system associated with the model of the originator and the two low-level models that, according to the *PNI* definition, must be compared through equivalence checking. In particular, the former low-level model is obtained by hiding all the high-level actions, i.e. $O/_{receive\_ack}/_{receive\_request}/^{q}_{receive\_stop}/_{snd\_msg}$. Note that the parameter of the hiding operator is not meaningful where we omitted it. Intuitively, by varying parameter q in the range $(0,1)$, we model all the possible recipients that execute the protocol and decide the probability distribution of each message sent to the originator. The latter low-level model is obtained by restricting the high-level actions, i.e. $O\backslash High$. Intuitively, such a model expresses the absence of the recipient, that means nothing (potentially insecure) happens. We point out that the class $\mathcal{A}$ of adversaries considered by *PNI* is adequate to reveal the security degree of the protocol. In particular, a history-dependent strategy followed by a malicious recipient would not be useful to gain an advantage as each protocol step does not depend on the previous ones.

If we limit ourselves to an exact verification of the security property, we obtain that *PNI* does not hold as the condition

$$O\backslash High \approx_{\mathrm{PB}} O/_{receive\_ack}/_{receive\_request}/^{q}_{receive\_stop}/_{snd\_msg} \ \forall q \in (0,1)$$

is not satisfied. The reason is that in the presence of a (potentially malicious) recipient the action *unfair* can be observed with a certain probability.

In order to quantitatively estimate the effectiveness of the most powerful adversary we resort to an approximate analysis. To this aim, we take the low-level views of the protocol considered by *PNI* (see Figure 1) and we calculate the difference between them for each possible relation R and for each value of q chosen by the recipient.

We start by considering relation $R_1 = \{\{1, 1', 2, 3, 5\}, \{4\}\}$, for which we obtain the following results: $|\, Prob(1, \tau^*, [4]) - Prob(1', \tau^*, [4]) \,| = p \cdot q \cdot \sum_{i=0}^{\infty}((1-p)\cdot(1-q))^i = \frac{p\cdot q}{1-(1-p)\cdot(1-q)} = \frac{p\cdot q}{p+q-p\cdot q}$,
$|\, Prob(2, \tau^*, [4]) - Prob(1', \tau^*, [4]) \,| = \frac{(1-q)\cdot p\cdot q}{p+q-p\cdot q}$,
$|\, Prob(3, \tau^*, [4]) - Prob(1', \tau^*, [4]) \,| = q$,
$|\, Prob(2, \tau^*, [4]) - Prob(1, \tau^*, [4]) \,| = \frac{p\cdot q^2}{p+q-p\cdot q}$,
$|\, Prob(3, \tau^*, [4]) - Prob(1, \tau^*, [4]) \,| = \frac{q^2-p\cdot q^2}{p+q-p\cdot q}$,
$|\, Prob(3, \tau^*, [4]) - Prob(2, \tau^*, [4]) \,| = \frac{q^2}{p+q-p\cdot q}$.
The same results follow if we consider the sequence $\tau^* unfair$ leading to class [5]. In order to compute $\delta_q^{R_1}$, we first point out that $\frac{p}{p+q-p\cdot q}$ cannot be greater than 1 (note that such a quantity corresponds to the probability of reaching

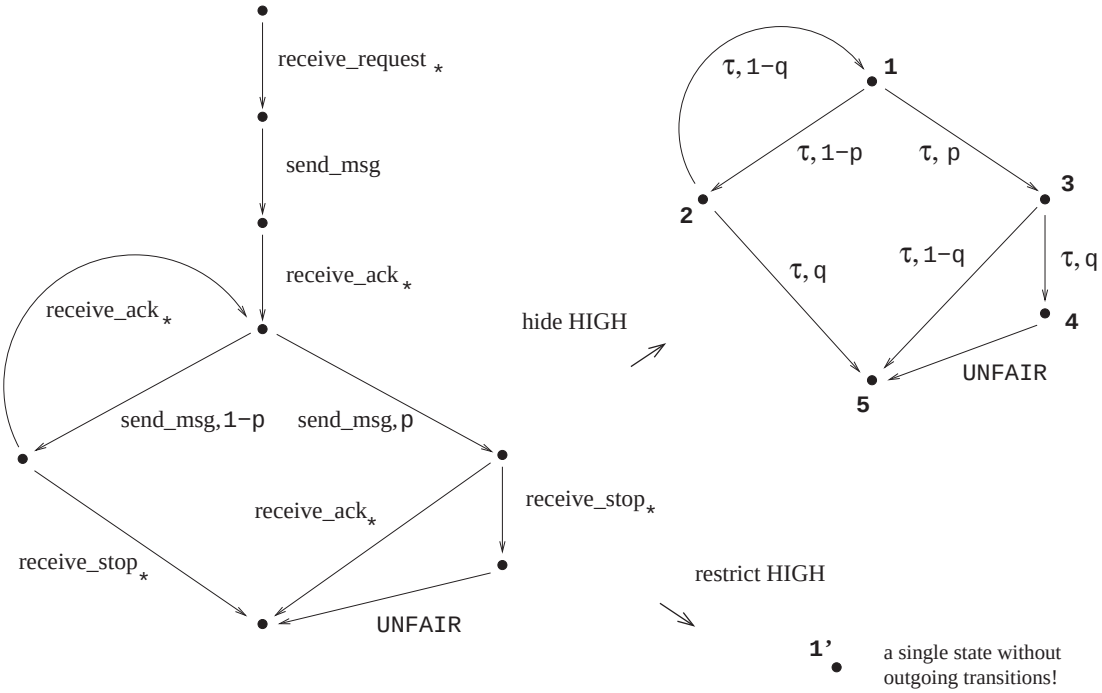

Fig. 1. Nonrepudiation protocol specification and models to be compared through equivalence checking.

state 3 from state 1 through a sequence τ^*). Similarly, we have that $\frac{q}{p+q-p\cdot q} \leq 1$. Therefore, it can be easily verified that $\delta_q^{R_1} = q$. Note that in the limiting scenario $q = 0$ we obtain $\delta_0^{R_1} = 0$, which, obviously, represents the worst case from the viewpoint of the adversary. Intuitively, in such a case we have that state 4 is not reachable and the protocol cannot terminate in an unfair way for the originator. Instead, as q tends to 1 we have that $\delta_q^{R_1}$ tends to 1, because of the difference between states $1'$ (which cannot reach state 4) and 3 (which reaches state 4 with probability q). On the basis of such considerations, we consider relation $R_2 = \{\{1, 1', 2, 5\}, \{3\}, \{4\}\}$, for which we obtain $\delta_q^{R_2} = \frac{p}{p+q-p\cdot q} = |\, Prob(1, \tau^*, [3]) - Prob(1', \tau^*, [3])\,|$.

We summarise the analysis of further relations as follows. For each relation R including the pair $(1', 4)$ it follows $\delta_q^R = 1$. Indeed, from state $1'$ it is not possible to reach class [5], while state 4 reaches state 5 with probability 1 through the action *unfair*. We can argue similarly if $(2, 3) \in R$, since from state 1 we reach class [2] with probability 1, while from state $1'$ we cannot reach class [2]. In the case $(2, 4) \in R$, we have that $Prob(4, \tau^*, [5]) = 0$, while $Prob(2, \tau^*, [5]) > q$, hence $\delta_q^R > q$. In the case $C = \{3, 4\}$ is a class in $\mathcal{G}/R$, then $\delta_q^R \geq \frac{p}{p+q-p\cdot q}$, since $Prob(1, \tau^*, C) = \frac{p}{p+q-p\cdot q}$ and $Prob(1', \tau^*, C) = 0$. We can argue similarly for $R = \{\{1, 1', 5\}, \{2\}, \{3\}, \{4\}\}$. For $R = \{\{1, 1', 3, 5\}, \{2\}, \{4\}\}$ we have $\delta_q^R \geq q$, since $Prob(3, \tau^*, [4]) = q$ and $Prob(1', \tau^*, [4]) = 0$. Finally, for each R such that $\{1, 1'\}$ is a class in $\mathcal{G}/R$ we can argue as in the previous cases.

Hence, we can conclude that $\varepsilon_q = \min\{\delta_q^{R_1}, \delta_q^{R_2}\} = \min\{q, \frac{p}{p+q-p\cdot q}\}$. In

particular, we have that $\varepsilon_q = q$ if $p \geq \frac{q^2}{1-q+q^2}$, otherwise $\varepsilon_q = \frac{p}{p+q-p \cdot q}$ if $p \leq \frac{q^2}{1-q+q^2}$. The value of ε_q expresses how similar can be the process that does nothing (which, by definition, is secure) and the process that executes the protocol with a recipient probabilistically modeled by parameter q. Obviously, the smaller ε_q is, the higher the security degree of the protocol. By varying parameter q, a malicious recipient can try to maximise the difference between the two low-level views of the system. Formally, the maximum value of ε_q obtained by varying parameter q can be calculated by solving a non-linear optimisation problem. As shown above, the value of ε_q strictly depends on the value of p. Since p is a parameter under the control of the originator, it can be verified that the maximal difference between the two low-level views of the system can be kept as small as desired by decreasing p. Obviously, the smaller p is chosen, the longer the duration of the protocol is. Therefore, a real implementation of the protocol becomes impractical if the participants require a very small tolerance to violations of the security property.

We now discuss an alternative estimation that is less coarse than that provided so far. In our approach to probabilistic protocol analysis, high-level actions model the protocol communication events, while low-level actions are extra signals that are observed by Low in order to infer the behaviour of the protocol run. Hence, Low distinguishes between two different protocol runs on the basis of such observations (i.e. the extra signals it can consume) rather than directly interacting with the protocol entities. That means the distance between pairs of states depends on the difference between the probabilities of the weak transitions of the form $\tau^* l$, where l is a visible action observed by Low, i.e. one of the extra signals emitted during the protocol run. Based on these considerations, we now employ a variant of Definition 2.4 that replaces the condition $a \in GAct$ by the condition $a \in GAct - \{\tau\}$. For R_1, we have $\delta_q^{R_1} = q = |\, Prob(3, \tau^* unfair, [5]) - Prob(1', \tau^* unfair, [5]) \,|$. For R_2, it follows $\delta_q^{R_2} = \frac{p \cdot q}{p+q-p \cdot q} = |\, Prob(1, \tau^* unfair, [5]) - Prob(1', \tau^* unfair, [5]) \,|$. Therefore, $\delta_q^{R_2} < \delta_q^{R_1}$. For each other relation R, δ_q^R is greater than (equal to) $\delta_q^{R_2}$. Therefore, $\varepsilon_q = \frac{p \cdot q}{p+q-p \cdot q}$, i.e. ε_q tends to the limiting value p as q tends to the limiting value 1. Again, the value of p determines, from the viewpoint of an external low-level observer, the difference between the system modelling the protocol and a secure system. In particular, p represents the maximal probability of observing an unfair execution, obtained when the adversary sets $q = 1$, i.e. the protocol is stopped after the execution of the first step.

4 Conclusion

We have applied a formal approach to estimating the security of a system to a case study, namely a probabilistic nonrepudiation protocol. This approach allows us to measure the amount of information leakage caused by the probabilistic adversaries of a class $\mathcal{A}$ defined by a probabilistic noninterference

property. The quantitative analysis is based on a notion of process similarity corresponding to an approximate version of the weak probabilistic bisimulation semantics [5]. Such an approximation provides an upper bound ε for the probability of observing a security violation caused by the most powerful adversary in $\mathcal{A}$. For a comparison with related work, the reader is referred to [1].

As we have discussed in Section 3.3, the cost for estimating ε can be reduced by considering a restricted set of interesting relations and by discarding those that cannot contribute to find ε. We intend to investigate the effectiveness of such a strategy since the number of potential relations to be checked factorially increases as the number of states increases (note that we have to take all the possible disjoint subsets of states that form a covering of the state space).

We have discussed an alternative definition of the quantity ε resulting from an analysis which takes into account the actual observational power of a low-level user in our specific case study. This yields a finer estimation of the protocol security. We are investigating other alternative definitions aiming at more precise estimations of the information leakage. In particular, when computing the distance between a pair of reachable states it could be useful for the analysis to take into account also the probability of reaching such states. For instance, the similarity between the initial states of two processes to be compared should not have the same weight as the similarity between two states that are reachable with a negligible probability. Thus, we believe that including such weights in the calculation of ε will increase the precision of the resulting estimation.

Finally, as a future work, we intend to apply the approach presented in this paper to other properties in order to extend the class of probabilistic adversaries that are considered. Indeed, more powerful adversaries are needed to reveal a larger spectrum of security problems of probabilistic protocols.

References

[1] Aldini, A., and A. Di Pierro, *A Quantitative Approach to Noninterference for Probabilistic Systems*, in ENTCS – Selected Papers from MIUR project MEFISTO "Formal Methods for Security" – to appear.
http://mefisto.web.cs.unibo.it

[2] Aldini, A., and R. Gorrieri, Security Analysis of a Probabilistic Nonrepudiation Protocol, in Proc. of *2nd Workshop on Process Algebra and Performance Modelling, Probabilistic Methods in Verification (PAPM-ProbMiV'02)*, Springer LNCS **2399**:17–36, 2002.

[3] Aldini, A., M. Bravetti, A. Di Pierro, R. Gorrieri, C. Hankin, and H. Wiklicky, Two Formal Approaches for Approximating Noninterference

Properties, *Foundations of Security Analysis and Design II – Tutorial Lectures*, Springer LNCS **2946**:1–43, 2004.

[4] Aldini, A., M. Bravetti, and R. Gorrieri, *A Process-algebraic Approach for the Analysis of Probabilistic Noninterference*, Journal of Computer Security **12**(2), 2004.

[5] Baier, C., and H. Hermanns, Weak Bisimulation for Fully Probabilistic Processes, in Proc. of *9th Int. Conf. on Computer Aided Verification*, Springer LNCS **1254**:119–130, 1997.

[6] Bravetti, M., and A. Aldini, *Discrete Time Generative-reactive Probabilistic Processes with Different Advancing Speeds*, Theoretical Computer Science **290**(1):355–406, 2003.

[7] Bravetti, M., and M. Bernardo. Compositional Asymmetric Cooperations for Process Algebras with Probabilities, Priorities, and Time, in Proc. of *1st Workshop on Models for Time-Critical Systems*, ENTCS **39**(3), 2000.

[8] Di Pierro, A., C. Hankin, and H. Wiklicky, *Approximate Non-Interference*, Journal of Computer Security, 12(1):37–81, 2004.

[9] Di Pierro, A., C. Hankin, and H. Wiklicky, *Probabilistic Confinement in a Declarative Framework*, in Declarative Programming – Selected Papers from AGP 2000 – ENTCS **48**, 1–23, 2001.

[10] Focardi, R., and R. Gorrieri, *A Classification of Security Properties*, Journal of Computer Security **3**(1):5–33, 1995.

[11] Glabbeek, R. J. van, S. A. Smolka, and B. Steffen. *Reactive, Generative and Stratified Models of Probabilistic Processes*, Information and Computation **121**:59–80, 1995.

[12] Goguen, J. A., and J. Meseguer, Security Policy and Security Models, in Proc. of *IEEE Symposium on Security and Privacy*, pp. 11–20, 1982.

[13] Gray III, J.W., Probabilistic Interference, in Proc. of *IEEE Symposium on Security and Privacy*, pp. 170–179, 1990.

[14] Markowitch, O., and Y. Roggeman, Probabilistic Non-Repudiation without Trusted Third Party. *2nd Conf. on Security in Comm. Networks*, 1999.

[15] Milner, R., "Communication and Concurrency", Prentice Hall, 1989.

[16] Sabelfeld, A., and D. Sands, Probabilistic Noninterference for Multi-threaded Programs, in Proc. *13th IEEE Computer Security Foundations Workshop (CSFW'00)*, pp. 200–214, 2000.

[17] Volpano, D., and G. Smith, Probabilistic Noninterference in a Concurrent Language, in Proc. of *11th IEEE Computer Security Foundations Workshop (CSFW'98)*, pp. 34–43, 1998.