Information Flow Security for Stochastic Processes

Information Flow Security for Stochastic Processes

Jane Hillston[1], Andrea Marin[2], Carla Piazza[3], and Sabina Rossi[2]

[1] University of Edinburgh, UK
Jane.Hillston@ed.ac.uk
[2] Università Ca' Foscari Venezia, Italy
{marin,sabina.rossi}@unive.it
[3] Università di Udine, Italy
carla.piazza@uniud.it

Abstract. In this paper we study an information flow security property for systems specified as terms of a quantitative process algebra, namely Performance Evaluation Process Algebra (PEPA). Intuitively, we propose a quantitative extension of the Non-Interference property used to secure systems from the functional point view by assuming that the observers are able to measure also the timing properties of the system, e.g., the response time or the throughput.
We introduce the notion of *Persistent Stochastic Non-Interference (PSNI)* and provide two characterizations of it: one based on a bisimulation-like equivalence relation inducing a lumping on the underlying Markov chain, and another one based on unwinding conditions which demand properties of individual actions. These two different characterizations naturally lead to efficient methods for the verification and construction of secure systems. A decision algorithm for *PSNI* is presented and an application of *PSNI* to a queueing system is discussed.

1 Introduction

In the last decades, security of information systems has become a crucial topic of research. Finding a formal characterisation of the various properties defined in the context of security, (e.g., confidentiality, anonymity, integrity, etc.) has been an active field of research. Beside numerous definitions of security have been proposed, very few results take into account the time behaviour of the analysed system. However, it is well-known that from the observation of the response times of a system, malicious observers can infer some characteristics that may help an attack to succeed (see, e.g., [2, 3, 5]). In this paper, we propose a first set of results to cover this gap. We consider systems specified as terms of a quantitative process algebra, namely Performance Evaluation Process Algebra (PEPA). In contrast with most the process algebras used in previous well-known results (e.g., the CCS used for the Non-Interference property [6]), PEPA allows us to specify random delays to model the quantitative properties of the system. Besides, the results that we present can be applied to any Markovian formalism with a synchronisation operator in the style of PEPA cooperation, e.g., the Kronecker's product for Stochastic Automata Networks (see [13, 12] and the references therein).

Intuitively, the idea that we propose is a quantitative extension of the Non-Interference property that has been widely used to secure systems from the functional point view [4, 6–8, 18, 16, 17, 14, 15]. Let us consider a system that performs some actions that are intended to be confidential and some others that are observable by an external, possibly malicious, user. Roughly speaking, in the standard, functional, definition of Non-Interference a system S is secure if any external observer is not able to distinguish the behaviour of S performing confidential, secret, activities from the behaviour of the same system but prevented from performing any secret action. In our setting, the definition does not change, however we assume that the observer is able to measure also the timing properties of that system, e.g., the response time or the throughput. In this paper we consider the strictest situation in terms of security requirements, i.e., the observer can see any observable execution path with its delays, i.e., he/she can see the transient behaviour of the system and study correlation properties, averages, etc. The request that for any execution path of the model that performs unobservable, private, actions there exists a corresponding execution path in the model that does not perform private actions (and vice versa) clearly implies that the two models are also indistinguishable when observed in steady-state. However, as shown in the example of Section 5, the opposite is in general not true.

We introduce a notion of *stochastic Non-Interference* which is *persistent* in the sense that if a system is secure then all its reachable states are secure too. We show that such property, named *Persistent Stochastic Non-Interference (PSNI)* can be charaterized in terms of a bisimulation-like equivalence relation, between the whole system and the system prevented from performing confidential activities. The property that we propose is strictly related to the lumping of Markov chains since the observation equivalence at the base of our definition relies on the notion of lumpability [10]. Moreover, we provide a characterization of *PSNI* in terms of unwinding conditions which demand properties of individual actions. These two different characterizations naturally lead to efficient methods for the verification and construction of secure systems. We prove that *PSNI* can be verified in polynomial time with respect to the number of states of a system.

We describe an application of *PSNI* to a simple queueing system in which at random instants some private internal operations are performed. Although the functionality of the system is not altered by these operations (and hence the standard Non-Interference is satisfied), the response time is worsen and hence private information can be leaked. We show a simple workaround that makes the system secure and discuss its implications in terms of overall performance.

Structure of the paper. The paper is organized as follows. In Section 2 we introduce the process algebra PEPA, its semantics, and the observation equivalence named *lumpable bisimilarity.* The notion of *Persistent Stochastic Non-Interference (PSNI)* and its characterizations are presented in Section 3. In Sec-

$$\frac{}{(\alpha,r).P \xrightarrow{(\alpha,r)} P} \qquad \frac{P \xrightarrow{(\alpha,r)} P'}{P+Q \xrightarrow{(\alpha,r)} P'} \qquad \frac{Q \xrightarrow{(\alpha,r)} Q'}{P+Q \xrightarrow{(\alpha,r)} Q'}$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P/L \xrightarrow{(\alpha,r)} P'/L}\ (\alpha \notin L) \qquad \frac{P \xrightarrow{(\alpha,r)} P'}{P/L \xrightarrow{(\tau,r)} P'/L}\ (\alpha \in L) \qquad \frac{P \xrightarrow{(\alpha,r)} P'}{A \xrightarrow{(\alpha,r)} P'}\ (A \overset{def}{=} P)$$

$$\frac{P \xrightarrow{(\alpha,r)} P'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,r)} P' \underset{L}{\bowtie} Q}\ (\alpha \notin L) \qquad \frac{Q \xrightarrow{(\alpha,r)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,r)} P \underset{L}{\bowtie} Q'}\ (\alpha \notin L)$$

$$\frac{P \xrightarrow{(\alpha,r_1)} P' \quad Q \xrightarrow{(\alpha,r_2)} Q'}{P \underset{L}{\bowtie} Q \xrightarrow{(\alpha,R)} P' \underset{L}{\bowtie} Q'} \qquad R = \frac{r_1}{r_\alpha(P)}\frac{r_2}{r_\alpha(Q)} \min(r_\alpha(P), r_\alpha(Q)) \quad (\alpha \in L)$$

Table 1: Operational semantics for PEPA components

tion 4 we describe an algorithm to decide whether a PEPA component is *PSNI*. Section 5 presents a simple example of a queueing system in which some private operations are preformed. Finally, Section 6 concludes the paper.

2 The Calculus

PEPA (Performance Evaluation Process Algebra) is a popular Markovian process algebra introduced in [9] that allows one to model and study the quantitative properties of systems. It consists of two basic elements: the *components* and the *activities*. Activities are pairs (α, r) where α is a label or *action type* belonging to a countable set $\mathcal{A}$, and $r \in \mathbb{R}^+ \cup \{\top\}$ is its *rate* The duration of an activity is a negative exponential distribution with mean r^{-1}. Action type $\tau \in \mathcal{A}$ is the *unknown* type. Activity rates may be $\top$ which should be read as *unspecified*. The syntax for PEPA terms follows the grammar:

$$P ::= P \underset{L}{\bowtie} P \mid P/L \mid S$$
$$S ::= (\alpha, r).S \mid S + S \mid A$$

where S denotes a *sequential component* and P denotes a *model component* which runs in parallel. Finally, A is a countable set of *constants* and $\mathcal{C}$ denotes the set of all possible components.

Operational semantics. Table 1 shows the operational semantics of PEPA. The component $(\alpha, r).P$ carries out the activity (α, r) of type α at rate r and subsequently behaves as P. When $a = (\alpha, r)$, the component $(\alpha, r).P$ may be written as $a.P$. $P + Q$ specifies a system which may behave either as P or as Q and where all the current activities of both P and Q are enabled. The first activity to complete distinguishes one of the components, P or Q. The other component of the choice is discarded. The component P/L behaves as P except that any activity of type within the set L are *hidden*, i.e., they are relabelled with the unknown type τ. The meaning of a constant A is given by a defining equation such as $A \overset{def}{=} P$ which gives the constant A the behaviour of the component P. The cooperation combinator $\underset{L}{\bowtie}$ is in fact an indexed family of combinators, one for each possible set of action types, $L \subseteq \mathcal{A} \setminus \{\tau\}$. The *cooperation set* L defines the action types on which the components must synchronise or *cooperate* (the unknown action type, τ, may not appear in any cooperation set). It is assumed that each component proceeds independently with the activities whose types do not occur in the cooperation set L (*individual activities*). However, activities with action types in L require the simultaneous involvement of both components. The shared activity will have the same action type as the two contributing activities and its rate is that of the slower component. If in a component an activity has rate $\top$, then we say that it is is passive with respect to that action type. In this case the rate of the shared activity will be that of the other component. For a given P and action type α, the *apparent rate* of α in P, denoted by $r_\alpha(P)$, is the sum of the rates of the α activities enabled in P.

The semantics of each term in PEPA is given via a labelled *multi-transition system* where the multiplicities of arcs are significant. In the transition system, a state or *derivative* corresponds to each syntactic term of the language and an arc represents the activity which causes one derivative to evolve into another. The set of reachable states of a model P is termed the *derivative set* of P ($ds(P)$) and constitutes the set of nodes of the *derivation graph* of P ($\mathcal{D}(P)$) obtained by applying the semantic rules exhaustively. We denote by $\mathcal{A}(P)$ the set of all the *current action types* of P, i.e., the set of action types which the component P may next engage in. We denote by $\mathcal{A}ct(P)$ the multiset of all the *current activities* of P. Finally we denote by $\boldsymbol{\mathcal{A}}(P)$ the union of all $\mathcal{A}(P')$ with $P' \in ds(P)$, i.e., the set of all action types syntactically occurring in P. For any component P, the *exit rate* from P will be the sum of the activity rates of all the activities enabled in P, i.e., $q(P) = \sum_{a \in \mathcal{A}ct(P)} r_a$, with r_a being the rate of activity a. If P enables more than one activity, $|\mathcal{A}ct(P)| > 1$, then the dynamic behaviour of the model is determined by a race condition. As a consqeuence, the nondeterministic branching of the pure process algebra is replaced by a probabilistic branching. Thanks to the exponential assumption, the probability that a particular activity completes is the ratio between its rate and the exit rate from P.

Underlying Markov Chain. Let $P \overset{def}{=} P_0$ with $ds(P) = \{P_0, \ldots, P_n\}$ be a finite PEPA model. Then, the stochastic process $X(t)$ on the space $ds(P)$ is a continuous time Markov chain [9].

The *transition rate* between two states P_i and P_j is denoted by $q(P_i, P_j)$ and corresponds to rate at which the system changes from behaving as component P_i to behaving as P_j, i.e., it is the sum of the activity rates labelling arcs which connect the node corresponding to P_i to the node corresponding to P_j in the derivation graph. Formally:

$$q(P_i, P_j) = \sum_{a \in \mathcal{A}ct(P_i|P_j)} r_a$$

with $P_i \neq P_j$ and $\mathcal{A}ct(P_i|P_j) = \{\!| a \in \mathcal{A}ct(P_i)|\ P_i \xrightarrow{a} P_j |\!\}$. When P_j is not a one-step derivative of P_i we set $q(P_i, P_j) = 0$. In the following, when possible, we will write q_{ij} instead of $q(P_i, P_j)$. In the definition of the infinitesimal generator $\mathbf{Q}$ of $X(t)$, q_{ij}, $i \neq j$, are the off-diagonal elements of the matrix whereas the diagonal elements are, as usual, the negative sum of the row non-diagonal elements, i.e., $q_{ii} = -q(P_i)$. For any finite and irreducible PEPA model P, the steady-state distribution $\Pi(\cdot)$ exists and it may be found by solving the probability normalising equation and the linear system of global balance equations: $\sum_{P_i \in ds(P)} \Pi(P_i) = 1$ and $\Pi\mathbf{Q} = \mathbf{0}$. Another notion that will be used in the paper is that of *conditional transition rate* from P_i to P_j via an action type α, denoted by $q(P_i, P_j, \alpha)$. This is the sum of the activity rates labelling arcs connecting the corresponding nodes in the derivation graph which are also labelled by the action type α. It is the rate at which a system behaving as component P_i evolves to behaving as component P_j as the result of completing a type α activity. The *total conditional transition rate* from P to $S \subseteq ds(P)$, denoted $q[P, S, \alpha]$, is defined as

$$q[P, S, \alpha] = \sum_{P' \in S} q(P, P', \alpha)$$

where $q(P, P', \alpha) = \sum_{P \xrightarrow{(\alpha, r_\alpha)} P'} r_\alpha$.

Observation Equivalence When we study a system by means of a process algebraic model, actions, rather than states, are used to capture its observable behaviour. Therefore, we introduce an equivalence notion in which components are regarded as equal if an external observer sees them performing exactly the same actions. In this section we recall a bisimulation-like relation, named *lumpable bisimulation*, for PEPA models that we previously introduced in [10].

Two PEPA components are *lumpably bisimilar* if there exists an equivalence relation between them such that, for any action type α different from τ, the total conditional transition rates from those components to any equivalence class, via activities of this type, are the same.

Definition 1. (Lumpable bisimulation) *An equivalence relation over PEPA components, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$, is a* lumpable bisimulation *if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$ such that*

- *either $\alpha \neq \tau$,*
- *or $\alpha = \tau$ and $P, Q \notin S$,*

it holds

$$q[P, S, \alpha] = q[Q, S, \alpha]\,.$$

Notice that, in contrast with the notion of strong equivalence [9], lumpable bisimulation allows arbitrary activities with type τ among components belonging to the same equivalence class, and therefore it is less strict.

We are interested in the relation which is the largest lumpable bisimulation, formed by the union of all lumpable bisimulations.

Definition 2. (Lumpable bisimilarity) *Two PEPA components P and Q are* lumpably bisimilar*, written $P \approx_l Q$, if $(P, Q) \in \mathcal{R}$ for some lumpable bisimulation $\mathcal{R}$, i.e.,*

$$\approx_l = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a lumpable bisimulation}\}.$$

$\approx_l$ is called lumpable bisimilarity *and it is the largest symmetric lumpable bisimulation over PEPA components.*

In [10] we proved that lumpable bisimilarity is a congruence for the so-called evaluation contexts, i.e., if $P_1 \approx_l P_2$ then

- $a.P_1 \approx_l a.P_2$;
- $P_1/L \approx_l P_2/L$;
- $P_1 \underset{L}{\bowtie} Q \approx_l P_2 \underset{L}{\bowtie} Q$ for all $L \subseteq \mathcal{A}$.

3 Persistent Stochastic Non-Interference

The security propery named *Persistent Stochastic Non-Interference (PSNI)* tries to capture every possible information flow from a *classified (high)* level of confidentiality to an *untrusted (low)* one. A strong requirement of this definition is that no information flow should be possible even in the presence of malicious processes that run at the classified level.

The definition of *PSNI* is based on the basic idea of Non-Interference [8]: "No information flow is possible from high to low if what is done at the high level *cannot interfere* in any way with the low level".

More precisely, the notion of *PSNI* consists of checking all the states reachable by the system against all high level potential interactions.

In order to formally define our security property, we partition the set $\mathcal{A} \setminus \{\tau\}$ of visible action types, into two sets, $\mathcal{H}$ and $\mathcal{L}$ of high and low level action types. A high level PEPA component H is a PEPA term such that for all $H' \in ds(H)$, $\mathcal{A}(H') \subseteq \mathcal{H}$, i.e., every derivative of H may next engage in only high level actions. We denote by $\mathcal{C}_H$ the set of all high level PEPA components.

A system P satisfies *PSNI* if for every state P' reachable from P and for every high level process H a low level user cannot distinguish P' from $P' \underset{\mathcal{H}}{\bowtie} H$. In other words, a system P satisfies *PSNI* if what a low level user sees of the system is not modified when it cooperates with any high level process H.

In order to formally define the *PSNI* property, we denote by $P \setminus \mathcal{H}$ the PEPA component $(P \underset{\mathcal{H}}{\bowtie} \bar{H})$ where $\bar{H}$ is any high level process that does not cooperate with P, i.e., for all $P' \in ds(P)$, $\mathcal{A}(P') \cap \mathcal{A}(\bar{H}) = \emptyset$. Intuitively $P \setminus \mathcal{H}$ denotes the component P prevented from performing high level actions.

First we prove that $P \setminus \mathcal{H}$ is well defined, i.e., it does not depend on $\bar{H}$. The proof follows by structural induction on P.

Lemma 1. *Let P be a PEPA component and $\bar{H}$ be a high level process that does not cooperate with P. $P \underset{\mathcal{H}}{\bowtie} \bar{H} \xrightarrow{(\alpha,r)} Q$ if and only if Q is of the form $P' \underset{\mathcal{H}}{\bowtie} \bar{H}$ and $P \xrightarrow{(\alpha,r)} P'$ with $\alpha \in \mathcal{L} \cup \{\tau\}$.*

Notice that the above lemma applies also to P' and more in general to all the processes in $ds(P)$, since they do not cooperate with $\bar{H}$.

Lemma 2. *Let P be a PEPA component. Let $\bar{H}_1$ and $\bar{H}_2$ be two high level processes that do not cooperate with P, i.e., for all $P' \in ds(P)$, $\mathcal{A}(P') \cap \mathcal{A}(\bar{H}_i) = \emptyset$ for $i = 1, 2$. The derivation graphs $\mathcal{D}(P \underset{\mathcal{H}}{\bowtie} \bar{H}_1)$ and $\mathcal{D}(P \underset{\mathcal{H}}{\bowtie} \bar{H}_2)$ are isomorphic as graphs with labels on the edges.*

The formal definition of *PSNI* is as follows.

Definition 3. *Let P be a PEPA component.*

$$P \in PSNI \textit{ iff } \forall P' \in ds(P),\, \forall H \in \mathcal{C}_H,$$

$$P' \setminus \mathcal{H} \approx_l (P' \underset{\mathcal{H}}{\bowtie} H)/\mathcal{H}\,.$$

We introduce a novel bisimulation-based equivalence relation over PEPA components, named $\approx_l^{hc}$, that allows us to give a first characterization of *PSNI* with no quantification over all the high level components H. In particular, we show that $P \in$ *PSNI* if and only if $P \setminus \mathcal{H}$ and P are not distinguishable with respect to $\approx_l^{hc}$. Intuitively, two processes are $\approx_l^{hc}$-equivalent if they can simulate each other in any possible high context, i.e., in every context $C[\_]$ of the form $(\_ \underset{\mathcal{H}}{\bowtie} H)/\mathcal{H}$ where $H \in \mathcal{C}_H$. Observe that for any high context $C[\_]$ and PEPA model P, all the states reachable from $C[P]$ have the form $C'[P']$ with $C'[\_]$ being a high context too and $P' \in ds(P)$.

We now introduce the concept of *lumpable bisimulation on high contexts*: the idea is that, given two PEPA models P and Q, when a high level context $C[\_]$ filled with P executes a cetain activity moving P to P' then the same context filled with Q is able to simulate this step moving Q to Q' so that P' and Q' are again lumpable bismilar on high contexts, and vice-versa. This must be true for every possible high context $C[\_]$. It is important to note that the quantification over all possible high contexts is re-itereted for P' and Q'.

We use the following notation. For a PEPA model P, $\alpha \in \mathcal{A}$, $S \subseteq ds(P)$ and a high context $C[\_]$ we define:

$$q_C(P, P', \alpha) = \sum_{C[P] \xrightarrow{(\alpha, r_\alpha)} C'[P']} r_\alpha$$

and

$$q_C[P, S, \alpha] = \sum_{P' \in S} q_C(P, P', \alpha)\,.$$

The notion of *lumpable bisimulation on high contexts* is defined as follows:

Definition 4. (Lumpable bisimilarity on high contexts) *An equivalence relation over PEPA components, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$, is a* lumpable bisimulation on high contexts *if whenever $(P, Q) \in \mathcal{R}$ then for all high context $C[\_]$, for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$ such that*

- *either $\alpha \neq \tau$,*
- *or $\alpha = \tau$ and $P, Q \notin S$,*

it holds

$$q_C[P, S, \alpha] = q_C[Q, S, \alpha]\,.$$

Two PEPA components P and Q are lumpably bisimilar on high contexts*, written $P \approx_l^{hc} Q$, if $(P, Q) \in \mathcal{R}$ for some lumpable bisimulation on high contexts $\mathcal{R}$, i.e.,*

$$\approx_l^{hc} = \bigcup \{\mathcal{R} \mid \mathcal{R} \textit{ is a lumpable bisimulation on high contexts}\}.$$

$\approx_l^{hc}$ *is called* lumpable bisimilarity on high contexts *and it is the largest symmetric lumpable bisimulation on high contexts over PEPA components.*

The next theorem provides a characterization of *PSNI* in terms of $\approx_l^{hc}$.

Theorem 1. *Let P be a PEPA component. Then*

$$P \in PSNI \textit{ iff } P \setminus \mathcal{H} \approx_l^{hc} P\,.$$

We now show how it is possible to give a characterization of *PSNI* avoiding both the universal quantification over all the possible high level components and the universal quantification over all the possible reachable states.

Before we have shown how the idea of “being secure in every state” can be directly moved inside the lumpable bisimulation on high contexts notion ($\approx_l^{hc}$). However this bisimulation notion implicitly contains a quantification over all possible high contexts. We prove that $\approx_l^{hc}$ can be expressed in a rather simpler way by exploiting local information only. This can be done by defining a novel equivalence relation which focuses only on observable actions that do not belong to $\mathcal{H}$. More in detail, we define an observation equivalence where actions from $\mathcal{H}$ *may* be ignored. We introduce the notion of *lumpable bisimilarity up to $\mathcal{H}$*.

Definition 5. (Lumpable bisimilarity up to $\mathcal{H}$) *An equivalence relation over PEPA components, $\mathcal{R} \subseteq \mathcal{C} \times \mathcal{C}$, is a* lumpable bisimulation up to $\mathcal{H}$ *if whenever $(P, Q) \in \mathcal{R}$ then for all $\alpha \in \mathcal{A}$ and for all $S \in \mathcal{C}/\mathcal{R}$*

- *if $\alpha \notin \mathcal{H} \cup \{\tau\}$ then*
$$q[P, S, \alpha] = q[Q, S, \alpha]\,,$$
- *if $\alpha \in \mathcal{H} \cup \{\tau\}$ and $P, Q \notin S$, then*
$$q[P, S, \alpha] = q[Q, S, \alpha]\,.$$

Two PEPA components P and Q are lumpably bisimilar up to $\mathcal{H}$, *written* $P \approx_l^{\mathcal{H}} Q$, *if* $(P, Q) \in \mathcal{R}$ *for some lumpable bisimulation up to* $\mathcal{H}$, *i.e.,*

$$\approx_l^{\mathcal{H}} = \bigcup \{\mathcal{R} \mid \mathcal{R} \textit{ is a lumpable bisimulation up to } \mathcal{H}\}.$$

$\approx_l^{\mathcal{H}}$ *is called* lumpable bisimilarity up to $\mathcal{H}$ *and it is the largest symmetric lumpable bisimulation up to* $\mathcal{H}$ *over PEPA components.*

The next theorem shows that the binary relations $\approx_l^{hc}$ and $\approx_l^{\mathcal{H}}$ are equivalent.

Theorem 2. *Let P and Q be two PEPA components. Then*

$$P \approx_l^{hc} Q \textit{ if and only if } P \approx_l^{\mathcal{H}} Q\,.$$

Theorem 2 allows us to identify a local property of processes (with no quantification on the states and on the high contexts) which is a necessary and sufficient condition for *PSNI*. This is stated by the following corollary:

Corollary 1. *Let P be a PEPA component. Then*

$$P \in PSNI \textit{ iff } P \setminus \mathcal{H} \approx_l^{\mathcal{H}} P\,.$$

Finally we provide a characterization of *PSNI* in terms of *unwinding conditions* which demand properties of individual activities. In practice, whenever a state P' of a *PSNI* PEPA model P may execute a high level activity leading it to a state P'', then P' and P'' are indistinguishable for a low level observer.

Theorem 3. *Let P be a PEPA component.*

$$P \in PSNI \textit{ iff } \forall P' \in ds(P),$$

$$P' \xrightarrow{(h,r)} P'' \textit{ implies } P' \setminus \mathcal{H} \approx_l P'' \setminus \mathcal{H}\,.$$

Using the equivalence relation $\approx_l^{\mathcal{H}}$ this can be reformulated as follows.

Theorem 4. *Let P be a PEPA component.*

$$P \in PSNI \textit{ iff } \forall P' \in ds(P),$$

$$P' \xrightarrow{(h,r)} P'' \textit{ implies } P' \approx_l^{\mathcal{H}} P''\,.$$

Theorems 2, 3 and 4 provide different characterizations of *PSNI* which naturally lead to efficient methods for the verification and construction of secure systems. We also prove some compositionality results that allow us to check the security of a system by only verifying the security of its subcomponents. In particular we prove that $PSNI$ is compositional with respect to the low prefix, hiding, and cooperation over a set of low actions.

Proposition 1. *Let P and Q be two PEPA components. If $P, Q \in PSNI$, then*

- $(\alpha, r).P \in PSNI$ *for all* $\alpha \in \mathcal{L} \cup \{\tau\}$*;*
- $P/L \in PSNI$ *for all* $L \subseteq \mathcal{A}$*;*
- $P \bowtie_L Q \in PSNI$ *for all* $L \subseteq \mathcal{L}$*.*

We also prove that if $P \in PSNI$ then the equivalence class $[P]$ with respect to lumpable bisimilarity $\approx_l$ is closed under *PSNI*.

Proposition 2. *Let* P *and* Q *be two PEPA components. If* $P \in PSNI$ *and* $P \approx_l Q$ *then also* $Q \in PSNI$*.*

4 A Decision Algorithm for *PSNI*

In this section we briefly describe an algorithm to decide whether a PEPA component is *PSNI*. We first exploit the characterization of *PSNI* given in Corollary 1, i.e., we provide an algorithm that given in input two PEPA components P and Q having finite derivative graphs allows one to decide whether $P \approx_l^{\mathcal{H}} Q$. In virtue of Corollary 1 this will allow us to decide whether a process is *PSNI*. As observed in [10] even if the set $\mathcal{C}$ of PEPA components is infinite, since we are interested in $P \approx_l^{\mathcal{H}} Q$ we can safely focus on the graph $\mathcal{D}(P) \cup \mathcal{D}(Q)$. We intend to exploit the algorithm introduced in [1] for solving the label-compatibility problem. To this aim we need to introduce the notion of directed labeled weighted graphs, the label-compatibility problem, and to show how our problem can be mapped into a label-compatibility one.

Definition 6. (Directed labeled weighted graph) *A directed labeled weighted graph is a tuple* $G = (V, Lab, E, w)$ *where:*

- V *is a finite set of vertices;*
- Lab *is a finite set of labels;*
- $E \subseteq V \times V \times Lab$ *is a finite set of labeled edges;*
- $w : E \to \mathbb{R}$ *is a weighting function that associates a value to each edge.*

Given $V' \subseteq V$, we denote by $w(v, V', a)$ the sum of the weights of the edges from v to V' having label a.

The following definition of compatibility introduced in [1] extends that of [19] to directed labeled weighted graphs.

Definition 7. (Label-Compatibility Problem) *Let* $G = (V, Lab, E, w)$ *be a directed labeled weighted graph and* $\mathcal{R} \subseteq V \times V$ *be an equivalence relation over* V*.* $\mathcal{R}$ *is said to be* label-compatible *with* G *if for each* $a \in Lab$*, for each* $C, C' \in V/\mathcal{R}$*, and for each* $v, v' \in C$ *it holds that* $w(v, C', a) = w(v', C', a)$*.*

Let $G = (V, Lab, E, w)$ *be a directed labeled weighted graph the* labeled weighted compatibility problem over G *requires to compute the largest equivalence relation label-compatible with* G*.*

In [1] it has been proved that the label-compatibility problem always has a unique solution. We now introduce the graph that allows us to map our problem of deciding $P \approx_l^{\mathcal{H}} Q$ into a label compatibility problem.

Definition 8. (Up to $\mathcal{H}$ Lumping Graph) *Let P and Q be PEPA components. The* up to $\mathcal{H}$ lumping graph *of $P \cup Q$ is the directed labelled weighted graph $\mathcal{LH}_{P\cup Q} = (V_{P\cup Q}, \mathcal{A}, E_{P\cup Q}, w_{P\cup Q})$, where:*

- *$V_{P\cup Q}$ is $ds(P) \cup ds(Q)$*
- *$E_{P\cup Q}$ is the set of labeled edges*

$$E_{P\cup Q} = \{(R, R', \alpha) \mid R \xrightarrow{(\alpha,r)} R'\} \cup \{(R, R, \alpha) \mid \ \textit{and } \alpha \in \mathcal{H} \cup \{\tau\}\}$$

with R and R' in $V_{P\cup Q}$

- *$w_{P\cup Q}$ is the function which associates to each edge in $E_{P\cup Q}$ the value*

$$w_{P\cup Q}(R, R', \alpha) = \begin{cases} q(R, R', \alpha) & \textit{if} \alpha \notin \mathcal{H} \cup \{\tau\} \vee R \neq R' \\ -q[R, V_{P\cup Q} \setminus \{R\}, \alpha] & \textit{otherwise} \end{cases}$$

When P and Q coincide we use $\mathcal{LH}_P$ to denote $\mathcal{LH}_{P\cup P}$.

Theorem 5. *Let P and Q be two PEPA components. It holds that $P \approx_l^{\mathcal{H}} Q$ if and only if in the largest equivalence relation label-compatible with $\mathcal{LH}_{P\cup Q}$ the vertices P and Q are equivalent.*

As an immediate consequence of the above theorem we get that we can directly exploit the algorithm presented in [1] with initial relation the total relation over $V_{P\cup Q}$ to decide $\approx_l^{\mathcal{H}}$ in polynomial time with respect to the size of the graph $\mathcal{D}(P) \cup \mathcal{D}(Q)$. We refer to such algorithm as $LCW(\_)$[4].

Corollary 2. *Let P and Q be two PEPA components. Let $\mathcal{LH}_{P\cup Q}$ be the up to $\mathcal{H}$ lumping graph of $P\cup Q$ and $LCW(\_)$ be the algorithm reported in the Appendix. $LCW(\mathcal{LH}_{P\cup Q})$ decides $P \approx_l^{\mathcal{H}} Q$ in time $O(|V_{P\cup Q}| + |E_{P\cup Q}| \log |V_{P\cup Q}|)$.*

Notice that we are interested in deciding whether P is *PSNI*, i.e., whether $P \setminus \mathcal{H} \approx_l^{\mathcal{H}} P$. Exploiting the above result together with Corollary 1 this can be done by computing both $\mathcal{D}(P \setminus \mathcal{H})$ and $\mathcal{D}(P)$. From these two $\mathcal{LH}_{(P\setminus\mathcal{H})\cup P}$ can be determined in linear time and then $LCW(\_)$ can be exploited. However, from Theorem 5 together with Theorem 4 we can decide whether P is *PSNI* by simply working on $\mathcal{D}(P)$ as stated in the following theorem.

Theorem 6. *Let P be a PEPA component. Let $Comp_P$ be the largest equivalence relation label-compatible with $\mathcal{LH}_P$. P is* PSNI *if and only if whenever $P' \xrightarrow{(h,r)} P''$ with $P' \in ds(P)$ and $h \in \mathcal{H}$ it holds that $(P', P'') \in Comp_P$.*

This last result lowers the multiplicative constants hidden in the complexity result of Theorem 5, since it avoids the computation and also the management of $\mathcal{D}(P \setminus \mathcal{H})$. Moreover, it substantially reduces the effective complexity of the computation for many non-*PSNI* processes. As a matter of fact during the computation of $Comp_P$ as soon as a split separates two vertices that are connected through a high level transition we can stop the computation and return $P \notin PSNI$. This also suggests strategies for correcting insecure processes.

[4] Given a graph $G = (V, Lab, E, w)$ the use of $LCW(G)$ in this paper corresponds to a call to $LCW(G, V \times V)$ in [1].

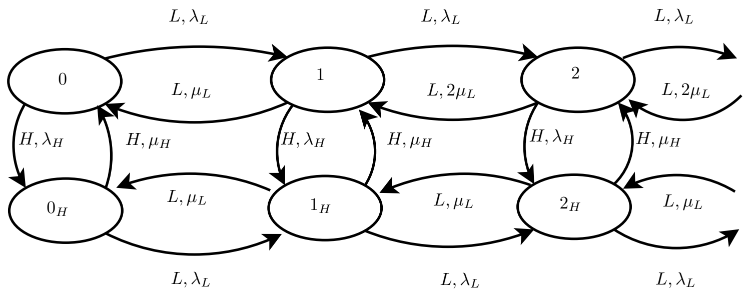

Fig. 1: LTS of the the model that does not satisfy $PSNI$.

5 Example

We consider a distributed system with $n \geq 2$ servers where ordinary jobs arrive according to a homogeneous Poisson process with intensity λ_L. Arrival and departures of ordinary jobs can be observed by a malicious user. The system has an internal job that alternates a phase of sleeping, whose duration is exponential with mean λ_H^{-1}, and a phase of working where it uses one of the n servers for an exponentially distributed time with mean μ_H^{-1}. Each of the ordinary customers requires a service time which is exponentially distributed with mean μ_L^{-1}. If the internal job becomes active and none of the servers is free, then one random ordinary job is preempted and the internal job is executed immediately. Given the exponential distribution of the service time, it is not necessary to discuss the resume policy for the preempted jobs. The waiting room has infinite capacity. The goal is that of hiding the state of the system when the internal process is being executed to the external, possibly malicious, observers. These know how the system works (including the value of μ_L) and the number of available servers.

Notice that in this setting the stability condition is given by:

$$\lambda_L < (n-1)\mu_L + \mu_L \frac{\mu_H}{\mu_L + \mu_H}$$

where the last factor is the probability that the internal process is not active. Fig. 1 shows the labelled transition system (LTS) of the PEPA specification of our model as it has been described so far for $n = 2$. States n and n_H denote the system when it contains n ordinary jobs and the internal process is not active (state n) and active (state n_H), respectively. It is interesting to observe that if the malicious user can only estimate the throughput of the ordinary jobs, then the system could be considered safe since this must be λ_L if the stability condition is met. Nevertheless, a smart observer could pay attention to the transient behaviour of the system, and hence could reasonably estimate the number of ordinary jobs in the system. For instance if $n = 2$, and in a time interval we have k arrivals and h departures, such that $k - h \geq 2$, then the next departure of an ordinary job should occur in an expected time of $(2\mu_L)^{-1}$ if the

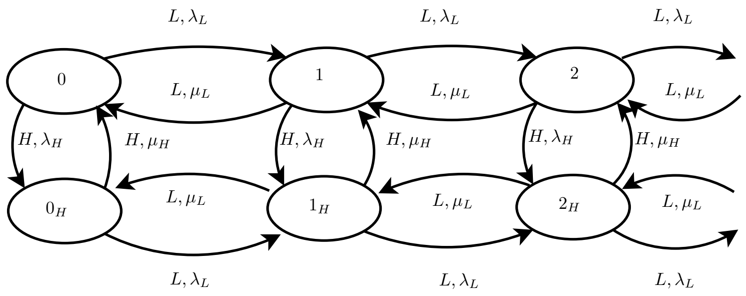

Fig. 2: LTS of the model that satisfies $PSNI$.

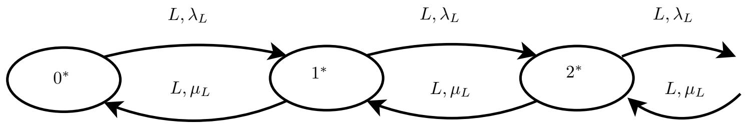

Fig. 3: LTS of the model as seen by an external observer.

internal job is not active and μ_L^{-1}, otherwise. In other words, the observer can apply some statistical methods to infer the probability that the internal job is active from the observation of the transient behaviour of the system.

Formally, we can say that the model of Fig. 1 does not satisfy the conditions of *PSNI*. In fact, the rate outgoing from state i_H to $(i-1)_H$ is different from from that going from i to $i-1$, where $i > 1$. One simple, but expensive, way to obtain a secure system according to *PSNI* is that of devoting one server to the execution of the internal process. The system of Fig. 1 can be modified to obtain that shown in Fig. 2. With these modification, the observer cannot distinguish the model of Fig. 2 from that of Fig. 3. However, in the general case of n servers, the stability condition becomes $\lambda_L < (n-1)\mu_L$, and the expected response time is higher than that of the original model. Finally, we notice that due to the independence between the internal process behaviour and the ordinary job service, in stability, the stationary probability π is:

$$\pi(i) = \begin{cases} (1-\lambda_L/\mu_L)\mu_H/(\lambda_L+\mu_H)(\lambda_L/\mu_L)^i & \text{if } i = 0, 1, \ldots \\ (1-\lambda_L/\mu_L)\lambda_H/(\lambda_L+\mu_H)(\lambda_L/\mu_L)^i & \text{if } i = 0_H, 1_H, \ldots \,. \end{cases}$$

Clearly, the stationary probability of the model of Fig. 3 is that of a M/M/1 queue, i.e., $\pi^*(i^*) = (1-\lambda_L/\mu_L)(\lambda_L/\mu_L)^i$ and we can observe that $\pi^*(i^*) = \pi(i) + \pi(i_H)$, as expected by lumping theory [11].

6 Conclusion

In this paper we presented a *persistent* information flow security property for stochastic processes specified as terms of a quantitative process algebra, namely Performance Evaluation Process Algebra (PEPA). Our property, named *Persistent Stochastic Non-Interference* (*PSNI*) is based on a bisimulation based observation equivalence for the PEPA terms which induces a lumping on the underlying Markov chain. The aim of our definition is that of protecting systems from maliciuos attachers which are able to measure also the timing properties of the system, e.g., the response time or the throughput.

In this paper we also deal with compositionality issues and prove that *PSNI* is compositional with respect to low prefix, cooperation on low actions and hiding.

As a future work we plan to relax the definition of Non-Interference by introducing metrics that allow us to measure the security degree of a system in terms of probabilities.

Acknowledgments: The work described in this paper has been partially supported by the Università Ca' Foscari Venezia - DAIS within the IRIDE program, by the Università di Udine PRID ENCASE project, and by GNCS-INdAM project Metodi Formali per la Verifica e la Sintesi di Sistemi Discreti e Ibridi.

References

1. G. Alzetta, A. Marin, C. Piazza, and S. Rossi. Lumping-based equivalences in markovian automata: Algorithms and applications to product-form analyses. *Information and Computation*, 260:99–125, 2018.
2. A. Bortz and D. Boneh. Exposing private information by timing web applications. In *Proc. of the 16th International Conference on World Wide Web (WWW)*, pages 621–628. ACM, 2007.
3. D. Brumley and D. Boneh. Remote timing attacks are practical. *Comput. Netw.*, 48(5):701–716, 2005.
4. S. Crafa and S. Rossi. Controlling information release in the pi-calculus. *Information and Computation*, 205(8):1235–1273, 2007.
5. E. W. Felten and M. A. Schneider. Timing attacks on web privacy. In *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS)*, pages 25–32. ACM, 2000.
6. R. Focardi and R. Gorrieri. Classification of Security Properties (Part I: Information Flow). In R. Focardi and R. Gorrieri, editors, *Proc. of Foundations of Security Analysis and Design (FOSAD'01)*, volume 2171 of *LNCS*, pages 331–396. Springer-Verlag, 2001.
7. H. Gao, C. Bodei, P. Degano, and H.R. Nielson. A formal analysis for capturing replay attacks in cryptographic protocols. In *Advances in Computer Science - ASIAN 2007. Computer and Network Security, 12th Asian Computing Science Conference*, pages 150–165, 2007.
8. J. A. Goguen and J. Meseguer. Security Policy and Security Models. In *Proc. of the 1982 Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, 1982.

9. J. Hillston. *A Compositional Approach to Performance Modelling.* Cambridge Press, 1996.
10. J. Hillston, A. Marin, C. Piazza, and S. Rossi. Contextual lumpability. In *Proc. of Valuetools 2013 Conf.*, pages 194–203. ACM Press, 2013.
11. J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* D. Van Nostrand Company, Inc., 1960.
12. A. Marin and S. Rossi. On the relations between Markov chain lumpability and reversibility. *Acta Informatica*, 54(5):447–485, 2017.
13. Andrea Marin and Sabina Rossi. On the relations between lumpability and reversibility. In *Proc. of MASCOTS 2014*, pages 427–432, 2014.
14. J. McLean. A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions. In *Proc. of the IEEE Symposium on Security and Privacy (SSP'94)*, pages 79–93. IEEE Computer Society Press, 1994.
15. P.Y.A. Ryan and S. Schneider. Process Algebra and Non-Interference. *Journal of Computer Security*, 9(1/2):75–103, 2001.
16. A. Sabelfeld and A. C. Myers. Language-Based Information-Flow Security. *IEEE Journal on Selected Areas in Communication*, 21(1):5–19, 2003.
17. G. Smith and D. M. Volpano. Secure Information Flow in a Multi-threaded Imperative Language. In *Proc. of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'98)*, pages 355–364. ACM Press, 1998.
18. D. Sutherland. A Model of Information. In *Proc. of the 9th National Computer Security Conference*, pages 175–183, 1986.
19. A. Valmari and G. Franceschinis. Simple O(m logn) Time Markov Chain Lumping. In *Proc. of Int. Conf. TACAS*, volume 6015, pages 38–52, Paphos, Cyprus, 2010. Springer Verlag.