A Classification of Time and/or Probability Dependent Security Properties

Ruggero Lanotte [1]

Dipartimento di Scienze della Cultura, Politiche e dell'Informazione,
Università dell'Insubria,
Via Valleggio 11, 22100 Como, Italy

Andrea Maggiolo-Schettini [2] Angelo Troina [3]

Dipartimento di Informatica, Università di Pisa,
Largo B. Pontecorvo 3, 56127 Pisa, Italy

Abstract

In multilevel systems it is important to avoid unwanted indirect information flow from higher levels to lower levels, namely the so called *covert channels*. Initial studies of information flow analysis were performed by abstracting away from time and probability. It is already known that systems that are considered to be secure may turn out to be insecure when time or probability are considered. Recently, work has been done in order to consider also aspects either of time or of probability, but not both. In this paper we propose a general framework, based on Probabilistic Timed Automata, where both *probabilistic* and *timing covert channels* can be studied. We define a Non-Interference security property that allows one to express information flow in a timed and probabilistic setting, and we compare the property with analogous properties defined in settings where either time or probability or none of them are taken into account. This allows to classify properties depending on their discerning power.

1 Introduction

In a multilevel system every agent is confined in a bounded security level; information can flow from a certain agent to another agent only if the level of the former is lower than the level of the latter. Access rules can be imposed by the system in order to control direct unwanted transmission from higher levels

[1] Email: ruggero.lanotte@uninsubria.it
[2] Email: maggiolo@di.unipi.it
[3] Email: troina@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

to lower levels; however, it could be possible to transmit information indirectly by using system side effects. Usually, this kind of indirect transmissions, called *covert channels*, do not violate the access rules imposed by the system.

The existence of covert channels has led to the more general approach of *information flow security*, which aims at controlling the way information may flow among different entities. The idea is to try to directly control the whole flow of information, rather than only the direct communication among agents. In [12] the authors introduce the notion of *Non-Interference*, stating, intuitively, that low level agents should not be able to deduce anything about the activity of high level agents. By imposing some information flow rules, it is possible to control direct and indirect leakages, as both of them give rise to unwanted information flows.

In the literature, there are many different definitions of security based on the information flow idea, and each is formulated in some system model (see, e.g., [12,20,13,10,11,9,5,1]). Most of the properties considered are based on analysis of information flow that does not take into consideration aspects of time or probability, and therefore they are not useful to check the existence of probabilistic or timing covert channels. To overcome this, a significant work has been done in order to extend the study by considering either time (see, e.g., [11,9,5]) or probability (see, e.g., [13,1,8]).

This has required the use of descriptive means for systems which allow expressing time and probability. Timed Automata have been introduced by Alur and Dill [3] as an extension of ω-Automata to describe real-time systems. Timed Automata are equipped with variables measuring time, called *clocks*. Transitions are guarded by *clock constraints*, which compare the value of a clock with some constant, and by *reset updates*, which reset a clock to the initial value 0. Extensions with probability have been proposed (e.g. in [2,6,15,16]). In this paper we are interested in a general framework where both probabilistic and timing covert channels can be studied. For the description of systems we introduce a particular class of Probabilistic Timed Automata (PTAs) well-suited for the analysis of information flow security properties.

The framework of PTAs allows specification of timed systems showing a probabilistic behavior in an intuitive and succinct way. Therefore, within the framework of PTAs, where time and probabilities are taken into consideration, the modeler can describe, in the same specification, different aspects of a system, and analyze on a single model real-time properties, performance and reliability properties (by using classical model checking techniques), and information flow security properties useful to detect both probabilistic and timing covert channels.

In Section 2 we present our model of Probabilistic Timed Automata, and we recall the definitions of Probabilistic Automata, Timed Automata and Nondeterministic Systems. We define bisimulation equivalences and operations for all these models. In Section 3 we define the Non-Interference security property

in a probabilistic timed setting described by Probabilistic Timed Automata. The concept of Non-Interference was proposed originally in a purely nondeterministic setting [12,20,10]. We show here that this concept, together with the analogous concepts for Probabilistic Automata and Timed Automata, can be expressed in a unique framework where both probability and time are considered. Time and probability allow discovering that systems that in a nondeterministic setting are considered to be secure, are instead insecure. Time only and probability only give incomparable discerning powers, while having both time and probability gives a discerning power greater than the ones given by each of them. In Section 4 we briefly discuss a property, *Non Deducibility on Composition*, which, at any of the levels of description considered, is stronger then Non-Interference. A classification of Non Deducibility on Composition properties analogous to the one for Non-Interference could be given.

2 Formalisms

We recall the definition of Probabilistic Timed Automata, operations and bisimulation for these automata, which we proposed in [17]. This model is inspired by models of Probabilistic Timed Automata in the literature (see, as examples, [6,15,2]). Probabilistic Automata are defined as a particular case. We recall also the definitions of Timed Automata ([3]) and of Nondeterministic Systems.

Let us assume a set X of positive real variables called *clocks*. A *valuation* over X is a mapping $v : X \to \mathbb{R}^{\geq 0}$ assigning real values to clocks. For a valuation v and a time value $t \in \mathbb{R}^{\geq 0}$, let $v+t$ denote the valuation such that $(v+t)(x) = v(x) + t$, for each clock $x \in X$.

The set of *constraints* over X, denoted $\Phi(X)$, is defined by the following grammar, where ϕ ranges over $\Phi(X)$, $x \in X$, $c \in \mathbb{Q}$ and $\sim \in \{<, \leq, =, \neq, >, \geq\}$:

$$\phi ::= x \sim c \,|\, \phi \wedge \phi \,|\, \neg\phi \,|\, \phi \vee \phi \,|\, true$$

We write $v \models \phi$ when *the valuation v satisfies the constraint ϕ*. Formally, $v \models x \sim c$ iff $v(x) \sim c$, $v \models \phi_1 \wedge \phi_2$ iff $v \models \phi_1$ and $v \models \phi_2$, $v \models \neg\phi$ iff $v \not\models \phi$, $v \models \phi_1 \vee \phi_2$ iff $v \models \phi_1$ or $v \models \phi_2$, and $v \models true$.

Let $B \subseteq X$; with $v[B]$ we denote the valuation resulting after resetting all clocks in B. More precisely, $v[B](x) = 0$ if $x \in B$, $v[B](x) = v(x)$, otherwise. Finally, with $\mathbf{0}$ we denote the valuation with all clocks reset to 0, namely $\mathbf{0}(x) = 0$ for all $x \in X$.

Definition 2.1 A Probabilistic Timed Automaton (PTA) is a tuple $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$, where:

- Σ is a finite alphabet of actions.
- X is a finite set of positive real variables called clocks.
- Q is a finite set of states and $q_0 \in Q$ is the initial state.

- $\delta \subseteq Q \times \Sigma \cup \{\tau\} \times \Phi(X) \times 2^X \times Q$ is a finite set of transitions. The symbol τ represents the silent or internal move. For a state q, we denote with $start(q)$ the set of transitions with q as source state, i.e. the set $\{(q_1, a, \phi, B, q_2) \in \delta \,|\, q_1 = q\}$.
- $Inv : Q \to \Phi(X)$ is a function assigning a constraint $\phi \in \Phi(X)$ (called *state invariant*) to each state in Q.
- $\Pi = \{\pi_1, \ldots, \pi_n\}$ is a finite set of probability distributions as functions $\pi_i : \delta \to [0, 1]$, for any $i = 1, \ldots, n$, where $\pi_i(e)$ is the probability of performing the transition e. We require that $\sum_{e \in start(q)} \pi_i(e) \in \{0, 1\}$ for any i and q.

Examples of PTAs with $|\Pi| = 1$ are in Figures 1, 2 and 3.

A *configuration* of A is a pair (q, v), where $q \in Q$ is a state of A, and v is a valuation over X. The initial configuration of A is represented by $(q_0, \mathbf{0})$ and the set of all the configurations of A is denoted with $\mathcal{S}_A$.

There is a discrete *transition step* from a configuration $s_i = (q_i, v_i)$ to a configuration $s_j = (q_j, v_j)$ through action $(a, \pi) \in (\Sigma \cup \{\tau\}) \times \Pi$, written $s_i \xrightarrow{(a,\pi)} s_j$, if there is a transition $e = (q_i, a, \phi, B, q_j) \in \delta$ such that $\pi(e) > 0$, $v_i \models \phi \wedge Inv(q_i)$, $v_j = v_i[B]$ and $v_j \models Inv(q_j)$.

There is a continuous *timed step* from a configuration $s_i = (q_i, v_i)$ to a configuration $s_j = (q_j, v_j)$ through time $t \in \mathbb{R}^{>0}$, written $s_i \xrightarrow{t} s_j$, if $q_j = q_i$, $v_j = (v_i + t)$ and $\forall t' \in [0, t]\ v_i + t' \models Inv(q_i)$.

Given a configuration $s = (q_i, v_i)$, $Adm(s) = \{(q_i, a, \phi, B, q) \in \delta \mid v_i \models \phi\}$ is the set of transitions executable by an automaton from configuration s; a transition in $Adm(s)$ is said to be *enabled* in s. Given two configurations $s_i = (q_i, v_i)$, $s_j = (q_j, v_j)$ and $(a, \pi) \in (\Sigma \cup \{\tau\}) \times \Pi$, $Adm(s_i, (a, \pi), s_j) = \{e = (q_i, a, \phi, B, q_j) \in \delta \mid \pi(e) > 0 \ \wedge \ v_i \models \phi \ \wedge \ v_j = v_i[B]\}$ is the set of transitions leading from configuration s_i to configuration s_j through a transition step labeled with (a, π). A configuration $s = (q_i, v_i)$ is *terminal* iff $Adm(s') = \emptyset$ for all $s' = (q_i, v_i + t)$ where $t \in \mathbb{R}^{\geq 0}$; S_T denotes the set of terminal configurations.

For configurations s_i, s_j, and $\alpha \in ((\Sigma \cup \{\tau\}) \times \Pi) \cup \mathbb{R}^{>0}$, the probability $P(s_i, \alpha, s_j)$ of reaching configuration s_j from configuration s_i through a step labeled with α, is defined as

$$P(s_i, \alpha, s_j) = \begin{cases} \frac{\sum_{e \in Adm(s_i, \alpha, s_j)} \pi(e)}{\sum_{e \in Adm(s_i)} \pi(e)} & \text{if } s_i \xrightarrow{\alpha} s_j,\ \alpha = (a, \pi) \in (\Sigma \cup \{\tau\}) \times \Pi \\ 1 & \text{if } s_i \xrightarrow{\alpha} s_j,\ \alpha \in \mathbb{R}^{>0} \\ 0 & \text{if } s_i \not\xrightarrow{\alpha} s_j \end{cases}$$

The probability of executing a transition step from a configuration s is chosen according to the values returned by the function π among all the transitions enabled in s, while the probability of executing a timed step la-

beled with $t \in \mathbb{R}^{>0}$ is set to the value 1. Intuitively, an automaton chooses non-deterministically a distribution π for executing a transition step (selected probabilistically among all the transitions enabled in s), or lets time pass performing a timed step.

An *execution fragment* starting from s_0 is a finite sequence of timed and transition steps $\sigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots \xrightarrow{\alpha_k} s_k$, where $s_0, s_1, \ldots, s_k \in \mathcal{S}_A$ and $\alpha_i \in ((\Sigma \cup \{\tau\}) \times \Pi) \cup \mathbb{R}^{>0}$. $ExecFrag$ is the set of execution fragments and $ExecFrag(s)$ is the set of execution fragments starting from s. We define $last(\sigma) = s_k$ and $|\sigma| = k$. The execution fragment σ is called *maximal* iff $last(\sigma) \in S_T$.

For any $j < k$, σ^j is the sequence of steps $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_j} s_j$.

If $|\sigma| = 0$ we put $P(\sigma) = 1$, else, if $|\sigma| = k \geq 1$, we define $P(\sigma) = P(s_0, \alpha_1, s_1) \cdot \ldots \cdot P(s_{k-1}, \alpha_k, s_k)$.

An *execution* is either a maximal execution fragment or an infinite sequence $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$, where $s_0, s_1 \ldots \in \mathcal{S}_A$ and $\alpha_1, \alpha_2, \ldots \in (\Sigma \cup \{\tau\} \times \Pi) \cup \mathbb{R}^{>0}$. We denote with $Exec$ the set of executions and with $Exec(s)$ the set of executions starting from s. Finally, let $\sigma\uparrow$ denote the set of executions σ' such that $\sigma \leq_{prefix} \sigma'$, where *prefix* is the usual prefix relation over sequences.

Executions and execution fragments of a PTA arise by resolving both the nondeterministic and the probabilistic choices [15]. To resolve the nondeterministic choices of a PTA, we introduce *schedulers* of PTAs.

A *scheduler* of a PTA $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$ is a function F from $ExecFrag$ to $\Pi \cup \mathbb{R}^{>0}$. With $\mathcal{F}_A$ we denote the set of schedulers of A. Given a scheduler $F \in \mathcal{F}_A$ and an execution fragment σ, we assume that F is defined for σ iff $\exists s \in \mathcal{S}_A$ and $a \in \Sigma \cup \{\tau\}$ such that $last(\sigma) \xrightarrow{F(\sigma)} s$ if $F(\sigma) \in \mathbb{R}^{>0}$ or $last(\sigma) \xrightarrow{(a,F(\sigma))} s$ if $F(\sigma) \in \Pi$.

For a scheduler $F \in \mathcal{F}_A$ we define $ExecFrag^F$ (resp. $Exec^F$) as the set of execution fragments (resp. executions) $\sigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$ of A such that, for any $0 < i < |\sigma|$:

- if $\alpha_i \in \mathbb{R}^{>0}$, then $F(\sigma^{i-1}) = \alpha_i$;
- if $\alpha_i = (a, \pi)$, then $F(\sigma^{i-1}) = \pi$.

A scheduler should also respect the *nonZeno* condition of divergent times. Formally we have that for any infinite sequence $\sigma = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots$ in $Exec^F$ the sum $\sum_{\alpha_i \in \mathbb{R}^{>0}} \alpha_i$ diverges.

Assuming the basic notions of probability theory (see e.g. [14]) we define the probability space on the executions starting in a given configuration $s \in \mathcal{S}_A$ as follows. Given a scheduler F, let $Exec^F(s)$ be the set of executions starting in s, $ExecFrag^F(s)$ be the set of execution fragments starting in s, and $\Sigma^F_{Field}(s)$ be the smallest sigma field on $Exec^F(s)$ that contains the basic cylinders $\sigma\uparrow$, where $\sigma \in ExecFrag^F(s)$. The probability measure $Prob^F$ is

the unique measure on $\Sigma^F_{Field}(s)$ such that $Prob^F(\sigma\uparrow) = P(\sigma)$.

Let A be a PTA, $F \in \mathcal{F}_A$, $\hat{\alpha}$ stand for a if $\alpha = (a, \pi)$ or $\alpha = a \in \mathbb{R}^{>0}$ and for ε (the empty string) if $\alpha = (\tau, \pi)$, $s \in \mathcal{S}_A$ and $\mathcal{C} \subseteq \mathcal{S}_A$. Given the set $Exec^F(\tau^*\hat{\alpha}, \mathcal{C})$ of executions that lead to a configuration in $\mathcal{C}$ via a sequence belonging to the set of sequences $\tau^*\hat{\alpha}$, we define $Exec^F(s, \tau^*\hat{\alpha}, \mathcal{C}) = Exec^F(\tau^*\hat{\alpha}, \mathcal{C}) \cap Exec^F(s)$. Given a scheduler F, we define the probability $Prob^F_A(s, \tau^*\hat{\alpha}, \mathcal{C}) = Prob^F_A(Exec^F(s, \tau^*\hat{\alpha}, \mathcal{C}))$.

Bisimilarity is widely accepted as the finest extensional behavioral equivalence one would want to impose on systems, and it may be used to verify a property of a system by assessing the bisimilarity of the system considered with a system one knows to enjoy the property.

The *bisimulation* of a system by another system is based on the idea of mutual step-by-step simulation. Intuitively, two systems A and A' are bisimilar, if whenever one of the two systems executes a certain action and reaches a configuration s, the other system is able to simulate this single step by executing the same action and reaching a configuration s' which is again bisimilar to s. A *weak bisimulation* is a bisimulation that does not take into account τ (internal) moves. Hence, whenever a system simulates an action of the other system, it can also execute some internal τ actions before and after the execution of that action. A *branching bisimulation* is, instead, a weak bisimulation where τ moves are allowed only before the execution of the action to simulate.

In order to abstract away from τ moves, Milner [22] introduces the notion of observable step, which consists of a single *visible* action α preceded and followed by an arbitrary number (including zero) of internal moves. Such moves are described by a *weak* transition relation $\Longrightarrow$, defined as $\stackrel{\alpha}{\Longrightarrow} = (\stackrel{\tau}{\longrightarrow})^* \stackrel{\alpha}{\longrightarrow} (\stackrel{\tau}{\longrightarrow})^*$, where $\longrightarrow$ is the classical strong relation, and $\stackrel{\tau}{\Longrightarrow} = (\stackrel{\tau}{\longrightarrow})^*$. It is worth noting that with such a definition a weak internal transition $\stackrel{\tau}{\Longrightarrow}$ is possible even without performing any internal action.

For the definition of weak bisimulation in the fully probabilistic setting, Baier and Hermanns [4] replace Milner's weak internal transitions $s \stackrel{\tau}{\Longrightarrow} s'$ by the probability $Prob(s, \tau^*, s')$ of reaching configuration s' from s via internal moves. Similarly, for visible actions α, Baier and Hermanns define $\stackrel{\alpha}{\Longrightarrow}$ by means of the probability $Prob(s, \tau^*\alpha, s')$. The probabilistic model we have chosen for PTAs is that of fully probabilistic systems. In such a model, as proved in [4], the two relations of weak bisimulation and branching bisimulation do coincide.

Definition 2.2 Let $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$ be a PTA. A *weak bisimulation* on A is an equivalence relation $\mathcal{R}$ on $\mathcal{S}_A$ such that, for all $(s, s') \in \mathcal{R}$, $\mathcal{C} \in \mathcal{S}_A/\mathcal{R}$ and schedulers $F \in \mathcal{F}_A$, there exists a scheduler $F' \in \mathcal{F}_A$ such that

$$Prob^F_A(s, \tau^*\alpha, \mathcal{C}) = Prob^{F'}_A(s', \tau^*\alpha, \mathcal{C}) \qquad \forall \alpha \in \Sigma \cup \{\tau\} \cup \mathbb{R}^{>0}$$

and vice versa.

Two configurations s, s' are called *weakly bisimilar* on A (denoted $s \approx_A s'$) iff $(s, s') \in \mathcal{R}$ for some weak bisimulation $\mathcal{R}$.
Two PTAs $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$ and $A' = (\Sigma', X', Q', q'_0, \delta', Inv', \Pi')$ such that $Q \cap Q' = \emptyset$, $X \cap X' = \emptyset$ and $\Pi \cap \Pi' = \emptyset$ are called weakly bisimilar (denoted by $A \approx A'$) if, given the PTA $\hat{A} = (\Sigma \cup \Sigma', X \cup X', Q \cup Q', q_0, \delta \cup \delta', \hat{Inv}, \Pi \cup \Pi')$ with

$$\hat{Inv}(q) = \begin{cases} Inv(q) & if\ q \in Q \\ Inv'(q) & if\ q \in Q' \end{cases}$$

and with $\mathcal{F}_{\hat{A}} = \mathcal{F}_A \cup \mathcal{F}_{A'}$, it holds $(q_0, \mathbf{0}) \approx_{\hat{A}} (q'_0, \mathbf{0})$, where the valuation $\mathbf{0}$ is defined over all clocks of the set $X \cup X'$.

Proposition 2.3 *It is decidable whether two PTAs are weakly bisimilar.*

Proof. It is easy to modify the algorithm given in [18] to deal with state invariants and with a finite set of probability functions instead of a single probability function.

We define operations of *restriction* and *hiding* on PTAs.
We assume a PTA $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$ and a set $L \subseteq \Sigma$ of actions.

Definition 2.4 The *restriction* of a PTA A with respect to the set of actions L is $A \setminus L = (\Sigma, X, Q, q_0, \delta', Inv, \Pi')$, where:

- $\delta' = \{(q, a, \phi, B, q') \in \delta \mid a \notin L\}$.
- $\pi' \in \Pi'$ iff $\pi \in \Pi$ where, for all $e = (q, a, \phi, B, q') \in \delta'$, $\pi'(e) = \frac{\pi(e)}{\sum_{e' \in \delta' \cap start(q)} \pi(e')}$.

The second condition is assumed in order to normalize the probability of each transition according to the ones remaining after the restriction. Thanks to this rule the condition $\sum_{e \in start(q)} \pi'(e) \in \{0, 1\}$ continues to be true for each state q of $A \setminus L$.

Definition 2.5 The *hiding* of a transition $e = (q, a, \phi, B, q')$ with respect to the set of actions L (written e/L) is defined as:

$$e/L = \begin{cases} e & if\ a \notin L \\ (q, \tau, \phi, B, q') & if\ a \in L \end{cases}$$

The hiding of a PTA A with respect to the set of actions L is given by $A/L = (\Sigma, X, Q, q_0, \delta', Inv, \Pi')$, where $\delta' = \{e/L \mid e \in \delta\}$, and $\Pi' = \{\pi' | \exists \pi \in \Pi. \forall e' \in \delta'\ \pi'(e') = \sum_{e \in \delta : e/L = e'} \pi(e)\}$.

Proposition 2.6 *Given a PTA A, $A \setminus L$ and A/L are PTAs for all $L \subseteq \Sigma$.*

We introduce Probabilistic Automata as a subcase of PTAs.

Definition 2.7 A *Probabilistic Automaton* (PA) is a PTA $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$, where $X = \emptyset$, $\phi = true$ for every $e = (q, a, \phi, B, q') \in \delta$ and $Inv(q) = true$ for every state $q \in Q$.

As $X = \emptyset$, there are no valuations of clocks, and therefore a configuration reduces to a state. Transitions and state invariants of a PA may have as a constraint only the condition *true*. Moreover, since for PA we abstract from time, we assume that for each execution σ of a PA there is no scheduler $F \in \mathcal{F}_A$ such that $F(\sigma) \in \mathbb{R}^{>0}$.

We recall the definitions of clock equivalence [3]. Clock equivalence is a finite index equivalence relation permitting to group sets of evaluations and to have decidability results.

Let A be a PTA; with C_A we denote the greatest constant that appears in A.

Let us consider the equivalence relation $\sim$ over clock valuations containing precisely the pairs (v, v') such that:

- for each clock x, either $\lfloor v(x) \rfloor = \lfloor v'(x) \rfloor$, or both $v(x)$ and $v'(x)$ are greater than C_A, with C_A the largest integer appearing in clock constraints over x;
- for each pair of clocks x and y with $v(x) \leq C_A$ and $v(y) \leq C_A$ it holds that $fract(v(x)) \leq fract(v(y))$ iff $fract(v'(x)) \leq fract(v'(y))$ (where $fract(\cdot)$ is the fractional part);
- for each clock x with $v(x) \leq C_A$, $fract(v(x)) = 0$ iff $fract(v'(x)) = 0$.

As proved in [3], $v \sim v'$ implies that, for any $\phi \in \Phi(X)$ with constants less or equal than C_A, $v \models \phi$ iff $v' \models \phi$.

With $[v]$ we denote the equivalence class $\{v' \mid v \approx v'\}$. The set of equivalence classes $V = \{[v] \mid v \text{ is a valuation}\}$ is finite, and with $|V|$ we denote its cardinality.

Given a PTA $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$, we call $untime(A)$ the PA obtained as the *region automaton* of A, with probability functions chosen according to Π. Intuitively, the region automaton (see [3]) is obtained by considering timed regions as states, where a timed region is a pair (q, ϕ), where $q \in Q$ and $\phi \in \Phi(X)$, containing every configuration (q, v_i) such that $v_i \models \phi$. Note that in the region automaton there is a step between regions r and r' with symbol (a, π) if and only if there is an admissible run $s \xrightarrow{t} s'' \xrightarrow{(a,\pi)} s'$ of the PTA such that $t \in \mathbb{R}^{>0}$ and where $s \in r$ and $s' \in r'$. More precisely, $untime(A) = (\Sigma \cup \{\lambda\}, \emptyset, Q \times [V], (q_0, [v_0]), \delta', Inv', \Pi')$ where:

- $((q, [v]), \lambda, true, \emptyset, (q', [v'])) \in \delta'$ iff $q = q'$, $v' = v + t$ for some time $t \in \mathbb{R}^{>0}$ and $v + t' \models Inv(q)\ \forall t' \in [0, t]$;
- $((q, [v]), a, true, \emptyset, (q', [v'])) \in \delta'$ iff $(q, a, \phi, B, q') \in \delta$, $v \models \phi \wedge Inv(q)$, $v' = v[B]$ and $v' \models Inv(q')$;
- $Inv'(q, [v]) = true\ \ \forall (q, [v]) \in Q \times [V]$;

- $\pi \in \Pi'$ iff either $\pi(e) = 1$ for some $e = ((q,[v]),\lambda,true,\emptyset,(q',[v']))$, or, there exists $\pi' \in \Pi$ such that, for all $e = ((q,[v]),a,true,\emptyset,(q',[v'])) \in \delta'$, $\pi(e) = \sum_{e' \in S} \pi'(e')$ where $S = \{(q,a,\phi,B,q') \in \delta \mid v \models \phi,\ \ v' = v[B]\}$.

We recall the definition of Timed Automata ([3]).

Definition 2.8 A *Timed Automaton* (TA) is a tuple $A = (\Sigma, X, Q, q_0, \delta, Inv)$, where Σ, X, Q, q_0, δ and Inv are defined as in Definition 2.1.

As for PTAs, a *configuration* of A is a pair (q,v), where $q \in Q$ is a state of A, and v is a valuation over X. The initial configuration of A is represented by $(q_0, \mathbf{0})$ and the set of all the configurations of A is denoted with $\mathcal{S}_A$.

There is a discrete *transition step* from a configuration $s_i = (q_i, v_i)$ to a configuration $s_j = (q_j, v_j)$ through action $a \in \Sigma \cup \{\tau\}$, written $s_i \xrightarrow{a} s_j$, if there is a transition $e = (q_i, a, \phi, B, q_j) \in \delta$ such that $v_i \models \phi$, and $v_j = v_i[B]$.

Continuous timed steps are as for PTAs.

Definition 2.9 Let $A = (\Sigma, X, Q, q_0, \delta, Inv)$ be a TA. A *weak bisimulation* on A is an equivalence relation $\mathcal{R} \subseteq \mathcal{S}_\mathcal{A} \times \mathcal{S}_\mathcal{A}$ such that for all $(s,r) \in \mathcal{R}$ it holds that $\forall \alpha \in \Sigma \cup \{\tau\} \cup \mathbb{R}^{>0}$:

- if $s \xrightarrow{\alpha} s'$, then there exists r' such that $r \stackrel{\alpha}{\Longrightarrow} r'$ and $(s', r') \in \mathcal{R}$;
- conversely, if $r \xrightarrow{\alpha} r'$, then there exists s' such that $s \stackrel{\alpha}{\Longrightarrow} s'$ and $(s', r') \in \mathcal{R}$.

Two configurations s, r are called *weakly bisimilar* on A (denoted $s \approx_A r$) iff $(s,r) \in \mathcal{R}$ for some weak bisimulation $\mathcal{R}$.
Two TAs $A = (\Sigma, X, Q, q_0, \delta, Inv)$ and $A' = (\Sigma', X', Q', q'_0, \delta', Inv')$ such that $Q \cap Q' = \emptyset$ and $X \cap X' = \emptyset$ are called weakly bisimilar (denoted by $A \approx A'$) if, given the TA $\hat{A} = (\Sigma \cup \Sigma', X \cup X', Q \cup Q', q_0, \delta \cup \delta', \hat{Inv})$, it holds $(q_0, \mathbf{0}) \approx_{\hat{A}} (q'_0, \mathbf{0})$, where $\hat{Inv}$ is defined as in Definition 2.2.

Proposition 2.10 *It is decidable whether two TAs are weakly bisimilar.*

Proof. See [19] and [7].

We define operations of *restriction* and *hiding* on TAs.

We assume a TA $A = (\Sigma, X, Q, q_0, \delta, Inv)$ and a set $L \subseteq \Sigma$ of actions.

Definition 2.11 The *restriction* of a TA A with respect to the set of actions L is $A \setminus L = (\Sigma, X, Q, q_0, \delta', Inv)$, where $\delta' = \{(q,a,\phi,B,q') \in \delta \mid a \notin L\}$.

Definition 2.12 The *hiding* of a TA A with respect to the set of actions L is given by $A/L = (\Sigma, X, Q, q_0, \delta', Inv)$, where $\delta' = \{e/L \mid e \in \delta\}$.

Proposition 2.13 *Given a TA A, $A \setminus L$ and A/L are TAs for all $L \subseteq \Sigma$.*

Given a PTA $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$, $unprob(A) = (\Sigma, X, Q, q_0, \delta, Inv)$ gives the TA obtained from A by removing Π.

We introduce Nondeterministic Systems as a subcase of TAs.

Definition 2.14 A *Nondeterministic System* (NS) is a TA $A = (\Sigma, X, Q, q_0, \delta, Inv)$, where $X = \emptyset$, $\phi = true$ for every $e = (q, a, \phi, B, q') \in \delta$ and $Inv(q) = true$ for every state $q \in Q$.

As $X = \emptyset$, there are no valuations of clocks, and therefore a configuration reduces to a state. Transitions and state invariants of a NS may have as a constraint only the condition *true*. The class of NSs coincides with the class of Nondeterministic Automata.

Operators and bisimulation defined for TAs reduce in the subcase of NSs to the analogous operators and bisimulation defined in [10].

Given a TA $A = (\Sigma, X, Q, q_0, \delta, Inv)$, we call $untime(A)$ the NS obtained as the *region automaton* of A, and by considering an empty set of clocks X. Note that in the region automaton there is a transition between regions r and r' if and only if there is an admissible run σ of the TA such that there is a step $s \xrightarrow{\alpha} s'$ in σ where $s \in r$ and $s' \in r'$.

We shall use the same terminology for operators and bisimulation in the different models when this does not give rise to ambiguity.

Lemma 2.15 *The following statements hold:*

(i) *Given PTAs A and A', $A \approx A' \Rightarrow unprob(A) \approx unprob(A')$*

(ii) *Given PTAs A and A', $A \approx A' \Rightarrow untime(A) \approx untime(A')$*

(iii) *Given PAs A and A', $A \approx A' \Rightarrow unprob(A) \approx unprob(A')$*

(iv) *Given TAs A and A', $A \approx A' \Rightarrow untime(A) \approx untime(A')$.*

Proof. For cases (i) and (iii), let us assume $A = (\Sigma, X, Q, q_0, \delta, Inv, \Pi)$, $A' = (\Sigma', X', Q', q'_0, \delta', Inv, \Pi')$ and $\hat{A}$ constructed as in Definition 2.2. Since $A \approx A'$ for a weak bisimulation $\mathcal{R}$, we have that for all $(s, r) \in \mathcal{R}$, $\mathcal{C} \in \mathcal{S}_{\hat{A}}/\mathcal{R}$ and schedulers F, there exists a scheduler F' such that $Prob^F_{\hat{A}}(s, \tau^*\alpha, \mathcal{C}) = Prob^{F'}_{\hat{A}}(r, \tau^*\alpha, \mathcal{C})$ $\forall \alpha \in \Sigma \cup \{\tau\} \cup \mathbb{R}^{>0}$. Now, if $Prob^F_{\hat{A}}(s, \alpha, s') > 0$ for some $s' \in \mathcal{C}$ there exists a configuration r' and a scheduler F' such that $Prob^{F'}_{\hat{A}}(r, \tau^*\alpha, r') = Prob^F_{\hat{A}}(s, \alpha, s') > 0$. Therefore if $s \xrightarrow{\alpha} s'$, then there exists r' such that $r \stackrel{\alpha}{\Longrightarrow} r'$ and, since s' and r' are in the same equivalence class, there exists also a bisimulation $\mathcal{R}'$ on $\mathcal{S}_{\hat{A}_{np}}$ such that $(s', r') \in \mathcal{R}'$, where $\hat{A}_{np}$ is constructed as in Definition 2.9 starting from $unprob(A)$ and $unprob(A')$. The same holds if we exchange the roles of s and r.

For cases (ii) and (iv), the implications hold by the construction of the region automaton. Actually, for each run of a PTA (or TA), there is an analogous run for the PA (or NS) obtained with $untime(A)$, with probabilities preserved by the normalizing operations. Weak bisimulations are, therefore, preserved. ∎

3 Security Properties

Given a system model with the basic operators of restriction and hiding, together with a notion of observational equivalence, it is easy to set up a framework for the analysis of information flow.

In all of the formalisms presented in Section 2, a finite alphabet Σ of visible actions is assumed. A multilevel system interacts with agents confined in different levels of clearance. In order to analyze the information flow between parties with different levels of confidentiality, the set of visible actions is partitioned into high level actions and low level actions. Formally, we assume the set of possible actions $\Sigma = \Sigma_H \cup \Sigma_L$, with $\Sigma_H \cap \Sigma_L = \emptyset$. In the following, with $l, l' \ldots$ and $h, h', \ldots$ we denote actions of Σ_L and Σ_H respectively. For simplicity, we specify only two-level systems; note, however, that this is not a real limitation, since it is always possible to deal with the case of more levels by iteratively grouping them in two clusters.

A low level agent is able to observe the execution of all the steps labeled with actions in Σ_L and all the timed steps. The basic idea of Non-Interference is that the high level does not interfere with the low level if the effects of high level communications are not visible by a low level agent. Finally, an important assumption when dealing with Non-Interference analysis is that a system is considered to be *secure* (no information flow can occur) if there is no interaction with high level agents (if high level actions are prevented).

We define Non-Interference properties, *Probabilistic Timed Non-Interference* (PTNI), *Probabilistic Non-Interference* (PNI), *Timed Non-Interference* (TNI) and *Nondeterministic Non-Interference* (NNI).

Definition 3.1 Given a system A in PTA (PA, TA, NS, resp.) A is PTNI (PNI, TNI, NNI, resp.)-secure if and only if $A/\Sigma_H \approx A \setminus \Sigma_H$.

In the definition above, $A \setminus \Sigma_H$ represents the isolated system, where all high level actions are prevented. As we have seen, such a system is considered secure due to the notion of Non-Interference. If the observational behavior of the isolated system is equal to the behavior of A/Σ_H, representing the system which communicates with high level agents in an invisible manner for the low agents point of view, A satisfies the security property.

Note that the PNI property is the BSPNI property defined in [1], the TNI property is an analogous of the tBSNNI property defined in [11], and NNI is the BSNNI property of [10].

The PTNI property, defined in an environment where both probability and time are studied, is able to detect information flow that may occur either due to the probabilistic behavior of the system or due to the time when an action occurs or due a combination of them.

Proposition 3.2 *It is decidable whether a PTA (PA, TA, NS, resp.) A satisfies the PTNI (PNI, TNI, NNI, resp.) property.*

Proof. The result derives directly by the decidability of weak bisimulation for all the models, and by the computable definitions of the operators of hiding and restriction. ∎

The security properties defined in the probabilistic and/or timed settings are conservative extensions of the security properties defined in the possibilistic and/or untimed settings.

Proposition 3.3 *The following implications hold:*

- $A \in PNI \Rightarrow unprob(A) \in NNI$
- $A \in TNI \Rightarrow untime(A) \in NNI$
- $A \in PTNI \Rightarrow unprob(A) \in TNI \wedge untime(A) \in PNI.$

Proof. The implications follow by the bisimulation based definitions of the security properties and by the conservativeness of the probabilistic weak bisimulation (see Lemma 2.15). ∎

The converse implications do not hold. The integration of probability and time adds new information that extends what is already known in the nondeterministic case. Therefore, systems considered to be secure in a purely possibilistic setting, may turn out to be insecure when considering aspects either of probability or of time. This is shown in Examples 3.4 and 3.5.

Example 3.4 In Figure 1 we show a case of probabilistic information flow presented in [1]. We assume $Inv(q_i) = true$ for every $i \in [0, 6]$. Abstracting away from probability, system A could be considered secure. In a purely possibilistic setting, in both systems $unprob(A)/\Sigma_H$ and $unprob(A) \setminus \Sigma_H$ a low level agent can observe the action l or the sequence ll' without further information about the execution of action h. It holds that $unprob(A)/\Sigma_H \approx unprob(A) \setminus \Sigma_H$ and, therefore, $unprob(A) \in NNI$ and $unprob(A) \in TNI$. In a probabilistic framework, given $\mathbf{p}+\mathbf{r}+\mathbf{q} = 1$, the high level action h interferes with the probability of observing either a single action l or the sequence ll'. Formally, in $A \setminus \Sigma_H$, a low level agent observes either the single action l with probability $\mathbf{p} + \mathbf{r}$ or the sequence ll' with probability $\mathbf{q}$. However, in A/Σ_H the single event l is observed with probability $\mathbf{p}$ and the sequence ll' with probability $\mathbf{r}+\mathbf{q}$. As a consequence we have $A/\Sigma_H \not\approx A \setminus \Sigma_H$, so that the PNI and the PTNI properties reveal the probabilistic covert channel.

Example 3.5 In Figure 2 we show a case of timing information flow. We assume $Inv(q_i) = true$ for $i \in [2, 3]$ and $Inv(q_i) = x \leq 5$ for $i \in [0, 1]$. Abstracting away from time, system A could be considered secure. In an untimed setting, in both systems $untime(A)/\Sigma_H$ and $untime(A) \setminus \Sigma_H$ a low level agent can observe only the action l executed with probability 1 without further information about the execution of action h. It holds that $untime(A)/\Sigma_H \approx untime(A) \setminus \Sigma_H$, and, therefore, $untime(A) \in PNI$. In a timed framework,

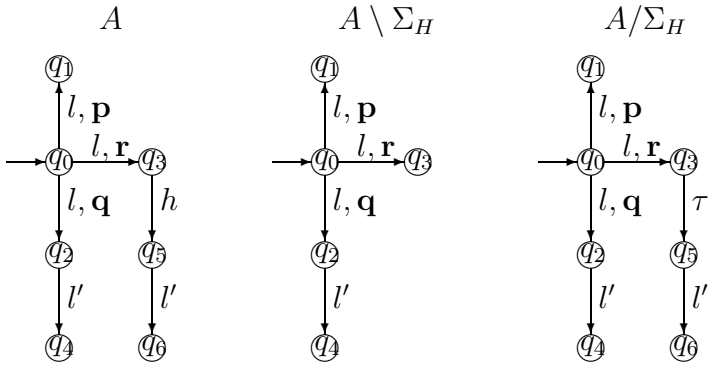

Fig. 1. A probabilistic covert channel.

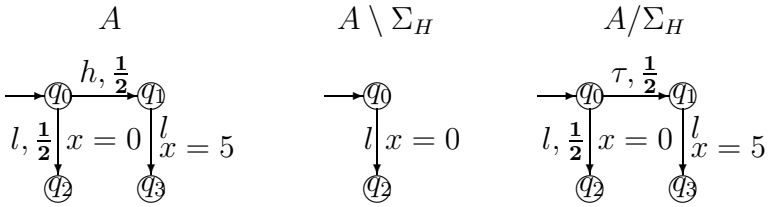

Fig. 2. A timing covert channel.

given a clock $x \in \mathbb{R}^{\geq 0}$, the high level action h interferes with the time of observing the action l. Formally, in $A \setminus \Sigma_H$, a low level agent observes the single action l executed immediately. However, in A/Σ_H the single event l could either be observed immediately or when the clock x reaches value 5. A low level agent, observing the event l when clock x has value 5 knows that action h has occurred. As a consequence, we have $A/\Sigma_H \not\approx A \setminus \Sigma_H$, so that the PTNI property reveals the timing covert channel. The same holds for $unprob(A)$; in this case the covert channel is detected by the TNI property.

For system A in Figure 1, $untime(A)$ is not PNI, but $unprob(A) \in TNI$. On the contrary, for system A in Figure 2, $unprob(A)$ is not TNI, but $untime(A) \in PNI$. This shows that the discerning powers of time and probability, as regards the Non-Interference property, are incomparable as stated in the next proposition.

Proposition 3.6 *The following implications hold:*

- $\exists\, PTA\ A:\ unprob(A) \in TNI \wedge untime(A) \notin PNI$
- $\exists\, PTA\ A\ : untime(A) \in PNI \wedge unprob(A) \notin TNI.$

If we can express both time and probability as in PTAs, we are able to describe systems exhibiting information flow that neither a formalism with only probability nor a formalism with only time can express. For such systems we are able to show that they are not PTNI, even if they are both PTI and TNI, and therefore we are able to reveal a new covert channel.

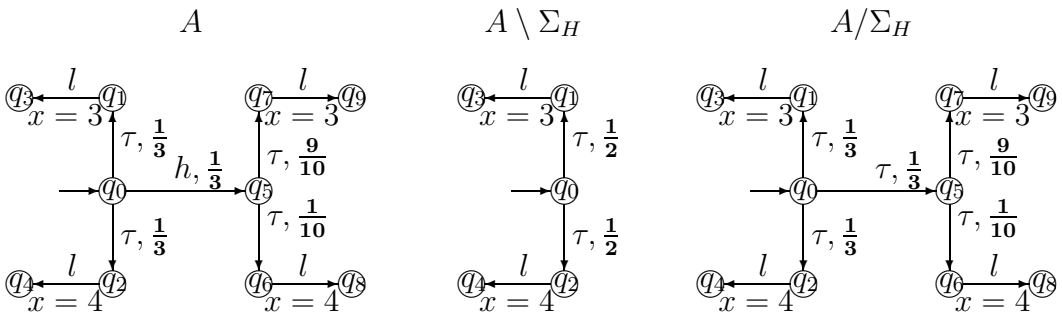

Fig. 3. A probabilistic timing covert channel.

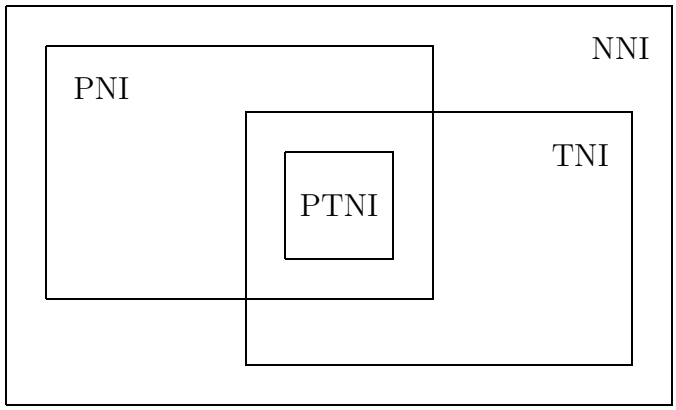

Fig. 4. Relations among Non-Interference security properties.

Proposition 3.7 $\exists A : A \notin PTNI \wedge unprob(A) \in TNI \wedge untime(A) \in PNI$.

Proof. Consider the PTA A in Figure 3. We assume $Inv(q_i) = true$ for $i \in \{3, 4, 8, 9\}$, $Inv(q_i) = x \leq 3$ for $i \in \{0, 1, 5, 7\}$ and $Inv(q_i) = x \leq 4$ for $i \in \{2, 6\}$. It is easy to see that $untime(A) \in PTI$. In both $untime(A)/\Sigma_H$ and $untime(A) \setminus \Sigma_H$, a low level agent observes the single event l taken with probability 1, and therefore $untime(A)/\Sigma_H \approx untime(A) \setminus \Sigma_H$. It is also easy to see that $unprob(A) \in TNI$. In both $unprob(A)/\Sigma_H$ and $unprob(A) \setminus \Sigma_H$, a low level agent could either observe the single event l taken when $x = 3$ or the event l taken when $x = 4$, and therefore $unprob(A)/\Sigma_H \approx unprob(A) \setminus \Sigma_H$. Finally, we show that A is not PTNI. In a probabilistic and timed framework, the high level action h interferes with the probability of observing the action l executed either when $x = 3$ or when $x = 4$. Formally, in $A \setminus \Sigma_H$, a low level agent observes the action l either when $x = 3$ or when $x = 4$ with probability $\frac{1}{2}$, respectively. However, in A/Σ_H the event l taken when $x = 3$ is observed with probability $\frac{19}{30}$, while the action l taken when $x = 4$ is observed with probability $\frac{11}{30}$. As a consequence, we have $A/\Sigma_H \not\approx A \setminus \Sigma_H$, so that the PTNI properties reveals the probabilistic timing covert channel. ∎

The diagram in Figure 4 summarizes our results.

4 Conclusions

Other security properties have been introduced in the literature in order to capture different behaviors of systems that have to be considered insecure. In [10] Focardi and Gorrieri promote the classification of a set of properties capturing the idea of information flow and Non-Interference. One of the most interesting and intuitive properties is the *Non Deducibility on Composition* (NDC), which states that a system A in isolation has not to be altered when considering all the potential interactions of A with the high level agents of the external environment. In [17] we defined for PTAs a concept of *parallel composition* and the property *Probabilistic Timed Non Deducibility on Composition* (PTNDC), and we have shown that $A \in \text{PTNDC} \Rightarrow A \in \text{PTNI}$. Analogous results could be proven for the properties of Non Deducibility on Composition and Non-Interference defined for PAs, TAs and NSs.

Moreover, implications similar to those of Proposition 3.3 could be proven for NDC security properties.

References

[1] A. Aldini, M. Bravetti, R. Gorrieri: *A Process-algebraic Approach for the Analysis of Probabilistic Non-interference*. Journal of Computer Security 12, 191–245, 2004.

[2] R. Alur, C. Courcoubetis, D. L. Dill: *Verifying Automata Specifications of Probabilistic Real-Time Systems*. Real-Time: Theory in Practice, Springer LNCS 600, 28–44, 1992.

[3] R. Alur, D. L. Dill: *A Theory of Timed Automata*. Theoretical Computer Science 126, 183–235, 1994.

[4] C. Baier, H. Hermanns: *Weak Bisimulation for Fully Probabilistic Processes*. Proc. of CAV'97, Springer LNCS 1254, 119–130, 1997.

[5] R. Barbuti, L. Tesei: *A Decidable Notion of Timed Non-interference*. Fundamenta Informaticae 54, 137–150, 2003.

[6] D. Beauquier: *On Probabilistic Timed Automata*. Theoretical Computer Science 292, 65–84, 2003.

[7] K. Cerans: *Decidability of Bisimulation Equivalences for Parallel Timer Processes*. Proc. of CAV'92, Springer LNCS 663, 302–315, 1992.

[8] A. Di Pierro, C. Hankin, H. Wiklicky: *Approximate Non-Interference*. Journal of Computer Security 12, 37-82, 2004.

[9] N. Evans, S. Schneider: *Analysing Time Dependent Security Properties in CSP Using PVS*. Proc. of Symp. on Research in Computer Security, Springer LNCS 1895, 222–237, 2000.

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

[11] R. Focardi, R. Gorrieri, F. Martinelli: *Information Flow Analysis in a Discrete-Time Process Algebra*. Proc. of 13th CSFW, IEEE CS Press, 170–184, 2000.

[12] J. A. Goguen, J. Meseguer: *Security Policy and Security Models*. Proc. of Symp. on Research in Security and Privacy, IEEE CS Press, 11–20, 1982.

[13] J. W. Gray III. *Toward a Mathematical Foundation for Information Flow Security*. Journal of Computer Security 1, 255–294, 1992.

[14] P. R. Halmos: *Measure Theory*. Springer-Verlag, 1950.

[15] M. Kwiatkowska, G. Norman, R. Segala, J. Sproston: *Automatic Verification of Real-time Systems with Discrete Probability Distribution*. Theoretical Computer Science 282, 101–150, 2002.

[16] M. Kwiatkowska, R. Norman, J. Sproston: *Symbolic Model Checking of Probabilistic Timed Automata Using Backwards Reachability*. Tech. rep. CSR-03-10, University of Birmingham, 2003.

[17] R. Lanotte, A. Maggiolo-Schettini, A. Troina: *Information Flow Analysis for Probabilistic Timed Automata*. Proc. of FAST'04, Springer IFIP 173, Toulouse, France, August 2004.

[18] R. Lanotte, A. Maggiolo-Schettini, A. Troina: *Weak Bisimulation for Probabilistic Timed Automata*.
`http://www.di.unipi.it/~lanotte/pub.html`.

[19] F. Laroussinie, K. G. Larsen, and C. Weise: *From Timed Automata to Logic - and Back*. Proc. of MFCS, Springer LNCS 969, 27–41, 1995.

[20] D. McCullough: *Noninterference and the Composability of Security Properties*. Proc. of Symp. on Research in Security and Privacy, IEEE CS Press, 177–186, 1988.

[21] J. K. Millen: *Hookup Security for Synchronous Machines*. Proc. of CSFW'90, IEEE CS Press, 84–90, 1990.

[22] R. Milner: *Communication and Concurrency*. Prentice Hall, 1989.