Fundamenta Informaticae 85 (2008) 1–15
IOS Press

Probabilistic Information Flow Security*

Damas P. Gruska†
Institute of Informatics, Comenius University,
Mlynska dolina, 842 48 Bratislava, Slovakia,
gruska@fmph.uniba.sk.

Abstract. A formal model for description of probabilistic timing attacks is presented and studied. It is based on a probabilistic timed process algebra, on observations (mappings which make visible only a part of system behavior) and on an information flow. The resulting security properties are studied and compared with other security concepts.

Keywords: probabilistic timed process algebras, timing attacks, information flow, opacity, security

1. Introduction

Several formulations of system security can be found in the literature. Many of them are based on a non-interference (see [15]) which assumes an absence of any information flow between private and public systems activities. More precisely, systems are considered to be secure if from observations of their public activities no information about private activities can be deduced. This approach has found many reformulations for different formalisms, computational models and nature or "quality" of observations. They try to capture some important aspects of systems behaviour with respect to possible attacks against systems security, often they are tailored to some types of attacks. Timing attacks have a particular position among attacks against systems security. They represent a powerful tool for "breaking" "unbreakable" systems, algorithms, protocols, etc. For example, by carefully measuring the amount of time required to perform private key operations, attackers may be able to find fixed Diffie-Hellman exponents, factor RSA keys, and break other cryptosystems (see [24]). This idea was developed in [7] where a timing attack against smart card implementation of RSA was conducted. In [22], a timing attack on the RC5 block encryption algorithm, in [28] the one against the popular SSH protocol and in [8] the one against web privacy are described.

*Work supported by the grant VEGA 1/3105/06.
†Address for correspondence: Institute of Informatics, Comenius University, Mlynska dolina, 842 48 Bratislava, Slovakia

The aim of this paper is to formalize opacity based passive and active (timing probabilistic) attacks by means of a particular probabilistic timed process algebra pTPA and by means of observations. The observations can hide some system actions (for example, internal actions, communications via encrypted channels, actions hidden by a firewall etc) but not elapsing of time.

In the literature several papers on formalizations of timing attacks can be found. Papers [10, 11, 14] express attacks in a framework of (timed) process algebras. In all these papers system actions are divided into private and public ones and it is required that there is not an interference between them. More precisely, in [10, 11] it is required that on a level of system traces one cannot distinguish between system which cannot perform private actions and system which can perform them but all of them are reduced to internal actions. In paper [14] a concept of public channels is elaborated. In the above mentioned papers also a slightly different approach to system security is presented - the system behaviour must be invariant with respect to composition with an attacker which can perform only private actions ([10, 11]) or with an attacker which can see only public communications ([14]).

In the presented approach actions are not divided to private and public ones on a system description level. Instead of this we work with special mappings (called observations) on a set of system actions. Since many of timing attacks described in the literature exploit also occurrences of "internal" actions we work also with this information what is not the case of the above mentioned papers. In this way we can describe timing attacks which could not be taken into account otherwise.

Moreover, since many attacks are based on statistical analyzes of system behaviour (see [24, 7, 22, 28]) instead of just "one single observation" (as it is done in for example [10, 11, 14, 16] or in [19] in case of process algebra for network communications [21]) we formulate information flow in terms of probabilities. So probabilistic version of Non-Information Flow property ($p\delta NIF$, for short) is presented and studied for passive and active attacks. Moreover, compositional properties of the presented security notions are presented and they are compared with other security properties - with Strong Nondeterministic Non-Interference, SNNI, for short (see [10]) and with persistent variant of Non-Deducibility on Composition, NDC for short, see in ([11]), as well.

The presented approach is different from the one which appeared in [26] and [1] where an information flow based security is studied in the framework of probabilistic Timed Automata and probabilistic process algebra, respectively. In the both papers security properties are based on bisimulation variants of SNNI (BSNNI). Actions are divided to public and private ones and the resulting security properties require the same probabilities for behavior containing and not containing private actions, while in the case of $p\delta NIF$ some difference ($\delta, \delta \in [0, 1]$) between them is allowed.

The paper is organized as follows. In Section 2 we describe the probabilistic timed process algebra which will be used as a basic formalism. In Section 3 we present and investigate the notion of probabilistic non-information flow property for the case of passive and active (timing) attacks.

2. Probabilistic Timed Process Algebra

In this section we define the Probabilistic Timed Process Algebra, pTPA for short. It will be done in two steps. First we define Timed Process Algebra (TPA) and later we will extend it with tools for expressing probabilities. TPA is based on Milner's CCS but the special time action t which expresses elapsing of (discrete) time is added. The presented language is a slight simplification of the Timed Security Process Algebra introduced in [10]. We omit the explicit idling operator ι used in tSPA and instead of this we

allow implicit idling of processes. Hence processes can perform either "enforced idling" by performing t actions which are explicitly expressed in their descriptions or "voluntary idling". But in the both cases internal communications have priority to action t in the case of the parallel operator. Moreover we do not divide actions into private and public ones as it is in tSPA. TPA differs also from the tCryptoSPA (see [14]). TPA does not use value passing and strictly preserves *time determinacy* in case of choice operator $+$ what is not the case of tCryptoSPA.

To define the language TPA, we first assume a set of atomic action symbols A not containing symbols τ and t, and such that for every $a \in A$ there exists $\overline{a} \in A$ and $\overline{\overline{a}} = a$. We define $Act = A \cup \{\tau\}, Actt = Act \cup \{t\}$. We assume that $a, b, \ldots$ range over A, $u, v, \ldots$ range over Act, and $x, y \ldots$ range over $Actt$. Assume the signature $\Sigma = \bigcup_{n\in\{0,1,2\}} \Sigma_n$, where

$$\begin{aligned}
\Sigma_0 &= \{Nil\} \\
\Sigma_1 &= \{x. \mid x \in A \cup \{t\}\} \cup \{[S] \mid S \text{ is a relabeling function}\} \\
&\quad \cup \{\backslash M \mid M \subseteq A\} \\
\Sigma_2 &= \{|, +\}
\end{aligned}$$

with the agreement to write unary action operators in prefix form, the unary operators $[S], \backslash M$ in postfix form, and the rest of operators in infix form. Relabeling functions, $S : Actt \to Actt$ are such that $\overline{S(a)} = S(\bar{a})$ for $a \in A, S(\tau) = \tau$ and $S(t) = t$.

The set of TPA terms over the signature Σ is defined by the following BNF notation:

$$P \ ::= \ X \ \mid \ op(P_1, P_2, \ldots P_n) \ \mid \ \mu X P$$

where $X \in Var$, Var is a set of process variables, $P, P_1, \ldots P_n$ are TPA terms, $\mu X-$ is the binding construct, $op \in \Sigma$.

The set of CCS terms consists of TPA terms without t action. We will use an usual definition of opened and closed terms where μX is the only binding operator. Closed terms which are t-guarded (each occurrence of X is within some subexpression $t.A$ i.e. between any two t actions only finitely many non timed actions can be performed) are called TPA processes. Note that Nil will be often omitted from processes descriptions and hence, for example, instead of $a.b.Nil$ we will write just $a.b$.

We give a structural operational semantics of terms by means of labeled transition systems. The set of terms represents a set of states, labels are actions from $Actt$. The transition relation $\to$ is a subset of TPA $\times$ $Actt$ $\times$ TPA. We write $P \xrightarrow{x} P'$ instead of $(P, x, P') \in \to$ and $P \not\xrightarrow{x}$ if there is no P' such that $P \xrightarrow{x} P'$. The meaning of the expression $P \xrightarrow{x} P'$ is that the term P can evolve to P' by performing action x, by $P \xrightarrow{x}$ we will denote that there exists a term P' such that $P \xrightarrow{x} P'$. We define the transition relation as the least relation satisfying the inference rules for CCS plus the following inference rules:

$$\frac{}{Nil \xrightarrow{t} Nil} \quad A1 \qquad \frac{}{u.P \xrightarrow{t} u.P} \quad A2$$

$$\frac{P \xrightarrow{t} P', Q \xrightarrow{t} Q', P \mid Q \not\xrightarrow{\tau}}{P \mid Q \xrightarrow{t} P' \mid Q'} \quad Pa1 \qquad \frac{P \xrightarrow{t} P', Q \xrightarrow{t} Q'}{P + Q \xrightarrow{t} P' + Q'} \quad S$$

Here we mention the rules that are new with respect to CCS. Axioms $A1, A2$ allow arbitrary idling. Concurrent processes can idle only if there is no possibility of an internal communication ($Pa1$). A run

of time is deterministic (S). Regarding behavioral relations we will work with the timed version of weak trace equivalence. Note that here we will use also a concept of observations which contain complete information which includes also τ actions and not just actions from A and t action as it is in [10]. For $s = x_1.x_2.\ldots.x_n, x_i \in Actt$ we write $P \xrightarrow{s}$ instead of $P \xrightarrow{x_1} \xrightarrow{x_2} \cdots \xrightarrow{x_n}$ and we say that s is a trace of P. The set of all traces of P will be denoted by $Tr(P)$. We will write $P \stackrel{x}{\Rightarrow} P'$ iff $P(\xrightarrow{\tau})^* \xrightarrow{x} (\xrightarrow{\tau})^* P'$ and $P \stackrel{s}{\Rightarrow}$ instead of $P \stackrel{x_1}{\Rightarrow} \stackrel{x_2}{\Rightarrow} \cdots \stackrel{x_n}{\Rightarrow}$. By ϵ we will denote the empty sequence of actions, by $Succ(P)$ we will denote the set of all successors of P and $Sort(P) = \{x | P \xrightarrow{s.x} \text{ for some } s \in Actt^\star\}$. If the set $Succ(P)$ is finite we say that P is finite state.

Definition 2.1. The set of weak timed traces of process P is defined as $Tr_w(P) = \{s \in (A \cup \{t\})^\star | \exists P'.P \stackrel{s}{\Rightarrow} P'\}$. Two process P and Q are weakly timed trace equivalent ($P \approx_w Q$) iff $Tr_w(P) = Tr_w(Q)$.

Now we add probabilities to TPA calculus. We will follow alternating model (the approach presented in [23]) which is neither reactive nor generative nor stratified (see [25]) but instead of that it based on separation of probabilistic and nondeterministic transitions and states. Probabilistic transitions are not associated with performing of actions but labeled only with probabilities. In so called probabilistic states a next transition is chosen according to probabilistic distribution. For example, process $a.(0.3.b.Nil \oplus 0.7.(a.Nil + b.Nil))$ can perform action a and after that it reaches the probabilistic state and from this state it can reach with probability 0.3 the state where only action b can be performed or with probability 0.7 it can reach the state where it can perform either a or b (see Fig. 1).

Note that resented approach slightly differs from the Calculus for Communicating with Time and Probabilities ([23]), where first probabilities are added to CCS and later time is added but without an explicit special time action.

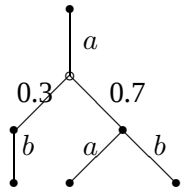

Figure 1. $a.(0.3.b.Nil \oplus 0.7.(a.Nil + b.Nil))$

Formally, to add probabilities to TPA calculus we introduce a new operator $\bigoplus_{i \in I} q_i.P_i, q_i$ being real numbers in $(0, 1]$ such that $\sum_{i \in I} q_i = 1$. Processes which can perform as the first action probabilistic transition will be called probabilistic processes or states (to stress that P is non-probabilistic process we will sometimes write P_N if necessary). Hence we require that all P_i processes in $\bigoplus_{i \in I} q_i.P_i$ and in $P_1 + P_2$ are non-probabilistic ones. By pTPA we will denote the set of all probabilistic and non-probabilist processes and all definitions and notations for TPA processes are extended for pTPA ones. We need new transition rules for pTPA processes. We mention only three rules which are significantly different from those ones for TPA.

$$\frac{}{P_N \xrightarrow{1} P_N} \quad A3 \qquad\qquad \frac{}{\bigoplus_{i \in I} q_i.P_i \xrightarrow{q_i} P_i} \quad A4$$

$$\frac{P \xrightarrow{q} P', Q \xrightarrow{r} Q'}{P \mid Q \xrightarrow{q.r} P' \mid Q'} \quad Pa2$$

For probabilistic choice we have the rule $A4$ and for a probabilistic transition of two processes running in parallel we have the rule $Pa2$. The technical rule $A3$ enables parallel run of probabilistic and non-probabilistic processes by allowing to non-probabilistic processes to perform $\xrightarrow{1}$ transition and hence the rule $Pa2$ could be applied.

Introducing probabilities to process algebras usually causes several technical complications. For example, an application of the restriction operator to probabilistic process may lead to unwanted deadlock states or to a situation when a sum of probabilities of all outgoing transitions is less than 1. A normalization is usually applied to overcome similar situations. We do not need to resolve such situations on the level of pTPA calculus since we will use only relative probabilities of sets of computations. To compute these probabilities normalization will be also exploited but only as the very last step.

3. Information Flow

In this section we will formalize a notion of passive and active timing attacks based on a non-probabilistic and later on probabilistic information flow between invisible (classified, private) and visible (public) system activities. We assume that an attacker is just an eavesdropper who can see a part of the system behavior and who tries to deduce from this some private information. In the case of timing attacks time of occurrences of observed events plays a crucial role, timing of actions represents a fundamental information.

To formalize the attacks we do not divide actions into public and private ones as it is done for non-interference properties, see for example in [14, 6] but instead of this we use a more general and more flexible concept of observations. This concept was recently exploited in [3] and [4] in a framework of Petri Nets and transition systems, respectively, where a concept of opacity is defined with the help of observations.

First we propose a concept of non-probabilistic Non-Information Flow (NIF) property which could be seen as a special case of the opacity property. The concept of opacity is rather strong and it is undecidable even for finite state processes. In the case of NIF property we restrict both power of observations and power of predicates over traces (see [4]). On the other side we get decidable security property for finite state systems. Note that NIF property is more general (see [16]) than Strong Nondeterministic Non-Interference property (see [10]).

Definition 3.1. An observation $\mathcal{O}$ is a mapping $\mathcal{O} : Actt \to Actt \cup \{\epsilon\}$ such that $\mathcal{O}(t) = t$ and for every $u \in Act, \mathcal{O}(u) \in \{u, \tau, \epsilon\}$.

An observation expresses what can an observer - eavesdropper see from a system behaviour. It cannot rename actions but only hide them completely ($\mathcal{O}(u) = \epsilon$) or indicate just a performance of some action but its name cannot be observed ($\mathcal{O}(u) = \tau$). Observations can be naturally generalized to sequences of actions. Let $s = x_1.x_2.\ldots.x_n, x_i \in Actt$ then $\mathcal{O}(s) = \mathcal{O}(x_1).\mathcal{O}(x_2).\ldots.\mathcal{O}(x_n)$. Since the observation expresses what an observer can see we will alternatively use both terms (observation - observer) with the same meaning. Note that in [4] observations defined in Definition 3.1 are called static, in contrast to dynamic or orwellian ones, for which an observation of an event might depend on previous events or on a (part) of a whole trace of actions, respectively. In that cases, an infinite memory is needed to compute observations.

3.1. Non-probabilistic Passive attacks

In general, systems respect the property of privacy if there is no leaking of private information, namely there is no *information flow* from the private level to the public level. This means that the secret behavior cannot influence the observable one, or, equivalently, no information on the observable behavior permits to infer information on the secret one. Moreover, in the case of timing attacks, timing of actions plays a crucial role. In the presented setting private actions are those that are hidden by observation $\mathcal{O}$, i.e. such actions a that $\mathcal{O}(a) \in \{\tau, \epsilon\}$ and for public actions we have $\mathcal{O}(a) = a$ i.e the observer can see them. Now we are ready to define Non-Information Flow property (NIF) for TPA processes. First some notations are needed. An occurrence of x action in a sequence of actions s we will indicate by $x \in s$ i.e. $x \in s$ iff $s = s_1.x.s_2$ for some $s_1, s_2 \in Actt^\star$ and for $\mathcal{S} \subseteq Actt$ we indicate $\mathcal{S} \cap s \neq \emptyset$ iff $x \in s$ for some $x \in \mathcal{S}$ otherwise we write $\mathcal{S} \cap s = \emptyset$. By $s|_{\mathcal{S}}$ we will denote string s restricted to the set of actions $\mathcal{S}$ i.e $s|_{Actt \setminus \mathcal{S}} = s$ if $\mathcal{S} \cap s = \emptyset$. Clearly, NIF property has to be parameterized by observation $\mathcal{O}$ and by a set of private actions $\mathcal{S}$ which occurrences are of interest. In other words, process P has NIF property if from its observation (given by $\mathcal{O}$) it cannot be deduced that some of given private actions ($\mathcal{S}$) were performed. We expect a consistency between $\mathcal{O}$ and $\mathcal{S}$ in the sense that the observation does not see actions from $\mathcal{S}$. The formal definition follows.

Definition 3.2. Let $\mathcal{O}$ be an observation and $\mathcal{S} \subseteq A$ such that $\mathcal{O}(a) \in \{\tau, \epsilon\}$ for $a \in \mathcal{S}$. We say that process P has $NIF^{\mathcal{S}}_{\mathcal{O}}$ property (we will denote this by $P \in NIF^{\mathcal{S}}_{\mathcal{O}}$) iff whenever $\mathcal{S} \cap s_1 \neq \emptyset$ for some $s_1 \in Tr(P)$ such that $\mathcal{O}(s_1) \neq \epsilon$ then there exists $s_2 \in Tr(P)$ such that $\mathcal{S} \cap s_2 = \emptyset$ and $\mathcal{O}(s_1) = \mathcal{O}(s_2)$.

Informally, process P has $NIF^{\mathcal{S}}_{\mathcal{O}}$ property if an observer with an observation given by $\mathcal{O}$ (note that (s)he can always see timing of actions) cannot deduce that process P has performed a sequence of actions which includes some private (secrete) actions from $\mathcal{S}$. In other words, $P \in NIF^{\mathcal{S}}_{\mathcal{O}}$ means that observer $\mathcal{O}$ cannot deduce anything about performance of actions from $\mathcal{S}$ and hence P is robust against corresponding attacks. By $NIF^{\mathcal{S}}_{\mathcal{O}}$ we will denote also the set of processes which have $NIF^{\mathcal{S}}_{\mathcal{O}}$ property. (Note that NIF property defined in [16] is slightly different from the presented one).

Example 3.1. Let $P = ((b.t.\bar{c} + a.\bar{c})|c) \setminus \{c\}$ and $\mathcal{O}(a) = \mathcal{O}(b) = \epsilon, \mathcal{O}(\tau) = \tau$. The observer given by $\mathcal{O}$ can detect occurrence of the action a but not b i.e. $P \in NIF^{\{b\}}_{\mathcal{O}}$but $P \notin NIF^{\{a\}}_{\mathcal{O}}$ since from observing just τ action (without any delay) it is clear that action a was performed. □

Example 3.2. Let $P = h.Nil$ and $\mathcal{O}(h) = \epsilon$. Clearly $P \in NIF^{\{\mathcal{S}\}}_{\mathcal{O}}$ for any $\{\mathcal{S}\}$ since P cannot perform any sequence of actions such that $\mathcal{O}(s) \neq \epsilon$. □

In many cases it seems to be sufficient to check occurrence of only one private action instead of a bigger set, i.e. the cases $\mathcal{S} = \{a\}$ for some $a \in A$. In these cases an observer tries to deduce whether confident action a was or was not performed. But even in this simplest possible case the NIF property is undecidable, but in general it is decidable for finite state processes. For the proof of the following theorem see [20].

Theorem 3.1. $NIF^{\{a\}}_{\mathcal{O}}$ property is undecidable but $NIF^{\mathcal{S}}_{\mathcal{O}}$ is decidable for finite state processes if $\mathcal{O}(x) \neq \epsilon$ for every $x \in Actt$.

3.2. Probabilistic Passive attacks

Now let us assume a process depicted on Fig. 2 which can perform only one action h (the one which is indicated). Let as assume that $\mathcal{O}$ is the identity function except that $\mathcal{O}(h) = \tau$. It can be checked that the process has $NIF_{\mathcal{O}}^{\{h\}}$ property since performing of action h is "hidden" by performing of τ action. On the other site if on observer can observe many times sequence $w.\tau.c$ it can be deduced with high probability that h has been performed. To formalize this kind of probabilistic information flow we have to reformulate the notion of NIF property. Roughly speaking, NIF property requires that every occurrence of a classified action is hidden by non-classified one. Probabilistic NIF property will require that relative probability of traces which contain that classified action differs by no more than by δ, where $0 \leq \delta \leq 1$, from probability of those traces which do not contain it but are observed in the same way.

To define probabilistic NIF formally, we need some preparatory work. Let P be a pTPA process and let $P \stackrel{x_1}{\rightarrow} P_1 \stackrel{x_2}{\rightarrow} P_2 \stackrel{x_3}{\rightarrow} \cdots \stackrel{x_n}{\rightarrow} P_n$, where $x_i \in Actt \cup (0,1]$ for every $i, 1 \leq i \leq n$. The sequence $P.x_1.P_1.x_2 \ldots x_n.P_n$ will be called a finite computational path of P (path, for short), its label is a subsequence of $x_1 \ldots . x_n$ consisting of those elements which belong to $Actt$ i.e. $label(P.x_1.P_1.x_2 \ldots x_n.P_n) = x_1 \ldots . x_n|_{Actt}$ and its probability is defined as a multiplication of all probabilities contained in it, i.e. $Prob(P.x_1.P_1.x_2 \ldots x_n.P_n) = 1 \times q_1 \times \cdots \times q_k$ where $x_1 \ldots . x_n|_{(0,1]} = q_1 \ldots g_k$. The multiset of finite pathes of P will be denoted by $Path(P)$. For example, the path $(0.5.a.Nil \oplus 0.5.a.Nil).0.5.(a.Nil).a.(Nil)$ is contained in $Path(0.5.a.Nil \oplus 0.5.a.Nil)$ two times. There exist a few techniques how to define this multiset. For example, in [27] a technique of schedulers are used to resolve the nondeterminism and in [13] all transitions are indexed and hence pathes can be distinguished by different indexes. In the former case, every scheduler defines (schedules) a particular computation path and hence two different schedulers determine different pathes, in the later case, the index records which transition was chosen in the case of several possibilities. The set of indexes for process P consists of sequences $i_1 \ldots i_k$ where $i_j \in \{0, \ldots, n\} \cup \{0, \ldots, n\} \times \{0, \ldots, n\}$ where n is the maximal cardinality of I for subterms of P of the form $\bigoplus_{i \in I} q_i.P_i$. An index records how a computation path of P could be derived, i.e. it records which process was chosen in case of several nondeterministic possibilities. If there is only one possible successor transitions are indexed by 1 (i.e. corresponding $i_l = 1$) If transition $P \stackrel{x}{\rightarrow} P'$ is indexed by k (i.e. corresponding $i_l = k$) then transition $P + Q \stackrel{x}{\rightarrow} P'$ is indexed by $k.1$ and transition $Q + P \stackrel{x}{\rightarrow} P'$ is indexed by $k.2$. If transition $P_i \stackrel{x}{\rightarrow} P'$ is indexed by k then transition $\bigoplus_{i \in I} q_i.P_i \stackrel{x}{\rightarrow} P'$ is indexed by $k.i$, and if transitions $P \stackrel{x}{\rightarrow} P'$ and $Q \stackrel{x}{\rightarrow} Q'$ are indexed by k and l, respectively, then transitions of $P|Q$ have indexes from $\{(k,0),(0,l),(k,l)\}$ depending on which transition rule for parallel composition was applied. Every index defines at most one path and the set of all indexes defines the multisets of pathes $Path(P)$. Let $C, C \subseteq Path(P)$ be a finite multiset. We define $Pr(C) = \sum_{c \in C} Prob(c)$ if $C \neq \emptyset$ and $Pr(\emptyset) = 0$.

Definition 3.3. Let $\mathcal{O}$ be an observation and $\mathcal{S} \subseteq A$ such that $\mathcal{O}(a) \in \{\tau, \epsilon\}$ for $a \in \mathcal{S}$. We say that process P has $p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ property (we will denote this by $P \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$) iff whenever $\mathcal{S} \cap s_1 \neq \emptyset$ for some $s_1 \in Tr(P), \mathcal{O}(s_1) \neq \epsilon$ then there exists $s_2 \in Tr(P)$ such that $\mathcal{S} \cap s_2 = \emptyset$ and $|p(T_1) - p(T_2)| \leq \delta$ for $T_1 = \{c | c \in Path(P), \mathcal{S} \cap label(c) \neq \emptyset, \mathcal{O}(label(c)) = \mathcal{O}(s_1)\}, T_2 = \{c | c \in Path(P), \mathcal{S} \cap label(c) = \emptyset, \mathcal{O}(label(c)) = \mathcal{O}(s_1)\}$ and $p(T_1) = Pr(T_1)/(Pr(T_1) + Pr(T_2)), p(T_1) = Pr(T_2)/(Pr(T_1) + Pr(T_2))$.

Roughly speaking, if there is trace s_1 which contains some classified action from $\mathcal{S}$ then the relative probability of the set of all traces observed exactly as s_1 and containing some classified actions (i.e.

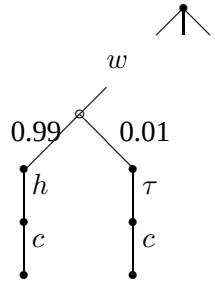

Figure 2. Process with $NIF^{\{h\}}_{\mathcal{O}} property$

$p(T_1)$) differs by no more than by δ from the probability of the set of all traces observed exactly as s_1 which do not contain any classified action (i.e. $p(T_2)$), i.e. $|p(T_1) - p(T_2)| \leq \delta$. Note that we normalize both $Pr(T_1)$ and $Pr(T_2)$ dividing them by $(Pr(T_1)+Pr(T_2))$ so that for the resulting relative probabilities we have ($0 \leq p(T_i) \leq 1$) and $p(T_1) + p(T_2) = 1$. Since we consider only t-guarded processes and elapsing of time is always observed, the multisets T_1, T_2 are finite.

Example 3.3. Let $P = a.(h.c.Nil + \tau.c.Nil)$ and $\mathcal{O}(a) = a, \mathcal{O}(c) = c, \mathcal{O}(h) = \mathcal{O}(\tau) = \tau$. The observer given by $\mathcal{O}$ cannot detect occurrence of the action h i.e. $P \in NIF^{\{h\}}_{\mathcal{O}}$. Now let as assume probabilistic version of P, $P' = a.(0.99.h.c.Nil \oplus 0.01.\tau.c.Nil)$ then $P \notin p\delta NIF^{\{h\}}_{\mathcal{O}}$ for $\delta < 0.98$. □

The probabilistic version of NIF property represents a stronger security property as its non-probabilistic variant as it is stated in the following Lemma.

Lemma 3.1. $p\delta NIF^{\mathcal{S}}_{\mathcal{O}} \subset NIF^{\mathcal{S}}_{\mathcal{O}}$ for any $\delta, 0 < \delta < 1$ and $\mathcal{O}, \mathcal{S}$ such that there exists $a \in Act$ such that $\mathcal{O}(a) \in \{\tau, \epsilon\}$.

Proof:
Let $P \in p\delta NIF^{\mathcal{S}}_{\mathcal{O}}$ form the first part of Definition 3.3 we have that $P \in NIF^{\mathcal{S}}_{\mathcal{O}}$ and hence $p\delta NIF^{\mathcal{S}}_{\mathcal{O}} \subseteq NIF^{\mathcal{S}}_{\mathcal{O}}$. To show that the inclusion is proper we can construct processes similar to the one described in Example 3.3. □

Moreover, for $\delta = 1$ probabilistic NIF and non-probabilistic NIF coincide and with smaller δ the resulting probabilistic NIF is stronger. The both properties are formulated by the following two Lemmas.

Lemma 3.2. $p1NIF^{\mathcal{S}}_{\mathcal{O}} = NIF^{\mathcal{S}}_{\mathcal{O}}$.

Proof:
It follows directly from Definition 3.3 that $p1NIF^{\mathcal{S}}_{\mathcal{O}} \subseteq NIF^{\mathcal{S}}_{\mathcal{O}}$. Let $P \in NIF^{\mathcal{S}}_{\mathcal{O}}$. For any s_1, $s_1 \in Tr(P)$ such that $\mathcal{S} \cap s_1 \neq \emptyset$ there exists $s_2, s_2 \in Tr(P)$ such that $\mathcal{S} \cap s_2 = \emptyset$ so we have that for the corresponding set T_2 (see Definition 3.3) we have $0 < p(T_2) \leq 1$. Hence $|p(T_1) - p(T_2)| \leq 1$ and then $P \in p1NIF^{\mathcal{S}}_{\mathcal{O}}$. □

Lemma 3.3. $p\delta_1 NIF^{\mathcal{S}}_{\mathcal{O}} \subset p\delta_2 NIF^{\mathcal{S}}_{\mathcal{O}}$ for $0 < \delta_1 < \delta_2 \leq 1$ and $\mathcal{O}, \mathcal{S}$ such that there exists $a \in Act$ such that $\mathcal{O}(a) \in \{\tau, \epsilon\}$.

Proof:
Directly from Definition 3.3 and by modification of process from Example 3.3 similarly as it was done in the proof of Lemma 3.1. □

For the probabilistic version of NIF we can formulate a similar property as that one which holds for NIF (see Theorem 3.1) and also its proof is similar.

Theorem 3.2. $p\delta NIF_{\mathcal{O}}^{\{a\}}$ property is undecidable but $p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ is decidable for finite state processes if $\mathcal{O}(x) \neq \epsilon$ for every $x \in Actt$.

Even if $p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ is decidable the corresponding algorithms are of exponential complexity. On the other side the property $p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ is compositional in the following sense.

Theorem 3.3. (Compositionality) Let $P, Q, P_i \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$, for $\imath \in I$. Then

$x.P \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ if $x \notin \mathcal{S}$

$P + Q \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$

$\bigoplus q_i.P_i \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ if it holds that whenever $s \in Tr(\bigoplus q_i.P_i)$, $\mathcal{S} \cap s \neq \emptyset$ than $s \in Tr(P_i)$ for every i

$P[f] \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ for any f such that $f(\mathcal{S}) \subseteq \mathcal{S}$

$P \setminus M \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ for any $M, M \subseteq \mathcal{S}$.

Proof:
We will prove the first three cases which are the most interesting ones.

(1) Let $P \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ and $\mathcal{S} \cap s_1 \neq \emptyset$ for some $s_1 \in Tr(x.P)$. Clearly $s_1 \neq x$ since $x \notin \mathcal{S}$. Hence let $s_1 = x.s_1'$, $\mathcal{S} \cap s_1' \neq \emptyset$ and $s_1' \in Tr(P)$. Since $P \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ there then $|p(T_1) - p(T_2)| \leq \delta$ for corresponding sets T_1 and T_2 (see Definition 3.3). But since $x \notin \mathcal{S}$ we have also $|p(T_1') - p(T_2')| \leq \delta$ for corresponding computation pathes of $x.P$ (clearly $p(T_1') = p(T_1)$ and $p(T_2') = p(T_2)$)).

(2) Let $P, Q \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ and $\mathcal{S} \cap s_1 \neq \emptyset$ for some $s_1 \in Tr(P+Q)$. Without lost of generality we can assume that $s_1 \in Tr(P)$. Since $P \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ there exists $s_2 \in Tr(P)$ such that $\mathcal{S} \cap s_2 = \emptyset$ and clearly $s_2 \in Tr(P + Q)$. To complete this part of the proof we need to show that $|p(T_1^{P+Q}) - p(T_2^{P+Q})| \leq \delta$. We have that $|p(T_1^{P+Q}) - p(T_2^{P+Q})| = |Pr(T_1^P \cup T_1^Q)/(Pr(T_1^P \cup T_1^Q) + Pr(T_2^P \cup T_2^Q)) - Pr(T_2^P \cup T_2^Q)/(Pr(T_1^P \cup T_1^Q) + Pr(T_2^P \cup T_2^Q))| = |Pr(T_1^P) + Pr(T_1^Q)/(Pr(T_1^P) + Pr(T_1^Q) + Pr(T_2^P) + Pr(T_2^Q)) - (Pr(T_2^P) + Pr(T_2^Q))/(Pr(T_1^P) + Pr(T_1^Q) + Pr(T_2^P) + Pr(T_2^Q))|$ where T_i^{P+Q}, T_i^Q, T_i^P are corresponding multisets of computational pathes. The rest of the proof follows from the inequations $|Pr(T_1^P) - Pr(T_2^P)| \leq \delta.(Pr(T_1^P) + Pr(T_2^P))$ and $|Pr(T_1^Q) - Pr(T_2^Q)| \leq \delta.(Pr(T_1^Q) + Pr(T_2^Q))$ given by the assumption that $P, Q \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$.

(3) Let $\mathcal{S} \cap s_1 \neq \emptyset$ for some $s_1 \in Tr(\bigoplus q_i.P_i)$ clearly there exists $s_2 \in Tr(\bigoplus q_i.P_i)$ such that $\mathcal{S} \cap s_2 = \emptyset$. By assumption we know that s_1 can be performed by all processes P_i and so we have $Pr(T_1^{\bigoplus q_i.P_i}) = q_1.Pr(T_1^{P_1}) + \cdots + q_k.Pr(T_1^{P_k})$ and $Pr(T_2^{\bigoplus q_i.P_i}) = q_1.Pr(T_2^{P_1}) + \cdots + q_k.Pr(T_2^{P_k})$. Clearly, we have $|p(T_1^{\bigoplus q_i.P_i}) - p(T_2^{\bigoplus q_i.P_i})| \leq \delta$ □

Note that in the previous theorem the requirement that $M \subseteq \mathcal{S}$ cannot be omitted in general. This follows from that fact that observations which completely hide also some actions not belonging to $\mathcal{S}$ are allowed. For example, $(0.5.h.Nil + 0.5.l.\tau.Nil) \setminus \{l\} \notin p\delta NIF_{\mathcal{O}}^{\{h\}}$ but $(0.5.h.Nil + 0.5.l.\tau.Nil) \in p\delta NIF_{\mathcal{O}}^{\{h\}}$ for $\mathcal{O}(h) = \mathcal{O}(\tau) = \tau, \mathcal{O}(l) = \epsilon$ and any δ from $[0, 1]$.

In [12] Focardi and Rossi defined a stronger (persistent) security property which allows to deal with possibly dynamic attackers and systems "being secure in every state". We can reformulate this concept for the probabilistic NIF property.

Definition 3.4. (Persistent Probabilistic NIF) $P \in Pp\delta\_NIF_{\mathcal{O}}^{\mathcal{S}}$ iff for every $P', P' \in Succ(P)$ we have $P' \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$.

It can be checked that $P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ is stronger than $p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ and hence we have the following property.

Theorem 3.4. $P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}} \subset p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ for any nonempty $\mathcal{S}$ and O being different from the identity function.

Proof:
Since $P \in Succ(P)$ we have $P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}} \subseteq p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$. Let $\mathcal{S} \neq \emptyset$ and $\mathcal{O}(h) = \tau$ (case when $\mathcal{O}(h) = \epsilon$ is similar) for some $h \in \mathcal{S}$ and let $a, b \notin \mathcal{S}$. Then for $P = \tau.(0.5.h.Nil \oplus 0.5.\tau.Nil)$ we have $P \in P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ but $h.Nil \notin p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ and hence $P_p\delta NIF_{\mathcal{O}}^{\mathcal{S}} \subset p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$n for any δ. □

In [16] NIF property is compared with Strong Nondeterministic Non-Interference (SNNI, for short). We recall its definition (see [10]). Suppose that all actions are divided in two groups, namely public (low level) actions L and private (high level) actions H i.e. $A = L \cup H, L \cap H = \emptyset$. Then process P has SNNI property if $P \setminus H$ behaves like P for which all high level actions are hidden for an observer. To express this hiding we introduce hiding= operator $P/M, M \subseteq A$, for which if $P \xrightarrow{a} P'$ then $P/M \xrightarrow{a} P'/M$ whenever $a \notin M \cup \bar{M}$ and $P/M \xrightarrow{\tau} P'/M$ whenever $a \in M \cup \bar{M}$. Formal definition of SNNI follows.

Definition 3.5. Let $P \in TPA$. Then $P \in SNNI$ iff $P \setminus H \approx_w P/H$.

Now we can compare $NIF_{\mathcal{O}}^{S}$ and $SNNI$ properties. Clearly, the former one is more general.

Theorem 3.5. $P \in SNNI$ iff $P \in NIF_{\mathcal{O}}^{H}$ for $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon$, $h \in H$ and $\mathcal{O}(x) = x$, $x \in L$.

Proof:
Part =>. Let $P \in SNNI$ and $P \xrightarrow{s_1}$ and $H \cap s_1 \neq \emptyset, \mathcal{O}(s_1) \neq \epsilon$. Hence we have that $P/H \xrightarrow{s_1'}$ where s_1' is equal to s_1 except that all actions from H are replaced by τ in s_1'. Since $P \in SNNI$ we have that $P \setminus H \approx_w P/H$ and so there exists s_2 such that $P \setminus H \xrightarrow{s_2}$ and $s_1'|_L = s_2|_L$ and $H \cap s_2 = \emptyset$. Clearly $P \xrightarrow{s_2}$ and hence $P \in NIF_{\mathcal{O}}^{H}$.

Part <=. Let $P \in NIF_{\mathcal{O}}^{H}$ and $P/H \xrightarrow{s_1'}$ i.e . $P \xrightarrow{s_1}$ for some s_1 such that $s_1|_L = s_1'|_L$.. Without loss of generality we can assume that $\mathcal{O}(s_1) \neq \epsilon$. Suppose that $H \cap s_1 \neq \emptyset$ (otherwise clearly $P/H \xrightarrow{s_1}$). Since $P \in NIF_{\mathcal{O}}^{H}$ we have that there exists $s_2, P \xrightarrow{s_2}$ such that $H \cap s_2 = \emptyset$ and $\mathcal{O}(s_1) = \mathcal{O}(s_2)$. i.e. $s_1|_L = s_2|_L$. From this we have that $P \setminus H \xrightarrow{s_2}$ i.e. every weak trace of P/H is the weak trace of $P \setminus H$. The converse is obvious and hence we have $P/H \approx_w P \setminus H$ i.e $P \in SNNI$. □

To compare probabilistic variant of NIF property with SNNI we exploit Lemma 3.1, Lemma 3.2, Lemma 3.3 and Theorem 3.5 and we get the following theorem which says that $p\delta NIF_{\mathcal{O}}^{\{H\}}$ is stronger security property as SNNI and that SNNI can be seen as a special case of $p\delta NIF_{\mathcal{O}}^{\{H\}}$ for $\delta = 1$.

Theorem 3.6. $P \in SNNI$ iff $P \in p\delta NIF_{\mathcal{O}}^{\{H\}}$ for some $\delta, 0 \leq \delta \leq 1$ and for $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon$, $h \in H$ and $\mathcal{O}(x) = x$, $x \in L$.

In [1] a bisimulation based variant of SNNI is defined for probabilistic calculus. We could define SNNI property for probabilistic calculus in a similar manner.

Definition 3.6. Let $P \in pTPA$. Then $P \in pSNNI$ iff for every $s \in Actt^{\star}$ we have $Pr(C_1) = Pr(C_2)$ where $C_1 = \{c | c \in Path(P \setminus H), label(c) = s\}$ and $C_2 = \{c | c \in Path(P/H), label(c) = s\}$.

A similar property as for SNNI with respect to probabilistic NIF property (see Theorem 3.5) holds also for pSNNI property and its proof is also similar.

Theorem 3.7. $P \in pSNNI$ iff $P \in p0NIF_{\mathcal{O}}^{\{H\}}$ for $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon$, $h \in H$ and $\mathcal{O}(x) = x$, $x \in L$.

We can define SNNI property up to probability δ similarly as it is done for NIF property and again the resulting property corresponds to $p\delta NIF_{\mathcal{O}}^{\{H\}}$ property for $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon$. .

Definition 3.7. Let $P \in pTPA$. Then $P \in p\delta SNNI$ iff for every $s \in Actt^{\star}$ we have $|(Pr(C_1) - Pr(C_2))/(Pr(C_1) + Pr(C_2))| \leq \delta$ where $C_1 = \{c | c \in Path(P \setminus H), label(c) = s\}$ and $C_2 = \{c | c \in Path(P/H), label(c) = s\}$ and $\delta \in [0, 1]$.

3.3. Active attacks

Till now we have considered the passive attacks. An intruder could only observe system behaviour. Now we will consider more powerful intruders which can employ some auxiliary processes to perform attacks. There is a natural restriction for such processes in the sense that they cannot perform public actions (see [9]). An alternative interpretation of this approach is such that it allows us to investigate security properties of processes as "contexts" for some private activities (expressed by those auxiliary processes). The context is secure if it does not permit information flow from "inside" to "outside". First we formulate the concept of so called active attacks (we will denote them by index a) in the framework of non-probabilistic NIF property.

Definition 3.8. (Active NIF) $P \in NIF_{a\mathcal{O}}^{\mathcal{S}}$ $(P\_NIF_{a\mathcal{O}}^{\mathcal{S}})$ iff $(P|A) \in NIF_{\mathcal{O}}^{\mathcal{S}}$ $(P\_NIF_{\mathcal{O}}^{\mathcal{S}})$ for every A such that $Sort(A) \subseteq \mathcal{S} \cup \{\tau, t\}$.

Active attacks are really more powerful than passive ones (see [16]).

Theorem 3.8. $NIF_{a\mathcal{O}}^{\mathcal{S}} \subset NIF_{\mathcal{O}}^{\mathcal{S}}$ and $P\_NIF_{a\mathcal{O}}^{\mathcal{S}} \subset P\_NIF_{\mathcal{O}}^{\mathcal{S}}$.

Again it can be proved (see [16]) that active NIF property is more general than another traces-based concept of active attacks called Non-Deducibility on Composition (see [11]). Now we we define probabilistic version of active NIF property.

Definition 3.9. (Active Probabilistic NIF) $P \in p\delta NIF_{a\mathcal{O}}^{\mathcal{S}}$ ($P\_p\delta NIF_{a\mathcal{O}}^{\mathcal{S}}$) iff $(P|A) \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ ($P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$) for every A such that $Sort(A) \subseteq \mathcal{S} \cup \{\tau, t\}$.

Active probabilistic attacks are really more powerful than passive ones.

Theorem 3.9. $p\delta NIF_{a\mathcal{O}}^{\mathcal{S}} \subset p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ and $P\_p\delta NIF_{a\mathcal{O}}^{\mathcal{S}} \subset P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$.

Proof:
Clearly $p\delta NIF_{a\mathcal{O}}^{\mathcal{S}} \subseteq p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ and $P\_p\delta NIF_{a\mathcal{O}}^{\mathcal{S}} \subseteq P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$. For the rest of the proof we construct processes P_1, P_2, A such that $P_1 \in p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ but $(P_1|A) \notin p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ and $P_2 \in P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ but $(P_2|A) \notin P\_p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$, respectively. For example, we can consider processes $P_1 = 0.5.h.l + 0.5.\tau.l$, $P_2 = \tau(0.5.h.l + 0.5.\tau.l)$ and $A = t.\bar{h}$ and let $\mathcal{O}(h) = \tau$. □

For the $p\delta NIF_{a\mathcal{O}}^{\mathcal{S}}$ we can formulate similar compositional theorem is the one which holds for $p\delta NIF_{\mathcal{O}}^{\mathcal{S}}$ (see Theorem 3.3). To compare active probabilistic NIF and active non-probabilistic NIF property we can formulate lemmas similar to Lemma 3.1, Lemma 3.2 and Lemma 3.3.

In [16] $P\_NIF_{\mathcal{O}}^{\mathcal{S}}$ is compared with persistent variant of Non-Deducibility on Composition (NDC for short, see in [11]). This property is based on the idea of checking the system against all high level potential interactions, representing every possible high level process i.e. a system is NDC if for every high level user A, the low level view of the behaviour of P is not modified (in terms of trace equivalence) by the presence of A. The idea of NDC can be formulated as follows.

Definition 3.10. (NDC) $P \in NDC$ iff for every $A, Sort(A) \subseteq H \cup \{\tau, t\}$

$$(P|A) \setminus H \approx_w P \setminus H$$

Similarly to Definition 3.4 we define persistent variant of NDC.

Definition 3.11. (Persistent NDC) $P \in P\_NDC$ iff for every $P', P' \in Succ(P)$ we have $P' \in NDC$.

The the proof of the following Theorem can be found in [16].

Theorem 3.10. $P \in P\_NDC$ iff $P \in NIF_{a\mathcal{O}}^{\mathcal{H}}$ for $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon$, for $h \in H$ and $\mathcal{O}(x) = x$ for $x \in L$.

Now, similarly to Theorem 3.6, we obtain a result which compares active probabilistic NIF property with persistent NDC property. More precisely, it says that $p\delta NIF_{a\mathcal{O}}^{\{H\}}$ is stronger security property as persistent NDC which can be seen as a special case of $p\delta NIF_{a\mathcal{O}}^{\{H\}}$ for $\delta = 1$.

Theorem 3.11. $P \in P\_NDC$ iff $P \in p\delta NIF_{a\mathcal{O}}^{\{H\}}$ for some $\delta, 0 \leq \delta \leq 1$ and for $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon$, $h \in H$ and $\mathcal{O}(x) = x$, $x \in L$.

3.4. Pure Probabilistic Timing attacks

Till now we have omitted a discussion on importance of time information. Probabilistic $p\delta NIF^{\mathcal{S}}_{\mathcal{O}}$ property says that there is no probabilistic information flow about occurrence of actions from $\mathcal{S}$ under observation $\mathcal{O}$. But in the case that there is an information flow we still cannot say whether this is due to time information contained in the process description or it is due to untimed part of the system behaviour. To distinguish these two cases let us consider untimed observations $\mathcal{O}_t$ for every observation $\mathcal{O}$ which differ from ordinary ones by ability to hide elapsing of time, i.e. $\mathcal{O}_t(t) = \epsilon, \mathcal{O}_t(x) = \mathcal{O}(x)$ for $x \neq t$. Now we can precisely define that system is open to pure timing attacks i.e. the attacks for which time information plays the crucial role.

Definition 3.12. We say that process P is open to pure probabilistic timing attacks under observation $\mathcal{O}$ to detect $\mathcal{S}$ iff $P \in p\delta NIF^{\mathcal{S}}_{\mathcal{O}_t}$ and $P \notin p\delta NIF^{\mathcal{S}}_{\mathcal{O}}$.

In other words system is open to pure timing attacks if there is not information flow only when timing information is not seen by an observer i.e. the observer cannot see time of action occurrences. Systems which are open to pure timing attacks might be considered to be safe if they are off-line or they are accessible only via slow networks.

Example 3.4. Let $P = 0.5.t.h.a.Nil \oplus 0.25.t.\tau.a.Nil \oplus 0.25.\tau.a.Nil$ and $\mathcal{O}(h) = \mathcal{O}(\tau) = \tau, \mathcal{O}(a) = a$. We have that $P \notin p\delta NIF^{\{h\}}_{\mathcal{O}}$ for $\delta < 1/3$ but $P \in NIF^{\{h\}}_{\mathcal{O}}$ and P is open to pure timing attacks. That means that this process is insecure only if we consider both probability and time. Otherwise it can be considered secure.

In practice it is easy to avoid pure timing attacks. Usually it is enough to put some random delays in critical sections. Actually some known pure timing attacks exploit "over-optimizations" of implementations of in general secure algorithms (see for example [24]). Another question is a time precision needed to perform a pure timing attack (or, on the other hand, how long should be the random delays protecting system security).

If an intruder cannot measure time with sufficient accuracy or only within a limited time window systems still might remain safe. In [17, 18] possibilities of an attacker which can observe systems only for a given limited time or can measure time elapsing between (two or more) actions only with some given precision are studied. The resulting security properties are more adequate if an attacker cannot measure time with absolute precision or cannot observe systems for an unlimited time.

4. Conclusions and further work

Timing attacks usually do not break system algorithms themselves but rather their bad, from security point of view, implementations. For example, such implementations, due to different optimizations, could result in non-probabilistic or probabilistic dependencies between time of computation and data to be processed, and as a consequence systems might become open to timing attacks. An attacker can deduce from time information also some information about private data, despite the fact that safe algorithms were used. Hence the importance of their study for privacy. In this paper we have presented a formal model which can express robustness of systems with respect to probabilistic timing attacks. This kind of attacks could not be modeled without probabilistic timed calculus and with ability exploit also

information on internal actions and hence now we can detect possibility of timing attacks which could not be detected otherwise. Note that probabilistic NIF properties for passive and active attacks are generalization of the Strong Nondeterministic Non-Interference property (see [10]) and Non-Deducibility on Composition property (see [11]), respectively.

Acknowledgement

The author thanks an anonymous referee who offered several insightful comments and helped to improve the paper.

References

[1] Aldini A., M. Bravetti and R. Gorrieri: A process-algebraic approach for the analysis of probabilistic noninterference, Journal of Computer Security archive Volume 12 , Issue 2, April, 2004.

[2] Bossi A., D. Macedonio, C. Piazza and S. Rossi. Information Flow in Secure Contexts. Journal of Computer Security, Volume 13, Number 3, 2005

[3] Bryans J., M. Koutny and P. Ryan: Modelling non-deducibility using Petri Nets. Proc. of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models, 2004.

[4] Bryans J., M. Koutny, L. Mazare and P. Ryan: Opacity Generalised to Transition Systems. In Proceedings of the Formal Aspects in Security and Trust, LNCS 3866, Springer, Berlin, 2006

[5] Bossi A., R. Focardi, C. Piazza and S. Rossi. Refinement Operators and Information Flow Security. Proc. of SEFM'03, IEEE Computer Society Press, 2003.

[6] Busi N. and R. Gorrieri: Positive Non-interference in Elementary and Trace Nets. Proc. of Application and Theory of Petri Nets 2004, LNCS 3099, Springer, Berlin, 2004.

[7] Dhem J.-F., F. Koeune, P.-A. Leroux, P. Mestre, J.-J. Quisquater and J.-L. Willems: A practical implementation of the timing attack. Proc. of the Third Working Conference on Smart Card Research and Advanced Applications (CARDIS 1998), LNCS 1820, Springer, Berlin, 1998.

[8] Felten, E.W., and M.A. Schneider: Timing attacks on web privacy. Proc. 7^{th} ACM Conference on Computer and Communications Security, 2000.

[9] Focardi, R. and R. Gorrieri: Classification of security properties. Part I: Information Flow. Foundations of Security Analysis and Design, LNCS 2171, Springer, Berlin, 2001.

[10] Focardi, R., R. Gorrieri, and F. Martinelli: Information flow analysis in a discrete-time process algebra. Proc. 13^{th} Computer Security Foundation Workshop, IEEE Computer Society Press, 2000.

[11] Focardi, R., R. Gorrieri, and F. Martinelli: Real-Time information flow analysis. IEEE Journal on Selected Areas in Communications 21 (2003).

[12] Focardi, R. and S. Rossi: Information flow security in Dynamic Contexts. Proc. of the IEEE Computer Security Foundations Workshop, 307-319, IEEE Computer Society Press, 2002.

[13] Glabbeek R. J. van, S. A. Smolka and B. Steffen: Reactive, Generative and Stratified Models of Probabilistic Processes Inf. Comput. 121(1): 59-80, 1995

[14] Gorrieri R. and F. Martinelli: A simple framework for real-time cryptographic protocol analysis with compositional proof rules. to appear at Science of Computer Programing.

[15] Goguen J.A. and J. Meseguer: Security Policies and Security Models. Proc. of IEEE Symposium on Security and Privacy, 1982.

[16] Gruska D.P.: Observation Based System Security. Fundamenta Informaticae, vol 79, Numbers 3-4, 2007

[17] Gruska D.P.: Information-Flow Attacks Based on Limited Observations. in Proc. of PSI'06, Springer Verlag, LNCS 4378, Berlin, 2007.

[18] Gruska D.P.: Information-Flow Security for Restricted Attackers. in Proc. of 8th International Symposium on Systems and Information Security, Sao Jose dos Campos, 2006

[19] D.P. Gruska, Network Information Flow, Fundamenta Informaticae, Volume 72, Numbers 1-3, pp 167-180, 2006

[20] Gruska D.P.: Information Flow in Timing Attacks. Proceedings CS&P'04, 2004.

[21] Gruska D.P. and A. Maggiolo-Schettini: Process algebra for network communication. Fundamenta Informaticae 45(2001).

[22] Handschuh H. and Howard M. Heys: A timing attack on RC5. Proc. Selected Areas in Cryptography, LNCS 1556, Springer, Berlin, 1999.

[23] Hansson, H. a B. Jonsson: A Calculus for Communicating Systems with Time and Probabilities. In Proceedings of 11th IEEE Real - Time Systems Symposium, Orlando, 1990.

[24] Kocher P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. Proc. Advances in Cryptology - CRYPTO'96, LNCS 1109, Springer, Berlin, 1996.

[25] López N. and Núñez: An Overview of Probabilistic Process Algebras and their Equivalences. In Validation of Stochastic Systems, LNCS 2925, Springer-Verlag, Berlin, 2004

[26] Lanotte R., A. Maggiolo-Schettini and A. Troina: A Classification of Time and/or Probability Dependent Security Properties. Electr. Notes Theor. Comput. Sci. 153(2): 177-193 (2006)

[27] Segala R. and N. Lynch: Probabilistic Simulations for Probabilistic Processes. Nord. J. Comput. 2(2): 250-273, 1995

[28] Song. D., D. Wagner, and X. Tian: *Timing analysis of Keystrokes and SSH timing attacks.* Pro.10th USENIX Security Symposium, 2001.