Fundamenta Informaticae 79 (2007) 1–12
IOS Press

Observation Based System 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 passive and active timing attacks is presented, studied and compared with other security concepts. It is based on a timed process algebra and on a concept of observations which make only a part of system behaviour visible. From this partial information which contains also timing of actions an intruder tries to deduce some private system activities.

Keywords: process algebras, timing attacks, information flow

1. Introduction

Several formulations of a notion of system security can be found in the literature. Many of them are based on a concept of non-interference (see [13.]) which assumes the 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 [21.]). This idea was developed

*Work supported by the grant VEGA 1/3105/06 and APVV-20-P04805.
†Address for correspondence: Institute of Informatics, Comenius University, Mlynska dolina, 842 48 Bratislava, Slovakia

in [6.] where a timing attack against smart card implementation of RSA was conducted. In [20.], a timing attack on the RC5 block encryption algorithm, in [23.] the one against the popular SSH protocol and in [7.] the one against web privacy are described.

Formal methods play a growing role not only in the design of (mainly critical) software applications and hardware components but also in checking their security properties. In practice, various formalisms are used. *Process algebras* are an important class of such formalisms. They successfully describe the behavior of systems as "communicating systems" and they usually abstract away many real properties of existing systems, such as duration and structure of actions, properties of communication networks, distribution of system components, and so on. If one wants to analyze systems with respect to timing attacks, a special process algebra which is enriched by time reasoning has to be used. The aim of this paper is to formalize a notion of passive and active (timing) attacks by means of a particular timed process algebra TPA and by a concept of observations. We assume that an intruder can observe a system (to be attacked) by means of the observations. These observations can hide some 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 [9.], [10.], [12.] 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 [9., 10.] it is required that on a level of system traces which do not contain internal actions 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 [12.] 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 ([9.], [10.]) or with an attacker which can see only public communications ([12.]).

In the presented approach actions are not divided to private and public ones on a system description level. Instead of this we work with a concept of observations. These are mappings on the set of actions which can hide some of actions but not elapsing of time. Since many of timing attacks described in the literature are based on observations of "internal" actions we work also with this information what is not the case of the above mentioned papers. In this way we can consider timing attacks which could not be taken into account otherwise. In this paper we continue with the work started in [17.]. Here Non-Information Flow property (NIF, for short) is presented for passive and active attacks. The resulting security concepts are compared with other concepts known in the literature. It is shown that Strong Nondeterministic Non-Interference (see [9.]) is a special case of NIF property for passive attacks and that Non-Deducibility on Composition (see [10.]) is a special case of NIF property for active attacks. Moreover, compositional properties of the presented security notions are presented.

The paper is organized as follows. In Section 2 we describe the timed process algebra which will be used as a basic formalism. In Section 3 we present and investigate the notion of non-information flow property (NIF) for the case of passive and active (timing) attacks. Moreover, we compare NIF properties with other security concepts known in the literature.

2. Timed Process Algebra

In this section we introduce the Timed Process Algebra, TPA for short. It is based on Milner's CCS (see [22.]) 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 [9.]. We omit the explicit idling operator ι used in tSPA and instead of this we use an alternative approach known in the literature and 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 [12.]) besides absence of value passing, by semantics of choice operator $+$ which in some cases abandons *time determinacy* which is strictly preserved in TPA.

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 are called 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 $P \xrightarrow{x} \!\!\!\!/$ 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 following inference rules:

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

$$\frac{}{Nil \xrightarrow{t} Nil} \quad A3 \qquad \frac{P \xrightarrow{u} P'}{P \mid Q \xrightarrow{u} P' \mid Q} \quad Pa1$$

$$\frac{P \xrightarrow{u} P'}{Q \mid P \xrightarrow{u} Q \mid P'} \quad Pa2 \qquad \frac{P \xrightarrow{a} P', Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \quad Pa3$$

$$\frac{P \xrightarrow{t} P', Q \xrightarrow{t} Q', P \mid Q \not\xrightarrow{\tau}}{P \mid Q \xrightarrow{t} P' \mid Q'} \quad Pa4 \qquad \frac{P \xrightarrow{u} P'}{P + Q \xrightarrow{u} P'} \quad S1$$

$$\frac{P \xrightarrow{u} P'}{Q + P \xrightarrow{u} P'} \quad S2 \qquad \frac{P \xrightarrow{t} P', Q \xrightarrow{t} Q'}{P + Q \xrightarrow{t} P' + Q'} \quad S3$$

$$\frac{P \xrightarrow{x} P'}{P \setminus M \xrightarrow{x} P' \setminus M}, (x, \overline{x} \notin M) \quad Res \qquad \frac{P[\mu X P/X] \xrightarrow{x} P'}{\mu X P \xrightarrow{x} P'} \quad Rec$$

$$\frac{P \xrightarrow{x} P'}{P[S] \xrightarrow{S(x)} P'[S]} \quad Rl$$

Here we mention the rules that are new with respect to CCS. Axioms $A2, A3$ allow arbitrary idling. Concurrent processes can idle only if there is no possibility of an internal communication ($Pa4$). A run of time is deterministic ($S3$). In the definition of the labeled transition system we have used negative premises (see $Pa4$). In general this may lead to problems, for example with consistency of the defined system. We avoid these dangers by making derivations of τ independent of derivations of t. For an explanation and details see [14.]. 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 [9.]. 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 $Tr_w(P) = Tr_w(Q)$.

3. Information Flow

In this section we will formalize a notion of passive and active timing attacks based on an information flow between invisible (private) and visible (public) system activities. We assume that an attacker is just

an eavesdropper who can see a part of the system behaviour 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 [12., 5.] but instead of this we use more general and flexible concept of observations. This concept was recently exploited in [2.] and [3.] in a framework of Petri Nets and transition systems, respectively, where a concept of opacity is defined with the help of observations.

We propose a concept of 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 [3.]). On the other side we get decidable security property for finite state systems. Later we will discuss a relationship between NIF and Strong Nondeterministic Non-Interference properties. Non-inference properties can be seen as a special case of opacity as well, but by NIF we can model, moreover, for example, attacks which are based on observing encrypted messages.

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 [3.] 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 the whole trace, respectively. In that cases to compute observations an infinite memory is needed.

3.1. 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 $\mathcal{S} \cap s = \emptyset$. 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 $s_1 \in Tr(P)$ 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.

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.

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 [17.].

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 $x \in Actt$.

Even if $NIF^{\mathcal{S}}_{\mathcal{O}}$ is decidable the corresponding algorithms are of exponential complexity. On way how to overcome this disadvantage is a bottom-up design of processes. Hence compositionality of $NIF^{\mathcal{S}}_{\mathcal{O}}$ plays an important role. We have the following property.

Theorem 3.2. (Compositionality) Let $P, Q \in NIF^{\mathcal{S}}_{\mathcal{O}}$. Then

$x.P \in NIF^{\mathcal{S}}_{\mathcal{O}}$ if $x \notin \mathcal{S}$

$P + Q \in NIF^{\mathcal{S}}_{\mathcal{O}}$

$P|Q \in NIF^{\mathcal{S}}_{\mathcal{O}}$

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

$P \setminus M \in NIF^{\mathcal{S}}_{\mathcal{O}}$ for any $M, M \subseteq \mathcal{S}$.

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

(1) Let $P \in NIF^{\mathcal{S}}_{\mathcal{O}}$ 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 $P \in NIF^{\mathcal{S}}_{\mathcal{O}}$ there exists $s'_2 \in Tr(P)$ such that $\mathcal{S} \cap s'_2 = \emptyset$ and $\mathcal{O}(s'_1) = \mathcal{O}(s'_2)$. Hence for $s_2, s_2 = x.s'_2$ we have $s_2 \in Tr(x.P)$ such that $\mathcal{S} \cap s_2 = \emptyset$ and $\mathcal{O}(s_1) = \mathcal{O}(s_2)$ and so $x.P \in NIF^{\mathcal{S}}_{\mathcal{O}}$.

(2) Let $P, Q \in NIF^{\mathcal{S}}_{\mathcal{O}}$ 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 $P \in NIF^{\mathcal{S}}_{\mathcal{O}}$ there exists $s_2 \in Tr(P)$ such that $\mathcal{S} \cap s_2 = \emptyset$ and $\mathcal{O}(s_1) = \mathcal{O}(s_2)$. But since $s_2 \in Tr(P + Q)$ we have $P + Q \in NIF^{\mathcal{S}}_{\mathcal{O}}$.

(3) Let $P, Q \in NIF_{\mathcal{O}}^{\mathcal{S}}$ but $P|Q \notin NIF_{\mathcal{O}}^{\mathcal{S}}$. Let s_1 is the shortest trace of $P|Q$ such that $\mathcal{S} \cap s_1 \neq \emptyset$ and for every trace s_2 such that $\mathcal{O}(s_1) = \mathcal{O}(s_2)$ we have $\mathcal{S} \cap s_2 \neq \emptyset$. Since s_1 is the shortest trace clearly only its last element belong to $\mathcal{S}$. This element was performed either by P or by Q. By case analysis and structural induction it can be shown that this leads to a contradiction with the assumption that $P, Q \in NIF_{\mathcal{O}}^{\mathcal{S}}$.

□

To compare NIF property with Strong Nondeterministic Non-Interference (SNNI, for short) we recall its definition (see [9.]). 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 3.3. 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 3.3. $P \in SNNI$ iff $P \in NIF_{\mathcal{O}}^{H}$ for $\mathcal{O}(h) = \tau$, $h \in H$ and $\mathcal{O}(x) = x$, $x \notin H$.

Proof:

Let $P \in SNNI$ i.e $P \setminus H \approx_w P/H$ and let $H \cap s_1 \neq \emptyset$ for some $s_1 \in Tr(P)$ If there is not $s_2 \in Tr(P)$ such that $H \cap s_2 = \emptyset$ and $\mathcal{O}(s_1) = \mathcal{O}(s_2)$ then clearly it would not hold $P \setminus H \approx_w P/H$ since $s_1' \in Tr(P/H)$ but $s_1' \notin Tr(P \setminus H)$ for s_1' obtained from s_1 by replacing all H actions by τ action.

Let $P \in NIF_{\mathcal{O}}^{H}$ and let $s_1 \in Tr(P/H)$. The only interesting case is such that $H \cap s_1 \neq \emptyset$ but then since $P \in NIF_{\mathcal{O}}^{H}$ there exist $s_2 \in Tr(P)$ such that $\mathcal{S} \cap s_2 = \emptyset$ and $\mathcal{O}(s_1) = \mathcal{O}(s_2)$. But since $\mathcal{O}(h) = \tau$, $h \in H$ and $\mathcal{O}(x) = x$, $x \notin H$ we have $s_1 \in Tr(P \setminus H)$. The other inclusion ($Tr(P \setminus H) \subseteq Tr(P/H)$) is straightforward.

□

In [11.] 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 NIF property.

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

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

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

Proof:
Since $P \in Succ(P)$ we have $P\_NIF_{\mathcal{O}}^{\mathcal{S}} \subseteq NIF_{\mathcal{O}}^{\mathcal{S}}$. Let $\mathcal{S} \neq \emptyset$ and $\mathcal{O}(x) = \tau$ (case when $\mathcal{O}(x) = \epsilon$ is similar) for some $x \in \mathcal{S}$ and let $a, b \notin \mathcal{S}$. Then for $P = \tau.(x.Nil + \tau.Nil)$ we have $P \in P\_NIF_{\mathcal{O}}^{\mathcal{S}}$ but $x.Nil \notin NIF_{\mathcal{O}}^{\mathcal{S}}$ and hence $P\_NIF_{\mathcal{O}}^{\mathcal{S}} \subset NIF_{\mathcal{O}}^{\mathcal{S}}$.

□

3.2. Active attacks

Till now we have considered the attacks which were the passive ones. 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 [8.]). 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". We formulate the concept of so called active attacks (we will denote them by index a) in the framework of NIF property.

Definition 3.5. (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 $Sort(A) \subseteq \mathcal{SS} \cup \{\tau, t\}$.

Active attacks are really more powerful than passive ones.

Theorem 3.5. $NIF_{a\mathcal{O}}^{\mathcal{S}} \subset NIF_{\mathcal{O}}^{\mathcal{S}}$ and $P\_NIF_{a\mathcal{O}}^{\mathcal{S}} \subset P\_NIF_{\mathcal{O}}^{\mathcal{S}}$.

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

□

For the $NIF_{a\mathcal{O}}^{\mathcal{S}}$ we can formulate similar compositional theorem is the one which holds for $NIF_{\mathcal{O}}^{\mathcal{S}}$

Theorem 3.6. (Compositionality) Let $P, Q \in NIF_{a\mathcal{O}}^{\mathcal{S}}$. Then

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

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

$P|Q \in NIF_{a\mathcal{O}}^{\mathcal{S}}$

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

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

Proof:
Similar to the proof of Theorem 3.2.

□

In the literature we can find another traces-based concept of active attacks called Non-Deducibility on Composition (NDC for short, see in [10.]). It 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 3.6. (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 3.7. (Persistent NDC) $P \in P\_NDC$ iff for every $P', P' \in Succ(P)$ we have $P' \in NDC$.

To compare $P\_NIF^{\mathcal{S}}_{\mathcal{O}}$ and P_NDC properties we need some preparatory work. The definition of persistent NDC property contains two universal quantifications (over all possible intruders and over all possible successors). To avoid them we exploit an idea introduced by Bossi, Focardi, Piazza and Rossi (see [4.]). First we define a low level observation equivalence (with respect to relation $\asymp$) which relates processes indistinguishable from the low level point of view.

Definition 3.8. (Equivalence on Low Actions) Let $\asymp$ be an equivalence relation over processes. We say that two processes P and Q are $\asymp$-equivalent on low actions, denoted by $P \asymp^l Q$, if $P \setminus H \asymp Q \setminus H$.

Now we can recall a notion of generalized unwinding condition. Roughly speaking, it requires that each high level action can be "simulated" in such a way that it is impossible for a low level user to infer which high level actions have been performed. All high level actions are required to be simulated in a way which is transparent to a low level user.

Definition 3.9. (Generalized Unwinding) Let $\asymp$ be an equivalence relation and $\longmapsto$ be a binary relation on processes. The unwinding class $(\mathcal{W}, \asymp^l, \longmapsto)$ is defined as $(\mathcal{W}, \asymp^l, \longmapsto) = \{P \in TPA | \forall Q \in Succ(P)$ if $Q \xrightarrow{h} R$ then $\exists R'$ such that $Q \longmapsto R'$ and $R \asymp^l R'\}$.

Now we get an alternative formulation of the persistent NDC property.

Theorem 3.7. $P \in P\_NDC$ iff $P \in (\mathcal{W}, \approx_w, \hat{\stackrel{\tau}{\Rightarrow}})$.

Proof:
Let $P \in P\_NDC$ and $P' \in Succ(P)$. Clearly $P' \in NDC$. Hence for any A we have $(P'|A) \setminus H \approx_w P' \setminus H$. Let $P' \xrightarrow{h} P_1$. Then it is easy to see that we can choose $A = \bar{h}.Nil$ such that $(P'|A) \setminus H \xrightarrow{\tau} P_1 \setminus H$. Since $(P'|A) \setminus H \approx_w P' \setminus H, P' \hat{\stackrel{\tau}{\Rightarrow}} P_2 \setminus H$ and $P_1 \setminus H \approx_w P_2 \setminus H$. Let $P' \xrightarrow{h} P_1$ then $P' \hat{\stackrel{\tau}{\Rightarrow}} P_2$ and $P_1 \approx_w P_2$.

It can be checked by case analysis that $Tr((P'|A) \setminus H) = Tr(P' \setminus H)$ for any $P' \in Succ(P)$ and $Sort(A) \subseteq H \cup \{\tau, t\}$ if $P \in (\mathcal{W}, \approx_w, \hat{\stackrel{\tau}{\Rightarrow}})$ and hence $P \in P\_NDC$.

□

Now, using Theorem 3.7 we can prove the following relation between the persistent NDC and active NIF. From this theorem it is clear that also (see Theorem 3.3) in case of active attacks the presented notion of NIF property is more general and powerful. Or alternatively, persistent NDC property is just a special case of active NIF property.

Theorem 3.8. $P \in P\_NDC$ iff $P \in NIF_a{}^{\mathcal{H}}_{\mathcal{O}}$ for $\mathcal{O}(h) = \tau$, for $h \in H$ and $\mathcal{O}(x) = x$ for $x \notin H$.

Proof:
Let $P \in P\_NDC$ and let $\mathcal{S} \cap s_1 \neq \emptyset$ for some $s_1 \in Tr(P|A)$. Suppose that there is not $s_2 \in Tr(P|A)$ such that $\mathcal{S} \cap s_2 = \emptyset$ and $\mathcal{O}(s_1) = \mathcal{O}(s_2)$. But then by Theorem 3.7 we would have that $P|A \notin (\mathcal{W}, \approx_w, \stackrel{\hat{\tau}}{\Rightarrow})$ what is in contradiction with $P \in P\_NDC$.

Now let $P \in NIF_{a\mathcal{O}}^{\mathcal{H}}$ for $\mathcal{O}(h) = \tau$, for $h \in H$ and $\mathcal{O}(x) = x$ for $x \notin H$. It can be checked that $P|A \in (\mathcal{W}, \approx_w, \stackrel{\hat{\tau}}{\Rightarrow})$ and hence by Theorem 3.7 it holds $P \in P\_NDC$. □

3.3. Pure Timing attacks

Till now we have omitted a discussion on importance of time information. $NIF_{\mathcal{O}}^{\mathcal{S}}$ property says that there is no 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 3.10. We say that process P is open to pure timing attacks under observation $\mathcal{O}$ to detect $\mathcal{S}$ iff $P \in NIF_{\mathcal{O}_t}^{\mathcal{S}}$ and $P \notin NIF_{\mathcal{O}}^{\mathcal{S}}$.

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.

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 [21.]). 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 [15., 16.] 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 can “break” systems which are often considered to be “unbreakable”. More precisely, the 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 dependency 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 timing attacks. This kind of attacks could not be modeled without 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. Moreover, we can precisely distinguish between timing attacks and non-timing attacks. This approach enables us to formulate not only the question whether a system is robust with respect to timing attacks but also other questions: how precise must be the measure of time to perform a successful attack, how to modify the system in such a way that attacks will be not possible etc. The presented formalism is compared with other concepts described in the literature and it is shown that it is more general and stronger in the sense that it can describe attacks which are not captured by the other concepts.

Further study will concern on more efficient decision algorithms, modeling of more elaborated active time attacks where an attacker can implement some less restricted processes to the system to be attacked (for example in the style of Trojan horse) to deduce some private activities. To have better described system activities (particularly to be able to perform traffic analysis), we consider to use formalism which can express also some network properties in style of [18.]. This approach was used in [19.] to study Bisimulation-based Non-deducibility on Composition which is an (stronger) alternative to SNNI. Since many of timing attacks are based on statistic behaviour it seem to be reasonable to exploit also some features of probabilistic process algebras. Than we could formulate probabilistic NIF property requiring that if there exists a sequence of actions (with a probability p) which contains some private actions then there should exists (with probability p') another sequence which does not contain them and the both sequences cannot be distinguish by an observer and $|p - p'| \leq e$ for some given $0 \leq e < 1$.

References

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

[2.] 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.

[3.] 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

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

[5.] 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.

[6.] 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.

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

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

[9.] 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.

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

[11.] 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.

[12.] 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.

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

[14.] Groote, J. F.: "Transition Systems Specification with Negative Premises". Baeten, J.C.M. and Klop, J.W. (eds.), *CONCUR'90*, Springer Verlag, Berlin, LNCS 458, 1990.

[15.] Gruska D.P.: Information-Flow Attacks Based on Limited Observations. in Proc. of PSI'06, Springer Verlag, LNCS 4378, Berlin, 2006.

[16.] 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

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

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

[19.] Gruska, D., Maggiolo-Schettini, A.: Nested Timing Attacks, Proceedings FAST 2003, 2003.

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

[21.] 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.

[22.] Milner, R.: *Communication and concurrency.* Prentice-Hall International, New York,1989.

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