Fundamenta Informaticae 109 (2011) 1–15
DOI 10.3233/FI-2011-508
IOS Press

Gained and Excluded Private Actions by Process Observations *

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

Abstract. Formalisms for description how much information on private actions can be obtained by observing public ones are presented. Two sets of private actions are considered. The set of actions which execution is guaranteed according to observations and the set of actions which execution is excluded according to observations. Since information flows could be realized also by means of different covert channels as time, termination and divergence this possibility is considered as well. Both qualitative and quantitative dimensions of the flow are considered.

Keywords: information flow, security, non-interference, termination, divergence, covert channel

1. Introduction

Traditional security properties are frequently criticized for being either too restrictive or too benevolent. For example, usually they consider a standard access control process to be insecure since there is always some (even very small) information flow for an attacker which tries to learn a password. There are several ways how to overcome these disadvantages. By means of the Shannon's information theory it could be expressed an amount of information leaked as it was done, for example, in [2, 3] for simple imperative languages and in [11] for process algebra. Another possibility is to exploit probabilistic theory as it was used for process algebras in [10]. Resulting techniques lead to expression how many bits of private information can leak or how probable is that an intruder can learn a secrete expressed by a predicate over processes traces.

The aim of this paper is to present an alternative formalisms for description how much information on private actions can be obtained by observing public ones. Here we will directly express an amount of secrete as a subset of private actions which occurrence (and not occurrence) can be deduced by an intruder who can observe public behaviour of the system. Since many attacks can be realized by exploiting

*Work supported by the grant VEGA 1/0688/10.
†Address for correspondence: Institute of Informatics, Comenius University, Mlynska dolina, 842 48 Bratislava, Slovakia

additional information coming from covert channels we will formalize also such situations. We will consider intruders capable to observe also termination as well as divergence. We show that this leads to different security properties. Since we will consider concepts as termination, divergence and idling we first define a simple timed extension of process algebra which enables us to describe precisely the above mentioned notions. In [1] terminations which could be exploited by an intruder are studied in the case of a deterministic imperative language. In the framework of process algebras we must use a slightly different approach. First, resulting transition systems are in general nondeterministic and concept of inputs/outputs cannot be directly applied and for second, termination, divergence and not termination has to be considered differently in case of timed process algebra.

The paper is organized as follows. In Section 2 we describe the timed process algebra TPA which will be used as a basic formalism. In Section 3 we present and investigate different notions of security based on an absence of information flow realized by direct observations of systems public actions and by exploiting additional covert channels (mainly termination and divergence). We will express a quality and quantity of private information leaked as subsets of private actions which occurrence can be deduced by an intruder and as a subset of private actions which occurrence can be excluded. More precisely, if an intruder sees process P to perform public actions $\tilde{l}$ we define a subset of private actions $g(P,\tilde{l})$, which occurrence can be deduced from such observation and subset $e(P,\tilde{l})$, which occurrence can be excluded according to that observation. We compare resulting security properties and present some of their properties. Moreover, we propose several numerical measures of systems security. In Section 4 we present a technique of monitors and we explain how the presented theory could be exploited for their constructions.

2. Timed Process Algebra

In this section we define Timed Process Algebra, TPA for short. TPA is based on Milner's CCS (see [16]) but the special time action t which expresses elapsing of (discrete) time is added (see also [9]). The presented language is a slight simplification of the Timed Security Process Algebra (tSPA) introduced in [5]. 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 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 [7]). 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 a 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 (see [16]) 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). In the definition of the labeled transition system we have used negative premises (see $Pa1$). 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 [8]. 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 [5]. 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 $P \stackrel{x}{\Rightarrow}_M P'$ for $M \subseteq A$ iff $P \xrightarrow{s_1} \xrightarrow{x} \xrightarrow{s_2} P'$ for $s_1, s_2 \in (M \cup \{\tau\})^\star$ and $P \stackrel{s}{\Rightarrow}_M$ instead of $P \stackrel{x_1}{\Rightarrow}_M \stackrel{x_2}{\Rightarrow}_M \cdots \stackrel{x_n}{\Rightarrow}_M$. Instead of $\Rightarrow_\emptyset$ we will write $\Rightarrow$ and instead of $\Rightarrow_{\{h\}}$ we will write $\Rightarrow_h$. By ϵ we will denote the empty sequence of actions and by $s \sqsubseteq s'$, $s, s' \in Actt^\star$ we will denote that s is a prefix of s'. By $s \sqsubseteq^t s'$ we will denote that s' can be obtained from s by inserting t actions to s. For example, $a.t.b \sqsubseteq^t a.t.t.b$. By $Sort(P)$ we will denote the set of actions (except τ) which can be performed by P i.e. $Sort(P) = \{x | P \xrightarrow{s.x}$ for some $s \in Actt^\star$ and $x \neq \tau\}$.

Let $s \in Actt^{\star}$. By $|s|$ we will denote the length of s i.e. a number of action contained in s. By $s|_B$ we will denote the sequence obtained from s by removing all actions not belonging to B. For example, $|s|_{\{t\}}|$ denote a number of occurrences of t in s, i.e. time length of s.

Definition 2.1. The set of weak timed traces of process P with respect to the set $M, M \subseteq A$ is defined as $Tr_{wM}(P) = \{s \in (A \cup \{t\})^{\star} | \exists P'.P \stackrel{s}{\Rightarrow}_M P'\}$. Instead of $Tr_{w\emptyset}(P)$ we will write $Tr_w(P)$.

Two processes P and Q are weakly timed trace equivalent with respect to M ($P \approx_{wM} Q$) iff $Tr_{wM}(P) = Tr_{wM}(Q)$ and process Q is a trace simulation of P with respect to M ($P \preceq_{wM} Q$) iff $Tr_{wM}(P) \subseteq Tr_{wM}(Q)$. Again we will write $\approx_w$ and $\preceq_w$ instead of $\approx_{w\emptyset}$ and $\preceq_{w\emptyset}$, respectively. We will write $P \preceq^t_{wM} Q$ if for every $s, s \in Tr_{wM}(P)$ there exists $s', s' \in Tr_{wM}(Q)$ such that $s \sqsubseteq^t s'$.

3. Non-interference

First we define termination-insensitive security (see for example [18]) for imperative programs. We suppose that the set of variables is divided into two parts - public and private ones.

Definition 3.1. (BTNI) A deterministic program C satisfies batch-job termination-insensitive noninterference (BTNI) if, for any memory (where program variables are stored) M and N that agrees on public (low) variables, the final memories produced by running C on M and N also agree on public variables (provided that both runs terminate successfully).

To translate this notion to process algebra setting we suppose that all actions are divided into two groups, namely public (low level) actions L and private (high level) actions H i.e. $A = L \cup H, L \cap H = \emptyset$. Moreover, we suppose that $H \neq \emptyset$ and $L \neq \emptyset$ and that for every $h \in H, l \in L$ we have $\overline{h} \in H, \overline{l} \in L$. To denote sequences of public actions, i.e sequences consisting of actions from $L \cup \{t\}$ and sequences of private actions from H, we will use notation $\tilde{l}, \tilde{l'}, \ldots$ for sequences from $(L \cup \{t\})^{\star}$ (note that elapsing of time - i.e. t action is also a public action) and $\tilde{h}, \tilde{h'}, \ldots$ for sequences from $H^{\star}$, respectively. The set of actions could be divided to more than two subsets, what would correspond into more levels of classification. All the following concepts could be naturally extended to such setting.

3.1. Gained private actions

First we define a set of private actions which occurrence can be learned by an intruder who see a process to perform a sequence of public actions $\tilde{l}$ (we will call such action as gained actions).

Definition 3.2. Let $P \in TPA$ and $\tilde{l} \in Tr_{wH}(P)$. Then the occurrence of the set of private action which can be gained about P by public observing $\tilde{l}$ is defined as follows:

$$g(P, \tilde{l}) = \{h | h \in H, P \stackrel{\tilde{l}}{\not\Rightarrow}_{H \setminus \{h\}}\}.$$

According to Definition 3.2 the set of private actions $g(P, \tilde{l})$ is the one which has to be performed by P if an intruder sees P to perform public actions $\tilde{l}$.

Example 3.1. Let $P = l_1.h.l_2.Nil + l_1.l_2.Nil$ and $P' = l_1.h.h'.l_2.Nil + l_1.h.l_2.Nil$. Let $\tilde{l} = l_1.l_2$ then we have $g(P, \tilde{l}) = \emptyset, g(P', \tilde{l}) = \{h\}$.

If the intruder can observe a longer sequence of public actions then the same or bigger set of private actions can be gained as it is stated in the following theorem.

Proposition 3.1. Let $\tilde{l}, \tilde{l'} \in Tr_{wH}(P)$ and $\tilde{l} \sqsubseteq \tilde{l'}$ then we have

$$g(P, \tilde{l}) \subseteq g(P, \tilde{l'}).$$

Proof:
Let $h \in g(P, \tilde{l})$. Suppose that $h \notin g(P, \tilde{l'})$ i.e. $P \stackrel{\tilde{l'}}{\Rightarrow}_{H\setminus\{h\}}$. Since $\tilde{l} \sqsubseteq \tilde{l'}$ we would have also $P \stackrel{\tilde{l}}{\Rightarrow}_{H\setminus\{h\}}$ but this would imply that $h \notin g(P, \tilde{l})$. □

As regards compositional properties some of them can be formulated for $g(P, \tilde{l})$ as they are stated in the next theorem.

Proposition 3.2. Let $P, Q \in TPA$ and $\tilde{l} \in Tr_{wH}(P) \cup Tr_{wH}(Q)$. Then the following holds:

$$\begin{aligned}
g(x.P, x.\tilde{l}) &= g(P, \tilde{l}) \text{ if } x \in L && (1)\\
g(\tau.P, \tilde{l}) &= g(P, \tilde{l}) && (2)\\
g(x.P, \tilde{l}) &= g(P, \tilde{l}) \cup \{x\} \text{ if } x \in H && (3)\\
g(P + Q, \tilde{l}) &\subseteq g(P, \tilde{l}) \cup g(Q, \tilde{l}) && (4)\\
g(P|Q, \tilde{l}) &\subseteq g(P, \tilde{l}) \cup g(Q, \tilde{l}) && (5)\\
g(P\setminus M, \tilde{l}) &= g(P, \tilde{l})\setminus M \text{ if } \tilde{l} \in Tr_{wH}(P\setminus M) && (6)\\
g(P[S], \tilde{l}) &= g(P, \tilde{l})[S] && (7)\\
g(\mu XP, \tilde{l}) &= g(P[\mu XP/X], \tilde{l}) && (8)
\end{aligned}$$

Proof:

(1) Let $h \in g(P, \tilde{l})$ i.e. $P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$. From this we have that $x.P \stackrel{x.\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$ and so $h \in g(x.P, x.\tilde{l})$ and vice versa.

(2) Clearly $\tau.P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$ iff $P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$.

(3) We have $\tilde{l} \in Tr_{wH}(x.P)$ and $x.P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{x\}}$ so $x \in g(x.P, \tilde{l})$. For $h, h \neq x$ it holds $h \in g(x.P, \tilde{l})$ iff $h \in g(P, \tilde{l})$.

(4) Let $h \in g(P + Q, \tilde{l})$. We have $P + Q \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$ i.e. $P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$ and $Q \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$ but $\tilde{l} \in Tr_{wH}(P)$ or $\tilde{l} \in Tr_{wH}(Q)$ and hence $h \in g(P, \tilde{l}) \cup g(Q, \tilde{l})$. Note that the equation does not hold: let $P = l.Nil, Q = h.l.Nil$, then $g(P + Q, \tilde{l}) = \emptyset$ and $g(P, \tilde{l}) \cup g(Q, \tilde{l}) = \{h\}$.

(5) Note that $\tilde{l} \in Tr_{wH}(P) \cup Tr_{wH}(Q)$ implies $\tilde{l} \in Tr_{wH}(P|Q)$. Let $h \in g(P|Q, \tilde{l})$ i.e. $P|Q \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$. Suppose that $h \notin g(P, \tilde{l}) \cup g(Q, \tilde{l})$ what means that $P \stackrel{\tilde{l}}{\Rightarrow}_{H\setminus\{h\}}$ and $Q \stackrel{\tilde{l}}{\Rightarrow}_{H\setminus\{h\}}$ but this would lead to contradiction with the assumption that $P|Q \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$. Note that $g(P|Q, \tilde{l}) \neq g(P, \tilde{l}) \cup g(Q, \tilde{l})$ in general. Let $P = l_1.h.l_2.Nil, Q = l_1.h.l_2.Nil + l_2.Nil$ then we have $g(P|Q, l_1.l_2) = \emptyset$ and $g(P, l_1.l_2) = g(Q, l_1.l_2) = \{h\}$.

(6) Let $h \in g(P\backslash M, \tilde{l})$ and suppose $h \notin g(P,\tilde{l})\backslash M$. Every execution of public actions $\tilde{l}$ performed by $P \setminus M$ contains action h and hence $h \notin M$. So it should hold that $h \notin g(P,\tilde{l})$ i.e. we have always $P \stackrel{\tilde{l}}{\Rightarrow}_{H\setminus\{h\}}$ but since restriction of actions from M has no influence to these executions we would have a contradiction with our assumption.

Let $h \in g(P,\tilde{l})\backslash M$. That means that $h \notin M$ and $P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$ but clearly we have also $P\backslash M \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$ so $h \in g(P\backslash M, \tilde{l})$.

(7,8) The proofs of these equations are straightforward.

□

Now we are prepared to formulate how much information can be gained by observing public activities of a process. The formal definition follows.

Definition 3.3. Let $P \in TPA$. By $g(P)$ we will denote the set of private actions which occurrence by P can be gained (detected) by an intruder observing sequence of public actions

$$g(P) = \bigcup_{\tilde{l}\in Tr_{wH}(P)} g(P,\tilde{l})$$

We say that no private information can be gained by observing P if $g(P) = \emptyset$.

Example 3.2. Let as consider the following process: $P = \sum h_i.l^i.Nil$ where $H = \{h_1,\ldots,h_n\}$. It is easy to see that $g(P,l_i) = \{h_i\}$ for every $i, 1 \leq i \leq n$ and $g(P) = H$ i.e. by observing l_i an attacker can learn that h_i was performed and an occurrence of every private action can be detected.

Example 3.3. Let as consider the following process: $P = h_1.l^1.\ldots.h_n.l^n.Nil$ where $H = \{h_1,\ldots,h_n\}$. Again, it is easy to see that $g(P,\tilde{l}) = g(P) = H$ where $\tilde{l} = l_1.\ldots.l_n$ i.e. by observing $\tilde{l}$ an attacker can learn that all private actions h_i were performed.

For $g(P)$ we could formulate similar composition properties as those ones formulated in Proposition 3.2.

Now we can show how the property "no private information can be gained by observing P" is related to another absence-of-information-flow property - Strong Nondeterministic Non-Interference (SNNI, for short). We recall its definition (see [5]). Process P has SNNI property (we will write $P \in SNNI$) 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 it holds if $P \stackrel{a}{\rightarrow} P'$ then $P/M \stackrel{a}{\rightarrow} P'/M$ whenever $a \notin M \cup \bar{M}$ and $P/M \stackrel{\tau}{\rightarrow} P'/M$ whenever $a \in M \cup \bar{M}$. Formal definition of SNNI follows.

Definition 3.4. Let $P \in TPA$. Then $P \in SNNI$ iff $P \setminus H \approx_w P/H$.

Theorem 3.1. If $P \in SNNI$ then $g(P) = \emptyset$.

Proof:
Let $P \in SNNI$ and suppose that $g(P) \neq \emptyset$. Hence there exists $h, h \in H$ and $\tilde{l}, \tilde{l} \in Tr_{wH}(P)$ such that $P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}}$. But then there exists sequence s which contains $\tilde{l}$ and h such that $s \in Tr_w(P/H)$ but $s \notin Tr_w(P \setminus H)$ i.e. $P \setminus H \not\approx_w P/H$. □

The inverse of the previous theorem does not hold as it shows the following example.

Example 3.4. Let $P = \sum_{1 \leq i \leq n} h_i.l.Nil$ and $H = \{h_1, \ldots, h_n\}$. Then $g(P) = \emptyset$ but P has not SNNI property since $P \setminus H \not\approx_w P/H$. Indeed $P \setminus H$ cannot perform the sequence of action $\tau.l$ while P/H can perform it and an intruder seeing l can deduce that a private action was performed. On the other side the intruder observing just action l can in fact learn nothing - it was clear from the beginning that some private action has to be performed so an intruder gained by observation no new knowledge and hence property SNNI could be considered stronger then property $g(P) = \emptyset$ or expressing something fundamentally different than $g(P)$ (this will be discussed in more detail in the following subsection). If we would consider process $P' = P + l.Nil$ then $P' \in SNNI$ and in this case an intruder learns really nothing observing l.

The amount of information about performed private actions can be naturally quantified as follows.

Definition 3.5. Let $P \in TPA$. Then the measure of gained private actions by observing P performing $\tilde{l}$ is defined as follows:

$$mg(P, \tilde{l}) = \frac{|g(P, \tilde{l})|}{|H|}$$

and

$$mg(P) = \frac{|g(P)|}{|H|}.$$

Clearly $P \in SNNI$ implies $mg(P) = 0$ and if $mg(P) = 1$ then all "secrete" could be discovered (i.e. $g(P) = H$). As a corollary of Proposition 3.1 we have that for $\tilde{l}, \tilde{l'} \in Tr_{wH}(P)$ and $\tilde{l} \sqsubseteq \tilde{l'}$ we have $mg(P, \tilde{l}) \leq mg(P, \tilde{l'})$.

In Definition 3.1 it is assumed that both runs terminate successfully. If we translate this concept to process algebra setting, i.e. if we would consider only terminating traces it would change the situation significantly.

Example 3.5. Let P be defined as follows $P = h_1.l^1. \ldots h_n.l^n.\mu X \tau.t.X$ where $H = \{h_1, \ldots, h_n\}$. Clearly, $g(P) = H$ but if we would consider observation of only terminating traces than an intruder would gain nothing.

In the following we will formalize the concept of considering only successfully terminating traces. We will express a termination on semantical level. We introduce a special symbol $\surd, \surd \notin Actt$ and a new derivation rule:

$$\frac{P \stackrel{x}{\not\rightarrow}, x \in Act \text{ and if } P \stackrel{t}{\rightarrow} P' \text{ then } P = P'}{P \stackrel{\surd}{\rightarrow} Nil}$$

and we extend the set of traces accordingly. Note that the requirement $P = P'$ from this transition rule is needed due to transition rules A1, A2 which allow idling for for arbitrary process.

Now we are ready to define the set of private actions which can be gained from P by public observing of terminating sequence $\tilde{l}.\surd$.

Definition 3.6. Let $P \in TPA$ and $\tilde{l}.\surd \in Tr_{wH}(P)$. Then the occurrence of the set of private actions which can be gained from P by public observing $\tilde{l}.\surd$ is defined as follows:

$$g_t(P, \tilde{l}.\surd) = \{h | P \stackrel{\tilde{l}.\surd}{\not\Rightarrow}_{H\setminus\{h\}}\}.$$

The above definition can be extended to all terminating sequences in style of Definition 3.3 and we get a set of private information which occurrence can be gained by observing terminate sequences i.e. $g_t(P) = \bigcup_{\tilde{l}.\surd \in Tr_{wH}(P)} g(P, \tilde{l}.\surd)$. Sets of private information gained by observing processes with or without termination detection are in general incomparable as it is stated in the following theorem.

Theorem 3.2. There exist processes P and Q such that $g_t(P) \subset g(P)$ and $g(Q) \subset g_t(Q)$.

Proof:
It can be checked that for process P from Example 3.5 we have $g_t(P) = \emptyset$ and $g(P) = H$. Let $Q = h.l.Nil + l.l'.Nil$ it is easy to check that $g(Q) = \emptyset$ and $g_t(Q) = \{h\}$. □

Example 3.6. Now let us consider the following process $P = l.Nil + h.\mu X\tau.t.X$. We have $g(P) = g_t(P) = \emptyset$ but occurrence of performing of private action h can be detected if an attacker can recognize a divergent behavior of the process.

The previous example leads us to a new concept of information gained by observing divergence (for example, by power consumption covert channel). First let as define divergence in TPA setting. We say that a process P diverge (this will be denoted as $P \Uparrow$) if it can perform an infinite sequence of actions containing only infinite number of τ actions and t actions. Note that since we consider only t-guarded processes any infinite sequence has to contain infinite number of t actions.

Definition 3.7. Let $P \in TPA$. Then the occurrence of the set of private action which can be gained from P by public observing $\tilde{l}$ and recognizing process divergence is defined as follows:

$$g_d(P, \tilde{l}) = \{h | P \stackrel{\tilde{l}}{\Rightarrow}_H P', P' \Uparrow \text{ and } P \stackrel{\tilde{l}}{\not\Rightarrow}_{H\setminus\{h\}} P'', P'' \Uparrow\}.$$

Again we define $g_d(P)$ a union of $g_d(P, \tilde{l})$ for all $\tilde{l}$ formally we have $g_d(P) = \bigcup_{\tilde{l} \in Tr_{wH}(P)} g_d(P, \tilde{l})$.

Example 3.7. Let $P = l.Nil + h.l.Nil.\mu X\tau.t.X$. It is easy to check that $g(P) = g_t(P) = \emptyset, g_d(P) = \{h\}$.

As regards a compositional property, similar theorem to Proposition 3.2 could be formulated for $g_d(P)$.

Also as regards a relationship between $g(P), g_t(P)$ and $g_d(P)$, a similar property as it is stated in Theorem 3.2 holds, i.e. all of them are different. The relationship can be depicted in the following figure 1.

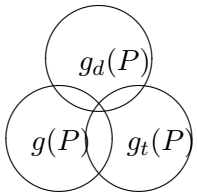

Figure 1. Gained actions relationship

3.2. Excluded private actions

In this subsection we will examine which private action could be excluded by an intruder observing a process. We start with a motivation example.

Example 3.8. (Access control process)
Let Psw be a set of all possible passwords. Let us consider a simple access control process defined as follows (the set of high level action H_{Psw} consists of actions $h_w, w \in Psw$ and actions $\bar{l}_{\text{login}}$, $\bar{l}_{\text{access denied}}, l_w, w \in Psw$ are low level actions).

$$P = l_v.h_v.\bar{l}_{\text{login}}.Nil + \sum_{u \in Psw, u \neq v} l_u.h_v.\bar{l}_{\text{access denied}}.Nil$$

This process could represent, for example, an access to safe-deposit where no name of a bank client is required just a private key (or pin code - i.e. some password, in general). An attacker tries to guess the correct password. (S)he enters u what is modeled by performing low level action l_u ((s)he can see/observe what he tries - a public action l_u could be "observed".) The guessed password (u) is compared with the correct one (v, represented by high level action h_v, which is unknown for the attacker). If the attacker observes public sequence $\tilde{l} = l_u.\bar{l}_{\text{access denied}}$ then (s)he can learn, that u is not the correct password so (s)he can gain some information about the correct one - since the correct one is from the reduced set $Psw \setminus \{u\}$. Note that $g(P, \tilde{l}) = g_d(P, \tilde{l}) = g_t(P, \tilde{l}) = \emptyset$ and hence to describe the knowledge obtained by an attacker observing $\tilde{l}$ we need a new concept.

Definition 3.8. Let $P \in TPA$. Then the occurrence of the set of private action which can be excluded by observing P performing public action $\tilde{l}$ (i.e. $\tilde{l} \in Tr_{wH}(P)$) is defined as follows:

$$e(P, \tilde{l}) = \bigcap_{P \stackrel{\tilde{l}}{\Rightarrow} M} H \setminus M$$

Example 3.9. (Access control process, continuation)
It is easy to check that $e(P, \tilde{l}) = \{h_u\}$. If we would consider the following modification of P

$$P' = \mu X(l_v.h_v.\bar{l}_{\text{login}}.t.X + \sum_{u \in Psw, u \neq v} l_u.h_v.\bar{l}_{\text{access denied}}.t.X)$$

then an intruder can exclude all wrong passwords. Usual solutions of this security hole are made either by limiting a number of attempts to enter a password or by putting longer time delays till the system is ready to read a new password.

Example 3.10. Let us consider process $R = h_1.l_1.Nil + h_2.l_1.Nil + h_3.l_2.Nil + h_4.l_2.Nil$ and $H = \{h_1, h_2, h_3, h_4\}$. We have $e(R, l_1) = \{h_3, h_4\}, e(R, l_2) = \{h_1, h_2\}$ but $g(R) = \emptyset$. This means that, in general, the concept of excluded actions by an observation cannot be covered by gained actions even if we consider all possible observations of a process ($g(R)$) (see also Theorem 3.3 for one observation).

If we have that $e(P, \tilde{l}) = \emptyset$ that means that an intruder after observing $\tilde{l}$ cannot exclude occurrence of any private action.

Note that an analogy to Proposition 3.1 does not hold for excluded actions as it is illustrated by the following example.

Example 3.11. Let $H = \{h\}$ and $P = l_1.h.l._2.Nil$. Then $e(P, \tilde{l}) = h, e(P, \tilde{l'}) = \emptyset$ for $\tilde{l} = l_1, \tilde{l'} = l_1.l_2$ i.e. $e(P, \tilde{l}) \not\subseteq e(P, \tilde{l'})$ for $\tilde{l} \sqsubseteq \tilde{l'}$.

On the other side, let us consider process $P = l_1.h.l_2.Nil + l_1.l_2.l_3.Nil$. For $\tilde{l} = l_1.l_2, \tilde{l'} = l_1.l_2.l_3$ we have $e(P, \tilde{l}) = \emptyset, e(P, \tilde{l'}) = \{h\}$ and so neither $e(P, \tilde{l}) \not\supseteq e(P, \tilde{l'})$ for $\tilde{l} \sqsubseteq \tilde{l'}$.

There is no direct correlation between sets $g(P, \tilde{l})$ and $e(P, \tilde{l})$ since there are processes such that for one is the former set empty and the later nonempty and vice versa. If both of them are empty, that means, that an intruder can learn practically nothing on private actions by observing process P and seeing it to perform $\tilde{l}$. In some sense $g(P, \tilde{l})$ and $e(P, \tilde{l})$ are complementary as it is stated in the following theorem.

Theorem 3.3. For every process P and every $\tilde{l}, \tilde{l} \in Tr_{wH}(P)$ it holds $g(P, \tilde{l}) \cap e(P, \tilde{l}) = \emptyset$ and $\emptyset \subseteq g(P, \tilde{l}) \cup e(P, \tilde{l}) \subseteq H$.

Proof:
Let $h \in g(P, \tilde{l})$. We now that every execution of sequence of visible action $\tilde{l}$ has to contain h i.e. if $P \stackrel{\tilde{l}}{\Rightarrow}_M$ then $h \in M$. That means $h \notin H \setminus M$ i.e. $h \notin e(P, \tilde{l})$. As regards the second part of the theorem let us consider process $P = l.Nil + h.l.Nil$. We have $g(P, l) = e(P, l) = \emptyset$. If we consider $H = \{h\}$ then we see that $g(P, \tilde{l}) \cup e(P, \tilde{l}) = H$ i.e. $\subseteq$ cannot be replaced by $\subset$. □

Now we can define a set of private actions which occurrence could be excluded by the set of observations O. Note, that for gained actions we considered all possible observations but here it seems to be more appropriate to have possibility to consider only a subset of them (see Example 3.12).

Definition 3.9. Let $P \in TPA$. Then the occurrence of the set of private action which executions could be excluded by the set of observations $O, O \subseteq Tr_{wH}(P)$ is defined as follows:

$$e_O(P) = \bigcup_{\tilde{l} \in O} e(P, \tilde{l})$$

Example 3.12. (Access control process, continuation)
Let us consider the following set of observations $O = \{l_u.\bar{l}_{\text{access denied}} | u \in H_{Psw}\}$. Then $e_O(P) = H_{Psw} \setminus \{h_v\}$ where h_v is the correct password. Hence an intruder can learn this password.

Example 3.13. Let us consider process R form Example 3.10 and $O = \{l_1, l_2\}$. Then we have $e_O(R) = H$ i.e. in this particular case $e_O(R)$ does not express anything useful for an intruder.

We can again quantify an amount of information on excluded private actions gained by observations. In this way we get a numerical expression of an appropriate level of security. High numbers (closer to 1) correspond to less secure systems.

Definition 3.10. Let $P \in TPA$. Then the measure of excluded private actions by observing P performing $\tilde{l}$ and $O, O \subseteq Tr_{wH}(P)$ is defined as follows:

$$me(P,\tilde{l}) = \frac{|e(P,\tilde{l})|}{|H|}$$

and

$$me(P,O) = \frac{|e_O(P)|}{|H|},$$

respectively.

Example 3.14. (Access control process, continuation)
Let $|H_{Psw}| = n$ and let an intruder observes actions $\tilde{l}, \tilde{l} = l_u.\bar{l}_{\text{access denied}}$ performed by P. Then $me(P,\tilde{l}) = \frac{1}{n}$ and $me(P,O) = (n-1)/n$. In general, smaller values of me correspond to higher levels of system security.

Also for sets of excluded private actions we could define an exploitation of additional covert channels as termination and divergence as it was done for sets of gained private actions. For resulting concepts we would have similar properties as they are presented in the previous subsection. Instead of that we will focus on elapsing of time.

3.3. Time Observations

Till now have not taken special care on elapsing of time. Considering time allows us to introduce an alternative measurements to process security as mg and me. Roughly speaking, an amount of gained information has to be related to time needed for corresponding observations (attacks).

Example 3.15. Let $P = l.t^n.h.l.Nil + l.t^{n-1}.l.Nil$ and $\tilde{l} = l.t^n.l$. Clearly we have g(P,$\tilde{l}$) $= \{h\}$ put for any sequence $\tilde{l'}, |\tilde{l'}|_{\{t\}}| < n$ we have $g(P,\tilde{l}) = \emptyset$.

The above mentioned example leads us to the following alternative measures of systems security.

Definition 3.11. Let $P \in TPA$, $\tilde{l} \in Tr_{wH}(P)$ and $M \subseteq H$. Then we define

$$tg(P,\tilde{l}) = \frac{|g(P,\tilde{l})|}{|\tilde{l}|_{\{t\}}|}$$

and

$$\begin{aligned} mtg(P,M) &= \min_k\{k | k = \sum |\tilde{l_i}|_{\{t\}}, \bigcup g(P,\tilde{l_i}) = M\} \text{ if such } \tilde{l_i} \text{ exists} \\ &= \infty \text{ otherwise .} \end{aligned}$$

Measure $tg(P,\tilde{l})$ expresses a relation between a number of private actions gained by an observation and (time) length of this observation. Let $P = l.t^m.h.l$, $\tilde{l} = l.l$ then $tg(P,\tilde{l}) = 1/m$. A smaller number expresses more secure processes. Measure $mtg(P,M)$ express minimal (time) length of observations to learn the set M of private actions. Again, a bigger number means more secure process (with respect to M). If $mtg(P,M) = \infty$ then M cannot be learned at all. Unfortunately, we have no "numeric" parallel to Proposition 3.1 as it is stated in the following theorem.

Theorem 3.4. There exist $P, Q \in TPA$, $\tilde{l}, \tilde{l'}, \in Tr_{wH}(P)$, $\tilde{l_1}, \tilde{l'_1}, \in Tr_{wH}(Q)$, $\tilde{l} \sqsubseteq \tilde{l'}$ and $\tilde{l_1} \sqsubseteq \tilde{l'_1}$ such that we have

$$tg(P,\tilde{l'}) < tg(P,\tilde{l}) \text{ and } tg(Q,\tilde{l_1}) < tg(Q,\tilde{l'_1}).$$

Proof:
Note that from Proposition 3.1 we know that $g(P,\tilde{l}) \subseteq g(P,\tilde{l'})$ but in this case time length of $\tilde{l}$ and $\tilde{l'}$ is not taken into account. Let us consider processes $P = l.h.Nil, Q = t^n.l_1.Nil + h.t^n.l_1.t.l_2.Nil$ and $\tilde{l} = l.t^n, \tilde{l'} = l.t^{n+k}, k > 0$ and $\tilde{l_1} = t^n.l_1, \tilde{l'_1} = t^n.l_1.t.l_2$. Then we have $g(P,\tilde{l}) = 1/n, g(P,\tilde{l'}) = 1/(n+k)$ and $g(Q,\tilde{l_1}) = 0, g(Q,\tilde{l'_1}) = 1/(n+1)$.

□

We can also easily describe timeless observations by putting t action between private actions i.e. an intruder does not see elapsing of time.

Example 3.16. Let $P = l.h.t.l.Nil + l.l.Nil$ and $\tilde{l} = l.l$. We have $g(P,\tilde{l}) = h$ but for an intruder who does not see elapsing of time we would have $g^t(P,\tilde{l}) = \emptyset$ where g^t is a modification of g such that we put t among private actions.

The concepts and definitions from previous subsections can be naturally translated for "timeless intruders". Resulting security properties are adequate if we can suppose that an intruder has no means to observe timed behavior of systems with reasonable accuracy.

4. Applications

The previously developed theory allows us to express how much information can be obtained by an intruder about process's private actions. Then we can change process design to fulfil some security target (expressed by, for example, values of functions mg, me, tg) i.e. we can construct a process which is more secure then the original one. We will call the new process as a reinforcement of the the old one. The formal definition follows.

Definition 4.1. We say that process P' is reinforcement of process P with respect to gained actions (denoted $P' < P$) (gained actions by termination observation ($P' <_t P$)/ gained actions by divergence observation ($P' <_d P$) / excluded actions ($P' <_e P$)) iff $P' \preceq_{wM} P$, for $M \subseteq L, M \cap Sort(P) = \emptyset$ and $g(P') < g(P)$ ($g_t(P') < g_t(P)$/ $g_d(P') < g_d(P)$/ $e(P') < e(P)$). We will speak about timed reinforcements (indicated by superscript t) if we replace $P' \preceq_{wM} P$ by $P' \preceq^t_{wM} P$ in the reinforcement definition.

In case that we cannot modify the original process we can use a technique similar to that of enforcing security policy (see [15, 17]). We exploit special processes called monitors. A monitor can modify behaviour of the process in such a way that the resulting process is, roughly speaking, more secure. Monitors can halt an "dangerous" execution or suspend or add some actions to trace of actions in such a way that resulting process (plus monitor) becomes more secure. In this case process does not communicate with an environment directly but through the monitor (see Fig.2). To define such monitors let us extend the set of public actions L by their ghost counterparts L' and we introduce bijection S between them. We expect from the monitoring that $((P[S]|Monitor) \setminus L)[S^{-1}]$ is (timed) reinforcement of process P.

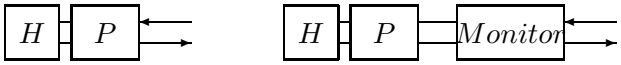

Figure 2. Direct and undirect communications

Example 4.1. (Access control process, reinforcement)
Let us consider access control process from Example 3.9 and the following monitor. $Monitor = \sum_{x_i \in L_{Psw}} \bar{x}'_1 x_1 . \bar{x}'_{\text{access denied}} . \bar{x}_{\text{access denied}} . \bar{x}'_2 x_2 . \bar{x}'_{\text{access denied}} . \bar{x}_{\text{access denied}} . x_{\text{access abborted}} \cdot$ The monitor block access after two failed attempts and hence significantly lower the value of me.

Example 4.2. (Timing attacks timed reinforcement)
Timing attacks are special type of attacks when an intruder exploits an information about timing of actions occurrences. These attacks are very powerful. 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 [13]). This idea was developed in [4] where a timing attack against smart card implementation of RSA was conducted. Attacks on web privacy are considered for example in [6]. One way how to avoid timing attacks is to introduce random delays (from the set I) between public actions. Let us consider the following process $Monitor = \mu X \sum_{x' \in L', i \in I} \bar{x}' . t^i . x . t . X$. By appropriate choice of I we can obtained time reinforcement of process P i.e. such that $((P[S]|Monitor) \setminus L)[S^{-1}] <^t P$ and hence we can lower the value of tg.

5. Conclusions

We have presented several security concepts based on an information flow. They express which set of private actions was performed (gained sets) or which set of private actions could be excluded by an intruder observing systems public actions (excluded sets). The concepts offer a finer security notion with respect to traditional ones which usually express only that an intruder can learn that a private action was performed (for example SNNI or opacity [12]). Moreover the notion excluded actions can be used for reduction of a space of possible private actions and if the reduction is significant then it really threatens systems security.

Concepts of gained and excluded sets of private actions are complementary. Roughly speaking, only systems for which both the sets - gained and excluded private actions are empty could be considered fully secure. But since this is a very rare situation we suggest also a numerical expression which offers us a quantification of security. That means, if the resulting measure is small enough the system can still be considered secure with respect to some given requirements. We have also discussed a measure of

security which depends on a minimal time length of public observation to obtain some given knowledge on private actions. This seems to be a realistic notion since it takes into account realistic possibilities for intruders since only rarely there are no restrictions on how long systems can be observed. The length of observations could be limited by system design (for example one can try to guess a password only for limited number of times) or naturally (say that an intruder can learn something significant only after an observation which takes hundreds of years).

We have investigated also several additional covert channels which could be exploited by an intruder. Particularly interesting are termination and divergence channels. They can be exploited by an intruder who can learn that the system is still working but does not react (for example, by power consumption). We have showed that such an intruder can learn an occurrence of different sets of private actions. It might happen, for example, that the system is completely secure if an intruder cannot see termination (or divergence) and vice versa.

Presented formalism allows us to express how much information can be obtained by an intruder about process private actions. Then we can change process design to fulfil some security target (expressed by, for example, values of functions mg, me, tg) i.e. we can construct a new process (called reinforced one) which is more secure then the original one. Another possibility is to use a technique of monitors which can block some ”dangerous” executions ore slightly change them so the process together with its monitor leaks less information. In fact, by blocking the dangerous executions some of the presented security properties are transformed to safety ones (note that non-interference is not a safety property in general).

References

[1] Askarov A., S. Hunt, A. Sabelfeld and D. Sands: Termination-Insensitive Noninterference Leaks More Than Just a Bit. Proc. of the 13th European Symposium on Research in Computer Security (ESORICS'2008), LNCS 5283, 2008.

[2] Clark D., S. Hunt and P. Malacaria: A Static Analysis for Quantifying the Information Flow in a Simple Imperative Programming Language. The Journal of Computer Security, 15(3). 2007.

[3] Clarkson, M.R., A.C. Myers, F.B. Schneider: Quantifying Information Flow with Beliefs. Journal of Computer Security, to appear, 2009.

[4] 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.

[5] 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.

[6] Focardi, R., R. Gorrieri, R. Lanotte, A. Maggiolo-Schettini, F. Martinelli, S. Tini and Enrico Tronci: Formal Models of Timing Attacks on Web Privacy. Electr. Notes Theor. Comput. Sci. (ENTCS) 62, 2001.

[7] Gorrieri R. and F. Martinelli: A simple framework for real-time cryptographic protocol analysis with compositional proof rules. Science of Computer Programming archive Volume 50, Issue 1-3, 2004.

[8] 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.

[9] Gruska D.P.: Process Algebra Contexts and Security Properties. Fundamenta Informaticae, vol. 102, Number 1, 2010.

[10] Gruska D.P.: Quantifying Security for Timed Process Algebras, Fundamenta Informaticae, vol. 93, Numbers 1-3, 2009.

[11] Gruska D.P.: Probabilistic Information Flow Security. Fundamenta Informaticae, vol. 85, Numbers 1-4, 2008.

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

[13] 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.

[14] Lowe G.: Quantifying information flow. In Proc. IEEE Computer Security Foundations Workshop, 2002.

[15] Martinelli F. and I. Matteucci: Through Modeling to Synthesis of Security Automata. Proc. of STM 2006, ENTCS Volume 179, 6 July 2007.

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

[17] Schneider, F. B.: Enforceable Security Policies. ACM Transactions on Information and System Security, Volume 3 Issue 1, Feb. 2000.

[18] Volpano D. M. and G. Smith: Eliminating Covert Flows with Minimum Typings. In Proc. CSFW, 1997.