Fundamenta Informaticae 120 (2012) 1–14
DOI 10.3233/FI-2012-765
IOS Press

Quantification of Positive and Negative Attacker's Information*

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

Abstract. Different techniques for expressing an amount of information on secrete data which can be obtained by a process observation are presented. They are based on information theory and they express certainty about sets of private actions which execution is guaranteed by a given observation and sets of actions which execution is excluded by a given observation. Moreover, the case when an intruder has same preliminary belief on secrete data is discussed. It is shown how the presented technique could be applied for such case. As regards working formalism, probabilistic process algebra is used for description of systems as well as attacker's belief.

Keywords: probabilistic process algebras, information flow, opacity, security, belief

1. Introduction

In [10] we have studied information flows by means of two sets - the set of private actions which execution is guaranteed by a given observation of public actions and the set of actions which execution is excluded by a given observation. This approach, similarly to traditional security properties, could be 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 - it at least (s)he can learn what is not the correct password. On the other side, it can happen that the set of excluded or gained (or both) passwords are empty but a membership of the password to some of them is very likely since not certain. There are several ways how to overcome these disadvantages. An amount of leaked information could be expressed by means of the

*Work supported by the grant VEGA 1/1333/12.
†Address for correspondence: Institute of Informatics, Comenius University, Mlynska dolina, 842 48 Bratislava, Slovakia

Shannon's information theory as it was done, for example, in [4, 5] 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 [12]. Resulting techniques lead to quantifications of how many bits of private information can leak or how probable is that an intruder can learn some secrete property over processes traces.

The aim of this paper is to enrich the formalism presented in [10] by expressing certainty about the sets of gained and excluded actions by means of information theory. In this way we can describe possible attacks or security holes which cannot be captured otherwise. For example, it might happen that either the set of gained or excluded (or both) private actions are empty what would lead to no leakage of private information but probability that some private action belongs to either of them is so high (but still not certain) that it could significantly help an attacker.

Moreover, we will also consider intruders with a preliminary belief on secret data/action and we will develop a way how to express leakage of information (information flow) in such cases. Traditional approach is not sufficient since if the intruder performs an attack according to his belief and the attack fails, entropy of private data could increase what would lead to the conclusion, that there is no leakage of private information (due to higher entropy after the attack then before it). On the other side, the attacker can still obtain some information and so there is some undescribed leakage if information. We will show how presented quantification techniques could be applied for such cases.

The paper is organized as follows. In Section 2 we describe the probabilistic process algebra pCCS which will be used as the basic formalism. In Section 3 we will quantify by means of information theory uncertainty of sets of gained and excluded private actions for a given observation of public actions. Moreover, we will present a way how to quantify particular leakage of partial information of complex secrete data - sequences of private actions. In section 4 we study and model attackers with belief.

2. Probabilistic Process Algebra

In this section we define the Probabilistic Process Algebra, pCCS for short, which is based on Milner's CCS (see [16]). First we assume a set of atomic action symbols A not containing symbol τ and such that for every $a \in A$ there exists $\overline{a} \in A$ and $\overline{\overline{a}} = a$. We define $Act = A \cup \{\tau\}$. We assume that $a, b, \ldots$ range over A and $u, v, \ldots$ range over Act. Assume the signature $\Sigma = \bigcup_{n\in\{0,1,2\}} \Sigma_n$, where

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

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, $S : Act \to Act$ are such that $\overline{S(a)} = S(\bar{a})$ for $a \in A$ and $S(\tau) = \tau$.

The set of CCS 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 CCS terms, $\mu X-$ is the binding construct, $op \in \Sigma$.

We will use an usual definition of opened and closed terms where μX is the only binding operator. Closed terms which are guarded (each occurrence of X is within some subexpression $u.A$ are called CCS processes. Note that Nil will be often omitted from processes descriptions and hence, for example, instead of $a.b.Nil$ we will write just $a.b$. Structural operational semantics for processes by given labeled transition systems. The set of terms represents a set of states, labels are actions from Act (see [16]).

The transition relation $\rightarrow$ is a subset of CCS $\times$ Act $\times$ CCS. We write $P \stackrel{x}{\rightarrow} P'$ instead of $(P, x, P') \in \rightarrow$ and $P \stackrel{x}{\not\rightarrow}$ if there is no P' such that $P \stackrel{x}{\rightarrow} P'$. The meaning of the expression $P \stackrel{x}{\rightarrow} P'$ is that the term P can evolve to P' by performing action x, by $P \stackrel{x}{\rightarrow}$ we will denote that there exists a term P' such that $P \stackrel{x}{\rightarrow} P'$.

Now we add probabilities to CCS calculus. We will follow alternating model (the approach presented in [14]) which is neither reactive nor generative nor stratified (see [15]) but instead of that it will be based on separation of probabilistic and nondeterministic transitions and states. Probabilistic transitions are not associated with actions but they are labeled 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).

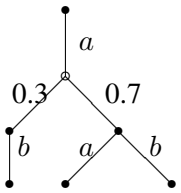

Figure 1. $a.(0.3.b.Nil \oplus 0.7.(a.Nil + b.Nil))$

Formally, to add probabilities to CCS 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 pCCS we will denote the set of all probabilistic and non-probabilistic processes and all definitions and notations for CCS processes are extended for pCCS ones. We need new transition rules for pCCS processes.

$$\frac{}{P_N \stackrel{1}{\rightarrow} P_N} \quad A1 \qquad\qquad \frac{}{\bigoplus_{i\in I} q_i.P_i \stackrel{q_i}{\rightarrow} P_i} \quad A2$$

$$\frac{P \stackrel{q}{\rightarrow} P', Q \stackrel{r}{\rightarrow} Q'}{P \mid Q \stackrel{q.r}{\rightarrow} P' \mid Q'} \quad Pa$$

For probabilistic choice we have the rule $A2$ and for a probabilistic transition of two processes running in parallel we have the rule Pa. The technical rule $A1$ enables parallel run of probabilistic and non-probabilistic processes by allowing to non-probabilistic processes to perform $\stackrel{1}{\rightarrow}$ transition and hence the rule Pa 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 pCCS 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.

To express what an observer can see from system behaviour we will define modified transitions $\stackrel{x}{\Rightarrow}_M$ which hide actions from M (except τ and probabilities). Formally, we will write $P \stackrel{x}{\Rightarrow}_M P'$ for $M \subseteq A$ iff $P \stackrel{s_1}{\rightarrow}\stackrel{x}{\rightarrow}\stackrel{s_2}{\rightarrow} P'$ for $s_1, s_2 \in (M \cup \{\tau\} \cup (0,1])^\star$ 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 $s \sqsubseteq s'$, $s, s' \in (Act \cup (0,1])^\star$ we will denote that s is a prefix of s'. By $Sort(P)$ we will denote the set of actions from A which can be performed by P i.e. $Sort(P) = \{x | P \stackrel{s.x}{\longrightarrow}$ for some $s \in (Act \cup (0,1])^\star$ and $x \in A\}$.

Let $s \in (Act \cup (0,1])^\star$. By $s|_B$ we will denote the sequence obtained from s by removing all actions not belonging to B and we will write $x \in s$ if the sequence s contains x as its element.

Definition 2.1. The set of weak traces of process P with respect to the set $M, M \subseteq A$ is defined as $Tr_{wM}(P) = \{s \in A^\star | \exists P'.P \stackrel{s}{\Rightarrow}_M P'\}$. Instead of $Tr_{w\emptyset}(P)$ we will write $Tr_w(P)$.

Two processes P and Q are weakly trace equivalent with respect to M ($P \approx_{wM} Q$) iff $Tr_{wM}(P) = Tr_{wM}(Q)$.

3. Non-interference

To define non-interference for 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 and sequences of private actions from H, we will use notation $\tilde{l}, \tilde{l}', \ldots$ for sequences from $L^\star$ and $\tilde{h}, \tilde{h}', \ldots$ for sequences from $H^\star$, respectively. The set of actions could be divided to more than into two subsets, what would correspond into more levels of classification. All the following concepts could be naturally extended for such setting.

3.1. Gained and excluded 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). Here we slightly modify the definition from [10].

Definition 3.1. Let $P \in CCS$ 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 3.1 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 3.2. 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 $g(P,\tilde{l}) = \emptyset, g(P',\tilde{l}) = \{h\}$.

By observing process an observer can obtain information not only about actions which had to be performed but also about actions which could be excluded (they could not be performed). We start with a motivation example taken from [10].

Example 3.3. (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 $g(P,\tilde{l}) = \emptyset$ and hence to describe the knowledge obtained by an attacker observing $\tilde{l}$ we need a new concept.

Definition 3.4. Let $P \in pCCS$. 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 3.5. 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 $g(R,\tilde{l}) = \emptyset$ for every $\tilde{l}$. This means that, in general, the concept of excluded actions cannot be covered by gained actions.

In some sense $g(P,\tilde{l})$ and $e(P,\tilde{l})$ are complementary as it is stated in the following theorem (for the proof see [10]).

Theorem 3.6. 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 $\emptyset \subseteq g(P,\tilde{l}) \cup e(P,\tilde{l}) \subseteq H$.

Now let us return to excluded actions. The following example gives a motivation for the next subsection.

Example 3.7. Let $P = l_1.h_1.l_2.Nil + .l_1.h_2.l_2.Nil$ and $P' = 0.99.l_1.h_1.l_2.Nil \oplus 0.01.l_1.h_2.l_2.Nil$. Let $\tilde{l} = l_1.l_2$ then we have $g(P, \tilde{l}) = g(P', \tilde{l}) = \emptyset$. If we take into account also probabilities, them performance of action h_2 in case of P' and observation $\tilde{l}$ is more likely then h_2. To precisely distinguish these two cases we will exploit information theory.

3.2. Information theory

To express quantity of information flow we will exploit Schannon information theory (see [18]). Let X be a discrete random variable and let x ranges over the set of values which X may take. By $p(x)$ we will denote probability that X takes the value x.

Self-information (or surprisal) is a measure of the information content associated with the outcome of the random variable X. It is defined as $\mathcal{H}(x) = \log_b \frac{1}{p(x)}$. We put $\mathcal{H}(x) = \infty$ if $p(x) = 0$. The information entropy (also called self-information or a measure of uncertainty) of the variable X is denoted $\mathcal{H}(X)$ and is defined as $\mathcal{H}(X) = \sum_x p(x).\log_b \frac{1}{p(x)}$. We put $p(x).\log_b \frac{1}{p(x)} = 0$ if $p(x) = 0$. We will work with the base b of $\log_b$ equal to 2 and hence the unit of the information entropy will be one bit. Sometimes we will write $\mathcal{H}(p_1, \ldots, p_n)$ instead of $\mathcal{H}(X)$ if probabilities of values of X are $p_1, \ldots, p_n$.

3.3. Quantification of gained and excluded actions

To define quantification of gained and excluded actions , we need some preparatory work. Let P be a pCCS process and let $P \stackrel{x_1}{\to} P_1 \stackrel{x_2}{\to} P_2 \stackrel{x_3}{\to} \cdots \stackrel{x_n}{\to} P_n$, where $x_i \in Act \cup (0, 1]$ 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 Act i.e. $label(P.x_1.P_1.x_2 \ldots x_n.P_n) = x_1 \ldots . x_n|_{Act}$ 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 [17] a technique of schedulers are used to resolve the nondeterminism and in [9] 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 $P \stackrel{x}{\to} P'$ is indexed by k (i.e. corresponding $i_l = k$) then transition $P + Q \stackrel{x}{\to} P'$ is indexed by $k.1$ and transition $Q + P \stackrel{x}{\to} P'$ is indexed by $k.2$. If transition $P_i \stackrel{x}{\to} P'$ is indexed by k then transition $\bigoplus_{i \in I} q_i.P_i \stackrel{x}{\to} P'$ is indexed by $k.i$, and if transitions $P \stackrel{x}{\to} P'$ and $Q \stackrel{x}{\to} Q'$ 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 3.8. Let $P \in pCCS$ and $\tilde{l} \in Tr_{wH}(P)$. We define

$$P_h^{\tilde{l}} = Pr(C), \text{ where } C = \{c | label(c) = s, s|_L = \tilde{l}, h \in s\}$$

and

$$P_{\neg h}^{\tilde{l}} = Pr(C), \text{ where } C = \{c | label(c) = s, s|_L = \tilde{l}, h \notin s\}.$$

Example 3.9. Let $P = 0.1.l_1.h.l_2.Nil \oplus 0.3.l_1.l_2.Nil \oplus 0.5.l_1.l_2.h.Nil \oplus 0.1.l_1.l_3.h.Nil$ and $\tilde{l} = l_1.l_2$. Then we have $P_h^{\tilde{l}} = 0.6$ and $P_{\neg h}^{\tilde{l}} = 0.3$.

Now we are ready to define surprisal ($\mathcal{H}(P_h^{\tilde{l}})$) of (private) action h in the case that sequence of (public) actions $\tilde{l}$ is observed.

Definition 3.10. We define surprisal $\mathcal{H}(P_h^{\tilde{l}})$ for process P and observation $\tilde{l}$ as

$$\mathcal{H}(P_h^{\tilde{l}}) = \log \frac{P_h^{\tilde{l}} + P_{\neg h}^{\tilde{l}}}{P_h^{\tilde{l}}}.$$

Example 3.11. Let us consider process P from Example 3.9. We have $\mathcal{H}(P_h^{\tilde{l}}) = 0.58$.

In general, high values of surprisal $\mathcal{H}(P_h^{\tilde{l}})$ mean lower probability that action h was performed when $\tilde{l}$ is observed i.e. that an intruder can learn less information about this private action.

There is a connection between surprisal as a quantitative property and sets of gained as qualitative properties as it is stated in the following theorem.

Theorem 3.12. $h \in g(P, \tilde{l})$ iff $\mathcal{H}(P_h^{\tilde{l}}) = 0$.

Proof:

$\Rightarrow$ Let $h \in g(P, \tilde{l})$ i.e. there exists at least one computational path c such that $label(c) = s, s|_L = \tilde{l}, h \in s$ and there is not a path c' such that $label(c') = s, s|_L = \tilde{l}, h \notin s$. That means that $P_h^{\tilde{l}} > 0$ and $P_{\neg h}^{\tilde{l}} = 0$. From that we $\mathcal{H}(P_h^{\tilde{l}}) = 0$.

$\Leftarrow$ We show that if $\mathcal{H}(P_h^{\tilde{l}}) > 0$ then $h \notin g(P, \tilde{l})$. $\mathcal{H}(P_h^{\tilde{l}}) > 0$ means that $P_h^{\tilde{l}} > 0$ and $P_{\neg h}^{\tilde{l}} > 0$ i.e. there exist two traces of P such that their public part is equal to $\tilde{l}$ and one contains h and the other one does not contain it, hence $h \notin g(P, \tilde{l})$. □

Now we can return to Example 3.7.

Example 3.13. (continuation of Example 3.7) Let $P = 0.99.l_1.h_1.l_2.Nil \oplus 0.01.l_1.h_2.l_2.Nil$. Let $\tilde{l} = l_1.l_2$ then we have $g(P, \tilde{l}) = \emptyset$ but $\mathcal{H}(P_{h_1}^{\tilde{l}}) \doteq 6.64$ and $\mathcal{H}(P_{h_2}^{\tilde{l}}) \doteq 0.14$.

In general, longer observations lead to bigger sets of gained actions as it is stated in the following proposition taken from [10].

Proposition 3.14. Let $\tilde{l}, \tilde{l'} \in Tr_{wH}(P)$ and $\tilde{l} \sqsubseteq \tilde{l'}$ then we have

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

As regards the quantification it does not hold that longer observations lead to smaller or higher surprisal as it is clear from the following example.

Example 3.15. Let $P = 0.3.h.l_1.Nil \oplus 0.1.h.l_1.l_2.Nil \oplus 0.3.l_1.Nil \oplus 0.3.l_1.l_2.Nil$. Then $\mathcal{H}(P_h^{l_1}) = 1.32$ and $\mathcal{H}(P_h^{l_1.l_2}) = 2$. That means that in this case the longer observation leads to higher surprisal of h. Now let as consider process $R = 0.7.h.l_1.Nil \oplus 0.3.l_1.Nil$. Then $\mathcal{H}(R_h^{l_1}) = 1.74$ and $\mathcal{H}(R_h^{l_1.l_2}) = 0$. Here, on the contrary, the longer observation leads to the lower surprisal of h.

Now we can show how quantification of 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 [6]). 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 $P \xrightarrow{a} P'$ 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 3.16. Let $P \in CCS$. Then $P \in SNNI$ iff $P \setminus H \approx_w P/H$.

Theorem 3.17. If $P \in SNNI$ then $g(P, \tilde{l}) = \emptyset$ for every $\tilde{l} \in L^\star$.

Proof:
The proof is based on the idea that if process has property SNNI then the set of gained actions has to be empty for for any public observation (see [10]). □

Corollary 1. If $P \in SNNI$ then $\mathcal{H}(P_h^{\tilde{l}}) > 0$ for every $\tilde{l} \in L^\star$ and $h \in H$.

Proof:
The proof follows from Theorem 3.12 and 3.17. Note that the inverse of the Theorem 3.17 as well as this corollary does not hold. Let $P = 0.5.h_1.l.Nil \oplus 0.5.h_2.l.Nil$. Then $P \notin SNNI$ but $\mathcal{H}(P_{h_1}^{l}) = \mathcal{H}(P_{h_2}^{l}) = 1$. □

Similarly to gained actions also excluded action could be quantified by surprisal of $\neg h$.

Definition 3.18. We define surprisal $\mathcal{H}(P_{\neg h}^{\tilde{l}})$ for process P and observation $\tilde{l}$ as

$$\mathcal{H}(P_h^{\tilde{l}}) = \log \frac{P_h^{\tilde{l}} + P_{\neg h}^{\tilde{l}}}{P_{\neg h}^{\tilde{l}}}.$$

For this quantification we have a similar property as it is stated in Theorem 3.12 holds and also its proof is similar.

Theorem 3.19. Let $h \in e(P, \tilde{l})$ then we have $\mathcal{H}(P_{\neg h}^{\tilde{l}}) = 0$.

Example 3.20. Let $P = 0.6.l_1.h_1.l_2.Nil \oplus 0.4.l_1.l_2.Nil$, $H = \{h_1, h_2\}$ and let $\tilde{l} = l_1.l_2$ then we have $e(P, \tilde{l}) = h_2$ and $\mathcal{H}(P_{\neg h_1}^{\tilde{l}}) \doteq 1.32$ and $\mathcal{H}(P_{\neg h_2}^{\tilde{l}}) = 0$.

An intruder can learn nothing (neither positive nor negative information) about occurrence of private action h by observing $\tilde{l}$ if $h \notin g(P,\tilde{l}) \cup e(P,\tilde{l})$. That means that it should hold $\mathcal{H}(P_h^{\tilde{l}}) > 0$ and $\mathcal{H}(P_{\neg h}^{\tilde{l}}) > 0$. Since both variables $P_h^{\tilde{l}}$ and $P_{\neg h}^{\tilde{l}}$ are correlated this means that minimal information can be obtained if both values are equal to 1.

Overall security of process P with respect to occurrence of private action h (it will be denoted by $S_h(P)$ can be expressed as

$$S_h(P) = \min\{\mathcal{H}(P_h^{\tilde{l}}) | \tilde{l} \in Tr_{wH}(P)\}.$$

If $S_h(P) = 0$ then there exists an observation such that an intruder can learn from that observation that h was performed. Moreover, we have the following corollary which follows from Theorem 3.12 and 3.17.

Corollary 2. If $S_h(P) = 0$ for some $h \in H$ then $P \notin SNNI$.

Note that definition of process security ($S_h(P)$) requires evaluation of $\mathcal{H}(P_h^{\tilde{l}})$ for all public traces $\tilde{l}$ of P. Actually if we want to check the security of finite state process (i.e. whether $S_h(P) = 0$) the number of traces which should be checked could be reduced. Moreover, it is meaningful to restrict possible lengths of observations hence obtain practically applicable measure of security.

Now suppose that an intruder tries to learn more than just an occurrence of a private action. For example, suppose that (s)he tries to learn a private key of length k (password, pin code etc - from now on we will call it as the private key). Each bit of the key represents a private data (action h_0 or h_1 and one and only one of them occurs in every execution trace). In the subsequent definitions we will show how the above mentioned techniques could be exploited for expression of how much information about the key can the intruder learn from observation(s) of public actions.

Definition 3.21. Let $P \in pCCS$, $H = \{h_{1_0}, h_{1_1} \ldots h_{k_0}, h_{k_1}\}$ and $\tilde{l} \in Tr_{wH}(P)$. Let n denotes the number corresponding to sequence of bits $h_{1_i} \ldots h_{k_i}$

$$P_n^{\tilde{l}} = Pr(C), \text{ where } C = \{c | label(c) = s, s|_L = \tilde{l}, s|_H = h_{i_1} \ldots h_{i_k}\}$$

Now we define a discrete random variable X corresponding to distribution of possible private keys. By $p(n)$ we denote probability that $X = n$.

Definition 3.22. Let $P \in pCCS$, $H = \{h_{1_0}, h_{1_1} \ldots h_{k_0}, h_{k_1}\}$ and $\tilde{l} \in Tr_{wH}(P)$. Let n denotes the number corresponding to sequence of bits $h_{1_i} \ldots h_{k_i}$. We define $p(n)$ as follows

$$p(n) = \frac{P_n^{\tilde{l}}}{\sum_{i=0}^{2^k} P_i^{\tilde{l}}}$$

Now we define entropy of variable X.

Definition 3.23. Let $P \in pCCS$, $H = \{h_{1_0}, h_{1_1} \dots h_{k_0}, h_{k_1}\}$ and $\tilde{l} \in Tr_{wH}(P)$. Let X is a discrete random variable X corresponding to probabilities $p(n)$. We define entropy $\mathcal{H}(P^{\tilde{l}})$ for process P and observation $\tilde{l}$ as

$$\mathcal{H}(P^{\tilde{l}}) = \sum_{i=0}^{2^k} p(i).\log\frac{1}{p(i)}$$

Suppose that at the beginning an attacker has no information about the value of the private key, i.e. all keys seem to be equally probable. This corresponds to maximal entropy (equal to k in this case). The entropy is lower after an observation from which the intruder can learn something at least about one bit of the key. This is formalized in the following theorem.

Theorem 3.24. Let $P \in pCCS$, $H = \{h_{1_0}, h_{1_1} \dots h_{k_0}, h_{k_1}\}$ and $\tilde{l} \in Tr_{wH}(P)$. Let $\mathcal{H}(P^{\tilde{l}}_{h_{i_0}}) < 1$ or $\mathcal{H}(P^{\tilde{l}}_{h_{i_1}}) < 1$ for some i. Then it holds $\mathcal{H}(P^{\tilde{l}}) < k$.

Proof:
If $\mathcal{H}(P^{\tilde{l}}_{h_{i_0}}) < 1$ or $\mathcal{H}(P^{\tilde{l}}_{h_{i_1}}) < 1$ that means that probabilities of i-th bit of the private key being one or zero are different hence resulting entropy is smaller than its maximal value k which is reached when all private keys i.e. also all bits of private keys are equally probable. □

By similar argument as it was used in the previous theorem we get the following theorem.

Theorem 3.25. Let $P \in pCCS$, $H = \{h_{1_0}, h_{1_1} \dots h_{k_0}, h_{k_1}\}$ and $\tilde{l} \in Tr_{wH}(P)$. $\mathcal{H}(P^{\tilde{l}}) < k$. Then there exists i such that $\mathcal{H}(P^{\tilde{l}}_{h_{i_0}}) < 1$ or $\mathcal{H}(P^{\tilde{l}}_{h_{i_1}}) < 1$.

If an intruder can gain or exclude at least one private action that overall entropy is decreased by 1.

Theorem 3.26. Let $P \in pCCS$, $H = \{h_{1_0}, h_{1_1} \dots h_{k_0}, h_{k_1}\}$ and $\tilde{l} \in Tr_{wH}(P)$. Let $h_{k_j} \in e(P, \tilde{l})$ or $h_{k_j} \in g(P, \tilde{l})$ for some k and $j \in \{0, 1\}$ then $\mathcal{H}(P^{\tilde{l}}) \leq k - 1$.

Proof:
Let $h_{k_j} \in g(P, \tilde{l})$ (for excluded actions is the proof similar). This means that an intruder can always learn h_k-the bit of the private key. Hence at most $k - 1$ bits are uncertain and hence maximal possible entropy is equal to $k - 1$. □

The previous theorem can be further generalized and hence we obtain the following corollary.

Corollary 3. Let $P \in pCCS$, $H = \{h_{1_0}, h_{1_1} \dots h_{k_0}, h_{k_1}\}$ and $\tilde{l} \in Tr_{wH}(P)$. Let $H' \subseteq e(P, \tilde{l})$ then $\mathcal{H}(P^{\tilde{l}}) \leq |H'|$.

All the above definitions and theorems could be extended from one observation $\tilde{l}$ of public actions to set of observations. In this way we could express how much information could be obtained from process observations. Now we will focus on intruders with a preliminary belief.

4. Attacker's preliminary belief

Suppose that an attacker can perform an attack (i.e. (s)he can force system to perform a sequence of public actions) from a given set $N, N \subseteq L^{\star}$. According to the previously developed technique it would be natural as a first attempt to chose $\tilde{l} \in N$ such that $\mathcal{H}(P)^{\tilde{l}}_h = \min\{\mathcal{H}(P)^{\tilde{l'}}_h | \tilde{l'} \in N\}$ if the attacker is interested in occurrence of action h. This uncertainty approach is not adequate for cases where an intruder has some preliminary belief about private/secrete data/actions. If an attack is performed according that belief/assumption but then it turns out that the assumption was incorrect uncertainty approach indicates no leak of private information despite the fact that some data/action can be excluded by the intruder (see [5] for details). In other words, if uncertainty on secrete data after observation (attack) is higher then before it, traditional approach would lead to a conclusion that there is no information flow.

One way how to overcome that disadvantage is to exploit concepts of excluded actions. For example, if an intruder believes that the most likely password is A, say, that with probability 0.9 and other passwords are equali probable, then if his/her belief turned out to be false, then entropy of possible passwords is higher after then before the first try with A. In this case we can still express obtained information by the set of excluded actions which now contains A.

In general, if attacker's belief could be expressed by probabilities associated with every $\tilde{l}_i, i \in N$. Let $p(\tilde{l}_i) \geq p(\tilde{l}_j)$, for $i < j$. We expect that the attacker first tries $\tilde{l}_1$ then $\tilde{l}_2$ and so on. Doing so some knowledge is accumulated, say in sets G_k and E_k of gained and excluded actions, respectively. Quality of the belief can be expressed by condition

$$g(P, \tilde{l}_{i+1}) \not\subset G_i \text{ or } e(P, \tilde{l}_{i+1}) \not\subset E_i$$

where G_i and E_i are sets of gained and excluded actions after i-th observation, respectively. This condition guaranties that some new knowledge (enlargement of G_i or E_i) is obtained by $\tilde{l}_{i+1}$.

On the other side, it is often not the case that a private action can be gained or excluded even if intruders belief is almost "correct". But we can still measure quality of the belief similarly by quantifications of $g(P, \tilde{l}_{i+1})$ and $e(P, \tilde{l}_{i+1})$.

Now we will further develop these ideas. We will formalize an assumption that an attacker can influence public behavior of systems. We will assume that (s)he can actively communicate with the system by means of chosen public input and output actions.

Let $\tilde{l_b}$ is subsequence of $\tilde{l}, \tilde{l} \in Tr_{wH}(P)$ i.e. $\tilde{l}$ can be obtained from $\tilde{l_b}$ by inserting public actions before, among and after elements of $\tilde{l_b}$ (we will denote this by $\tilde{l_b} \preceq \tilde{l}$). We suppose that actions contained in $\tilde{l_b}$ are those ones which are under the control of the intruder.

Suppose that an intruder tries to learn whether private action h is performed and (s)he believes that best way is to try $\tilde{l_b}$ first. Now we define a way how to decide whether his/her belief is appropriate. First we need some preparatory work.

Definition 4.1. Let $P \in pCCS$. We define

$$P^{\tilde{l_b}}_h = Pr(C), \text{ where } C = \{c | label(c) = s, \tilde{l_b} \preceq s|_L, h \in s\}$$

and

$$P^{\tilde{l_b}}_{\neg h} = Pr(C), \text{ where } C = \{c | label(c) = s, \tilde{l_b} \preceq s|_L, h \notin s\}.$$

We say that belief $\tilde{l_b}$ is appropriate if $\mathcal{H}(P_h^{\tilde{l_b}}) < 1$ where $\mathcal{H}(P_h^{\tilde{l_b}})$ is defined with by means of $P_h^{\tilde{l_b}}$ and $P_{\neg h}^{\tilde{l_b}}$.

In fact, if an intruder tries $\tilde{l_b}$ then occurrence of h is more likely - before this trial it was half to half.

Now we will generalize this idea even further. Suppose that an intruder ties to learn a private key. Following the technique introduced in definitions 3.22 and 3.23 we will suppose that a complete secrete is a sequence s of private actions/data (for example, sequence of bits of a private key, sequence of characters of a password). We can express this by pCCS process $S = s.Nil$. We will express intruder's belief again as pCCS process B. Both these process are parts of a system to be observed by an intruder A (see Fig. 2).

Figure 2. Attacker with and without preliminary belief

Now we will show a way how to formalize how much information an intruder with the belief can learn. Let U, B, S are pCCS process such that $Sort(U) \cup Sort(B) \cup Sort(S) \subseteq H$ and probabilities of all possible secrete sequences for U are equal (uniform distribution). Intruders belief expressed by process B and S do not need to have uniform distributions.

Definition 4.2. Let $P, B, U, S \in pCCS$ and $\tilde{l} \in Tr_{wH}(P|U)$. We say that intruder's belief B is appropriate as regards h and $\tilde{l}$ if $\mathcal{H}((P|U)_h^{\tilde{l}}) \geq \mathcal{H}((P|B)_h^{\tilde{l}}) \geq \mathcal{H}((P|S)_h^{\tilde{l}})$.

Note that attacker's belief has meaning only if the belief consists only of several "suspicious" sequences - at least their number should be much smaller than a number of all possible sequences or belief's entropy (if a probability is associated with each sequence of belief) is much smaller than entropy of all possible sequences with uniform distribution.

5. Conclusions and related works

We have presented quantification of several security concepts based on an information flow caused by observations of public actions. They express certainty on the set of private actions which were performed (the gained sets) or certainty of the set of private actions which could be excluded by an intruder observing systems public actions (the 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 as in property SNNI and opacity [13]).

The notion of 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. The presented formalism can capture also cases when the membership to such a set is not certain but only very likely what is information, which could help intruder very significantly. Note that without "quantification" of the set membership, we can consider a system to be secure despite the fact that a successful attack could be possible.

Concepts of the 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. The same holds for their quantified variants.

Moreover we have presented a way how to handle intruders with preliminary belief on secrete data what is rather common case (for example, some possibilities are believed to be more probable and hence the intruder tries to disprove or prove them first). We define a way how to measure quality of attacker's belief and we can exploit also sets of excluded actions and surprisal of its membership and hence results of even unsuccessful attacks bring always some measurable information for the attacker. Note, that this is not the case of traditional uncertainty approach by which after a failed attack entropy of secrete might be higher then before it, what would traditionally lead to a conclusion, that the attacker learned nothing by the attack. As an extension of the presented work we plan to elaborate more sophisticated ways how to chose a next attempt for an attack (for attackers with and without preliminary beliefs) if the attack fails.

As regards related works, the presented results could be compared with other approaches which can be found in the literature. We have shown direct relation between security property Strong Nondeterministic Non-Interference (see [6]) and surprisal $\mathcal{H}(P_h^{\tilde{l}})$ as well as security $S_h(P)$ of process P by Corollary 1 and 2. Hence presented quantification could be viewed also as "quantification" of SNNI property. Security property Non-Deducibility on Composition (NDC for short, see [7]) represents an alternative approach. Process has this property if for every high level user A (i.e. the one capable to perform only private i.e. high level actions), the low level view of the behaviour (seeing only public i.e. low level actions) of P is not modified (in terms of weak trace equivalence) by the presence of A, i.e. $(P|A) \setminus H \approx_w P \setminus H$. Since clearly $Tr_w(P) \subseteq Tr_w(P|A)$ a quantification of NDC could be done by quantification of difference between these two sets for every high level action or set of high level actions by similar techniques we have presented here. In [1] an alternative approach, probabilistic noninterference, is studied.

In [12] property Non-information Flow (NIF) is quantified for (timed) process algebra by means of probability theory. P has NIF property if from its observation it cannot be deduced that some of given private actions $\mathcal{S}$ were performed. For a special case if an observer does not see neither high level actions nor action τ, NIF property could model the set of gained actions. Its quantification expresses probability of an occurrence of a high level action. One of the most general security properties is opacity ([2] and [3]) i.e. many other properties could be viewed as special cases of it. It assumes predicate ϕ over process traces which express some property (an execution of one or more classified actions, an execution of actions in a particular classified order which should be kept hidden, etc.). The predicate is opaque if an observer cannot deduce the validity of ϕ i.e. if there are always two traces $w, w' \in Actt^\star$ such that $\phi(w), \neg\phi(w')$ and the traces cannot be distinguished by the observer i.e. $\mathcal{O}(w) = \mathcal{O}(w')$, where $\mathcal{O}(w)$ express what the observer can sees if process performs w. In [11] quantification of opacity for timed process algebra is defined. It express how much information on predicate validity can observer obtain. If the predicate express that a trace contains (or does not contain) a particular high level action h we could quantify (by quantified opacity) also certainty of h belonging to the set of gained (or excluded) actions. On the other side, by the presented technique we can naturally and simply cover also entropy for occurrences of sequences of high level actions which would be rather clumsy with quantified opacity. As regards quantification for intruders with belief we do not know any other work as the one in [5] but this deals only with imperative languages.

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] 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] 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.

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

[6] 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.

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

[8] 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.

[9] 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

[10] Gruska D.P.: Gained and Excluded Private Actions by Process Observations, Fundamenta Informaticae, vol. 109, No. 3, 2011. 281-295

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

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

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

[14] 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.

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

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

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

[18] Shannon, C. E.: A mathematical theory of communication. Bell System Technical Journal, vol. 27, 1948.