Quantification of Attacks with and without Belief *

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 of 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. Probabilistic process algebra is used for description of systems as well as attackers belief.

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

1 Introduction

In [Gru11] 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 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. By means of the Shannon's information theory it could be expressed an amount of information leaked as it was done, for example, in [CHM07,CMS09] for simple imperative languages and in [Gru08] for process algebra. Another possibility is to exploit probabilistic theory as it was used for process algebras in [Gru09]. 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 [Gru11] by expressing certainty about the sets of gained and excluded actions by means

* Work supported by the grant VEGA 1/0688/10.

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 182-193

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 of information.

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 the 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. We end the section by modeling attackers with belief.

2 Probabilistic Process Algebra

In this section we define the Probabilistic Process Algebra, pCCS for short, which will is based on Milner's CCS (see [Mil89]). 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

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

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, $S : Act \to Act$ 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 [Mil89]).

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 $P \stackrel{x}{\rightarrow} P'$.

For $s = x_1.x_2.\ldots.x_n, x_i \in Act$ we write $P \stackrel{s}{\rightarrow}$ instead of $P \stackrel{x_1}{\rightarrow}\stackrel{x_2}{\rightarrow} \ldots \stackrel{x_n}{\rightarrow}$ 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 $P(\stackrel{\tau}{\rightarrow})^* \stackrel{x}{\rightarrow} (\stackrel{\tau}{\rightarrow})^* P'$ and $P \stackrel{s}{\Rightarrow}$ instead of $P \stackrel{x_1}{\Rightarrow}\stackrel{x_2}{\Rightarrow} \ldots \stackrel{x_n}{\Rightarrow}$. By ϵ we will denote the empty sequence of actions, by $Succ(P)$ we will denote the set of all successors of P and $Sort(P) = \{x | P \stackrel{s.x}{\longrightarrow}$ for some $s \in Act^\star\}$.

Now we add probabilities to CCS calculus. We will follow alternating model (the approach presented in [HJ90]) which is neither reactive nor generative nor stratified (see [LN04]) 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).

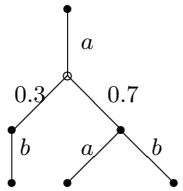

Fig. 1. $a.(0.3.b.Nil \oplus 0.7.(a.Nil + b.Nil))$

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-probabilist processes and all definitions and notations for CCS processes are extended for pCCS ones. We need new transition rules for pCCS processes.

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

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

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 $\xrightarrow{1}$ 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 $P \xrightarrow{s_1} \xrightarrow{x} \xrightarrow{s_2} P'$ for $s_1, s_2 \in (M \cup \{\tau\} \cup (0,1])^\star$ and $P \stackrel{s}{\Rightarrow}_M$ instead of $P \stackrel{x_1}{\Rightarrow}_M \stackrel{x_2}{\Rightarrow}_M \ldots \stackrel{x_n}{\Rightarrow}_M$. 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 (except τ) which can be performed by P i.e. $Sort(P) = \{x | P \xrightarrow{s.x}$ for some $s \in (Act \cup (0,1])^\star$ and $x \neq \tau\}$.

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 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 $Tr_{wM}(P) = Tr_{wM}(Q)$. Again we will write $\approx_w$ instead of $\approx_{w\emptyset}$.

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 $L \cup \{t\}$ and sequences of private actions from H, we will use notation $\tilde{l}, \tilde{l'}, \ldots$ for sequences from $(L \cup \{t\})^\star$ 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 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 [Gru11].

Definition 2. *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 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 1. Let $P = l_1.h.l_2.Nil + l_1.l_2.Nil$ and $P' = l_1.h.h'.l_2.Nil + l_1.h.h''.l_2.Nil$. Let $\tilde{l} = l_1.l_2$ then we have $g(P, \tilde{l}) = \emptyset, g(P', \tilde{l}) = \{h\}$.

An observer by observing process 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 for sure). We start with a motivation example taken from [Gru11].

Example 2 (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 $\overline{l}_{\text{login}}$, $\overline{l}_{\text{access denied}}, l_w, w \in Psw$ are low level actions).

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

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. *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:*

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

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 [Gru11]).

Theorem 1. *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. 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.

3.2 Information theory

To express quantity of information flow we will exploit Schannon information theory (see [Sh48]). 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 the following:

$$\mathcal{H}(x) = \log_b \frac{1}{p(x)}.$$

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 the following:

$$\mathcal{H}(X) = \sum_x p(x).\log_b \frac{1}{p(x)}.$$

We define $p(x).\log_b \frac{1}{p(x)} = 0$ 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 $P \stackrel{x_1}{\to} P_1 \stackrel{x_2}{\to} P_2 \stackrel{x_3}{\to} \ldots \stackrel{x_n}{\to} P_n$, 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. $Prob(P.x_1.P_1.x_2 \ldots x_n.P_n) = 1 \times q_1 \times \ldots \times q_k$ 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 [SL95] a technique of schedulers are used to resolve the nondeterminism and in [**?**] 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 4. *Let 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

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

Definition 5. *We define surprisal* $\mathcal{H}(P^{\tilde{l}}_h)$ *for process* P *and observation* $\tilde{l}$ *as*

$$\mathcal{H}(P^{\tilde{l}}_h) = \log \frac{P^{\tilde{l}}_h + P^{\tilde{l}}_{\neg h}}{P^{\tilde{l}}_h}.$$

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

Theorem 2. *Let* $h \in g(P, \tilde{l})$ *then we have* $\mathcal{H}(P^{\tilde{l}}_h) = 0$.

Proof. Sketch. Let $h \in g(P, \tilde{l})$. Then we know that there do not exist pathes observed as $\tilde{l}$ such that one contains h and another one does not contain h and so $\mathcal{H}(P^{\tilde{l}}_h) = 0$.

Note that a higher surprisal means that an intruder can learn less information about private actions. Now we can return to Example 3.

Example 4. 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 $\mathcal{H}(P^{\tilde{l}}_{h_1}) \doteq 6.64$ and $\mathcal{H}(P^{\tilde{l}}_{h_2}) \doteq 0.14$.

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 [FGM00]). 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 6. *Let* $P \in CCS$. *Then* $P \in SNNI$ *iff* $P \setminus H \approx_w P/H$.

Theorem 3. *If* $P \in SNNI$ *then* $g(P, \tilde{l}) = 0$ *for every* $\tilde{l} \in L^\star$.

Proof. Sketch. 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 [Gru11]).

To quantify excluded actions we define surprisal of $\neg h$.

Definition 7. *We define surprisal* $\mathcal{H}(P^{\tilde{l}}_{\neg h})$ *for process* P *and observation* $\tilde{l}$ *as*

$$\mathcal{H}(P^{\tilde{l}}_h) = \log \frac{P^{\tilde{l}}_h + P^{\tilde{l}}_{\neg h}}{P^{\tilde{l}}_{\neg h}}.$$

For this quantification we have a similar property as it is stated in Theorem 2 holds and also its proof is similar.

Theorem 4. *Let $h \in e(P, \tilde{l})$ then we have $\mathcal{H}(P^{\tilde{l}}_{\neg h}) = 0$.*

Example 5. 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 $\mathcal{H}(P^{\tilde{l}}_{\neg h_1}) \doteq 1.32$ and $\mathcal{H}(P^{\tilde{l}}_{\neg h_2}) = 0$.

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 private key of length k. Each bit of the key represents a private data (action h_0 or h_1). In the subsequent definitions we will show how the above mentioned techniques could be exploited for expression of how much information about the key the intruder can learn from observation(s) of public actions.

Definition 8. *Let 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 $h_{1_i} \ldots h_{k_i}$*

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

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

Definition 9. *Let 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*

$$p(n) = \frac{P^{\tilde{l}}_n}{\sum_{i=0}^{2^k} P^{\tilde{l}}_i}$$

Now we can define entropy of variable X.

Definition 10. *Let 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 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 5. *Let 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 $\mathcal{H}(P^{\tilde{l}}_{h_{i_0}}) \neq 0$ or $\mathcal{H}(P^{\tilde{l}}_{h_{i_1}}) \neq 0$ for some i. Then it holds $\mathcal{H}(P^{\tilde{l}}) < k$.*

Proof. The main idea. Surprisal different from zero means that some information about corresponding bit can be learned and hence a probabilities of the keys are not equal.

Theorem 6. *Let 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 $\mathcal{H}(P^{\tilde{l}}) < k$. Then there exists i such that $\mathcal{H}(P^{\tilde{l}}_{h_{i_0}}) \neq 0$ or $\mathcal{H}(P^{\tilde{l}}_{h_{i_1}}) \neq 0$.*

Proof. Sketch. If entropy is different from its maximal possible value (k) after observation $\tilde{l}$ then at least for one bit some information has to by learned by $\tilde{l}$.

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 at all. Due to the lack of space we omit that and instead we will focus an intruders with a preliminary belief.

3.4 Observation with belief

Suppose that an attacker can perform an attack (i.e. 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 e first attempt to chose $\tilde{l} \in N$ such that $\mathcal{H}(P)^{\tilde{l}}_{h} = \max\{\mathcal{H}(P)^{\tilde{l}'}_{h} | \tilde{l}' \in N\}$ 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 [CMS09] 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.

In this subsection we will assume intruders with the following belief scenarios. 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. 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).

Fig. 2. Attacker with and without preliminary belief

Now we will show two ways how to formalize how much information an intruder with 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 11. *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 $\mathcal{H}((P|U)^{\tilde{l}}_h) \leq \mathcal{H}((P|B)^{\tilde{l}}_h) \leq \mathcal{H}((P|S)^{\tilde{l}}_h)$.*

Now we present an approach which compares probabilities of sequences of public actions for uniform distribution and belief.

Definition 12. *Let Let $R \in pCCS$ and $\tilde{l} \in Tr_{wH}(R)$. We define*

$$[R]^{\tilde{l}} = Pr(C), \text{ where } C = \{c | label(c) = s, s|_L = \tilde{l}\}$$

Let U, B are pCCS process such that $Sort(U) \cup Sort(B) \subseteq H$ and probabilities of all possible secrete sequences for U are equal (uniform distribution). Intruders belief expressed by process B has no uniform distribution.

Definition 13. *Let $P, B, U \in pCCS$ and $\tilde{l} \in Tr_{wH}((P|U) \setminus H)$. We say that an intruder with belief B can learn something about private sequence s by $\tilde{l}$ if $[(P|U) \setminus H]^{\tilde{l}} \neq [(P|B) \setminus H]^{\tilde{l}}$.*

In all above mentioned attacker's scenarios 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.

4 Conclusions

We have presented quantification of several security concepts based on an information flow which can be detected 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 by property SNNI and opacity [Gru07]).

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. But the presented formalism can capture also cases when the membership to such a set us 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 have presented a way how to measure quality of attacker's belief and moreover we can exploit also sets of excluded actions and surprisal of its membership and in this way 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.

References

[CHM07] 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.

[CMS09] Clarkson, M.R., A.C. Myers, F.B. Schneider: Quantifying Information Flow with Beliefs. Journal of Computer Security, to appear, 2009.

[FGM00] 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.

[Gru11] Gruska D.P.: Gained and Excluded Private Actions by Process Observations. To apear in Fundamenta Informaticae, 2011.

[Gru09] Gruska D.P.: Quantifying Security for Timed Process Algebras, Fundamenta Informaticae, vol. 93, Numbers 1-3, 2009.

[Gru08] Gruska D.P.: Probabilistic Information Flow Security. Fundamenta Informaticae, vol. 85, Numbers 1-4, 2008.

[Gru07] Gruska D.P.: Observation Based System Security. Fundamenta Informaticae, vol. 79, Numbers 3-4, 2007.

[HJ90] 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.

[LN04] López N. and Núñez: An Overview of Probabilistic Process Algebras and their Equivalences. In Validation of Stochastic Systems, LNCS 2925, Springer-Verlag, Berlin, 2004

[Mil89] Milner, R.: *Communication and concurrency.* Prentice-Hall International, New York,1989.

[SL95] Segala R. and N. Lynch: Probabilistic Simulations for Probabilistic Processes. Nord. J. Comput. 2(2): 250-273, 1995

[Sh48] Shannon, C. E.: A mathematical theory of communication. Bell System Technical Journal, vol. 27, 1948.