Fundamenta Informaticae 93 (2009) 1–15
DOI 10.3233/FI-2009-86
IOS Press

Quantifying Security for Timed Process Algebras[*]

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

Abstract. A quantification of process's security by quantification of an amount of information flow is defined and studied in the framework of timed process algebras. The resulting quantified security is compared with other (qualitative) security notions. Unprecise and limited observations are defined and discussed.

Keywords: information flow, information theory, opacity, surprisal, uncertainty, security, unprecise and limited observations

1. Introduction

The aim of this paper is to quantify an amount of information flow in the framework of timed process algebras. To express the information flow we will use observation functions and opacity. The observation functions express what an intruder can observe from systems behaviour. They can hide some system activities (for example, internal actions, communications via encrypted channels, actions hidden by a firewall etc) or they can express unprecise observations for which an outcome of an observation is not precisely given i.e. the outcome might be a set of possible results or a random variable. The information flow will be expressed by opacity. It is a qualitative property. A predicate is opaque if from observation of system activities an observer cannot deduce whether the predicate holds or it does not hold. For many applications this property is too restrictive. Predicates (properties) which are not opaque are considered to be insecure since an intruder can detect validity of the predicates by observing system behavior. On

[*] Work supported by European Science Foundation program AutoMathA.

[†] Address for correspondence: Institute of Informatics, Comenius University, Mlynska dolina, 842 48 Bratislava, Slovakia

the other side they are considered to be insecure also in the case that predicates validity can be deduced form observations only with a very low probability or such deductions require an unrealistic number of observations (for example, usually access control processes exhibit some information flow showing which password is not correct but they are still considered to be secure under reasonable password policy). Hence there is a need to quantify an amount of information flow which can be gained from the observations. For this we use Shannon's information theory which enable us to quantify an amount of information which could be obtained about validity of given predicate ϕ and later an amount of direct information flow between processes's inputs and outputs. We will use concepts as *surprisal, uncertainty, mutual information, conditional mutual information* and prove some of security properties based on them. Some comparison with qualitative security notions known in the literature will be presented. Later we will study the case when an observer cannot precisely or unlimitedly long observe systems behavior and the case when information flows are not based on opacity but are based on a conditional mutual information between private and public inputs/outputs.

As regards quantifying information flow there is a number of papers devoted to its analysis in the framework of imperative languages (see [4] for an overview). In [15] an information flow is studied in the framework of process algebras. Particularly, it is investigated how much information i.e. a number of bits can be transmitted by observing some timed system activities. Here we start with a quantification of opacity (opacity was introduced in [1, 2]). Opacity is very general notion and many security properties can be viewed as special cases of opacity (see for example [11], where it was proved that many security properties defined for process algebras can be seen as special cases of opacity and moreover other, stronger security properties can be defined by means of opacity). A weaker form of "probabilistic" opacity developed in the framework of probabilistic timed process algebra has been studied in [10]. The model in which an observer could observe time elapsing between two actions only with limited precision or during a limited time window is studied in [12, 13]. In these papers security properties are considered to be qualitative but capabilities of an intruder have some quantitative aspects but his observations are still precise what represents a different approach that one studied here.

In this paper we will work with the timed process algebra instead of a probabilistic timed process algebra. This decision was taken for the sake of simplicity but all definitions can be easily translated to probabilistic setting (see a discussion in the last section) and hence the notions we developed could be applied for any type of probabilistic process algebra. While in [10] a special probabilistic process algebra was chosen as a basic formalism and we measured probability with which an attacker can learn validity of a predicate over processes traces, here we quantify an amount of information about validity of such predicate which can be gained by attacker who can observe processes traces. We quantify also an amount of information about private inputs which can be gained by an attacker which can see public inputs and outputs.

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 quantified information flow for different observation functions and security requirements. We introduce surprisal and uncertainty of security properties which could be expressed by means of predicates over system activities and mutual information flow between system (private/public) inputs and (public) outputs, respectively.

2. Timed Process Algebra

In this section we define Timed Process Algebra, TPA for short. TPA is based on Milner's CCS but the special time action t which expresses elapsing of (discrete) time is added. The presented language is a slight simplification of Timed Security Process Algebra introduced in [5]. We omit an explicit idling operator ι used in tSPA and instead of this we allow implicit idling of processes. Hence processes can perform either ”enforced idling” by performing t actions which are explicitly expressed in their descriptions or ”voluntary idling”. But in the both cases internal communications have priority to action t in the case of the parallel operator. Moreover we do not divide actions into private and public ones as it is in tSPA. TPA differs also from the tCryptoSPA (see [9]). 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

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

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

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

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

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

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

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

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

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

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

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

3. Information Flow

To formalize an information flow we do not divide actions into public and private ones at the system description level, as it is done for example in [9, 3], but we use a more general concept of observation and opacity. This concept was recently exploited in [1] and [2] in a framework of Petri Nets and transition systems, respectively.

First we define observation function on sequences from $Actt^\star$.

Definition 3.1. (Observation)
Let Θ be a set of elements called observables. Any function $\mathcal{O} : Actt^\star \rightarrow \Theta^\star$ is an observation function. It is called static /dynamic /orwellian / m-orwellian $(m \geq 1)$ if the following conditions hold respectively (below we assume $w = x_1 \ldots x_n$):

- static if there is a mapping $\mathcal{O}' : Actt \rightarrow \Theta \cup \{\epsilon\}$ such that for every $w \in Actt^\star$ it holds $\mathcal{O}(w) = \mathcal{O}'(x_1) \ldots \mathcal{O}'(x_n)$,
- dynamic if there is a mapping $\mathcal{O}' : Actt^\star \rightarrow \Theta \cup \{\epsilon\}$ such that for every $w \in Actt^\star$ it holds $\mathcal{O}(w) = \mathcal{O}'(x_1).\mathcal{O}'(x_1.x_2) \ldots \mathcal{O}'(x_1 \ldots x_n)$,
- orwellian if there is a mapping $\mathcal{O}' : Actt \times Actt^\star \rightarrow \Theta \cup \{\epsilon\}$ such that for every $w \in Actt^\star$ it holds $\mathcal{O}(w) = \mathcal{O}'(x_1, w).\mathcal{O}'(x_2, w) \ldots \mathcal{O}'(x_n, w)$,

- m-orwellian if there is a mapping $\mathcal{O}' : Actt \times Actt^\star \rightarrow \Theta \cup \{\epsilon\}$ such that for every $w \in Actt^\star$ it holds $\mathcal{O}(w) = \mathcal{O}'(x_1, w_1).\mathcal{O}'(x_2, w_2) \ldots \mathcal{O}'(x_n, w_n)$ where $w_i = x_{max\{1,i-m+1\}}.x_{max\{1,i-m+1\}+1} \cdots x_{min\{n,i+m-1\}}$.

In the case of the static observation function each action is observed independently from its context. In the case of the dynamic observation function an observation of an action depends on the previous ones, in the case of the orwellian and m-orwellian observation function an observation of an action depends on the all and on m previous actions in the sequence, respectively. The static observation function is the special case of m-orwellian one for $m = 1$. Note that from the practical point of view the m-orwellian observation functions are the most interesting ones. An observation expresses what an observer - eavesdropper can see from a system behaviour and we will alternatively use both the terms (observation - observer) with the same meaning.

3.1. Opacity

Now suppose that we have some security property. This might be an execution of one or more classified actions, an execution of actions in a particular classified order which should be kept hidden, etc. Suppose that this property is expressed by predicate ϕ over process traces. We would like to know whether an observer can deduce the validity of the property ϕ just by observing sequences of actions from $Actt^\star$ performed by given process.

The observer cannot deduce the validity of ϕ if there are 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. $\mathcal{O}(w) = \mathcal{O}(w')$. We formalize this concept by opacity.

Definition 3.2. (Opacity)
Given process P, a predicate ϕ over $Actt^\star$ is opaque w.r.t. the observation function $\mathcal{O}$ if for every sequence w, $w \in Tr(P)$ such that $\phi(w)$ holds and $\mathcal{O}(w) \neq \epsilon$, there exists a sequence $w', w' \in Tr(P)$ such that $\neg\phi(w')$ holds and $\mathcal{O}(w) = \mathcal{O}(w')$. The set of processes for which the predicate ϕ is opaque with respect to $\mathcal{O}$ will be denoted by $Op^\phi_\mathcal{O}$.

The notion of opacity is rather general. With its help many other security properties can be defined (anonymity, non-interference etc. see [2]). On the other side, opacity is undecidable even for the simplest possible observation function, namely for the constant one, and for finite state processes (see [14]).

Example 3.1. Let $\mathcal{O} : Actt \rightarrow Actt \cup \{\epsilon\}$ such that $\mathcal{O}(a) = \mathcal{O}(b) = \epsilon, \mathcal{O}(\tau) = \tau$ and let $P = ((b.t.\bar{c} + a.\bar{c})|c) \setminus \{c\}$. Let $\phi_1(s)$ if s contains b and $\phi_2(s)$ if s contains a. Then the observer given by $\mathcal{O}$ can detect occurrence of the action a but not b i.e. $P \in Op^{\phi_1}_\mathcal{O}$ but $P \notin Op^{\phi}_\mathcal{O}$ since from observing just τ action (without any delay) it is clear that action a was performed. □

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

In [11] $Op^\phi_\mathcal{O}$ property is compared with Strong Nondeterministic Non-Interference (SNNI, for short). We recall its definition (see [5]). Suppose that all actions are divided in two groups, namely public (low level) actions L and private (high level) actions H i.e. $A = L \cup H, L \cap H = \emptyset$. Then process P has SNNI property if $P \setminus H$ behaves like P for which all high level actions are hidden for an observer. To express this hiding we introduce hiding= operator $P/M, M \subseteq A$, for which if $P \xrightarrow{a} P'$ then $P/M \xrightarrow{a} P'/M$ whenever $a \notin M \cup \bar{M}$ and $P/M \xrightarrow{\tau} P'/M$ whenever $a \in M \cup \bar{M}$. Formal definition of SNNI follows.

Definition 3.3. Let $P \in TPA$. Then $P \in SNNI$ iff $P \setminus H \approx_w P/H$.

Now we can compare $NIF^{\phi}_{\mathcal{O}}$ and $SNNI$ properties. Clearly, the former one is more general (see [11]).

Theorem 3.1. $P \in SNNI$ iff $P \in Op^{\phi}_{\mathcal{O}}$ for $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon$, $h \in H$, $\mathcal{O}(x) = x$, $x \in L$ and $\phi(s)$ iff s contains an element from H.

Note that by $Op^{\phi}_{\mathcal{O}}$ we can model situations which cannot be described by SNNI. For example, the definition of SNNI expects that everything what cannot be observed is private. So it cannot model situations when we do not see some actions and we are not interested in their occurrences at all (see examples from the beginning of the next subsection).

3.2. Quantifying opacity

Opacity, as it is defined in Definition 3.2, is frequently criticized from the both side - as a qualitative property it might happen that sometimes it is too weak or that in other cases it can be too strong.

Example 3.3. Let us consider process $P_1 = \sum_{i=1}^{2^k} h_i.(\sum_{j=1, j\neq i}^{2^k} l_j.\bar{l}_{refused} + l_i.\bar{l}_{accepted})$ $\mathcal{O}(h_i) = \epsilon$, for all i, $\mathcal{O}(x) = x$ for other actions, and predicate ϕ_i such that $\phi_i(s)$ holds iff s contains action h_i for some given i. Clearly $P_1 \notin Op^{\phi_i}_{\mathcal{O}}$ since an occurrence of action h_i is not "hidden" by any other action and so by observing $l_i.\bar{l}_{accepted}$ we have certainty that h_i has been performed. But P_1, which represents a simple access control process, is considered to be secure if k is big enough. So in this case for sufficiently big k opacity looks like to be too strong security property. On the other side if we consider $\phi'_i = \neg\phi_i$ we have again $P_1 \notin Op^{\phi'_i}_{\mathcal{O}}$ but validity of ϕ'_i is much more likely under observation $l_i.\bar{l}_{refused}$ and security of P_1 with respect to ϕ'_i is in this case much more lower.

Example 3.4. Let us consider process $Q = \sum_{i\in\{0,1\}} h_{1i}.h_{2i}.\dots.h_{(k-1)i}.(l_n + h_{ki}.\tau)$ where $n = \sum_{j=1}^{k} h_{ji}.2^{j-1}$. Let us suppose that $\mathcal{O}(h_{ji}) = \epsilon$, for all j, i and $\mathcal{O}(x) = x$ for the other actions and $\phi_n(s)$, for $2^{(k-1)} < n \leq 2^k$, is true iff the h actions contained in s represent binary notation of n (see definition of l_n). Clearly $Q \in Op^{\phi_n}_{\mathcal{O}}$ but an attacker can learn all bits of n, say of a long private key, except of the last one. Such system is usually not considered to be secure since the space of all possible keys of size 2^k can be reduced to only two possibilities and hence the private key can be, in practice, discovered.

Example 3.5. Now let us consider process $P_2 = \sum_{i=1}^{2^k} h_i.\mu X(\sum_{j=1, j\neq i}^{2^k} l_j.\bar{l}_{refused}.X + l_i.\bar{l}_{accepted}.X)$. If an attacker can observe this process for an unlimited time and can influence public data l_i than (s)he can learn validity of ϕ for any k with absolute certainty. But under limited length of observations process P_2 can be still considered to be secure for k being big enough.

To overcame an insufficiency of (qualitative) opacity illustrated in the previous examples we will define quantitative measure of opacity. To do so we need some preparatory work. The multiset of finite traces of P will be denoted by $MTr(P)$. For example, the trace $a.b$ is contained in $MTr(a.bNil + a.b.c.Nil)$ two times. There exist a few techniques how to define this multiset, originally developed for probabilistic process algebras (but here we will assume that all sequences have the same probability). For example, in [16] a technique of schedulers are used to resolve the nondeterminism and in [8] 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 $i_j \in \{0,1,2\} \cup \{0,1,2\} \times \{0,1,2\}$. An index records how a computation path of P could be derived, i.e. it records which process was chosen in case of nondeterminism. If there is only one possible successor then transitions are indexed by 1 (i.e. corresponding $i_l = 1$) If transition $P \xrightarrow{x} P'$ is indexed by k (i.e. corresponding $i_l = k$) then transition $P + Q \xrightarrow{x} P'$ is indexed by $k.1$ and transition $Q + P \xrightarrow{x} P'$ is indexed by $k.2$. If transitions $P \xrightarrow{x} P'$ and $Q \xrightarrow{x} Q'$ 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 the parallel composition was applied. Every index defines at most one trace and the set of all indexes defines the multisets of traces $MTr(P)$.

3.3. Information theory

To express quantity of information flow we will exploit Schannon information theory (see [17]). 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 $p_1, \ldots, p_n$.

Given two random variables X and Y, the mutual information between them, written $\mathcal{I}(X;Y)$, is defined as follows:

$$\mathcal{I}(X;Y) = \sum_x \sum_y p(x,y). \log \frac{p(x,y)}{p(x).p(y)}.$$

It can be easily shown that $\mathcal{I}(X;Y) = \mathcal{H}(X) + \mathcal{H}(Y) - \mathcal{H}(X,Y) = \mathcal{H}(X) - \mathcal{H}(X|Y) = \mathcal{H}(Y) - \mathcal{H}(Y|X)$.

Conditional entropy of X given knowledge of Y is defined as follows:

$$\mathcal{H}(X|Y) = \sum_y p(y).\mathcal{H}(X|Y=y),$$

and conditional mutual information between X and Y given knowledge of Z is defined as follows:

$$\mathcal{I}(X;Y|Z) = \mathcal{H}(Y|Z) - \mathcal{H}(Y|X,Z).$$

3.4. Surprisal and uncertainty of security properties

First we express quantification of an amount of information flow by means of the simplest concepts and later we develop more elaborated ones. Let $\mathcal{O}$ be an observation function and ϕ be a predicate over traces. Let $o \in Actt^\star$. We denote $MTr(P)^{\mathcal{O}=o} = \{s|s \in MTr(P), \mathcal{O}(s) = o\}$ and $MTr(P)^{\mathcal{O}=o}_\phi = \{s|s \in MTr(P), \phi(s) \wedge (\mathcal{O}(s) = o)\}$. We define

$$p(MTr(P)^o_\phi) = |MTr(P)^{\mathcal{O}=o}_\phi|/|MTr(P)^{\mathcal{O}=o}|.$$

Definition 3.4. We define surprisal $\mathcal{H}(P^{\mathcal{O}=o}_\phi)$ of ϕ for process P and observation $o, o \neq \epsilon$ as

$$\mathcal{H}(P^{\mathcal{O}=o}_\phi) = \log \frac{1}{p(MTr(P)^{\mathcal{O}=o}_\phi)}.$$

Example 3.6. For processes $P_1 = h.c.Nil, P_2 = h.c.Nil + \tau.c.Nil,$ $P_3 = h.c.Nil + h.c.Nil + h.c.Nil + h.c.Nil + \tau.c.Nil$ $P_4 = h.c.Nil + \tau.c.Nil + \tau.c.Nil + \tau.c.Nil + \tau.c.Nil$, $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon, \mathcal{O}(c) = c$ and predicate ϕ such that $\phi(s)$ holds iff s contains action h. We have $\mathcal{H}(P^{\mathcal{O}=c}_{1\phi}) = 0$, $\mathcal{H}(P^{\mathcal{O}=c}_{2\phi}) = 1, \mathcal{H}(P^{\mathcal{O}=c}_{3\phi}) = \log(5/4) = 0.32, \mathcal{H}(P^{\mathcal{O}=c}_{4\phi}) = \log(5) = 2.32$.

Example 3.7. Let us consider process $P = \tau.t.t.c.Nil + h.t.c.Nil$ and let $\mathcal{O}(h) = \mathcal{O}(\tau) = \tau$, $\mathcal{O}(t) = t, \mathcal{O}(c) = c$ and $\phi(s)$ iff s contains h. It is easy to see that $\mathcal{H}(P^{\mathcal{O}=c}_\phi) = 0$. Now let us consider another observer which cannot see elapsing of time shorter than 2 time units which can be easily modeled by a dynamic observation function. For such the observer we have $\mathcal{H}(P^{\mathcal{O}=c}_\phi) = 1$.

As it is stated in the following theorem there is a correspondence between a value of $\mathcal{H}(P^{\mathcal{O}=o}_\phi)$ and predicate opacity and so surprisal can be seen as a quantification of opacity.

Theorem 3.2. $P \in Op^\phi_{\mathcal{O}}$ iff $\mathcal{H}(P^{\mathcal{O}=o}_\phi) > 0$ for every o such that $\mathcal{O}(o) \neq \epsilon$.

Proof:
Let $P \in Op^\phi_{\mathcal{O}}$ and let $w \in Tr(P)$ such that $\phi(w)$ and $\mathcal{O}(w) = o, o \neq \epsilon$. Then there exists w', $w' \in Tr(P)$ such that $\neg\phi(w')$ and $\mathcal{O}(w) = \mathcal{O}(w')$. From this we have $p(MTr(P)^{\mathcal{O}=o}_\phi) < 1$ i.e. $\mathcal{H}(P^{\mathcal{O}=o}_\phi) > 0$.

Let $\mathcal{H}(P^{\mathcal{O}=o}_\phi) > 0$ for every o such that $\mathcal{O}(o) \neq \epsilon$. an let $w \in Tr(P)$ such that $\phi(w)$ and $\mathcal{O}(w) = o$. Since $\mathcal{H}(P^{\mathcal{O}=o}_\phi) > 0$ we have that $p(MTr(P)^{\mathcal{O}=o}_\phi) < 1$ i.e. there exists $w', w' \in Tr(P)$ such that $\neg\phi(w')$ and $\mathcal{O}(w) = \mathcal{O}(w')$ i.e. $P \in Op^\phi_{\mathcal{O}}$. □

So if $\mathcal{H}(P_\phi^{\mathcal{O}=o}) = 0$ then from observation o we have certainty that for corresponding trace(s) of P predicate ϕ holds. If $\mathcal{H}(P_\phi^{\mathcal{O}=o}) \geq 1$ then it is equally or more probable that ϕ does not hold than it holds.

For processes we have the following compositionality property.

Theorem 3.3. Let $\mathcal{H}(P_\phi^{\mathcal{O}=o}) = e_1$, $\mathcal{H}(Q_\phi^{\mathcal{O}=o}) = e_2$ then

$$\min\{e_1, e_2\} \leq \mathcal{H}((P+Q)_\phi^{\mathcal{O}=o}) \leq \max\{e_1, e_2\}.$$

Proof:
Let $|MTr(P)_\phi^{\mathcal{O}=o}| = n_1, |MTr(P)^{\mathcal{O}=o}| = m_1$ and $|MTr(Q)_\phi^{\mathcal{O}=o}| = n_2$, $|MTr(Q)^o| = m_2$. Without loss of generality we can assume that $e_1 \leq e_2$ i.e. $n_1/m_1 \leq n_2/m_2$. From that we get $n_1.m_2 \leq n_2.m_1$. We have that $|MTr(P+Q)_\phi^{\mathcal{O}=o}| = n_1 + n_2$ and $|MTr(P+Q)^{\mathcal{O}=o}| = m_1 + m_2$ and so $n_1/m_1 \leq (n_1+n_2)/(m_1+m_2) \leq n_2/m_2$. □

Many security properties, including SNNI, are based on the following idea. The system is considered to be secure if an attacker cannot learn by observing its behaviour whether some private activity was performed. This property can be easily expressed by opacity and by a special type of predicate over processes traces. Such predicate is valid if the trace contain some private activity form a set H of private activities. We will call such predicate set defined. The formal definition follows.

Definition 3.5. Predicate ϕ over $Actt^\star$ will be called set defined if there exists a set $H, H \subset Actt$ such that $\phi(s)$ iff there exists $h, h \in H$ such that $s = x_1.h.x_2$ for $x_1, x_2 \in Actt^\star$.

For the set defined predicates we have the following simple compositional property. The proof is straightforward.

Theorem 3.4. Let ϕ be a set defined predicate and let $\mathcal{H}(P_\phi^o) = e$. Then $\mathcal{H}(x.P_\phi^o) = e$ if $\neg\phi(x)$ and $\mathcal{H}(x.P_\phi^o) = 0$ if $\phi(x)$.

There is no correlation between length of observation and the resulting surprisal. See the following theorem.

Theorem 3.5. There exist process P and P' and observations o, o' such that o is the prefix of o', i.e. $o' = o.s$ for some s, such that $\mathcal{H}(P_\phi^{\mathcal{O}=o}) < \mathcal{H}(P_\phi^{\mathcal{O}=o'})$ and $\mathcal{H}({P'}_\phi^{\mathcal{O}=o'}) < \mathcal{H}({P'}_\phi^{\mathcal{O}=o})$.

Proof:
$P = \tau.c.d.Nil + h.c.(d.Nil + d.Nil)$, $P' = \tau.c.(d.Nil + d.Nil) + h.c.d.Nil$ and $\mathcal{O}(h) = \mathcal{O}(\tau) = \epsilon, \mathcal{O}(c) = c, \mathcal{O}(d) = d$, $o = c, o' = c.d$. □

The definition of opacity (see Definition 3.2) of predicate ϕ is asymmetric in the sense that if $\phi(w)$ does not hold than it is not required that there exist another trace for which it holds (in general $Op_\mathcal{O}^\phi \neq Op_\mathcal{O}^{\neg\phi}$). This means that opacity says something to an intruder which tries to detect only validity of ϕ (if it is opaque, than validity cannot be detected) but not its non-validity i.e. it says nothing about predicate $\neg\phi$.

To overcome this disadvantage we introduce a measure of uncertainty of ϕ under observation o. The uncertainty expresses an amount of information which can be learned by attacker about predicate ϕ.

Definition 3.6. We define uncertainty $\mathcal{H}_u(P_\phi^{\mathcal{O}=o})$ of ϕ for process P and observation $o, o \neq \epsilon$ as $\mathcal{H}_u(P_\phi^{\mathcal{O}=o}) = p(MTr(P)_\phi^{\mathcal{O}=o}). \log \frac{1}{p(MTr(P)_\phi^{\mathcal{O}=o})} + (1 - p(MTr(P)_\phi^{\mathcal{O}=o})). \log \frac{1}{1-p(MTr(P)_\phi^{\mathcal{O}=o})}$.

The uncertainty expresses how uncertain is predicate ϕ under observation o. It reaches maximal value (equal to 1) when probabilities that ϕ holds and that ϕ does not hold are equal. It reaches minimal value (equal to 0) it validity of ϕ or $\neg\phi$ is certain.

The uncertainty has a similar relationship to opacity as the suprisal (see Theorem 3.2). Also the proof is similar.

Theorem 3.6. If $P \in Op_{\mathcal{O}}^{\phi}$ than $\mathcal{H}_u(P_\phi^{\mathcal{O}=o}) > 0$ for every o such that $\mathcal{O}(o) \neq \epsilon$.

The inverse implication in Theorem 3.6 does not hold but we have the following property. Its proof is straightforward.

Theorem 3.7. If $\mathcal{H}_u(P_\phi^{\mathcal{O}=o}) = 0$ then $P \notin Op_{\mathcal{O}}^{\phi}$ or $P \notin Op_{\mathcal{O}}^{\neg\phi}$.

Till now we have quantified an amount of information flow by means of surprisal and uncertainty for a given observation o. But to get an appropriate quantification of security of processes we have to limit a power of an attacker. First we start with a restriction on a length of observations the attacker can performed. Let $s \in Actt^\star$. By $|s|_t$ we will denote the number of occurrences of action t contained in s. Note that every process can perform only finite number of actions different from t between any two action t. Let us suppose that an attacker can observe process behavior for no longer that for n time units. Then we put $\mathcal{H}_u(P_\phi^n) = \min_{|o|_t \leq n} \mathcal{H}_u(P_\phi^{\mathcal{O}=o})$. Note that we know (see Theorem 3.5) that value of $\mathcal{H}_u(P_\phi^{\mathcal{O}=o})$ does not need to be correlated with the length of o. Moreover the value of $\mathcal{H}_u(P_\phi^n)$ should be related to $\log |MTr(P)^{\mathcal{O}=o}|$. If $\mathcal{H}_u(P_\phi^n) = 0$ then there exist an observation o which gives us certainty about validity of ϕ or $\neg\phi$. On the other side we can ask about the minimal n such that $\mathcal{H}_u(P_\phi^n) = 0$ (if such n exits). For example, let $P_2 = \sum_{i=1}^{2^k} h_i.\mu X(\sum_{j=0, j\neq i}^{2^k} l_j.t.X + l_i.t.\bar{l}_{accepted}.X)$, $\mathcal{O}(h_i) = \epsilon$, for all i, $\mathcal{O}(x) = x$ for other actions, and predicate ϕ_i such that $\phi_i(s)$ holds iff s contains action h_i for some given i. For an attacker which can influence a performance of actions l_j it takes at most 2^k time unites to learn the private input h_i.

3.5. Mutual information flow

Non-Deducibility on Composition (NDC for short, see in [6]) is a widely studied security property. It is based on the idea of checking the system against all high level potential interactions, representing every possible high level process. System is NDC if for every high level user A, the low level view of the behaviour of P is not modified (in terms of trace equivalence) by the presence of A. The idea of NDC can be formulated as follows.

Definition 3.7. **(NDC)** $P \in NDC$ iff for every $A, Sort(A) \subseteq H \cup \{\tau, t\}$

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

Now we will define a quantified variant of NDC. Let A be a finite subset of $Actt^*, A \neq \emptyset$. X_A be a corresponding discrete random variable with range A and uniform probability. Let P be a process and let Y_P be a random variable which ranges over $\bigcup_{s \in A} MTr((P|s) \setminus H))$ with uniform probability (string s is considered to be process $s.Nil$).

We define the mutual information between X_A and Y_P as follows:

$$\mathcal{F}(A \rightsquigarrow P) = \mathcal{I}(X_A, Y_P).$$

We illustrate mutual information by the following example. Note that if two variables are independent then mutual information is equal to zero.

Example 3.8. Let $P = h.c.Nil + d.Nil$, $A = \{\epsilon, \bar{h}\}$. We have that $\mathcal{F}(A \rightsquigarrow P) = \mathcal{H}(X_A) + \mathcal{H}(Y_P) - \mathcal{H}(X_A, Y_P) = 1 + 1 - 1,58$ = 0,42.

Again mutual information can be viewed as a quantification of NDC as it is stated by the following theorem.

Theorem 3.8. Let $P \notin NDC$ then there exists A such that $\mathcal{F}(A \rightsquigarrow P) > 0$.

Proof:

Let $P \notin NDC$. That means that there exists process H and $s \in Tr((P|H) \setminus H$ such that $s \notin Tr(P \setminus H)$. Let $h_1. \dots .h_n$ is a sequence of actions which participates on s and are performed by H. For the rest of the proof we chose $A = \{\epsilon, h_1. \dots .h_n\}$. Clearly we have $\mathcal{F}(A \rightsquigarrow P) > 0$. □

Also an inverse of the previous theorem holds.

Theorem 3.9. Let for every A, $A \subset Actt^*, A \neq \emptyset$ we have $\mathcal{F}(A \rightsquigarrow P) > 0$ for some P. Then $P \notin NDC$.

3.6. Conditional mutual information flow

Now suppose that we are interested not only whether some private action was performed (see Definition 3.3) but also which one was performed. It can be modeled by opacity considering predicates ϕ_a such that $\phi_a(s)$ is valid if s contains private action a. Here we offer an alternative approach which can exploit an additional knowledge which might have an attacker at disposal.

We will assume that process P receives some private input from the set $\{h_1, \dots, h_n\}$, some public input from the set $\{l_1, \dots, l_m\}$ and produces an output from the set $\{\bar{l}_1, \dots, \bar{l}_k\}$. (Note that this assumption could be naturally generalized to several inputs/outputs.) The process can perform also other actions but those are out of interest.

Suppose that distributions of possible private and public inputs and the resulting distribution of a corresponding public output are given by discrete random variables H_{in}, L_{in} and L_{out}, respectively.

Following an approach advocated in [4] define conditional mutual information flow ($\mathcal{F}_P(H \rightsquigarrow L)$) between private inputs and public outputs, knowing public inputs. It expresses an amount of information on private inputs which can be learned by attacker who can see public inputs and outputs. If there is no information flow between private inputs and public outputs (knowing public inputs) then this conditional mutual information is equal to zero, i.e. public and private data are independent and attacker can learn nothing.

Definition 3.8.

$$\mathcal{F}_P(H \rightsquigarrow L) = \mathcal{I}(H_{in}, L_{out}|L_{in})$$

Note that for system in which the output (L_{out}) is uniquely given by the inputs H_{in}, L_{in} (it means that the system is deterministic with respect to these inputs and that the output does not depend on any other inputs) then we have that $\mathcal{F}_P(H \rightsquigarrow L) = \mathcal{I}(H_{in}, L_{out}|L_{in}) = \mathcal{H}(L_{out}|L_{in})$ (see [4]).

Example 3.9. Let us consider $P_1 = \sum_{i=1}^{2^k} h_i.(\sum_{j=1,j\neq i}^{2^k} l_j.\bar{l}_{refused} + l_i.\bar{l}_{accepted})$ and suppose that passwords (actions h_i) are distributed with uniform probability. So the same holds for attacker's guesses. Then we have $\mathcal{F}_{P_1}(H \rightsquigarrow L) = \mathcal{H}(L_{out}|L_{in}) = \sum_{i=0}^{2^k} p(l_i).\mathcal{H}(L_{out}|L_{in} = l_i)$ what is roughly $k/2^k$. So an amount of information flow is rather low for a bigger value of k. Now suppose that the password is more likely (say, with probability 2^{10} times higher) to be a word from a dictionary of size $2^n, n < k$ (dictionary attack). Then we have $\mathcal{F}_{P_1}(H \rightsquigarrow L) = \mathcal{H}(L_{out}|L_{in}) = (2^k - 2^n).(1/p).H(1/p, 1 - 1/p) + 2^{(n+10)}/p.H(2^{10}/p, 1 - 2^{10}/p)$, for $p = 1/(2^n.(2^{10} - 1) + 2^k)$. In this case the amount of information flow is rather high for $n \ll k$.

The concept mutual information flow is different from the previous ones and cannot be directly compared with them. Here, in fact, we consider only processes which do not belong to SNNI so we cannot expect results similar to Theorem 3.6 and 3.7.

Restriction to just one input sent through a private channel and one input sent through a public channel and one output sent through a public channel might be too restrictive. Let discrete random variables H^i_{in}, L^j_{in} and L^k_{out} correspond to i, j and k inputs/outputs, respectively. Then we define conditional mutual information flow as follows:

$$\mathcal{F}^u_P(H \rightsquigarrow L) = \max_{i,j,k} \mathcal{I}(H^i_{in}, L^j_{out}|L^k_{in}).$$

Now we can ask whether P is insecure (or secure) knowing value of $\mathcal{F}^u_P(H \rightsquigarrow L)$. If $\mathcal{F}^u_P(H \rightsquigarrow L) = 0$ we know that public outputs and private inputs (knowing public inputs) are independent variables and hence the process could be considered secure. On the other side, let us consider two processes $P_1 = \sum_{i=1}^{2^k} h_i.(\sum_{j=1,j\neq i}^{2^k} l_j.\bar{l}_{refused} + l_i.\bar{l}_{accepted})$ and its recursive version
$P_2 = \sum_{i=1}^{2^k} h_i.\mu X(\sum_{j=1,j\neq i}^{2^k} l_j.\bar{l}_{refused}.X + l_i.\bar{l}_{accepted}.X)$.
Clearly we have $\mathcal{F}_{P_1}(H \rightsquigarrow L) = \mathcal{F}_{P_2}(H \rightsquigarrow L)$ but P_2 is less secure because the an attacker can try to guess a password infinitely many times.

Now let us consider process
$P_3 = \tau.(\sum_{i=1}^{2^k} h_i.(\sum_{j=1}^{2^k} l_j.(\sum_{j=1}^{2^k} \bar{l}_j))) + \tau.(\sum_{i=1}^{2^k} h_i.(\sum_{j=1}^{2^k} l_j.\bar{l}_{((i+j) \bmod 2^k)}))$. It consists of two subprocesses. The first one produces a random output l_j and the second one produces an output fully determinated by the inputs. The both subprocesses are prefixed by τ actions which might represent a lack of knowledge of system behaviour. But it might be that there is a way how to force the process to chose the second subprocess which gives full information on the private inputs to an attacker. We propose a solution to above mentioned situations: first we realistically limit a maximal length of observations (to maximum of n time units), and then we encode every computational paths by sequences $p, p \in \{0,1,2\}^\star$ and put

$$\mathcal{F}^s_P(H \rightsquigarrow L) = \max_{0 \leq |p|_t \leq n} \mathcal{F}_{P_p}(H \rightsquigarrow L)$$

where $\mathcal{F}_{P_p}(H \rightsquigarrow L)$ denotes the information flow for process P and its computational path p and by $|p|_t$ we denote length of a trace corresponding to computational path p. In this way we obtain a more realistic quantification of process security than it is given just by $\mathcal{F}_P(H \rightsquigarrow L)$.

3.7. Unprecise observations

Let as assume that on observer cannot observe time of action occurrences with an absolute precision. That means that a possible outcome of an observation function might be not just a single observation but a set of different observations. Each element of the set represents a possible observation. We can model this kind of observations directly with observation functions (see Definition 3.1) but instead of that we generalize definition of observational functions. We will consider functions $\mathcal{O} : Actt^\star \rightarrow 2^{\Theta^\star}$. If $\mathcal{O}(w)$ is always a singleton that we get the previous concept of observational functions. We assume that for every $w, w \in Actt^\star$ we have that $o_1|_{Act} = o_2|_{Act}$ (i.e. o_1, o_2 restricted to actions from Act are equal) for every $o_1, o_2 \in \mathcal{O}(w)$. That means that members of $\mathcal{O}(w)$ differ only in timing of actions. We now modify the definition of opacity.

Definition 3.9. (Unprecise Opacity)
Given process P, a predicate ϕ over $Actt^\star$ is unprecise opaque w.r.t. the observation function $\mathcal{O}$ if for every sequence w, $w \in Tr(P)$ such that $\phi(w)$ holds and $\mathcal{O}(w) \neq \{\epsilon\}$, there exists a sequence $w', w' \in Tr(P)$ such that $\neg\phi(w')$ holds and $\mathcal{O}(w) \cap \mathcal{O}(w') \neq \emptyset$. The set of processes for which the predicate ϕ is unprecise opaque with respect to $\mathcal{O}$ will be denoted by $uOp^\phi_\mathcal{O}$.

Example 3.10. Let as consider process
$P = c.t.t.t.h.c.Nil + c.t.\tau.c.Nil$, $\mathcal{O}(c.t.t.t.h.c) = \{c.t.t.t.t.\tau.c, c.t.t.t.\tau.c, c.t.t.\tau.c, c.t.\tau.c\}$ and $\mathcal{O}(c.t.\tau.c) = \{c.t.t.\tau.c, c.t.\tau.c, c.\tau.c\}$ and predicate ϕ such that $\phi(s)$ iff s contains action h. Clearly $P \in uOp^\phi_\mathcal{O}$ but for $P \notin Op^\phi_{\mathcal{O}'}$ for precise observation which only hides h action.

We have to modify accordingly definitions of $MTr(P)^{\mathcal{O}=o}$ so we will have $MTr(P)^{\mathcal{O}=o} = \{s | s \in MTr(P), o \in \mathcal{O}(s)\}$. Corresponding definitions for quantified information flow under unprecise observations are the same as for the case of precise observations. In this way we obtain more realistic security properties with respect to so called timing attacks which are based on timing information on systems behaviour. Many system, which are in theory opened to that type of attacks, are reasonable safe if an attacker capability to observe precisely elapsing of time is limited. Hence in that case there is no need to modify such systems by, for example, adding random delays between some actions, to obtain their security.

With this technique we could define also other kinds of unprecise observations not only those ones when elapsing of time cannot be observed with absolute precision. For example, we can model observers who sometimes cannot precisely distinguish some actions particularly if many actions are performed between two time unites and so on.

4. Discussion and future work

We have developed the concept of quantified information flow in timed process algebras for different settings. For the sake of simplicity we have used a timed process algebra instead of a probabilistic timed process algebra (depending on an application one can chose between reactive, generative or stratified probabilistic calculi, see [8]). Using that kind of algebras we could have more adequate tools for expressing probabilities of traces. Instead of that we have used uniform probability distribution but all the concepts and results could be easily translated to a probabilistic calculus. Actually we would have different definitions of $p(MTr(P)^{o}_{\phi})$ which appears in Definition 3.4 and 3.6 of surprisal and uncertainty.

In the case of mutual information flow ($\mathcal{F}(A \rightsquigarrow P)$) we could associate probabilities also with elements of set A (i.e. X_A could have also other than uniform probability distribution) and with Y_P in $\mathcal{I}(X_A, Y_P)$.

In the case of the unprecise observations, we could generalize this concept as follows. We could permit that the observations can be unprecise not only with respect to elapsing of time but they could be unprecise in general. We can model this by observational functions which map every trace of actions to a discrete random variable which ranges over strings from $\Theta^{\star}$. Moreover we can associate some probability distribution to resulting possible observations for a given ”unprecise” observer. For example, we can model Gaussian (or any other) distribution of errors for different kinds of observation etc.

As regards the future work, besides considering probabilistic process algebra, we plan to investigate changes of a level of security of process P by putting it to different contexts and by composing it with other processes.

References

[1] 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.

[2] 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

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

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

[7] 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.

[8] 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

[9] 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.

[10] Gruska D.P.: Probabilistic information flow security. Fundamenta Informaticae, Vol. 85, No. 1-4, 2008.

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

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

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

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

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

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

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