Differential Privacy and Security [⋆]

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 differential privacy is defined and studied in the framework of probabilistic process algebras. The resulting (quantitative) security properties are studied and compared with other (qualitative) security notions.

Keywords: differential privacy, probabilistic process algebra, information flow, security, opacity

1 Introduction

Several formulations of system security can be found in the literature. Many of them are based on a non-interference (see [GM82]) which assumes an absence of any information flow between private and public systems activities. More precisely, systems are considered to be secure if from observations of their public activities no information about private activities can be deduced. This approach has found many reformulations for different formalisms, computational models and nature or "quality" of observations. For many applications such properties could be criticized for being either too restrictive or too benevolent. They are too restrictive in the case that there exists some information flow between public and private activities (or data) but this flow is reasonable small. For example, usually access control processes exhibit some information flow (mostly) showing which password is not correct but they are still considered to be secure under reasonable password policy: it is not meaningful to consider such systems insecure in the case that a number of possible passwords is sufficiently large. On the other side, qualitative security properties could be too benevolent. For example, if an intruder cannot learn the whole secrete (password, private key, etc) they could consider a system to be safe despite the fact, that the intruder could still learn almost all the secrete (for example, significant number of bits of private key). Hence there is a need to quantify an amount of information flow which can be gained from the observations of public system activities.

An amount of possibly leaked information could be expressed by means of Shannon's information theory as it was done, for example, in [CHM07,CMS09] for simple imperative languages and in [Gru08] for process algebras. Another possibility is to exploit probabilistic theory as it was used for process algebras in

[⋆] Work supported by the grant VEGA 1/1333/12.

[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 on processes. In [L02] 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. In [Gru11] it is investigated which private actions can gained or excluded by observations of public actions.

The aim of this paper is to quantify an amount of information flow by differential privacy (see [D08]) in the framework of probabilistic process algebras. The concept of differential privacy was originally developed to "provide means to maximize the accuracy of queries from statistical databases while minimizing the chances of identifying its records". Later on it was used also for other applications. In [X14] differential privacy is studied for probabilistic automata and in [X12] it has been exploited in the framework of probabilistic process algebra by comparing probabilities of a given output produced by inputs which differ in one position. Here we extend and further develop this approach and we propose several other security properties based on ϵ-differential privacy for a (different) probabilistic process algebra. We show how these properties are related as well as how they are related to some traditional qualitative security properties (namely, Non-Deducibility on Composition [FGM03] and opacity [BKR04,BKMR06]). Moreover, we show some of their compositionality properties as well as undecidability and decidability results.

The paper is organized as follows. In Section 2 we describe our working formalism - probabilistic process algebra. In Sections 3 we recall some (qualitative) security properties based on an absence of information flow which will serve as a motivation for our work. Section 4 is devoted to differential privacy. Here we define and investigate various security properties based on ϵ-differential privacy.

2 Probabilistic Process Algebra

In this section we define the Probabilistic Process Algebra, pCCS for short, which 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.

To 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]). 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 .

Formally, 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 assume the signature $\Sigma = \bigcup_{n\in N} \Sigma_n$, where

$$\begin{aligned}
&\Sigma_0 = \{Nil\}\\
&\Sigma_1 = \{x. \mid x \in Act\} \cup \{[S] \mid S \text{ is a relabeling function}\}\\
&\qquad \cup\{\backslash M \mid M \subseteq A\}\\
&\Sigma_2 = \{|\}\\
&\Sigma_n = \{\bigoplus_{i\in I} q_i, |I| = n\}
\end{aligned}$$

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

The set of pCCS 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 pCCS terms, $\mu X-$ is the binding construct, $op \in \Sigma$. We require that all P_i processes in $\bigoplus_{i\in I} q_i.P_i$ 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 (see [Mil89]) are extended for pCCS ones. Structural operational semantics is given by labeled transition systems. The transition relation $\rightarrow$ is a subset of pCCS $\times$ $Act \cup (0,1] \times$ pCCS. We just mention the new transition rules for probabilitis.

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

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

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.

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 pCCS 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 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 $P \xrightarrow{x} P'$.

To express what an observer can see from system behaviour we will define modified transitions $\overset{x}{\Rightarrow}$ which hide the action τ and probabilities. Formally, we will write $P \overset{x}{\Rightarrow} P'$ iff $P \overset{s_1}{\rightarrow}\overset{x}{\rightarrow}\overset{s_2}{\rightarrow} P'$ for $s_1, s_2 \in (\{\tau\} \cup (0,1])^\star$ and $P \overset{s}{\Rightarrow}$ instead of $P \overset{x_1}{\Rightarrow}\overset{x_2}{\Rightarrow} \ldots \overset{x_n}{\Rightarrow}$. We will write $P \overset{x}{\Rightarrow}$ if there exists P' such that $P \overset{x}{\Rightarrow} P'$. 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. $Sort(P) = \{x | P \xrightarrow{s.x}$ for some $s \in (Act \cup (0,1])^\star$ and $x \in A\}$.

As regards behaviorial semantics, we will work with the weak trace equivalence.

Definition 1. *The set of weak traces of process P is defined as $Tr_w(P) = \{s \in A^\star | \exists P'.P \overset{s}{\Rightarrow} P'\}$. Two processes P and Q are weakly trace $(P \approx_w Q)$ iff $Tr_w(P) = Tr_w(Q)$.*

We conclude this section with a definition of probabilities of traces for a given process. Let P be a pCCS process and let $P \overset{x_1}{\rightarrow} P_1 \overset{x_2}{\rightarrow} P_2 \overset{x_3}{\rightarrow} \ldots \overset{x_n}{\rightarrow} P_n$, 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 paths 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 [GSS95] all transitions are indexed and hence paths 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 paths, 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 $P_i \xrightarrow{x} P'$ is indexed by k then transition $\bigoplus_{i \in I} q_i.P_i \xrightarrow{x} P'$ is indexed by $k.i$, and 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 parallel composition was applied. Every index defines at most one path and the set of all indexes defines the multisets of paths $Path(P)$. 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 $Pr(\emptyset) = 0$. For

$s \in Tr_w(P)$ we will denote by $Pr(s)$ the probability of performing s (i.e. it is the sum of probabilities of all paths $c \in Path(P)$ such that $label(c) = s$).

3 Information Flow

In this section we recall two (qualitative) security properties for CCS (i.e. non-probabilistic process algebra). The first inspiration for our work is the security property Non-Deducibility on Composition (NDC for short, see in [FGM03]). 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 property NDC if for every high level user A, the low level view of the behaviour of P is not modified (in terms of weak trace equivalence) by the presence of A. The idea of NDC can be formulated as follows.

Definition 2. (NDC) *$P \in NDC$ iff for every $A, Sort(A) \subseteq H \cup \{\tau\}$*

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

Now we introduce another information flow notion, which is based on a more general concept of observation and opacity. This concept was exploited in [BKR04] and [BKMR06] in a framework of Petri Nets and transition systems, respectively. First we assume an observation function $\mathcal{O} : Act^\star \to Act^\star$.

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's traces. Contrary to the original definition we do not require that the predicate is total. We would like to know whether an observer can deduce the validity of the property ϕ just by observing sequences of actions from $Act^\star$ performed by given process. The observer cannot deduce the validity of ϕ if there are two traces $w, w' \in Act^\star$ 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 (Opacity). *Given process P, a predicate ϕ over $Act^\star$ is opaque w.r.t. the observation function $\mathcal{O}$ if for every sequence w, $w \in Tr_w(P)$ such that $\phi(w)$ holds and $\mathcal{O}(w) \neq \epsilon$, there exists a sequence $w', w' \in Tr_w(P)$ 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 $Op^\phi_\mathcal{O}$.*

Now we are prepared to define several quantitative security properties based on differential privacy. Actually, as we will see later, two of them are really quantitative counterparts of the above mentioned qualitative properties.

4 Differential Privacy

Differential privacy was originally developed for privacy protection of statistical databases (see [D08]). In the original definition, a query mechanism A is ϵ-differentially private if for any two databases D_1 and D_2 which differ only for one individual (one raw, for example, data of one person), and any property S, the probability distributions of $A(D_1), A(D_1)$ differ on S at most by e^ϵ, namely,

$$\Pr(\mathcal{A}(D_1) \in S) \leq e^\epsilon \times \Pr(\mathcal{A}(D_2) \in S).$$

Now we will reformulate ϵ-differential privacy for our process algebra framework. Every sequence of high level actions s (i.e. $s \in H^*$) represents a secrete input. The public output o is a sequence of low level actions (i.e. $o \in L^*$). First we start with formulation of ϵ-differential privacy for the given secrete input and public output. Note that this definition is similar to the one which appeared in [X12]. We will write for a given process P conditional probability $Pr(o|s)$ as probability $Pr(o)$ for process $(P|s.Nil) \setminus H$.

Definition 4. *$P \in DF_\epsilon(o,s)$ iff $o \in Tr_w((P|s.Nil) \setminus H)$ and*

$$Pr(o|s) \leq e^\epsilon \times Pr(o|s')$$

for every $s' \in H^*$ which differs from s in one position.

Note that in the previous definition we assume that if $s = x_1 \dots x_n$ $s' = x'_1 \dots x'_n$ then there exists j such that $x_j \neq x'_j$ and $x_i = x'_i$ for $i \neq j$. The property $DF_\epsilon(o,s)$ says that by observing the public output o an intruder cannot be pretty sure (expressed by ϵ) whether the secrete input was s or s'. Note that for $\epsilon = 0$ the inputs s and s' do not lead to different probabilities for the corresponding output. Now we will formulate several properties of differential privacy. First, differential privacy is not sensitive to a length of the observation (public output) i.e. a longer observation can leak less as well as more on private inputs as it is stated by the following proposition.

Proposition 1. *For every ϵ there exist processes P, P', $s \in H^*$ and $o_1, o_2, o_3, o_4 \in L^*$ such that $o_1 \sqsubseteq o_2$ and $o_3 \sqsubseteq o_4$ and such that $P \in DF_\epsilon(o_1,s)$, $P \notin DF_\epsilon(o_2,s)$ and $P \in DF_\epsilon(o_4,s)$, $P' \notin DF_\epsilon(o_3,s)$.*

Proof. Let $P = (1-\epsilon)/2.(h_1.l_1.(p.l_2.Nil \oplus (1-p).l_3.Nil)) \oplus ((1+\epsilon)/2.h_1.l_1.l_2.Nil)$, $s = h_1$ and $o_1 = l_1$, $o_2 = l_1.l_2$. By appropriate choice of p we get $P \in DF_\epsilon(o_1,s)$, $P \notin DF_\epsilon(o_2,s)$. The second case is similar.

Differential privacy is neither sensitive to a length of secrete as it is stated by the following proposition, its proof is similar to the proof of previous proposition.

Proposition 2. *For every ϵ there exist processes P, P', $o \in L^*$ and $s_1, s_2, s_3, s_4 \in H^*$ such that $s_1 \sqsubseteq s_2$ and $s_3 \sqsubseteq s_4$ and such that $P \in DF_\epsilon(o,s_1)$, $P \notin DF_\epsilon(o,s_2)$ and $P \in DF_\epsilon(o,s_4)$, $P' \notin DF_\epsilon(o,s_3)$.*

Now we will formulate and prove some compositional properties of $DF_\epsilon(o,s)$ property.

Proposition 3. *$P \in DF_\epsilon(o,s)$ then $l.P \in DF_\epsilon(l.o,s)$ and $h.P \in DF_\epsilon(o,h.s)$.*

Proof. Clearly, every observation of the process $l.P$ has to start with l and probabilities of all traces with the proper prefix l do not change. Similarly for the process $h.P$.

Proposition 4. *Let us assume processes P_i and let $p = \min(q_1.Pr(o|s)_1, \ldots q_n.Pr(o|s)_n)$ and let us suppose that $p = q_i.Pr(o|s)_i$, and $p' = \max(q_1.Pr(o|s)_1, \ldots q_n.Pr(o|s)_n)$ and let us suppose that $p' = q_j.Pr(o|s)_j$, where $Pr(o|s)_i$ is the corresponding probability for the process P_i. Let $P = \bigoplus_{i \in \{1,\ldots,n\}} q_i.P_i$ then $P \in DF_{\ln(p'/p)}(o, s)$.*

Proof. The main idea. The process P can output o with the input s by performing P_i and can output o with the input s' by performing P_j. The rest of the proof could be done by computing the corresponding probability.

Proposition 5. *Let S be a bijection on L and on H and $P \in DF_\epsilon(o, s)$. Then $P[S] \in DF_\epsilon(S((o), S(s))$ and $P \setminus M \in DF_\epsilon(o, s)$.*

Proof. The first part follows directly from the definition of relabeling. The second part follows from the fact that the restriction either has no influence on performing o and hence the corresponding probabilities are not changed or $M \cup Sort(o.Nil) \neq \emptyset$ and in this case probabilities are equal to 0.

As regards the recursion we need an auxiliary definition.

Definition 5. *Process variable X is sequential in P if every subterm of P containing X (except X itself) is of a form $y.P'$ or $\sum P_i$. Let $M \subseteq Act$. Process variable X is M-guarded in P if it is contained in a subterm of P of the form $u.P'$, $u \in M$.*

Proposition 6. *Let $P \in DF_\epsilon(o, s)$ and $Pr(o|s) \neq 0$ for P and P is sequential and process variable X is M-guarded in P for some nonempty M such that $Sort(o.Nil) \cap M = \emptyset$. Then $\mu X.P \in DF_\epsilon(o, s)$.*

Proof. Sketch. We have to eliminate the case when o could be produced by application of the recursion what is satisfied by proposition's requirements. The rest follows directly from the definitions of $DF_\epsilon(o, s)$ and recursion.

Now we can define the property expressing security of the input s with respect to ϵ-differential privacy. Process has this property if there is no observation (output) which could distinguish between the input s and input s' (which differs from s in one element). The formal definition is the following.

Definition 6. *$P \in DF_\epsilon(s)$ if for every $o \in L^*$ it holds $P \in DF_\epsilon(o, s)$.*

The property $DF_\epsilon(s)$ is rather strong but in general it is undecidable as it is stated by the following proposition.

Proposition 7. *Property $DF_\epsilon(s)$ is undecidable.*

Proof. The main idea. We exploit Turing power of pCCS and hence we reduce the property to the halting problem. Let R be an arbitrary process and let $T = \mu X. \sum_{y \in Act} y.X$. By deciding $(P|((R|T) \setminus Act)) \in DF_\epsilon(s)$ we could decide halting problem for R.

We could put some restrictions on processes in such a way that the property $DF_\epsilon(s)$ is decidable for them.

Proposition 8. *Property $DF_\epsilon(s)$ is decidable for finite processes and for processes which are sequential and H-guarded.*

Proof. Sketch. Only the case of infinite processes is interesting. If a process is sequential and H-guarded this process can produce public outputs only by reading secrete inputs and hence we can limit length of possible outputs o i.e. there are only finitely many cases to be checked.

Now we define which observations could leak something about the secrete s with respect to ϵ-differential privacy.

Definition 7. $DF(P, \epsilon, s) = \{o|Pr(o|s) > e^\epsilon \times Pr(o|s')$ *and* $o \in Tr_w((P|s.Nil) \setminus H)\}$.

Clearly, $P \in DF_\epsilon(s)$ iff $DF(P, \epsilon, s) = \emptyset$. On the other side, if $DF(P, \epsilon, s) \neq \emptyset$ we can ask what is the minimal length of $o, o \in DF(P, \epsilon, s)$. Usually, longer o (a higher value of $|o|$) means that the secrete s could be considered safer.

Similarly to the previous definition, we can specify which secretes could by leak (with respect to ϵ-differential privacy) by the given observation o.

Definition 8. $DF(P, \epsilon, o) = \{s|Pr(o|s) > e^\epsilon \times Pr(o|s')$ *and* $o \in Tr_w((P|s.Nil) \setminus H)\}$.

There is a simple relation between sets from Definition 7 and 8, namely, $o \in (P, \epsilon, s)$ iff $s \in (P, \epsilon, o)$. Another generalization of above mentioned concepts is overall security of processes with respect to ϵ-differential privacy which requires that processes are secure with respect to every secrete input and public output. The formal definition follows.

Definition 9. $DF(\epsilon) = \{P|P \in DF_\epsilon(o, s)$ *for every* $o \in L^*, s \in H^*\}$.

Note that for $P \in DF(\epsilon)$ it holds that $DF(P, \epsilon, o) = DF(P, \epsilon, s) = \emptyset$ i.e. for such the process there is no secret which could leak by any observation.

Naturally, all above mentioned sets depend on value of ϵ as corresponding "security" level. So it is meaningful to define "highest" security as the minimal ϵ such that by observing o an intruder cannot be sure (in terms of ϵ differential privacy) about the value of s.

Definition 10. $PDF(P, o, s) = \min\{\epsilon|P \in DF_\epsilon(o, s)\}$.

Clearly, for $\epsilon_1 < \epsilon_2$ it holds $DF(P, \epsilon_1, s) \subseteq DF(P, \epsilon_2, s)$ and $DF(P, \epsilon_1, o) \subseteq DF(P, \epsilon_2, o)$. Hence for $PDF(P, o, s)$ we obtain the smallest sets $DF(P, \epsilon, o)$, $DF(P, \epsilon, s)$ and $DF(\epsilon)$. As regards "length" of observations and secrets we have the following proposition.

Proposition 9. *There exist P, $s \in H^*$ and $o_1, o_2, o_3, o_4 \in L^*$ such that $o_1 \subset o_2$ and $o_3 \subset o_4$ such that $PDF(P, o_1, s) < PDF(P, o_2, s)$ and $PDF(P, o_4, s) < PDF(P, o_3, s)$.*

Proof. The proof follows from Proposition 1.

Till now we have investigated an impact of probability distributions for two secret inputs which differ only in one position. This approach could be too restrictive in many cases so we extend it now. We assume a metric ρ on the set of secretes, i.e. sequences of high level actions. Hence we can relate probabilities of the output o produced by arbitrary secretes s, s' not only those ones which differ only in one position.

Definition 11. *$P \in DF_{\epsilon,\rho}(o, s)$ iff $o \in Tr_w((P|s.Nil) \setminus H)$ and*

$$Pr(o|s) \leq e^{\epsilon \times \rho(s,s')} \times Pr(o|s')$$

Similarly to Definition 9 we can define the set of secure properties with respect to metrics ρ and ϵ-differential privacy.

Definition 12. *$DF(\epsilon, \rho) = \{P | P \in DF_\epsilon(o, s)$ for every $o \in L^*, s \in H^*\}$.*

Now we can relate qualitative security property NDC to quantitative one, namely ϵ-differential privacy.

Proposition 10. *Let P be a process and ρ be a metric on sequences of H actions. Then if $P \in NDC$ then for every $o \in L^*$, $s \in H^*$ there exists ϵ such that $P \in DF_{\epsilon,\rho}(o, s)$. Moreover, if $P \in DF(\epsilon, \rho)$ for some ϵ and ρ is such that $\rho(x, y) \neq 0$ whenever $x \neq y$, then $P \in NDC$.*

Proof. Let $P \in NDC$, i.e. $(P|A) \setminus H \approx_w P \setminus H$ for every A such that $Sort(A) \subseteq H \cup \{\tau\}$. This means that also $(P|s.Nil) \setminus H \approx_w (P|s'.Nil) \setminus H$ and so $Pr(o|s) = 0$ iff $Pr(o|s') = 0$ for every o i.e. it cannot happen that one of these probabilities is non-zero and another one is equal to zero, hence there exists ϵ such that $P \in DF_{\epsilon,\rho}(o, s)$.

Now suppose that for every $o \in L^*$, $s \in H^*$ there exists ϵ such that $P \in DF_{\epsilon,\rho}(o, s)$. This means that for any two secretes if one could output o then also another one can do the same and hence $P \in NDC$.

As regards the metric, there are several meaningful choices how to measure a distance between two secrets. First we consider a variant of Hamming distance.

Definition 13. *Let $s, s' \in Act^*$ and $s = x_1.x_2.\ldots.x_n$, $s' = x'_1.x'_2.\ldots.x'_m$. We define metrics ρ_0 as a number of positions where s and s' differ, i.e. $\rho_0(s, s') = |m - n| + \sum_{i=1, x_i \neq x'_i}^{\min(n,m)} 1$.*

For the metric ρ_0 we have the following result which relates $DF_\epsilon(o, s)$ and $DF_\epsilon(o, s)$ properties.

Proposition 11. *Let $P \in DF_\epsilon(o, s)$ for every $s \in H^*$. Then $P \in DF_{\epsilon,\rho_0}(o, s)$.*

Proof. Suppose that $\rho_0(s, s') = n$, then there exist $s_1, \ldots, s_{n-1}$ such that s_i, s_{i+1} differ by one element as well as s, s_1 and s_{n-1}, s'. Since we have $P \in DF_\epsilon(o, s)$, $P \in DF_\epsilon(o, s_i)$ for all $i, 1 \leq n-1$ we have $Pr(o|s) \leq e^{\epsilon \times n} \times Pr(o|s')$.

The metric ρ_0 does not take into account the length of inputs. If we have two completely different inputs of length 2 and inputs which differ in two positions but both of length 128, in both cases the metric is 2 what does not express an amount of secrecy which could leak or is protected. In the first case the whole secrete is protected and in the second case only a fraction of secrecy could be protected if $P \in DF_{\epsilon,\rho_0}(o, s)$. Hence we could consider more elaborated metrics, for example $\rho_{min}(s, s') = (\rho_o + \min |s|, |s'|)/\min(|s|, |s'|)$, $\rho_{max}(s, s') = (\rho_o + \max |s|, |s'|)/\max(|s|, |s'|)$, $\rho_{sum}(s, s') = (\rho_o + |s| + |s'|)/(|s| + |s'|)$ etc.

Now we can reformulate Definition 7 and 8 taking into account a given metric. We illustrate this by generalization of the set $DF(P, \epsilon, o)$.

Definition 14. *$DF(P, \epsilon, \rho, \delta, o) = \{s|Pr(o|s) > e^{\epsilon \times \delta} \times Pr(o|s')$ and $o \in Tr_w((P|s.Nil) \setminus H)$ and $\rho(s, s') = \delta\}$.*

The sets of secretes $DF(P, \epsilon, \rho, \delta, o)$ represents those secrets which could (at least partially) leak under the observation o. The amount of leakage is given by ρ and δ. It is easy to check that $DF(P, \epsilon, \rho_0, 1, o) = DF(P, \epsilon, o)$. Similarly, we could generalize the set $DF(P, \epsilon, s)$.

Now we have taken into account a more appropriate distance between two secrets but we have omitted a length of observations. It makes a difference if a secrete could leak by short observation or it could leak only by very long observations. For example, if $s_1 \in DF(P, \epsilon, \rho, \delta, o)$ and $|o|$ is small but $s_2 \in DF(P, \epsilon, \rho, \delta, o')$ only for a very big $|o'|$ then s_2 should be considered safer. This leads us to further generalization of ϵ-differential privacy. We consider function f which could take into account a distance between secrete inputs, their length, as well as length of outputs. Moreover, it can incorporate also a cost of observations (it could be different from it length) and other relations.

Definition 15. *$P \in DF_{\epsilon,f}(o, s)$ iff $o \in Tr_w((P|s.Nil) \setminus H)$ and*

$$Pr(o|s) \leq e^{\epsilon \times f(s,s',o)} \times Pr(o|s').$$

We believe that by appropriate choice of the function f we obtain more realistic security properties based on ϵ-differential privacy but we leave this for the further research. But now we turn to another generalization of ϵ-differential privacy which is inspired by opacity (see Definition 3).

Definition 16. *Suppose that we have the predicate ϕ over secrets. Then we define $P \in oDF_{\epsilon,\phi}(o, s)$ if for $o \in Tr_w((P|s.Nil) \setminus H)$ where s is such that $\phi(s)$ holds we have*

$$Pr(o|s) \leq e^{\epsilon} \times Pr(o|s')$$

for some $s' \in H^*$ such that $\neg\phi(s')$.

There is a clear relationship between qualitative property "opacity" Op^{ϕ}_{O} and its quantitative variant based on ϵ-differential privacy.

Proposition 12. *Suppose that for every $o \in L^*, s \in H^*$ there exists ϵ such that $P \in oDF_{\epsilon,\phi}(o, s)$. Then $P \in Op^{\phi}_{\mathcal{O}}$ for $\mathcal{O}$ which maps high level actions, probabilities as well as τ action to empty sequence, and vice versa.*

Proof. The main idea. Let us assume that $P \in oDF_{\epsilon,\phi}(o, s)$. This means that for every secrete s for which ϕ holds there exists s' for which ϕ does not hold. Since we consider the observation function $\mathcal{O}$ which "does not see" high level actions and τ, we have $P \in Op^{\phi}_{\mathcal{O}}$. The proof of the opposite implication is similar.

We can relate $oDF_{\epsilon,\phi}(o, s)$ also to the property $oDF_{\epsilon}(o, s)$.

Proposition 13. *Let us assume that $P \in oDF_{\epsilon}(o, s)$. Then $P \in oDF_{\epsilon,\phi_s}(o, s)$ where $\phi_s(s')$ holds if $s = s'$ and does not hold if s and s' differ in one position.*

Proof. The main idea. Let us assume that $P \in oDF_{\epsilon}(o, s)$. This means that probability of the output o with the secrete input s' which differs form s in one position (i.e. $\phi_s(s), \neg\phi_s(s')$ hold) is non zero and hence $P \in oDF_{\epsilon,\phi_s}(o, s)$.

Note that it is easy to prove that the most of the above mentioned properties (sets) are undecidable in general (it follows from undecidable result stated by Proposition 7). We leave for further work to specify conditions for which they are decidable.

5 Conclusions

We have presented several (quantitative) security concepts based on ϵ-differential privacy. They could be seen as quantifications of some qualitative properties, namely non-deducibility on composition [FGM03] and opacity [BKR04,BKMR06]). They express how secure is the secrete input s with respect to the public output o, which secrete could leak by observing the public output o, which output could leak the secrete s or which processes are completely safe i.e. there is no secrete and output which could leak it. Even very basic of these properties are undecidable in general but we have shown under which conditions they become decidable. But since also in this case complexity remains very high we propose some compositional properties to manage it at least somehow. We propose also some metrics on inputs which could be exploited to obtain more realistic security properties. As it was mentioned, one should consider also length of inputs and relate it to the length of public outputs. Without this we could obtain too restrictive security notions. The price of leakage - as a relation between amount of leaked secrecy with respect to the length of observation is a crucial security characterization. Otherwise no access control process based on passwords would be considered safe (if a number of attempts to guess the password is not limited).

As regards the future work, besides already mentioned plans, we also plan to exploit information theory to express how much information on secrete inputs could leak with a given probability. This is particularly interesting if secrete inputs have qualities which cannot be simply captured. Then we will use differences between entropy of inputs as a metric. Moreover, we plan to concentrate on efficient techniques for checking of above proposed security properties.

References

[BKR04] 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.

[BKMR06] 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.

[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.

[D08] Dwork C.: Differential Privacy: A Survey of Results. Proc. Theory and Applications of Models of Computation, LNCS 4978, 2008.

[FGM03] Focardi R., R. Gorrieri, and F. Martinelli: Real-Time information flow analysis. IEEE Journal on Selected Areas in Communications 21 (2003).

[GSS95] 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.

[GM82] Goguen J.A. and J. Meseguer: Security Policies and Security Models. Proc. of IEEE Symposium on Security and Privacy, 1982.

[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.

[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.

[L02] Lowe G.: Quantifying information flow". In Proc. IEEE Computer Security Foundations Workshop, 2002.

[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

[X12] Xu L.: Modular reasoning about differential privacy in a probabilistic process calculus. In TGC, pages 198212, 2012.

[X14] Xu L., K. Chatzikokolakis, H. Lin and Catuscia Palamidessi: Metrics for Differential Privacy in Concurrent Systems, In Proceedings of HotSpot, 2014.