Measuring Information Flow in Reactive Processes

Chunyan Mu

King's College London,
The Strand, London WC2R 2LS
{chunyan.mu}@kcl.ac.uk

Abstract. This paper outlines an approach for measuring information flow within reactive probabilistic systems. First, we present the probabilistic model of reactive labelled transition system with input-output actions. Second, we present the language and semantics for simple reactive processes, and investigate the quantified information flow analysis over this semantics. Third, we define a metric over the semantics and then present a method to compute the leakage in reactive processes. The metric we considered is the square root of the Jensen-Shannon divergence: the quantitative information is contained in the distance between state transformations given by a process metric. Finally, we show that there is a connection between our leakage definition and mutual information in the framework of information theory.

1 Introduction

Information flow measurement has recently become an attractive research topic in the security community. The goal of information flow security in this content is to guarantee that information propagates throughout the execution environment without security violations such that not too much secure information is leaked to public outputs. Traditionally, the approach of information flow security was based on *non-interference* [1], which enforces that there is no secure information about the high inputs can be deduced by observing the low outputs. However, non-interference is too restrictive, and it is too hard to write useful programs in the real world. We therefore consider a new policy to relax non-interference: the program is secure if the amount of information flow from high (confidential) to low (public) is *not too much* from a quantitative point of view. The precursor for this work was that of Denning in the early 1980's. Denning [2] suggested that the data manipulated by a program can be typed with security levels, and first explored the use of information theory as the basis for a quantitative analysis of information flow in programs. However, she did not suggest how to automate the analysis or attempt to make the analysis formal and complete. Millen [3] first built a formal correspondence between non-interference and mutual information, and established a connection between information theory and state-machine models of information flow in computer systems. Wittbold and Johnson [4] gave an analysis of certain combinatorial theories of computer

S. Qing, C.J. Mitchell, and G. Wang (Eds.): ICICS 2009, LNCS 5927, pp. 211–225, 2009.

security from information-theoretic perspective and introduced non-deducibility on strategies due to feedback and internal non-determinism. There has been much recent work in the information theoretic based foundations of quantitative information flow computation [5,6]. Most of the work in this area to date has concentrated on simple programs in simple imperative languages. However, real world programs normally allow input/output, and behave as a reactive system. It is important to consider a quantitative analysis over reactive systems in the computational world. There also have been several attempts on probabilistic and concurrent systems: Di Pierro, Hankin and Wiklicky [7] gave a definition of probabilistic measures on flows in a probabilistic concurrent constraint language where the interference came via probabilistic operators. However, the approach of approximate non-interference presented in this paper is based on the specific probabilistic declarative language PCCP. It seems difficult to automate in other framework. Gavin Lowe [8] measured information flow in CSP by counting refusals. He devised a formal definition of information quantity transmitted from a high level user to a low level user in a computing system. The definition was based on the number of different behaviours of High that can be distinguished from Low's point of view. Like other quantitative definitions, Lowe's definition was based on Shannon's information theory. However, this approach did not consider probabilistic behaviours. Boreale [9] studied the quantitative models of information leakage in the process calculi by applying an information theoretic framework. The *absolute leakage* measured in bits, present the absolute leakage of zero precisely when it satisfies secrecy. The *rate of leakage*, measured in bits per action, presented the maximum information extracted by repeated experiments coincided with the absolute leakage of the process. A weakness of the ratio formulation was that it was difficult to apply to recursive processes.

All to work to date suffers several problems: some of the works provided reasonable analysis on simple program in simple imperative languages, but did not work for programs with complex behaviours like interactions [5,6,10]; some of the works was able to process interactions but the reasonability and completeness of the approach was somewhat weak [8,7]. In this paper, we consider quantitative information flow in reactive systems with input, output, and probabilistic behaviours. The basic concept of our work is that the quantity of information flow is considered by looking at the different behaviours of a high user from a low user's distribution-based observations. We introduce a method to provide a quantitative analysis of information flow for reactive processes due to metric spaces on the process domain. A metric space is built over the execution of the programs via the semantics defined, and the information flow is measured via metrics. The metric we choose here corresponds to the framework of information theory. The attack model in our system considers situations in which a sequence of confidential inputs can be fed into the processes or programs. The attacker can communicate with the program via a set of input-output behaviours. The input-output actions are guarded by probabilistic choices which are following probability distributions. In other words, to capture the secure information flows, we consider the input-output actions with different security levels: high and low

which are governed by high level users and low level users respectively. Low level users are not allowed to observe high level actions but not vice versa. Executing the program produces a set of distribution-based traces in which only low level input-output actions are visible. By observing the visible traces of the program, the attacker tries to collect and deduce some confidential information via the observations. The model we applied for measuring secure information flow within reactive processes is based on probabilistic labelled transition system. The notation of observations is used to provide a basis for recording the history traces of behaviours from the view of low users. Intuitively, the system produces a set of weighted observation trees. Next, inspired by the methodology introduced by [11,12], we define a process domain based on metric spaces, and the metric is with respect to Jensen-Shannon divergence. We use this metric to compute the distance between the different views of the low users due to different behaviours of the high user. We then introduce a method to quantify the secure information flow within processes based on such distances: the quantitative information is contained in the distance within state transformations of the tree set given by a process metric. There are many metrics can be used to measure the distance of distributions, we show that the Jensen-Shannon divergence is a suitable measure of the information flow quantity. To show the intuition behind our method, we discuss that there is a connection between our definition and mutual information in the framework of information theory. We believe our approach provides a reasonable measurement on secure information flow in processes.

The rest of the paper is organized as follows. Section 2 explains the probabilistic model of reactive processes. In Section 3, we present a simple language and semantics for reactive probabilistic processes. Section 4 introduces the method for leakage computation over reactive processes. Finally, we draw conclusions in Section 5.

2 Reactive Probabilistic Labelled Transition System

This section presents a model of reactive probabilistic labelled transition systems. We consider our probabilistic model to be reactive in the sense that the system can react to the environment if fed with a set of high inputs equipped with a probability distribution: by executing a set of low level input-output actions, the system produces a set of observation trees in the way of resulting distributions to the outside.

2.1 Reactive Probabilistic Labelled Transition System

First of all, the model of quantitative reactive systems considered here is based on Probabilistic Labelled Transition Systems (PLTS). In order to consider probabilistic behaviour and information flow measurement, we consider probabilistic labelled transition systems incorporating probability distributions. A *probability distribution* on a set M is a function $f : M \to [0, 1]$ such that the set $\{m \in M | f(m) > 0\}$ is finite and $\sum_{m \in M} f(m) = 1$. Intuitively, probabilistic

labelled transition systems are labelled transition systems with probabilities attached to each transition, such that transitions are considered as $P \xrightarrow{a}_{\mu} Q$, denoting P performing an a labelled transition and then behaving as the state Q with probability μ. A formal definition based on Larsen and Skou's [13] probabilistic model is presented as follows.

Definition 1 (Probabilistic Labelled Transition System). *The probabilistic labelled transition system is given as a triple* $PLTS = (T, \Sigma, \mu)$*, where* T *is a set of states,* Σ *is a set of actions,* μ *is a family of probability distributions, such that* $\mu : T \to \Sigma \to (T \to [0,1])$*. Specifically,* $\mu_{p,a} : T \to [0,1]$*,* $\mu_p : \Sigma \to T \to [0,1]$*, where for any* $a \in \Sigma$ *and* p *is a state that can perform the action* a*, indicating the possible next states and their probabilities after* p *has performed* a*,* i.e. $\mu_{p,a}(q) = \lambda$ *means that the probability that* p *becomes* q *after performing* a *is* λ*. Furthermore,* $\forall p \in T$ *and can perform action* a*,* $\sum_{p' \in T} \mu_{a,p}(p') = 1$*,* i.e.*,* $\mu_{p,a}$ *is a probability distribution.*

Second, to allow reactive behaviours, following [12], we consider that the transition relation $\to$ is between a set of states and *certain sets*. The sets are defined as a set of a pair consisting actions (Σ) and probability distribution on states ($\mu(\mathcal{R})$): $\{\Sigma \times \mu(\mathcal{R})\}$, which must satisfy the *reactiveness condition*: $X \subseteq \Sigma \times \mu(\mathcal{R})$ is said to satisfy the reactive condition if for any $(a_1, s_1), (a_2, s_2) \in X$ either $a_1 \neq a_2$ or $(a_1, s_1) = (a_2, s_2)$. The definition of the reactive probabilistic labelled transition system is presented as follows on the basis of Norman's definition [14].

Definition 2 (Reactive Probabilistic Labelled Transition System). *A simple reactive probabilistic transition system is defined as a tuple* $(\mathcal{R}, \Sigma, \to)$*, where* $\mathcal{R}$ *is a set of states,* Σ *is a finite set of actions, and* $\to$ *is a transition relation*

$$\to \subseteq \mathcal{R} \times \wp(\Sigma \times \mu(\mathcal{R}))$$

satisfying: for all $E \in \mathcal{R}$ *there exists* $S \in \wp(\Sigma \times \mu(\mathcal{R}))$ *such that* $(E, S) \in \to$*, written as* $E \to S$*, where* $\wp(\cdot \times \cdot)$ *denotes the power set of operators restricted to only finite subsets of Cartesian products satisfying the reactiveness condition.*

Let us consider an example of the application of the RPLTS. Consider any $S \in \wp(\Sigma \times \mu(\mathcal{R}))$ as a reactive probabilistic process in which the first move of its behaviour (input action) is made by external choice under a distribution. The initial high input actions set $?H = \{h_1, \ldots, h_m\}$ feeds into the system. For any $1 \leq i \leq m$ and $F \in \mathcal{R}$, the probability of $S = \{(h_1, w_1), \ldots, (h_m, w_m)\}$ performing the action h_i as their initial move and then behaving as F_i is also given by $w_i(F)$ where $w_i(F) \in \mu(R)$. Intuitively, the RPLTS produces a set of trees. Note that $\{w_i | 1 \leq i \leq m\}$ is a probability distribution thus $\sum_{i=1}^{m} w_i = 1$. The action for a particular channel is incorporated into a distribution function on the events that occur on the channel. The execution of the reactive processes can be viewed as action-guarded and is on the basis of the probability distributions. The resulting distributions are obtained by executing a sequence of input-output actions, and can be viewed as the reaction of the system in the way of sets of probabilistic

synchronisation sub-trees due to each high-input triggered interaction. We thus define distribution-based observations to capture the transformation of each visible interaction step of the processes. At the end of execution a full description of all the sub-trees is obtained in the form of observations based on probability distributions. The notation of observations will be further discussed in Section 2.3.

2.2 The Security Model

The environment is high and low users: low can not observe high inputs and outputs, but high can observe low. In addition, low knows text or description of the program. The way of interaction between users and the system is based on the input-output actions over channels with security levels. We consider two levels: H and L, where H denotes high-level confidentiality and L denotes low-level confidentiality. On the other hand, in order to simply concentrate on the quantitative analysis of secure information flow, we consider a partition over the actions (labels) as: input $?A$, output $!A$ and internal τ. Put two kind of partitions together we have: $\Sigma ::= ?A_H \mid !A_H \mid ?A_L \mid !A_L \mid \tau$. Internal action τ can not be seen from the outside and happens automatically. Low level input and output actions are visible to the external environment.

2.3 Observations

Assume there are a sequence of high inputs to the RPLTS, the low observations are the probability distribution on the low traces due to the high inputs. Information on the projection of the high inputs from the trace can be deduced from these observations. Observations are used to record the history of *observable* transformations of the system on each interaction step during the executions due to the high inputs, *i.e.* a sequence of visible communications that the system might communicate. The observation set is generated by the system, which is defined as a map from a set of states $\mathcal{R}$ to a probability distribution of observable behaviours on the experiments by performing a set of input actions $?H$: $\mathcal{R} \to (?H.T \to [0,1])$. We put $?H.\bot = 1$, where $\bot$ denotes the case of inactive processes. Each set of high inputs $?H$ introduces an interaction step. During this interaction step, low users are communicating with the system via input-output actions. The system therefore produce an observation tree due to such behaviours of the low traces. The next turn of high inputs $?H'$ starts another interaction step and so on. We present the definition of an interaction step and the observation due to each interaction as follows.

Definition 3 (Interaction unit). *We define an interaction unit as the set of the computation steps of processes due to one set of distribution-based high inputs $?H$. The system thus produces a set of visible computational behaviours as a reaction due to such high inputs: $\mathcal{R} \to (?H.T \to [0,1])$, where $\mathcal{R}$ is a set of states, T is the experiment starting with $?H$, and $?H$ is the set of high inputs which starts this interaction and follows a distribution $\{w_i | 1 \leq i \leq m\}$, w_i denotes the probability of each input: $0 < w_i \leq 1$, $\sum_{i=1}^{m} w_i = 1$, and m is the size of the inputs.*

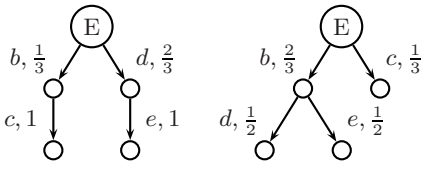
Definition 4 (Observation). *Observation of one interaction unit on a set of states is defined as the set of all E's finite visible history traces, and is described as a distribution set $\{\pi_i | 1 \leq i \leq n\}$ due to each high input h_i with its weight (probability) w_i at the beginning of this interaction, where π_i is a distribution obtained by the probabilistic computation tree started by $?h_i$. Note that high input h_i also follows a distribution, i.e., $\sum_{i=1}^{n} w_i = 1$. Let $?H = \{h_i | 1 \leq i \leq n$, we have,*

$$\mathcal{O}(?H.E) = \{w_i \cdot \sum_{j=1}^{m_i} p_{ij}.L_{ij}.\bot | 1 \leq i \leq n\}$$

where L_{ij} denotes a set of visible labels (actions) set over the channel L, $\bot$ denotes the action leads to an inactive state. Note that, $\pi_i = \{p_{ij} | 1 \leq i \leq n, 1 \leq j \leq m_i\}$ is a distribution, $\sum_{j=1}^{m_i} p_{ij} = 1$, $\sum_{i=1}^{n} w_i = 1$, $\sum_{i=1}^{n} w_i \cdot \sum_{j=1}^{m_i} p_{ij} = 1$.

Let us consider an example to show how the observation works.

Example 1. Consider simple process E with one interaction in Figure 1: $?h_1.E = \frac{1}{3}b.c.\bot + \frac{2}{3}d.e.\bot$, $?h_2.E = \frac{2}{3}b.(\frac{1}{2}d + \frac{1}{2}e).\bot + \frac{1}{3}c.\bot$, and assume $?H = \{h_1, h_2\}$ (where h_1 with weight $\frac{1}{3}$, h_2 with weight $\frac{2}{3}$), and b, c, d, e are visible actions which can be low inputs/outputs.



$$\mathcal{O}(h_1.E) = \frac{1}{3}b.c.\bot + \frac{2}{3}d.e.\bot$$
$$\mathcal{O}(h_2.E) = \frac{2}{3}b.(\frac{1}{2}d + \frac{1}{2}e).\bot + \frac{1}{3}c.\bot$$

$$\mathcal{O}(?H.E) = \{\frac{1}{3}h_1[\frac{1}{3}b.c.\bot + \frac{2}{3}d.e.\bot],\ \frac{2}{3}h_2[\frac{2}{3}b.(\frac{1}{2}d + \frac{1}{2}e).\bot + \frac{1}{3}c.\bot]\}$$

Fig. 1. Example of observations due to interaction unit

3 The Language and Its Semantics of Process Algebra

We consider a CSP-like probabilistic input-output security process algebra, which includes deterministic probabilistic choice, synchronous concurrency, and recursion.

3.1 The Language

The syntax of all expressions is given as follows:

$$F ::= \bot \mid x \mid \sum_{i \in I} a_i.p_i.F_i \mid F_1 \| F_2 \mid \mu x.F$$

where $\perp$ denotes the inactive process that does nothing or a state of termination; x denotes a variable, $\sum_{i=1}^{n} a_i.p_i.F_i$ denotes action-guarded probabilistic choice, where a is a parameterised action set: $a = \bigcup a_i \in \{?H, !H, ?L, !L\}$, p_i is the probability of performing a_i and then behaving as F_i, $p_i \in (0, 1]$, $\sum_{i=1}^{n} p_i = 1$; $F_1 \| F_2$ denotes synchronous parallel composition; $\mu x.F$ denotes recursion.

3.2 Operational Semantics

The set of states of the probabilistic labelled transition system $(\mathcal{R}, \Sigma, \rightarrow)$ can be considered as the set of a pair consisting of an action and a probability distribution: $\Sigma \times \mu(\mathcal{R})$, the transition relation is defined as: $\rightarrow \subseteq \mathcal{R} \times (\Sigma \times \mu(\mathcal{R})) \cup \{\emptyset\}$. The pair element of the set is written as: $\pi.a = \sum_{i=1}^{n} p_i.a_i$, where π denotes the probability distribution, a denotes the parameterised action set: $a \in \{?H, !H, ?L, !L\}$, $\sum_{i=1}^{n} p_i = 1$, $a_i \in a$, p_i denotes the probability of a_i. The action τ is invisible to the external environment and happens automatically. The probabilistic labelled transition system with a set of prefixing inputs can therefore be unwound into a synchronisation tree. The semantics are presented as follows in Table 1, where $E\{F/x\}$ denotes the result of replacing all free occurrences of x in E by F.

Probabilistic choices $a.E$ are guarded by the probabilistic actions following a probability distribution. Consider high inputs as $?H = \{h_i | 1 \leq i \leq m\}$, where m is the size of the high inputs, and the probability of performing h_i is w_i. Process E performs h_i and then behaves as E_i. According to the semantics rules, the process is written as $?H.E = \sum_{i=1}^{m} w_i.h_i.E_i$. The relative probabilistic labelled transition system thus produces a set of probabilistic synchronise trees: $\{w_i.h_i.E_i | 1 \leq i \leq m\}$. In the parallel operator, for visible action a, some $\pi_1, \pi_2 \in \mu(\mathcal{R})$: if $E_1 \xrightarrow{a}_{\pi_1} E_1'$ and $E_2 \xrightarrow{a}_{\pi_2} E_2'$. It is clear that $\pi = \pi_1\pi_2 \in \mu(\mathcal{R})$ is still a distribution. In practice, most recursions are guarded. The first n steps of the behaviour of a guarded recursion $\mu x.F(x)$ can be obtained by unwinding the recursion n times: $F^n(\mu x.F(x))$.

Proposition 1. *For any process, the synchronisation tree produced by its operational semantics forms a RPLTS.*

For proof see our techniqual report [15].

Table 1. Operational Semantics

$$\texttt{Act}\ \frac{E \xrightarrow{a_i} p_i\ E_i}{E \xrightarrow{a}_{\pi} \sum_{i=1}^{n} p_i.a_i.E_i} \qquad \pi \text{ denotes the probability distribution: } \{p_i | 1 \leq i \leq n\}$$
$$p_i \in [0, 1],\ \sum_{i=1}^{n} p_i = 1,\ a = \{a_i | 1 \leq i \leq n\} \in \{?H, !H, ?L, !L\}$$

$$\texttt{Par}\ \frac{E_1 \xrightarrow{\tau} E_1'}{E_1 \| E_2 \xrightarrow{\tau} E_1' \| E_2} \quad \frac{E_2 \xrightarrow{\tau} E_2'}{E_1 \| E_2 \xrightarrow{\tau} E_1 \| E_2'} \quad \frac{E_1 \xrightarrow{a}_{\pi_1} E_1' \quad E_2 \xrightarrow{a}_{\pi_2} E_2'}{E_1 \| E_2 \xrightarrow{a}_{\pi_1, \pi_2} \pi_1\pi_2.a.(E_1' \| E_2')} (a \neq \tau)$$

$$\texttt{Rec}\ \frac{}{\mu x.E \xrightarrow{\tau} E[\mu x.E/x]}$$

3.3 Observing Behaviour and Equivalence Relations

A set of observable process traces can therefore be extracted from its operational semantics. For our purpose of quantitative security analysis, a coarser equivalence is considered more satisfactory as it will only distinguish processes that can be distinguished by external low-level observations.

Definition 5 (Probabilistic low bi-simulation). *The low bi-simulation* $\sim_L$ *is a relation on the set of processes* $\mathcal{R}$*:* $\{E_i | 1 \leq i \leq m\}$ *produced by the probabilistic transition system due to a high inputs set* $?H = \{w_i.h_i | 1 \leq i \leq m\}$*, such that, if* $E_i \sim_L E_j$ *(*$1 \leq i, j \leq m$ *and* $i \neq j$*) then:*

$$\forall a \in \{?L, !L\}. \forall S \in L^{\sim}. E_i \xrightarrow{a}_{\mu} S \Leftrightarrow E_j \xrightarrow{a}_{\mu} S$$

where $L^{\sim}$ *denotes the set of low bi-simular classes of* $\mathcal{R}$ *and* $E_i \xrightarrow{a}_{\mu} S$ *if and only if* $\mu = \sum\{\mu' | E_i' \in S\}$ *and* $E_i \xrightarrow{a}_{\mu'} E_i'$*. Probabilistic processes* E_i *and* E_j *are called probabilistic low bi-simular if* (E_i, E_j) *is contained in some probabilistic low bi-simulations.*

4 Information Flow Measurement

We propose to introduce a method for measuring the quantity of information flowed from high-level inputs to public visible observations with respect to the observation tree sets.

4.1 A Metric for Probabilistic Processes

Inspired by the metric space construction for denotational semantics defined by De Bakker & Zucker in [11], and Kwiatkowska & Norman in [12], we introduce a metric with respect to the square root of Jensen-Shannon divergence for the purpose of secure information flow measurement.There are several reasons we choose the JSD as a measure of the difference between distributions. First, JSD is related to other information-theoretical functionals, such as the relative entropy or Kullback Leibler distance. It therefore shares their mathematical properties as well as their intuitive interpretability [16]. Unlike the Kullback Leibler (KL) distance, it is symmetric, always well defined and bounded ($0 \leq \mathtt{JSD} \leq 1$). Second, our system produces a set of transition trees with regard to the weighted high input actions, which may contains more than two elements. JSD can be generalised to measure the distance between more than two distributions, and the compared distributions can be weighted. Third, the square root of JSD ($\sqrt{D_{\mathtt{JS}}}$) is a true metric for the probabilistic distributions space. The $\sqrt{D_{\mathtt{JS}}}$ verifies the triangle inequality, which provides us a potential way to consider the leakage bounds for reactive processes.

Consider m distributions $P^{(1)}, P^{(2)}, \ldots, P^{(m)}$ and let $w^{(1)}, w^{(2)}, \ldots, w^{(m)}$ denote the corresponding weights. The Jensen-Shannon distance [17] between the m distributions $P^{(1)}, \ldots, P^{(m)}$ with weights $w^{(1)}, \ldots, w^{(m)}$ is given by

$$D_{\mathtt{JS}}(P^{(1)}, P^{(2)}, \ldots, P^{(m)}) = \mathcal{H}(\sum_{j=1}^{m} w^{(j)} P^{(j)}) - \sum_{j=1}^{m} w^{(j)} \mathcal{H}(P^{(j)})$$

To build a true metric space over the denotational semantics of process calculi with respect to the information flow measurement, we consider the *square root* of the Jensen-Shannon divergence as a metric as follows:

Definition 6. *For a set of processes $f_1, \ldots, f_m \in \mathcal{R}$, $d(f_1, \ldots, f_m)$ is defined as the square root of the JSD among m distribution trees $P^{(1)}, \ldots P^{(m)}$ with weights $w^{(1)}, \ldots, w^{(m)}$,* i.e.

$$d(f_1, \ldots, f_m) = \sqrt{\mathcal{H}(\sum_{j=1}^{m} w^{(j)} P^{(j)}) - \sum_{j=1}^{m} w^{(j)} \mathcal{H}(P^{(j)})}$$

Proposition 2. *For any processes $f_1, \ldots, f_m \in \mathcal{R}$, $d(f_1, \ldots, f_m)$, $d(f_1, \ldots, f_m) = 0$ iff $P^{(1)} \sim_L \ldots \sim_L P^{(m)}$.*

For proof see our techniqual report [15].

4.2 Quantity of the Information Flow

We have already built a metric space for the system, which can be used to measure the distances within processes. In this section, we introduce a definition of information flow quantity for reactive processes over the metric space. The definition of leakage needs to capture how much secure information contained in the high input is released to the public output. First let us concentrate on one interaction step. Consider a set of input actions $?H = \{h_1, h_2, \ldots, h_m\}$ with distribution π upon process E: $\pi(E)$ assigns a set of weight/probability $(w_1, w_2, \ldots, w_m)$ on a set of process trees. The system start to move with the high input actions and then execute a set of actions based on the structure of the transition system, produce a set of weighted trees denoted by $P^{(1)}, P^{(2)}, \ldots, P^{(m)}$, and thus generate an observation set: $\{\mathcal{O}_i(E) | 0 \leq i \leq m\}$ due to the tree set. The observation set maps the process E into a resulting distribution. The information leakage is defined as the square of the distance between the resulting distributions on the tree set: $\mathcal{L} = d^2(P^{(1)}, P^{(3)}, \ldots, P^{(m)})$.

Definition 7 (Leakage of one interaction). *Assume one high input action set $?H = \{h_1, \ldots, h_m\}$ operates on processes $\mathcal{R}$. The observation tree set produced by the system is denoted by $(P^{(1)}, \ldots, P^{(m)})$, where $P^{(i)} = h_i.\mathcal{R}_{1 \leq i \leq m}$ with weight w_i describes the probabilistic transformations of each observed tree. The leakage on one interaction due to such processes is defined as the square of the metric between the observed tree set:*

$$\mathcal{L} = d^2(P^{(1)}, \ldots, P^{(m)}) = \mathcal{H}(\sum_{i=1}^{m} w^{(i)} P^{(i)}) - \sum_{i=1}^{m} w^{(i)} \mathcal{H}(P^{(i)})$$

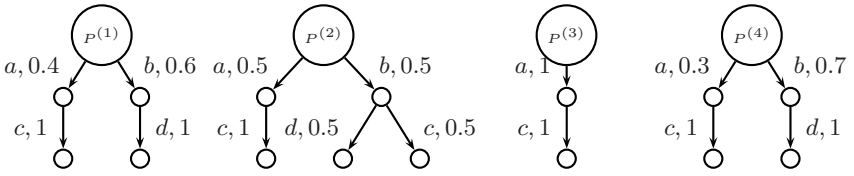


Fig. 2. Example of the leakage computation

Example 2. Assume a high input action set $?H = \{h_i \rightarrow w_i | 1 \leq i \leq 4\}$ as: $h_i \rightarrow 0.2, h_2 \rightarrow 0.4, h_3 \rightarrow 0.1, h_4 \rightarrow 0.3$. By accepting the set of input actions as the initial move, the system produces a set of process trees as follows:

$$\mathcal{O}(P^{(1)}) = 0.4 \cdot a.c.\bot + 0.6 \cdot b.d.\bot$$
$$\mathcal{O}(P^{(2)}) = 0.5 \cdot a.c.\bot + 0.25 \cdot b.d.\bot + 0.25 \cdot b.c.\bot$$
$$\mathcal{O}(P^{(3)}) = 1 \cdot a.c.\bot$$
$$\mathcal{O}(P^{(4)}) = 0.3 \cdot a.c.\bot + 0.7 \cdot b.d.\bot$$

We know that the weight of the process trees are: $w^{(1)} = 0.2$, $w^{(2)} = 0.4$, $w^{(3)} = 0.1$, $w^{(4)} = 0.3$. According to our definition of information flow quantity over processes, we have:

$$\mathcal{L} = d^2(P^{(1)}, P^{(2)}, P^{(3)}, P^{(4)}) = \mathcal{H}(\sum_{i=1}^{4} w^{(i)} P^{(i)}) - \sum_{i=1}^{4} w^{(i)} \mathcal{H}(P^{(i)}) = 0.31$$

Clearly, zero leakage implies the intuition that for different high inputs, the observer can not tell the difference between them by observing the process trees, and therefore it satisfies non-interference. Bigger leakage implies the observer can tell more difference of the process trees due to the high inputs, and thus more secure information is leaked to the public.

4.3 Measuring Information Flow Over Interaction Steps

One interaction step in the reactive labelled transition systems produces a pair set: $\Sigma \times \mu(\mathcal{R})$. Assume the initial high inputs $?H_0$ start the first interaction step as: $?H_0 = \{(w_{01}.h_{01}), \ldots, (w_{0m_0}, h_{0m_0})\}$, where $\sum_{i=1}^{m_0} w_{0i} = 1$. Due to such initial high inputs set, the system produces a set of probabilistic sub-trees. Each computation step of each sub-tree is described as a pair set $\Sigma \times \mu(E_0)$, where E_0 denotes the set of the states taking $?H_0$ as the initial move. During the execution of the program, we may have a sequence of high inputs sets as: $\{?H_0, ?H_1, \ldots, ?H_k\}$, where $?H_0 = \{(w_{01,h01}), \ldots, (w_{0m_0}, h_{0m_0})\}$, $\ldots$, $?H_k = \{(w_{k1}, h_{k1}), \ldots, w_{km_k}, h_{km_k})\}$. For each interaction, we have obtained the metric space $(\xi[i], d_i)$, where $\xi[i]$ denotes the set of probabilistic computation sub-trees due to $?H_i$, d_i denotes the distance between the computational sub-trees, and $0 \leq i \leq k$. We therefore consider the collection of the metric spaces: $\{(\xi[i], d_i) | 0 \leq i \leq k\}$. Each interaction tree may be incorporated with

a probability of this interaction happens and behaves as an active process. The probability of the process taking $?H_i$ can be considered as the product of the probabilities from the root to the current state which is going to take $?H_i$ as the next move, denoted as q_i, where $0 \leq i \leq k$. Such collection describes the history of the sequence of interaction trees with their probabilities. We then define the maximum information flow leakage due the sequence of high inputs sets as the square of the probabilistic sum of the distance between the sub-trees of each interaction step, *i.e.*, $\mathcal{L} \leq (\sum_{i=1}^{k}(q_i \cdot d_i))^2$.

Example 3. Consider we have a sequence of high inputs set with two elements: the initial one is $?H_0 = \{\frac{1}{3}h_{01}, \frac{2}{3}h_{02}\}$ which is input at the beginning of the program, another one is $?H_1 = \{\frac{1}{2}h_{11}, \frac{1}{2}h_{12}\}$ which is input into the process during the execution of the program. Assume actions a, b, c are visible. Consider the total distribution tree obtained due to the program is as follows:

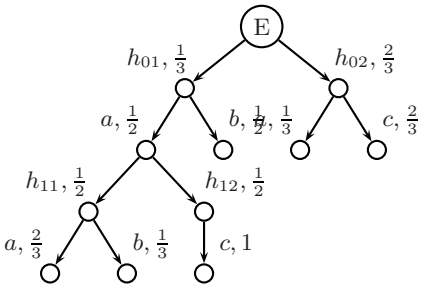


Fig. 3. Example of interactions

$$\mathcal{O}(\xi[0]) = \{\frac{1}{3}h_{01} \cdot [\frac{1}{2}a.\bot + \frac{1}{2}b.\bot],\ \frac{2}{3}h_{02} \cdot [\frac{1}{3}a.\bot + \frac{2}{3}c.\bot\}$$
$$\mathcal{O}(\xi[1]) = \frac{1}{6} \cdot \{\frac{1}{2}h_{11} \cdot [\frac{2}{3}a.\bot + \frac{1}{3}b.\bot],\ \frac{1}{2}h_{12} \cdot [c.\bot]\}$$

where $\xi[0]$ denotes the truncation of the tree of the process after taking the initial high inputs set but before the second high inputs getting in, $\xi[1]$ denotes the truncation of the tree after taking the second high inputs set with its probability $\frac{1}{6}$ due to current state. Let d_0 denote the distance within $\xi[0]$, and d_1 denote the distance within $\xi[1]$, the maximum information flow quantity from the sequence of high inputs set $?H_0, ?H_1$ to the outside is computed by: $d_0 = \sqrt{0.459} = 0.677$, $d_1 = 1$, $\mathcal{L} \leq (d_0 + \frac{1}{6} \cdot d_1)^2 = 0.712$.

Definition 8 (Leakage upper bound over multi-interaction steps). *Assume we have a sequence of high inputs sets, which are fed into the program at the beginning of and during the execution of the program:* $\{?H_0, ?H_1, \ldots, ?H_k\}$, $?H_0 = \{(w_{01}.h_{01}), \ldots, (w_{0m_0}.h_{0m_0})\}$, …, $?H_k = \{(w_{k1}.h_{k1}), \ldots, (w_{km_k}, h_{km_k})\}$, *assume the probabilities of taking* $?H_0, \ldots, ?H_k$ *and behaving as an active process are* $q_0, q_1, \ldots, q_k$. *The observation tree sets obtained are:*

$$\left\{\{P^{(01)},\ldots,P^{(0m_0)}\},\ \{P^{(11)},\ldots,P^{(1m_1)}\},\ \ldots\ \{P^{(k1)},\ldots,P^{(km_k)}\}\right\}$$

We define the information leakage upper bound of this program from the sequence of high inputs sets to the public observer as:

$$\mathcal{L}_{\mathtt{lub}} = \left(\sum_{i=0}^{k} q_i \cdot d(P^{(i1)},\ldots,P^{(im_i)})\right)^2$$

4.4 Relationship with Information Theoretic Based Definition

Inspired by the discussion in [16], in this section, we consider a connection between our leakage computation and mutual information to show some intuitions of our method. When only one interaction happens in our reactive probabilistic transition system, the process can be viewed as a batch program produced a set of public outputs in the way of distribution given a set of high inputs under any distribution, *i.e.* a distribution transformer over one interaction from the denotational point of view [10]. Let us concentrate on the case of one interaction step. We have discussed that the observation set provides a set of weighted resulting distributions. Let us denote the observation set due to the truncation of the viewed interaction step as a random variable $\mathcal{O} = \{o_1, o_2, \ldots, o_k\}$. Suppose that the sequence of $\mathcal{O}$ is divided into m subsequences: $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(m)}$ with probability $w^{(1)}, w^{(2)}, \ldots, w^{(m)}$ based on the distribution of the high input action set which starts this interaction. Let us consider a random vector (o, t) where random variables $o \in \mathcal{O}$ and $t \in \{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(m)}\}$ are generated as follows: o denotes the observing element locates at t, t denotes the high subsequence that leads to the process tree containing observing point o. Let p_{ij} denotes the joint probability $o = o_i$ and $t = \mathcal{T}^{(j)}$ for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, m$, and it can be viewed as: $\begin{pmatrix} p_{11}\ p_{12}\ \cdots\ p_{1m} \\ p_{21}\ p_{22}\ \cdots\ p_{2m} \\ \cdots\ \cdots \\ p_{k1}\ p_{12}\ \cdots\ p_{km} \end{pmatrix}$. Then we get that the random variable o assuming the values $o_1, o_2, \ldots, o_k$ with probability $p_1, p_2, \ldots, p_k$. We also get that the random variable t if assuming the values $\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, \ldots, \mathcal{T}^{(m)}$ with probability $w^{(1)}, w^{(2)}, \ldots, w^{(m)}$ where the marginal probability p_i and $w^{(j)}$ are given by $p_i = \sum_{j=1}^{m} p_{ij},\ w^{(j)} = \sum_{i=1}^{k} p_{ij}$, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, m$. Intuitively, $p_i = \sum_{j=1}^{m} w^{(j)} p_i^{(j)}$ defines the probability of finding o_i in the whole tree sequence, *i.e.* $p_i^{(j)} = \sum_{i=1}^{k} \frac{p_{ij}}{w^{(j)}}$ is the normalised probability of finding o_i in $\mathcal{T}^{(j)}$. Therefore $P^{(j)} = \sum_{i=1}^{k} p_i^{(j)}$. We consider the Jensen-Shannon distance of $P^{(1)}, P^{(2)}, \ldots, P^{(m)}$ due to the truncation of one interaction:

$$D_{\mathtt{JS}}(P^{(1)}, P^{(2)}, \ldots, P^{(m)}) = \mathcal{H}(\sum_{j=1}^{m} w^{(j)} P^{(j)}) - \sum_{j=1}^{m} w^{(j)} \mathcal{H}(P^{(j)})$$

$$= \sum_{j=1}^{m} w^{(j)} \sum_{i=1}^{k} p_i^{(j)} \log_2 p_i^{(j)} - \sum_{i=1}^{k} (\sum_{j=1}^{m} p_i) \log_2 (\sum_{j=1}^{m} p_i)$$

Intuitively, $P^{(j)} = \sum_{i=1}^{k} p_i^{(j)}$ is the probability of finding o in all subsequence of t. If $P^{(1)} \sim_L P^{(2)} \sim_L \ldots \sim_L P^{(m)}$ then it is easy to understand that the identity of o does not tell us anything about the identity of high subsequence t from which observing point o is observed, as the probability distribution of o is identical in all subsequence t, *i.e.* by observing visible actions, we can not get any knowledge about high-level input.

Next, let us consider the mutual information in o about sequence t due to high input, which quantifies the amount of information we obtained from learning the identity of the observing element o about the identity of that subsequence t from which element o was observed. It can be mathematically proven [16] that the mutual information in o about t is identical to the mutual information in t about o, and hence we can state that the Jensen-Shannon divergence $D_{\mathtt{JS}}$ quantifies the amount of information we obtain from learning the identity of the chosen element o about the identity of the subsequence t. The mutual information $\mathcal{I}$ in o about t is defined by Shannon [18]:

$$\mathcal{I} = \sum_{i=1}^{k} \sum_{j=1}^{m} p_{ij} \log_2 \frac{p_{ij}}{w^{(j)} p_i} = \sum_{i=1}^{k} \sum_{j=1}^{m} w^{(j)} p_i^{(j)} \log_2 \frac{p_i^{(j)}}{p_i}$$

$$= \sum_{j=1}^{m} w^{(j)} \sum_{i=1}^{k} p_i^{(j)} \log_2 p_i^{(j)} - \sum_{i=1}^{k} p_i \log_2 p_i = D_{\mathtt{JS}}(P^{(1)}, P^{(2)}, \ldots, P^{(m)})$$

Therefore, $D_{\mathtt{JS}}(P^{(1)}, P^{(2)}, \ldots, P^{(m)})$ over one interaction is equal to the mutual information of o about t, and we obtain Proposition 3.

Proposition 3. *Each interaction truncation of the process can be viewed as a batch program which is given a distribution based high inputs and produces distribution based low observations. For this case, our definition is equivalent to the mutual information between high inputs and low observations.*

Example 4. Let us consider an example with one interaction to see the intuitions. Assume we have two possible high input actions with weights $w^{(1)} = 0.4$, $w^{(2)} = 0.6$ as the initial move of the process. The system thus produces trees $t \in \{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}\}$ due to such inputs. Assume that the observation over the first tree $\mathcal{T}^{(1)}$ with weight $w^{(1)} = 0.4$ is as

$$\mathcal{O}(P^{(1)}) = 0.2a.b.\bot + 0.3b.\bot + 0.5a.b.c.\bot, \quad \textit{i.e.} \quad \begin{array}{cccc} p_1^{(1)} & p_2^{(1)} & p_3^{(1)} & p_4^{(1)} \\ 0.2 & 0.3 & 0.5 & 0 \end{array}$$

and the observation over the second tree $\mathcal{T}^{(2)}$ with weight $w^{(2)} = 0.6$ is as

$$\mathcal{O}(P^{(2)}) = 0.3a.b.\bot + 0.4b.\bot + 0.2a.b.c.\bot + 0.1d.\bot, \quad \textit{i.e.} \quad \begin{array}{cccc} p_1^{(2)} & p_2^{(2)} & p_3^{(2)} & p_4^{(2)} \\ 0.3 & 0.4 & 0.2 & 0.1 \end{array}$$

Therefore, the leakage according to our definition can be computed by:

$$D_{\mathtt{JS}}(P^{(1)}, P^{(2)}) = \mathcal{H}(\sum_{j=1}^{2} w^{(j)} P^{(j)}) - \sum_{j=1}^{2} w^{(j)} \mathcal{H}(P^{(j)}) = 0.1034$$

On the other hand, since the joint probability p_{ij} of (o, t) obtained by the distribution trees can be considered as: $\begin{pmatrix} 0.08 & 0.12 & 0.20 & 0 \\ 0.18 & 0.24 & 0.12 & 0.06 \end{pmatrix}$, we then can compute the public output p_i as $(0.26 \quad 0.36 \quad 0.32 \quad 0.06)$. Therefore, mutual information in o about t, *i.e.* the mutual information between public output view $Y = \{p_1, p_2, p_3, p_4\}$ and high input with weight $X = \{w^{(1)}, w^{(2)}\}$ is computed by:$\mathcal{I}(X;Y) = \sum_{i=1}^{4} \sum_{j=1}^{2} p_{ij} \log_2 \frac{p_{ij}}{w^{(j)} p_i} = 0.1034$.

The example illustrates Proposition 3, and thus shows the intuition of the connection between our method of leakage analysis of interactive processes and the information theoretic based definition for batch programs discussed above.

5 Conclusions

We have introduced a method to measure the information flow within reactive system. The probabilistic system we considered is based on probabilistic labelled transition systems. We apply the framework of Kwiatkowska and Norman's metric probabilistic semantics [12], and investigate the quantified security properties over this semantics. We define a metric over the semantics and develop a method to compute the information flow quantity over interaction steps in reactive processes. It is shown that there is a connection between our leakage definition and the framework of information theory and non-interference. We have come out with a novel way of binding the leakage from reactive program system which has RPLTS semantics. This is a big step forward. The outcome is that we get estimating an upper bound on the leakage on reactive processes. However, our observer is very strong. Similar to Boreale's work [9], the observers can observe all the possible actions of the system. Another weakness of our approach is that the leakage is a function of the semantics, in general, it is not executable. A suitable approximation is required. For future work, we plan to look a way to weaken the observers, *e.g.* the observers only can observe low output or some other restrictions. We also want to investigate developing algorithm which is able to compute approximation on the leakage upper bound.

Acknowledgments. I am grateful to David Clark for helpful comments on this work. This work is funded by the EPSRC grant EP/C009967/1 Quantitative Information Flow and Royal Society Project Information Flow in Process Algebras. Also thanks to CREST centre at King's College London for their support.

References

1. Goguen, J., Meseguer, J.: Security policies and security models. In: IEEE Symposium on Security and privacy, pp. 11–20. IEEE Computer Society Press, Los Alamitos (1982)

2. Denning, D.E.R.: Cryptography and Data Security. Addison-Wesley, Reading (1982)
3. Millen, J.: Covert channel capacity. In: Proceeding IEEE Symposium on Resarch in Security and Privacy. IEEE Computer Society Press, Los Alamitos (1987)
4. Wittbold, J.T., Johnson, D.M.: Information flow in nondeterministic systems. In: IEEE Symposium on Security and Privacy, pp. 144–161 (1990)
5. Clark, D., Hunt, S., Malacaria, P.: Quantitative analysis of the leakage of confidential data. ENTCS, vol. 59. Elsevier, Amsterdam (2002)
6. Malacaria, P.: Assessing security threats of looping constructs. In: POPL, Nice, France, pp. 225–235. ACM Press, New York (2007)
7. Pierro, A.D., Hankin, C., Wiklicky, H.: Approximate non-interference. In: CSFW, pp. 3–17 (2002)
8. Lowe, G.: Quantifying information flow. In: Proceedings IEEE Computer Security Foundations Workshop, pp. 18–31 (2002)
9. Boreale, M.: Quantifying information leakage in process calculi. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 119–131. Springer, Heidelberg (2006)
10. Mu, C., Clark, D.: Quantitative analysis of secure information flow via probabilistic semantics. In: ARES, pp. 49–57. IEEE Computer Society Press, Los Alamitos (2009)
11. de Bakker, J.W., Zucker, J.I.: Processes and the denotational semantics of concurrency. Information and Control 54, 70–120 (1982)
12. Kwiatkowska, M., Norman, G.: Probabilistic metric semantics for a simple language with recursion. In: Penczek, W., Szałas, A. (eds.) MFCS 1996. LNCS, vol. 1113, pp. 419–430. Springer, Heidelberg (1996)
13. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing (preliminary report). In: POPL, pp. 344–352 (1989)
14. Norman, G.: Metric Semantics for Reactive Probabilistic Processes. PhD thesis, School of Computer Science, University of Birmingham (1997)
15. Mu, C.: Jensen-shannon divergency as a measure of information flow in reactive processes. Technical Report TR-09-07, Department of Computer Science, King's College London (2009)
16. Grosse, I., Bernaola-Galván, P., Carpena, P., Román-Roldán, R., Oliver, J., Stanley, H.E.: Analysis of symbolic sequences using the jensen-shannon divergence. Phys. Rev. E 65, 041905 (2002)
17. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information theory 37, 145–151 (1991)
18. Shannon, C.E.: A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. 5, 3–55 (1948)