Quantifying Opacity

Béatrice Bérard[1] * John Mullins[2] † Mathieu Sassolas[1] ‡

[1]Université Pierre & Marie Curie
LIP6/MoVe, CNRS UMR 7606
Paris, France

[2]École Polytechnique de Montréal
Campus of the Université de Montréal
Montreal (Quebec), Canada, H3C 3A7

Abstract

In this paper we propose two dual notions of quantitative information leakage in probabilistic systems, both related to opacity for non probabilistic systems. The liberal one measures the probability for an attacker observing a random execution of the system, to be able to gain information he can be sure about. We show that a null value for this measure corresponds to a secure system, in the usual sense of opacity. On the other hand, restrictive opacity is defined as the complement of the information-theoretic notion of mutual information. It measures the level of certitude in the information acquired by an attacker observing the system: we prove that a null value for this second measure corresponds to non opacity. We also show how these measures can be computed for regular secrets and observations. We finally apply them to the dining cryptographers problem and to the crowd anonymity protocol.

1 Introduction

Motivation. Opacity [14] is a very general framework allowing to specify a wide range of security properties a system has to assume when interacting with a passive attacker. The general idea behind it is that an attacker should not gain information by observing the system from the outside. The approach, as most existing information flow-theoretic approaches, is possibilistic. We mean by this that non determinism is used as a feature to model the random mechanism generation for all possible system behaviors. As such, opacity is not accurate enough to take into account two orthogonal aspects of security properties both regarding evaluation of the information gained by the attacker.

The first aspect regards the quantification of security properties. If executions leaking information are negligible with respect to the rest of executions, the overall security might not be compromised. For example if an error may leak information, but appears only in 1% of cases, the program could still be considered safe. The definitions of opacity [6, 2] capture the existence of at least one perfect leak, but do not grasp such a measure.

The other aspect regards the category of security properties a system has to assume when interacting with an attacker able to infer from experiments on the base of statistical analysis. For example, if every time the system goes *bip*, there is 99% chances that action a has been carried out by the server, then every *bip* can be guessed to have resulted from an a. Since more and more security protocols make use of randomization to reach some security objectives [8, 15], it becomes important to extend specification frameworks in order to cope with it.

Contribution. We define two generalizations of possibilistic opacity in the setting of probabilistic automata without non-determinism. One is more liberal than opacity while the other is more restrictive, from a security point of view. Moreover, as opacity itself, they can be instantiated into several probabilistic security properties such as probabilistic non-interference and anonymity. The first notion measures the quantity of information leaked that is to say, the probability for the system to yield *perfect* information. The second, defined in terms of the information theoretic notion of mutual information, provides a measure of the accuracy of a guess that is, the quantity of information which can be inferred by an observer from the secret property. We also show how to compute these values in some regular cases and apply the method to the dining cryptographers problem and the crowd protocols, re-confirming in passing the correctness result of Reiter and Rubin [15].

Related Work. The notion of opacity was introduced recently with the aim to provide a uniform description for

[*] Author partially supported by project DOTS (ANR-06-SETI-003) (French Government).

[†] Author partially supported by the NSERC of Canada under discovery grant No. 13321-2009.

[‡] Author partially supported by project CoChaT (DIGITEO-2009-27HD) (Région Île de France).

possibilistic security properties like non-interference and anonymity [6]. Up to now, probabilistic approaches were mostly centered on verifying specific security properties or computing information leakage and few works tried to extend opacity to a probabilistic setting.

In [16], the author discusses several measures of information leakage for deterministic or probabilistic programs with probabilistic input. These measures quantify the information concerning the input gained by a passive attacker observing the output. Exhibiting programs for which the value of entropy is not meaningful, the author proposes to consider instead the notions of vulnerability and min-entropy.

The authors of [1] study information leakage in systems modeled by process algebras, but they address the specific point of view of probabilistic non-interference and do not relate it to information theory. In [5], an information-theoretic point of view is adopted to measure information leakage in process algebras, but no relation is made with probabilistic security properties.

In [7], or more recently [3], the authors present a probabilistic version of anonymity, also using the tools of information theory, which is computed using regular expressions. Although anonymity can be seen as an instantiation of opacity, these approaches are focused only on anonymity.

In [13], a notion of probabilistic opacity is defined, but restricted to properties whose satisfaction depends only on the initial state of the run. The opacity there corresponds to the probability for an observer to guess from the observation whether the predicate holds for the run. In that sense our restrictive opacity (Section 4) is close to that notion. However, the definition of [13] lacks clear ties with the possibilistic notion of opacity.

Organization of the paper. Section 2 presents the underlying model and recalls definitions from the security and probability fields. Section 3 and 4 present respectively the notions of liberal and restrictive probabilistic opacity and their applications. Section 5 discusses the power and limitations of these definitions and concludes.

2 Preliminaries

In this section, we recall the notions of opacity, entropy, probabilistic automata, and the way to compute the probability of regular events in such automata.

2.1 Possibilistic opacity

The original definition of opacity was given in [6] for transition systems.

Recall that a transition system is a tuple $\Pi = \langle \Sigma, Q, \Delta, I \rangle$ where Σ is a set of actions, Q is a set of states, $\Delta \subseteq Q \times \Sigma \times Q$ is a set of transitions and $I \subseteq Q$ is a subset of initial states. A *run* in Π is a sequence of transitions written as: $\rho = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \cdots \xrightarrow{a_n} q_n$. For such a run, $\mathrm{fst}(\rho)$ (resp. $\mathrm{lst}(\rho)$) denotes q_0 (resp. q_n). We will also write $\rho \cdot \rho'$ for the run obtained by concatenating runs ρ and ρ' whenever $\mathrm{lst}(\rho) = \mathrm{fst}(\rho')$. The set of finite runs starting in state q is denoted by $Run_q(\Pi)$ and $Run(\Pi)$ denotes the set of finite runs starting from an initial state.

Opacity qualifies a predicate φ, given as a subset of $Run(\Pi)$ (or equivalently as its characteristic function $\mathbf{1}_\varphi$), with respect to an *observation function* $\mathcal{O}$ from $Run(\Pi)$ onto a (possibly infinite) set *Obs* of *observables*. Two runs ρ and ρ' are equivalent w.r.t. $\mathcal{O}$ if they produce the same observable: $\mathcal{O}(\rho) = \mathcal{O}(\rho')$. The set $\mathcal{O}^{-1}(o)$ is called an *observation class*. We sometimes write $[\rho]_\mathcal{O}$ for $\mathcal{O}^{-1}(\mathcal{O}(\rho))$.

A predicate φ is opaque on Π for $\mathcal{O}$ if for every run ρ satisfying φ, there is a run ρ' not satisfying φ equivalent to ρ. However, detecting whether an event *did not* occur gives as much information as the detection that the same event *did* occur. In addition, the asymmetry of this definition makes it impossible to use with refinement [2]: opacity would not be ensured in a system derived from a secure one in a refinement-driven engineering process. Hence we use the symmetric notion of opacity, where a predicate is symmetrically opaque if it is opaque as well as its negation. More precisely:

Definition 1 (Symmetrical opacity)**.** *Let Π be a transition system and $\mathcal{O} : Run(\Pi) \to Obs$ a surjective function called observation. A predicate $\varphi \subseteq Run(\Pi)$ is* symmetrically opaque *on Π for $\mathcal{O}$ if, for any $o \in Obs$, the following holds:*

$$\mathcal{O}^{-1}(o) \not\subseteq \varphi \text{ and } \mathcal{O}^{-1}(o) \not\subseteq \overline{\varphi}.$$

2.2 Information-theoretic measures

Recall that, for a countable set Ω, a *discrete distribution* (or *distribution* for short) is a mapping $\mu : \Omega \to [0, 1]$ such that $\sum_{\omega \in \Omega} \mu(\omega) = 1$. For any subset E of Ω, $\mu(E) = \sum_{\omega \in E} \mu(\omega)$. The set of all discrete distributions on Ω is denoted by $\mathcal{D}(\Omega)$. A *discrete random variable* with values in a set Γ is a mapping $Z : \Omega \to \Gamma$.

For a discrete random variable Z on Ω, the *entropy* of Z is a measure of the uncertainty or dually, information about Z, defined by the expected value of $\log(\mu(Z))$:

$$H(Z) = -\sum_z \mu(Z = z) \cdot \log(\mu(Z = z))$$

where $[Z = z]$ is the event $\{\omega \in \Omega \mid Z(\omega) = z\}$ and log is the base 2 logarithm.

For two random variables Z and Z' on Ω, the *joint entropy* of (Z, Z') is given by

$$H(Z, Z') = -\sum_z \sum_{z'} \left(\begin{array}{c} \mu(Z = z, Z' = z') \\ \cdot \log(\mu(Z = z, Z' = z')) \end{array} \right)$$

where $[Z = z, Z' = z']$ is the event $\{\omega \in \Omega \mid Z(\omega) = z \text{ and } Z'(\omega) = z'\}$.

The *conditional entropy* of Z *given the event* $[Z' = z']$ such that $\mu(Z' = z') \neq 0$ is defined by:

$$H(Z|Z' = z') = -\sum_{z} \left(\begin{array}{l} \mu(Z = z|Z' = z') \\ \quad \cdot \log(\mu(Z = z|Z' = z')) \end{array} \right)$$

where $\mu(Z = z|Z' = z') = \frac{\mu(Z=z, Z'=z')}{\mu(Z'=z')}$.

The *conditional entropy* of Z *given the random variable* Z' is defined by:

$$\begin{array}{rcl} H(Z|Z') & = & \sum_{z'} \mu(Z' = z') \cdot H(Z|Z' = z') \\ & = & -\sum_{z'} \mu(Z' = z') \\ & & \quad \cdot \left[\sum_{z} \begin{pmatrix} \mu(Z = z|Z' = z') \\ \cdot \log(\mu(Z = z|Z' = z')) \end{pmatrix} \right] \\ & = & -\sum_{z} \sum_{z'} \left(\begin{array}{l} \mu(Z = z, Z' = z')) \\ \quad \cdot \log(\mu(Z = z|Z' = z') \end{array} \right) \end{array}$$

which can be interpreted as the average entropy of Z that remains after the observation of Z'.

The *mutual information* between Z and Z' is given by:

$$\begin{array}{rcl} I(Z; Z') & = & H(Z) - H(Z|Z') \\ & = & H(Z') - H(Z'|Z) = I(Z'; Z) \end{array}$$

and measures the decrease of uncertainty about Z resulting from the observation of Z' or dually, the information gained about Z from the observation of Z'. See [10] for further properties of entropy and mutual information and [16] for a discussion.

2.3 Probabilistic automata

In this work, systems are modeled using probabilistic automata behaving as finite automata where non-deterministic choices for the next action and state or deadlock are randomized.

Recall that a finite automaton (FA) is a tuple $\Pi = \langle \Sigma, Q, \Delta, I, F \rangle$ where $\langle \Sigma, Q, \Delta, I \rangle$ is a finite transition system and $F \subseteq Q$ is a subset of final states. The automaton is deterministic if I is a singleton and for all $q \in Q$ and $a \in \Sigma$, the set $\{q' \mid (q, a, q') \in \Delta\}$ is a singleton. Runs in Π, $Run_q(\Pi)$ and $Run(\Pi)$ are defined like in a transition system. A run of an FA is *accepting* if it ends in a state of F. The *trace* of a run $\rho = q_0 \xrightarrow{a_1} q_1 \cdots \xrightarrow{a_n} q_n$ is the word $\mathrm{tr}(\rho) = a_1 \cdots a_n \in \Sigma^*$. The *language* of Π, written $\mathcal{L}(\Pi)$, is the set of traces of accepting runs starting in an initial state.

Replacing in a FA non-deterministic choices by choices based on a discrete distribution results in a *fully probabilistic automaton* (FPA). Consistently with the standard notion of substochastic matrices, we also consider a more general class of automata, *substochastic automata* (SA), which allow to describe subsets of behaviors from FPAs, see Figure 1 for examples. In both models, no non-determinism remains, thus the system is to be considered as autonomous: its behaviors do not depend on an exterior probabilistic agent acting as a scheduler for non-deterministic choices.

Definition 2 (Substochastic automaton)**.** *Let* $\surd$ *be a new symbol representing a termination action. A* substochastic automaton *(SA) is a tuple* $\langle \Sigma, Q, \Delta, q_0 \rangle$ *where*

- Σ *is a finite set of actions,*
- Q *is a finite set of states,*
- $\Delta : Q \to ((\Sigma \times Q) \uplus \{\surd\} \to [0, 1])$ *is a mapping such that for any* $q \in Q$,

  $$\sum_{x \in (\Sigma \times Q) \uplus \{\surd\}} \Delta(q)(x) \leq 1$$

  Δ *defines substochastically the action and successor from the current state, or the termination action* $\surd$*,*
- q_0 *is the initial state.*

A *fully probabilistic automaton* (FPA) is a particular case of SA where for all $q \in Q$, $\Delta(q) = \mu$ is a distribution in $\mathcal{D}((\Sigma \times Q) \uplus \{\surd\})$ *i.e.*

$$\sum_{x \in (\Sigma \times Q) \uplus \{\surd\}} \Delta(q)(x) = 1.$$

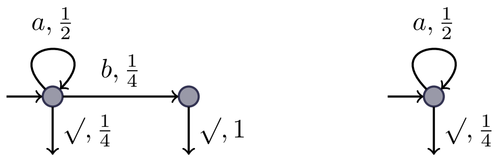
In SA or FPA, we write $q \to \mu$ for $\Delta(q) = \mu$ and $q \xrightarrow{a} r$ whenever $q \to \mu$ and $\mu(a, r) > 0$. We also write $q \cdot \surd$ whenever $q \to \mu$ and $\mu(\surd) > 0$. In the latter case, q is said to be a *final state*.

The notation above allows to define a run for an SA like in a transition system as a finite sequence of transitions written $\rho = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \cdots \xrightarrow{a_n} q_n$. The sets $Run_q(\Pi)$ and $Run(\Pi)$ are defined like in a transition system. A *complete run* is a sequence denoted by $\rho \cdot \surd$ where ρ is a run and $\Delta(\mathrm{lst}(\rho))(\surd) > 0$. The set $CRun(\Pi)$ denotes the set of complete runs starting from the initial state.

The *trace* of a run for an SA Π is defined like in finite automata. The *language* of a substochastic automaton Π, written $\mathcal{L}(\Pi)$, is the set of traces of complete runs starting in an initial state.

For an SA Π, a mapping $\mathbf{P}_\Pi$ into $[0, 1]$ can be defined inductively on the set of complete runs by:

$$\begin{array}{rcl} \mathbf{P}_\Pi(q\surd) & = & \mu(\surd) \\ \mathbf{P}_\Pi(q \xrightarrow{a} \rho) & = & \mu(a, r) \cdot \mathbf{P}_\Pi(\rho) \end{array}$$



(a) FPA $\mathcal{A}_1$ (b) SA $\mathcal{A}_2$

Figure 1. $\mathcal{A}_2$ is the restriction of $\mathcal{A}_1$ to a^*.

where $q \to \mu$ and $\text{fst}(\rho) = r$.

When Π is clear from the context, $\mathbf{P}_\Pi$ will simply be written $\mathbf{P}$. Since $\mathbf{P}_\Pi$ is a (sub-)probability on $CRun(\Pi)$, for any predicate $\varphi \subseteq CRun(\Pi)$, we have $\mathbf{P}(\varphi) = \sum_{\rho\in\varphi} \mathbf{P}(\rho)$. The measure is extended to languages $K \subseteq \mathcal{L}(\Pi)$ by $\mathbf{P}(K) = \mathbf{P}\left(\text{tr}^{-1}(K)\right) = \sum_{\text{tr}(\rho)\in K} \mathbf{P}(\rho)$.

In the examples of Figure 1, restricting the runs of $\mathcal{A}_1$ to those satisfying $\varphi = \{\rho \mid \text{tr}(\rho) \in a^*\}$ yields the SA $\mathcal{A}_2$, and $\mathbf{P}_{\mathcal{A}_1}(\varphi) = \mathbf{P}_{\mathcal{A}_2}(CRun(\mathcal{A}_2)) = \frac{1}{2}$.

A non probabilistic version of any SA is obtained by forgetting any information about probabilities.

Definition 3. *Let $\Pi = \langle \Sigma, Q, \Delta, q_0\rangle$ be an SA. The (non-deterministic) finite automaton $unProb(\Pi) = \langle \Sigma, Q, \Delta', q_0, F\rangle$ is defined by:*

- $\Delta' = \{(q, a, r) \in Q \times \Sigma \times Q \mid q \to \mu,\ \mu(a, r) > 0\}$,
- $F = \{q \in Q \mid q \to \mu, \mu(\surd) > 0\}$ *is the set of final states.*

It is easily seen that $\mathcal{L}(unProb(\Pi)) = \mathcal{L}(\Pi)$.

An observation function $\mathcal{O} : CRun(\Pi) \to Obs$ can also be easily translated from the probabilistic to the non probabilistic setting. For $\Pi' = unProb(\Pi)$, we define $unProb(\mathcal{O})$ on $Run(\Pi')$ by:
$unProb(\mathcal{O})(q_0 \xrightarrow{a_1} q_1 \cdots q_n) = \mathcal{O}(q_0 \xrightarrow{a_1} q_1 \cdots q_n \surd)$.

2.4 Computing the probability of a substochastic automaton

Given an SA Π, a system of equations can be derived on the probabilities for each state to yield an accepting run. This allows to compute the probability of all complete runs of Π by a technique similar to those used in [9, 12, 4] for probabilistic verification.

Definition 4 (Linear system of a substochastic automata)**.** *Let $\Pi = \langle \Sigma, Q, \Delta, q_0\rangle$ be a substochastic automaton. The* linear system associated with Π *is the following system $\mathcal{S}_\Pi$ of linear equations over $\mathbb{R}$:*

$$\mathcal{S}_\Pi = \left(X_q = \sum_{q'\in Q} \alpha_{q,q'} X_{q'} + \beta_q \right)_{q\in Q}$$

$$\textit{where} \quad \alpha_{q,q'} = \sum_{a\in\Sigma} \Delta(q)(a, q') \textit{ and } \beta_q = \Delta(q)(\surd)$$

When non-determinism is involved, for instance in Markov Decision Processes [9, 4], two systems of inequations are needed to compute maximal and minimal probabilities. Here, without non-determinism, both values are the same, hence the probability can be computed in polynomial time by solving the linear system associated with the SA.

Lemma 1. *Let $\Pi = \langle \Sigma, Q, \Delta, q_0\rangle$ be a substochastic automaton and define for all $q \in Q$, $L_q^\Pi = \mathbf{P}(CRun_q(\Pi))$. Then $(L_q^\Pi)_{q\in Q}$ is the unique solution of the system $\mathcal{S}_\Pi$.*

3 Relaxing opacity through probabilities

3.1 Definition and properties

One of the aspects in which the definition of opacity could be extended to probabilistic automata is by relaxing the universal quantifiers of Definition 1. Instead of wanting that *all* run satisfying φ have a similar (w.r.t. $\mathcal{O}$) run not in φ, we can just require that *almost all* of them do. To obtain this, we give a measure for the set of runs leaking information. To express properties of probabilistic opacity in an FPA Π, $\mathcal{O}$ is considered as a random variable. The characteristic function $\mathbf{1}_\varphi$ of φ is also considered as a random variable.

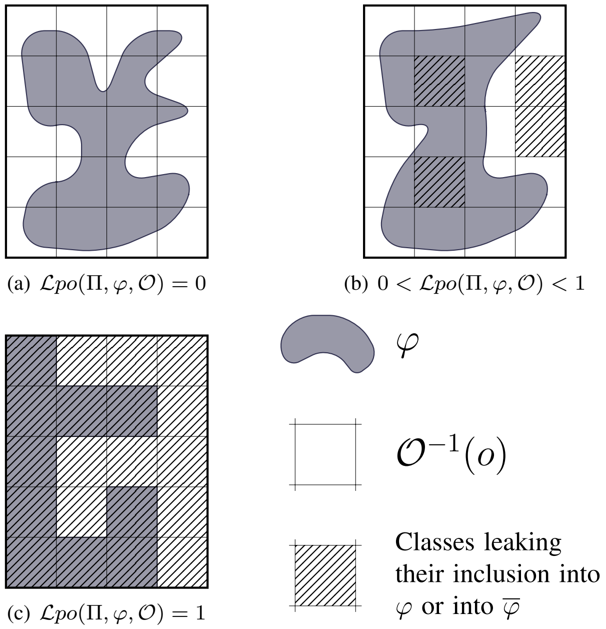
Definition 5 (Liberal probabilistic opacity)**.** *The* liberal probabilistic opacity *(LPO) of predicate φ on FPA Π, with respect to observation function $\mathcal{O}$ is defined by:*

$$\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = \sum_{\substack{o\in Obs\\ \mathcal{O}^{-1}(o)\subseteq\varphi}} \mathbf{P}(\mathcal{O} = o) + \sum_{\substack{o\in Obs\\ \mathcal{O}^{-1}(o)\subseteq\overline{\varphi}}} \mathbf{P}(\mathcal{O} = o)$$

This definition provides a measure of how insecure the system is.

The following proposition shows that a null value for this measure coincides with symmetrical opacity for the system, which is then secure. In this case, each equivalence class $\mathcal{O}^{-1}(o)$ overlaps both φ and $\overline{\varphi}$ as in Figure 2(a). On the other hand, the system is totally insecure when, observing through $\mathcal{O}$, we have all information about φ. In that case, the predicate φ is a union of equivalence classes $\mathcal{O}^{-1}(o)$ as in Figure 2(c) and this can be interpreted in terms of conditional entropy relatively to $\mathcal{O}$. The intermediate case occurs when some, but not all, observation classes contain only runs satisfying φ or only runs not satisfying φ, as in Figure 2(b).

Proposition 1.
(1) $0 \le \mathcal{L}po(\Pi, \varphi, \mathcal{O}) \le 1$
(2) $\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = 0$ *if and only if φ is symmetrically opaque on $unProb(\Pi)$ with respect to $unProb(\mathcal{O})$.*


φ
$\mathcal{O}^{-1}(o)$
Classes leaking their inclusion into φ or into $\overline{\varphi}$

(a) $\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = 0$

(b) $0 < \mathcal{L}po(\Pi, \varphi, \mathcal{O}) < 1$

(c) $\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = 1$

Figure 2. Liberal probabilistic opacity.

(3) $\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = 1$ *if and only if* $H(\mathbf{1}_\varphi|\mathcal{O}) = 0$.

Proof.

(1) The considered events are mutually exclusive, hence the sum of their probabilities never exceeds 1.

(2) First observe that a complete run $r_0 a \ldots r_n \surd$ has a non null probability in Π iff $r_0 a \ldots r_n$ is a run in $unProb(\Pi)$. Suppose $\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = 0$. Then there is no observable o with non-null probability such that $\mathcal{O}^{-1}(o) \subseteq \varphi$ (resp. $\overline{\varphi}$). Hence for each observable o, $\mathcal{O}^{-1}(o) \not\subseteq \varphi$ (resp. $\overline{\varphi}$). Conversely, if φ is opaque on $unProb(\Pi)$, there is no observable $c \in Obs$ such that $\mathcal{O}^{-1}(c) \subseteq \varphi$ (resp. $\overline{\varphi}$), hence the null value for $\mathcal{L}po(\Pi, \varphi, \mathcal{O})$.

(3) $H(\mathbf{1}_\varphi|\mathcal{O}) = 0$ iff

$$\sum_{\substack{o \in Obs \\ i \in \{0,1\}}} \mathbf{P}(\mathbf{1}_\varphi = i|\mathcal{O} = o) \cdot \log(\mathbf{P}(\mathbf{1}_\varphi = i|\mathcal{O} = o)) = 0$$

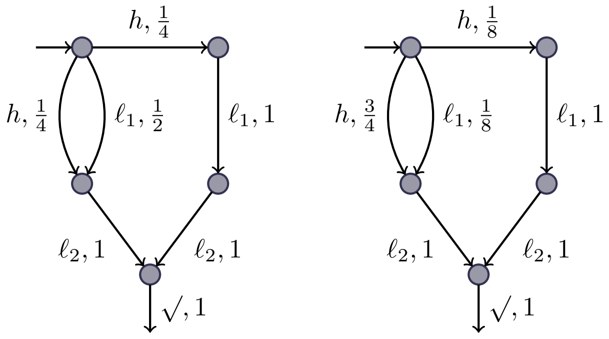
Since all the terms have the same sign, this sum is null if and only if each of its term is null. Setting for every $o \in Obs$, $f(o) = \mathbf{P}(\mathbf{1}_\varphi = 1|\mathcal{O} = o) = 1 - \mathbf{P}(\mathbf{1}_\varphi = 0|\mathcal{O} = o)$, we have: $H(\mathbf{1}_\varphi|\mathcal{O}) = 0$ iff $\forall\, o \in Obs$, $f(o) \cdot \log(f(o)) + (1 - f(o)) \cdot \log(1 - f(o)) = 0$. Since the equation $x \cdot \log(x) + (1 - x) \cdot \log(1 - x) = 0$ only accepts 1 and 0 as solutions, it means that for every observable o, either all the runs ρ such that $\mathcal{O}(\rho) = o$ are in φ, or they are all not in φ. Therefore $H(\mathbf{1}_\varphi|\mathcal{O}) = 0$ iff for every observable o, $\mathcal{O}^{-1}(o) \subseteq \varphi$ or $\mathcal{O}^{-1}(o) \subseteq \overline{\varphi}$, which is equivalent to $\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = 1$. □

Example. Consider the systems $\mathcal{A}_3$ and $\mathcal{A}_4$ of Figure 3. On these systems we define the predicate φ_{NI} which is true if the trace of a run contains letter h. In both cases the observation function $\mathcal{O}_L$ returns the projection of the trace onto the alphabet $\{\ell_1, \ell_2\}$. Remark that this example is an interference property [11] seen as opacity. Considered unprobabilistically, both systems are interferent since an ℓ_2 not preceded by an ℓ_1 betrays the presence of an h. However, they differ by how often this case happens.

The runs of $\mathcal{A}_3$ and $\mathcal{A}_4$ and their properties are displayed in Table 1. Then we can see that $[\rho_1]_{\mathcal{O}_L} = [\rho_2]_{\mathcal{O}_L}$ overlaps both φ_{NI} and $\overline{\varphi_{NI}}$, while $[\rho_3]_{\mathcal{O}_L}$ is contained totally in φ. Hence the LPO can be computed for both systems:

$$\mathcal{L}po(\mathcal{A}_3, \varphi_{NI}, \mathcal{O}_L) = \frac{1}{4} \qquad \mathcal{L}po(\mathcal{A}_4, \varphi_{NI}, \mathcal{O}_L) = \frac{3}{4}$$

Therefore $\mathcal{A}_3$ is more secure than $\mathcal{A}_4$. Indeed, the run that is interferent occurs more often in $\mathcal{A}_4$, leaking information more often.



(a) FPA $\mathcal{A}_3$

(b) FPA $\mathcal{A}_4$

Figure 3. Interferent FPAs $\mathcal{A}_3$ and $\mathcal{A}_4$.

tr(ρ)	$\mathbf{P}_{\mathcal{A}_3}(\rho)$	$\mathbf{P}_{\mathcal{A}_4}(\rho)$	$\in \varphi_{NI}$?	$\mathcal{O}_L(\rho)$
tr(ρ_1) $= \ell_1\ell_2\surd$	1/2	1/8	0	$\ell_1\ell_2$
tr(ρ_2) $= h\ell_1\ell_2\surd$	1/4	1/8	1	$\ell_1\ell_2$
tr(ρ_3) $= h\ell_2\surd$	1/4	3/4	1	ℓ_2

Table 1. Runs of $\mathcal{A}_3$ and $\mathcal{A}_4$.

3.2 Computation of LPO

We now show how LPO can be computed for regular predicates and simple observation functions. The method relies on a synchronized product between an SA Π and a deterministic FA $\mathcal{K}$, similarly to [9]. This product (which can be considered pruned of its unreachable states and states not reaching a final state) constrains the unprobabilistic version of Π by synchronizing it with $\mathcal{K}$. The probability of $\mathcal{L}(\mathcal{K})$

is then obtained by solving the associated system of equations. The computation of LPO results in applications of this operation with several FPAs.

Definition 6 (Synchronized product)**.**
Let $\Pi = \langle \Sigma, Q, \Delta, q_0 \rangle$ *be a substochastic automaton and let* $\mathcal{K} = \langle Q \times \Sigma \times Q, Q_K, \Delta_K, q_K, F \rangle$ *be a deterministic finite automaton. The synchronized product* $\Pi || \mathcal{K}$ *is the substochastic automaton* $\langle \Sigma, Q \times Q_K, \Delta', (q_0, q_K) \rangle$ *where transitions in* Δ' *are defined by: if* $q_1 \to \mu \in \Delta$*, then* $(q_1, r_1) \to \nu \in \Delta'$ *where for all* $a \in \Sigma$ *and* $(q_2, r_2) \in Q \times Q_K$*,*

$$\nu(a, (q_2, r_2)) = \begin{cases} \mu(a, q_2) & \textit{if } r_1 \xrightarrow{q_1, a, q_2} r_2 \in \Delta_K \\ 0 & \textit{otherwise} \end{cases}$$

$$\textit{and} \quad \nu(\surd) = \begin{cases} \mu(\surd) & \textit{if } r_1 \in F \\ 0 & \textit{otherwise} \end{cases}$$

Lemma 2. *Let* $\Pi = \langle \Sigma, Q, \Delta, q_0 \rangle$ *be an SA and* K *a regular language over* $Q \times \Sigma \times Q$ *accepted by a deterministic finite automaton* $\mathcal{K} = \langle Q \times \Sigma \times Q, Q_K, \Delta_K, q_K, F \rangle$*. Then*

$$\mathbf{P}_\Pi(K) = L^{\Pi || \mathcal{K}}_{(q_0, q_K)}$$

Proposition 2. *Let* φ *be a regular predicate in an FPA* Π*, i.e. such that* φ *is a regular subset of* $(Q \times \Sigma \times Q)^*$*, and let* $\mathcal{O}$ *be an observation function satisfying:*

- *the set of observables Obs is finite*
- *for each* $o \in Obs$*, the subset* $\mathcal{O}^{-1}(o)$ *of* $CRun(\Pi)$ *is a regular set.*

Then $\mathcal{L}po(\Pi, \varphi, \mathcal{O})$ *can be computed.*

Proof. The computation of $\mathcal{L}po(\Pi, \varphi, \mathcal{O})$ proceeds with the following steps:

- From the conditions on $\mathcal{O}$, a deterministic finite automaton $\mathcal{A}_o$ accepting $\mathcal{O}^{-1}(o)$ can be associated with each $o \in Obs$. Synchronizing this automaton with Π and pruning it yields a substochastic automaton $\Pi || \mathcal{A}_o$. By Lemma 2, the probability $\mathbf{P}(\mathcal{O} = o)$ is then computed by solving the linear system associated with $\Pi || \mathcal{A}_o$.
- On the other hand, since φ is regular, there is a deterministic finite automaton $\mathcal{A}_\varphi$ (respectively $\mathcal{A}_{\overline{\varphi}}$) accepting the runs which satisfy φ (respectively $\mathcal{A}_{\overline{\varphi}}$). By testing language emptiness for $unProb(\Pi || \mathcal{A}_o || \mathcal{A}_\varphi)$ (resp. $unProb(\Pi || \mathcal{A}_o || \mathcal{A}_{\overline{\varphi}})$), it can be decided if $\mathcal{O}^{-1}(o) \subseteq \overline{\varphi}$ (resp. $\mathcal{O}^{-1}(o) \subseteq \varphi$). The value of $\mathcal{L}po(\Pi, \varphi, \mathcal{O})$ is then obtained by adding probabilities $\mathbf{P}(\mathcal{O} = o)$ when $\mathcal{O}^{-1}(o)$ is contained in φ or its complement. □

4 Tightening opacity through information theory

4.1 Definition and properties

The completely opposite direction that can be taken to define a probabilistic version is a more paranoid one: how much information is leaked through the system's uncertainty? For example, on Figure 2(a), even though each observation class contains a run in φ and one in $\overline{\varphi}$, some classes are *nearly* in φ. In some other classes the balance between the runs satisfying φ and the ones not satisfying φ is more even. We would like to measure globally the balance between φ and $\overline{\varphi}$ in each observation class. Hence, for a run $\rho \in \varphi$, we will not ask if *there exists* a similar run not in φ, but *how many* there are, with a probabilistic measure taking into account the likelihood of such runs. We adopt an information-theoretic view: how much information is transmitted from the predicate φ to the observation function $\mathcal{O}$?

Definition 7 (Restrictive probabilistic opacity)**.** *Let* φ *be a predicate on the complete runs of an FPA* Π *and* $\mathcal{O}$ *an observation function. The* restrictive probabilistic opacity *(RPO) of* φ *on* Π*, with respect to* $\mathcal{O}$*, is defined by*

$$\mathcal{R}po(\Pi, \varphi, \mathcal{O}) = 1 - I(\mathbf{1}_\varphi; \mathcal{O})$$

Proposition 3.

(1) $0 \leq \mathcal{R}po(\Pi, \varphi, \mathcal{O}) \leq 1$

(2) If $\mathcal{R}po(\Pi, \varphi, \mathcal{O}) = 0$*, then* φ *is not opaque on* $unProb(\Pi)$ *with respect to* $unProb(\mathcal{O})$*.*

Proof.

(1) Since $\mathbf{1}_\varphi$ can take only two different values and entropy decreases with knowledge, $0 \leq H(\mathbf{1}_\varphi | \mathcal{O}) \leq H(\mathbf{1}_\varphi) \leq log(2) = 1$.

(2) This case is reached only when $H(\mathbf{1}_\varphi) = 1$ and $H(\mathbf{1}_\varphi | \mathcal{O}) = 0$. When $H(\mathbf{1}_\varphi | \mathcal{O}) = 0$, by Proposition 1 case (3), $\mathcal{L}po(\Pi, \varphi, \mathcal{O}) = 1 > 0$, then φ is not opaque on $unProb(\Pi)$ with respect to $unProb(\mathcal{O})$.□

4.2 Computation of RPO

Computing RPO can be done in a similar way with the same hypotheses as for LPO:

Proposition 4. *Let* φ *be a regular predicate for an FPA* Π *and let* $\mathcal{O}$ *be an observation function with a finite number of observation classes which are all regular sets of runs. Then* $\mathcal{R}po(\Pi, \varphi, \mathcal{O})$ *can be computed.*

Proof. Again we consider a finite deterministic automaton $\mathcal{A}_\varphi$ (resp. $\mathcal{A}_{\overline{\varphi}}$) accepting runs in φ (resp in $\overline{\varphi}$) and, for each $o \in Obs$, a finite deterministic automaton $\mathcal{A}_o$ accepting $\mathcal{O}^{-1}(o)$. We successively compute:

- $\mathbf{P}(\varphi) = \mathbf{P}(\mathbf{1}_\varphi = 1)$ (respectively $\mathbf{P}(\overline{\varphi}) = \mathbf{P}(\mathbf{1}_\varphi = 0)$) by synchronizing Π with $\mathcal{A}_\varphi$ (respectively $\mathcal{A}_{\overline{\varphi}}$), pruning it, and computing the solution of the associated linear system;
- the probability $\mathbf{P}(\mathcal{O} = o)$, for each $o \in \mathit{Obs}$, with similar techniques but with $\mathcal{A}_o$, and the probabilities $\mathbf{P}(\varphi \cap [\mathcal{O} = o]) = \mathbf{P}(\mathbf{1}_\varphi = 1, \mathcal{O} = o)$ (respectively $\mathbf{P}(\overline{\varphi} \cap [\mathcal{O} = o]) = \mathbf{P}(\mathbf{1}_\varphi = 0, \mathcal{O} = o)$), using synchronization of Π, $\mathcal{A}_\varphi$ (respectively $\mathcal{A}_{\overline{\varphi}}$), and $\mathcal{A}_o$;
- the conditional probabilities $\mathbf{P}(\mathbf{1}_\varphi = 1|\mathcal{O} = o) = \frac{\mathbf{P}(\mathbf{1}_\varphi=1,\mathcal{O}=o)}{\mathbf{P}(\mathcal{O}=o)}$ and $\mathbf{P}(\mathbf{1}_\varphi = 0|\mathcal{O} = o) = \frac{\mathbf{P}(\mathbf{1}_\varphi=0,\mathcal{O}=o)}{\mathbf{P}(\mathcal{O}=o)}$; entropies $H(\mathbf{1}_\varphi)$ and $H(\mathbf{1}_\varphi|\mathcal{O})$ using probabilities computed above and finally, mutual information between $\mathbf{1}_\varphi$ and $\mathcal{O}$ and $\mathcal{R}po(\Pi, \varphi, \mathcal{O})$. □

4.3 Application and examples

4.3.1 The Dining Cryptographers

Introduced in [8], this problem involves three cryptographers C_1, C_2 and C_3 dining in a restaurant. At the end of the meal, their master secretly tells each of them if they should be paying: $p_i = 1$ iff cryptographer C_i pays, and $p_i = 0$ otherwise. Wanting to know if one of the cryptographers paid or if the master did, they follow the following protocol. They flip a coin with each of their neighbor, the third one not seeing the result of the flip, marking $f_{i,j} = 0$ if the coin flip between i and j was heads and $f_{i,j} = 1$ if it was tails. Then each cryptographer C_i, for $i \in \{1, 2, 3\}$, announces the value of $r_i = f_{i,i+1} \oplus f_{i,i-1} \oplus p_i$ (where '$3+1 = 1$', '$1-1 = 3$' and '$\oplus$' represents the XOR operator). If $\bigoplus_{i=1}^{3} r_i = 0$ then no one (*i.e.* the master) paid, if $\bigoplus_{i=1}^{3} r_i = 1$, then one of the cryptographers paid, but the other two do not know who he is.

Here we will use a simplified version of this problem to limit the size of the model. We consider that some cryptographer paid for the meal, and adopt the point of view of C_1 who did not pay. The anonymity of the payer is preserved if C_1 cannot know if C_2 or C_3 paid for the meal. In our setting, the predicate φ_2 is, without loss of symmetry, "C_2 paid". The observation function lets C_1 know the results of its coin flips ($f_{1,2}$ and $f_{1,3}$), and the results announced by the other cryptographers (r_2 and r_3). We also assume that the coin used by C_2 and C_3 has a probability of q to yield heads, and that the master flips a fair coin to decide if C_2 or C_3 pays. It can be assumed that the coins C_1 flips with its neighbors are fair, since it does not affect anonymity from C_1's point of view. In order to limit the (irrelevant) interleaving, we have made the choice to fix the ordering between the coin flips.

The corresponding FPA $\mathcal{D}$ is depicted on Figure 4 where all $\surd$ transitions with probability 1 have been omitted from final (rectangular) states. On $\mathcal{D}$, the runs satisfying predicate φ_2 are the ones where action p_2 appears. The observation function $\mathcal{O}_1$ takes a run and returns the sequence of actions over the alphabet $\{h_{1,2}, t_{1,2}, h_{1,3}, t_{1,3}\}$ and the final state reached, containing the value announced by C_2 and C_3.

There are 16 possible complete runs in this system, that yield 8 equiprobable observables:

$$\begin{aligned}\mathit{Obs} = \{&(h_{1,2}h_{1,3}(r_2 = 1, r_3 = 0)),\\ &(h_{1,2}h_{1,3}(r_2 = 0, r_3 = 1)),\\ &(h_{1,2}t_{1,3}(r_2 = 0, r_3 = 0)),\\ &(h_{1,2}t_{1,3}(r_2 = 1, r_3 = 1)),\\ &(t_{1,2}h_{1,3}(r_2 = 0, r_3 = 0)),\\ &(t_{1,2}h_{1,3}(r_2 = 1, r_3 = 1)),\\ &(t_{1,2}t_{1,3}(r_2 = 1, r_3 = 0)),\\ &(t_{1,2}t_{1,3}(r_2 = 0, r_3 = 1))\,\}\end{aligned}$$

Moreover, each observation results in a run in which C_2 pays and a run in which C_3 pays, this difference being masked by the secret coin flip between them. For example, runs $\rho_h = h_{1,2}h_{1,3}h_{2,3}p_2(r_2 = 1, r_3 = 0)$ and $\rho_t = h_{1,2}h_{1,3}t_{2,3}p_3(r_2 = 1, r_3 = 0)$ yield the same observable $o_0 = h_{1,2}h_{1,3}(r_2 = 1, r_3 = 0)$, but the predicate is true in the first case and false in the second one. Therefore, if $0 < q < 1$, the unprobabilistic version of $\mathcal{D}$ is opaque. However, if $q \neq \frac{1}{2}$, for each observable, one of them is *more likely* to be lying, therefore paying. In the aforementioned example, when observing o_0, ρ_h has occurred with probability q, whereas ρ_t has occurred with probability $1 - q$. RPO can measure this advantage globally.

For the next RPO computation, we write $\mathbf{1}_\varphi$ instead of $\mathbf{1}_{\varphi_2}$ and $\mathcal{O}$ instead of $\mathcal{O}_1$.

$$\begin{aligned} I(\mathbf{1}_\varphi; \mathcal{O}) &= H(\mathbf{1}_\varphi) - H(\mathbf{1}_\varphi|\mathcal{O}) \\ &= 1 + \mathbf{Q} \end{aligned}$$

where

$$\mathbf{Q} = \sum_{\substack{o \in Obs \\ i \in \{0,1\}}} \mathbf{P}(\mathcal{O} = o) \cdot \mathbf{P}(\mathbf{1}_\varphi = i|\mathcal{O} = o) \cdot \log(\mathbf{P}(\mathbf{1}_\varphi = i|\mathcal{O} = o))$$

For each observable o, $\mathbf{P}(\mathbf{1}_\varphi = 1|\mathcal{O} = o) = 1 - \mathbf{P}(\mathbf{1}_\varphi = 0|\mathcal{O} = o)$. In addition, $\mathbf{P}(\mathbf{1}_\varphi = 1|\mathcal{O} = o)$ is either q or $1 - q$. This allows to compute the RPO, parametrized by q:

$$\mathcal{R}po(\mathcal{D}, \varphi_2, \mathcal{O}_1) = -(q \cdot \log(q) + (1 - q) \cdot \log(1 - q))$$

On this expression we can see that $\mathcal{R}po(\mathcal{D}, \varphi_2, \mathcal{O}_1) = 1$ if $q = \frac{1}{2}$, and $\mathcal{R}po(\mathcal{D}, \varphi_2, \mathcal{O}_1) = 0$ if $q = 0$ or $q = 1$. The variations of the RPO when changing the bias on q are depicted in Figure 5.

4.3.2 Crowds protocol

The anonymity protocol known as *crowds* was introduced in [15] and recently studied in the probabilistic framework

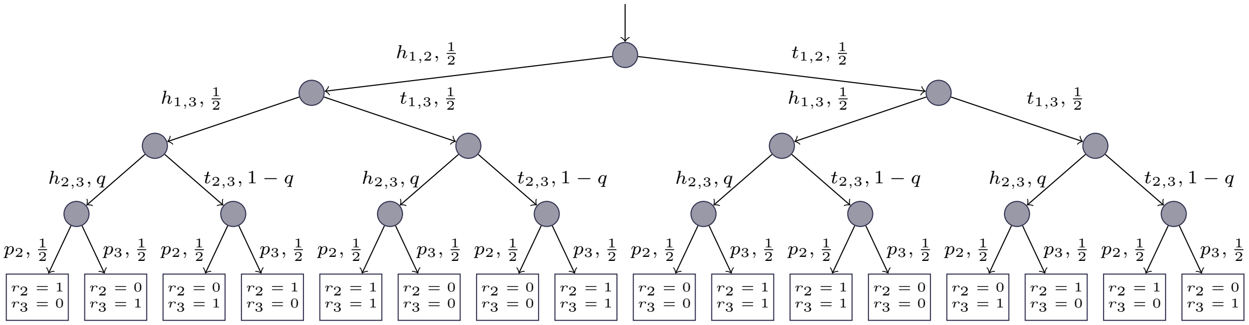


Figure 4. The FPA corresponding to the Dining Cryptographers protocol.

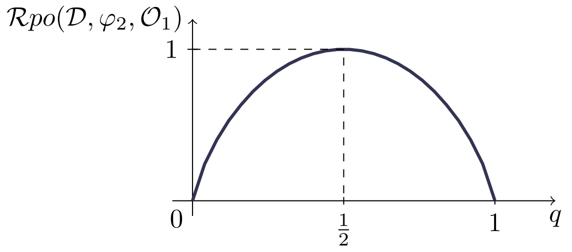


Figure 5. Evolution of the restrictive probabilistic opacity of the Dining Cryptographers protocol when changing the bias on the coin.

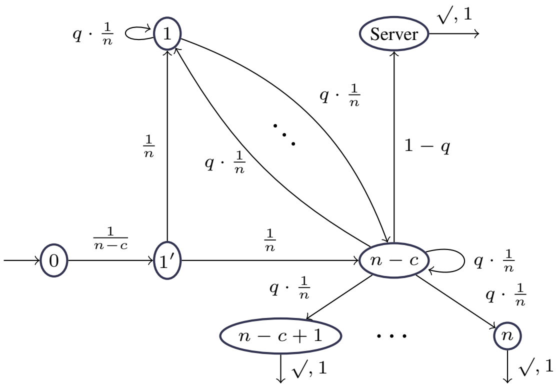


Figure 7. SA $\mathcal{C}_n^c||\mathcal{A}_{1\rightsquigarrow(n-c)}$ corresponding to runs where user 1 initiates the protocol and user $(n-c)$ is detected

in [7, 3]. When a user wants to send a message (or request) to a server without the latter knowing the origin of the message, the user routes the message through a crowd of n users. To do so, it selects a user randomly in the crowd (including himself), and sends him the message. When a user receives a message to be routed according to this protocol, it either sends the message to the server with probability $1-q$ or forwards it to a user in the crowd, with probability q. The choice of a user in the crowd is always equiprobable. Under these assumptions, this protocol is known to be secure (indeed, its RPO is 1). However, there can be c corrupt users in the crowd which divulge the identity of the person that sent the message to them. In that case, if a user sends directly a message to a corrupt user, its identity is no longer protected. RPO can measure the security of this system, depending on n and c.

First, consider our protocol as the system in Figure 6. The predicate we want to be opaque is φ_i that contains all the runs in which i is the initiator of the request. The observation function $\mathcal{O}$ returns the penultimate state of the run, *i.e.* the honest user that will be seen by the server or a corrupt user.

For sake of brevity, we write '$i \rightsquigarrow$' to denote the event "a request was initiated by i" and '$\rightsquigarrow j$' when "j was detected by the adversary" $i \rightsquigarrow \wedge \rightsquigarrow j$ is abbreviated in $i \rightsquigarrow j$. Notation '$\neg i \rightsquigarrow$' means that "a request was initiated by someone else than i"; similarly, combinations of this notations are used in the sequel. We also use the Kronecker symbol δ_{ij} defined by $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

Computing probabilities. All probabilities $\mathbf{P}(i \rightsquigarrow j)$ can be automatically computed using the method described in Section 4.2. For example, $\mathbf{P}(1 \rightsquigarrow (n-c))$, the probability for the first user to initiate the protocol while the last honest user is detected, can be computed from substochastic automaton $\mathcal{C}_n^c||\mathcal{A}_{1\rightsquigarrow(n-c)}$ depicted on Figure 7. The associated system is represented in Table 2 where L_S corresponds to the "Server" state. Resolving it yields, $L_i = \frac{q}{n}$ for all $i \in \{1, \dots, n-c-1\}$, $L_{n-c} = 1 - \frac{q\cdot(n-c-1)}{n}$, $L_{1'} = \frac{1}{n}$, and $L_0 = \frac{1}{(n-c)\cdot n}$. Therefore, $\mathbf{P}(1 \rightsquigarrow (n-c)) = \frac{1}{(n-c)\cdot n}$.

In this case, simple reasoning on the symmetries of the model allows to derive other probabilities $\mathbf{P}(i \rightsquigarrow j)$. Remark that the probability for a message to go directly from initiator to a corrupt user or the server is $\frac{c}{n}$: it only happens if a corrupt user is chosen by the initiator. If a honest user is chosen by the initiator, then the length will be greater, with probability $\frac{n-c}{n}$. By symmetry all honest users have equal probability to be the initiator, and equal probability to be

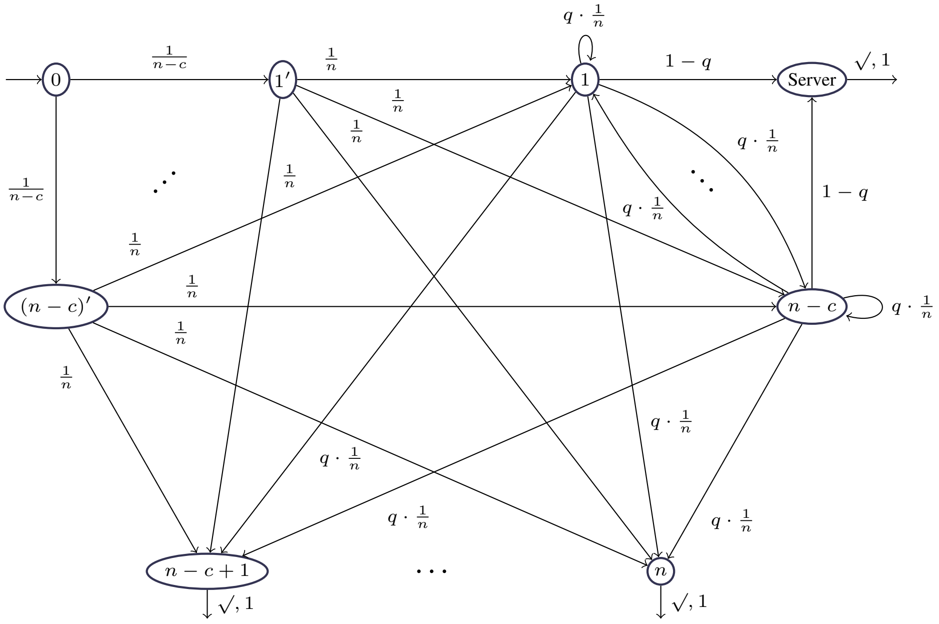


Figure 6. FPA $\mathcal{C}_n^c$ for Crowds protocol with n users, among whom c are corrupted.

$$\begin{cases} L_0 &= \frac{1}{n-c} \cdot L_{1'} \\ L_{1'} &= \sum_{i=1}^{n-c} \frac{1}{n} \cdot L_i \\ L_1 &= \sum_{i=1}^{n-c} \frac{q}{n} \cdot L_i \\ &\vdots \\ L_{n-c-1} &= \sum_{i=1}^{n-c} \frac{q}{n} \cdot L_i \\ L_{n-c} &= (1-q) \cdot L_S + \sum_{i=1}^{n} \frac{q}{n} \cdot L_i \\ L_{n-c+1} &= 1 \\ &\vdots \\ L_n &= 1 \\ L_S &= 1 \end{cases}$$

Table 2. Linear system associated to SA $\mathcal{C}_n^c || \mathcal{A}_{1 \rightsquigarrow (n-c)}$ of Figure 7

detected. Hence $\mathbf{P}(i \rightsquigarrow) = \mathbf{P}(\rightsquigarrow j) = \frac{1}{n-c}$.

Event $i \rightsquigarrow j$ occurs when i is chosen as the initiator (probability $\frac{1}{n-c}$), and either (1) if $i = j$ and i chooses a corrupted user to route its message, or (2) if a honest user is chosen and j sends the message to a corrupted user or the server (the internal route between honest users before j is irrelevant). Therefore

$$\begin{aligned} \mathbf{P}(i \rightsquigarrow j) &= \frac{1}{n-c} \cdot \left(\delta_{ij} \cdot \frac{c}{n} + \frac{1}{n-c} \cdot \frac{n-c}{n} \right) \\ \mathbf{P}(i \rightsquigarrow j) &= \frac{1}{n-c} \cdot \left(\delta_{ij} \cdot \frac{c}{n} + \frac{1}{n} \right) \end{aligned}$$

The case when i is not the initiator is derived from this probability:

$$\begin{aligned} \mathbf{P}(\neg i \rightsquigarrow j) &= \sum_{\substack{k=1 \\ k \neq i}}^{n-c} \mathbf{P}(k \rightsquigarrow j) \\ \mathbf{P}(\neg i \rightsquigarrow j) &= \frac{1}{n-c} \cdot \left((1 - \delta_{ij}) \cdot \frac{c}{n} + \frac{n-c-1}{n} \right) \end{aligned}$$

Conditional probabilities thus follow:

$$\mathbf{P}(i \rightsquigarrow | \rightsquigarrow j) = \frac{\mathbf{P}(i \rightsquigarrow j)}{\mathbf{P}(\rightsquigarrow j)} = \delta_{ij} \cdot \frac{c}{n} + \frac{1}{n}$$

$$\mathbf{P}(\neg i \rightsquigarrow | \rightsquigarrow j) = \frac{\mathbf{P}(\neg i \rightsquigarrow j)}{\mathbf{P}(\rightsquigarrow j)} = (1 - \delta_{ij}) \cdot \frac{c}{n} + \frac{n-c-1}{n}$$

Interestingly, these probabilities do not depend on q.

Computing RPO. We finally compute RPO (tedious calculi being omitted due to space constraints), denoting by $\mathbf{1}_i$ the random variable $\mathbf{1}_{\varphi_i}$ and by $\mathcal{O}$ the observation function of the penultimate state of the run:

$$\begin{aligned} -H(\mathbf{1}_i | \mathcal{O}) &= \sum_{j=1}^{n-c} \begin{pmatrix} \mathbf{P}(i \rightsquigarrow j) \cdot \log(\mathbf{P}(i \rightsquigarrow | \rightsquigarrow j)) \\ + \mathbf{P}(\neg i \rightsquigarrow j) \cdot \log(\mathbf{P}(\neg i \rightsquigarrow | \rightsquigarrow j)) \end{pmatrix} \\ &= \frac{1}{n-c} \cdot \begin{pmatrix} \frac{(n-c-1)\cdot(n-1)}{n} \cdot \log(n-1) \\ + \frac{n-c-1}{n} \cdot \log(n-c-1) \\ + \frac{c+1}{n} \cdot \log(c+1) \end{pmatrix} - \log(n) \end{aligned}$$

On the other hand

$$\begin{aligned} H(\mathbf{1}_i) &= \mathbf{P}(i \rightsquigarrow) \cdot \log(\mathbf{P}(i \rightsquigarrow)) \\ &\quad + \mathbf{P}(\neg i \rightsquigarrow) \cdot \log(\mathbf{P}(\neg i \rightsquigarrow)) \\ &= \log(n-c) - \frac{n-c-1}{n-c} \cdot \log(n-c-1) \end{aligned}$$

Hence

$$\begin{aligned}\mathcal{R}po(\mathcal{C}_n^c, \varphi_i, \mathcal{O}) &= 1 + \log(n) - \log(n-c)\\ &\quad + \frac{n-c-1}{n-c} \cdot \log(n-c-1)\\ &\quad - \frac{1}{n-c} \cdot \left(\begin{array}{l} \frac{(n-c-1)\cdot(n-1)}{n} \cdot \log(n-1) \\ + \frac{n-c-1}{n} \cdot \log(n-c-1) \\ + \frac{c+1}{n} \cdot \log(c+1) \end{array} \right)\end{aligned}$$

Remark that in the case where there is no corrupt user (*i.e.* when $c = 0$), we obtain $\mathcal{R}po(\mathcal{C}_n^0, \varphi_i, \mathcal{O}) = 1$, thus re-confirming in passing the result from [15] stating that the crowds protocol is secure. It can also be noted that, as expected, more corrupt users decrease the security, while more honest users increase it.

5 Discussion and conclusion

In this paper we introduced two dual notions of probabilistic opacity. The liberal one measures the probability for an attacker observing a random execution of the system to be able to gain information he can be sure about. The restrictive one measures the level of certitude in the information acquired by an attacker observing the system. The extremal cases of both these notions coincide with the possibilistic notion of opacity, which evaluates the existence of a leak of sure information.

The definition of these notions through probability and information theory allows to inherit from all the results in these fields when necessary. However, probabilistic opacity is not always easy to compute, especially if there are an infinite number of observables. Nevertheless, automatic computation is possible when dealing with regular predicates and finitely many regular observation classes.

In future work we plan to investigate other measures along the line of [16] and explore more of the properties of probabilistic opacity, to instantiate it to known security measures (anonymity, non-interference, etc.). Furthermore, we want to address the more general case of probabilistic automata in which the non-determinism has not been resolved.

Acknowledgments. We wish to thank the anonymous reviewers for their insightful comments and suggestions.

References

[1] A. Aldini, M. Bravetti, and R. Gorrieri. A process-algebraic approach for the analysis of probabilistic noninterference. *Journal of Computer Security*, 12(2):191–245, 2004.

[2] R. Alur, P. Černý, and S. Zdancewic. Preserving secrecy under refinement. In *Proc. of the 33rd Intl. Colloquium on Automata, Languages and Programming (ICALP'06)*, volume 4052 of *LNCS*, pages 107–118. Springer, 2006.

[3] M. E. Andrés, C. Palamidessi, P. van Rossum, and G. Smith. Computing the leakage of information-hiding systems. In *Proc. 16th Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'10)*, volume 6015 of *LNCS*, pages 373–389. Springer, March 2010.

[4] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. 15th Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'95)*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[5] M. Boreale. Quantifying information leakage in process calculi. *Information and Computation*, 207(6):699–725, 2009.

[6] J. W. Bryans, M. Koutny, L. Mazaré, and P. Y. A. Ryan. Opacity generalised to transition systems. *Intl. Jour. of Information Security*, 7(6):421–435, 2008.

[7] K. Chatzikokolakis, C. Palamidessi, and P. Panangaden. Anonymity protocols as noisy channels. *Information and Computation*, 206(2-4):378–401, Feb. 2008.

[8] D. Chaum. The dining cryptographers problem: unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[9] C. Courcoubetis and M. Yannakakis. Markov Decision Processes and Regular Events. *IEEE Transactions on Automatc Control*, 43(10):1399–1418, 1998.

[10] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, July 2006.

[11] J. A. Goguen and J. Meseguer. Security policy and security models. In *Proc. of IEEE Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, 1982.

[12] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[13] Y. Lakhnech and L. Mazaré. Probabilistic opacity for a passive adversary and its application to Chaum's voting scheme. Technical Report 4, Verimag, 2 2005.

[14] L. Mazaré. Decidability of opacity with non-atomic keys. In *Proc. 2nd Workshop on Formal Aspects in Security and Trust (FAST'04)*, volume 173 of *Intl. Federation for Information Processing*, pages 71–84. Springer, 2005.

[15] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[16] G. Smith. On the foundations of quantitative information flow. In *Proc. 12th Intl. Conf. on Foundations of Software Science and Computational Structures (FOSSACS '09)*, pages 288–302. Springer-Verlag, 2009.