Research Report on Quantifying Opacity

Béatrice Bérard[1⋆], John Mullins[2⋆⋆], and Mathieu Sassolas[1⋆⋆⋆]

[1] Université Pierre & Marie Curie, LIP6/MoVe, CNRS UMR 7606, Paris, France
[2] École Polytechnique de Montréal, Dept. of Comp. & Soft. Eng., Montreal (Quebec), Canada

Abstract. Opacity is a general language-theoretic framework in which several security properties of a system can be expressed. Its parameters are a predicate, given as a subset of runs of the system, and an observation function, from the set of runs into a set of observables. The predicate describes secret information in the system and, in the possibilistic setting, it is opaque if its membership cannot be inferred from observation.
In this paper, we investigate several notions of quantitative opacity for probabilistic systems, where the predicate and the observation function are seen as random variables. Our aim is to measure (i) the probability of opacity leakage relative to these random variables and (ii) the level of uncertainty about membership of the predicate inferred from observation. We show how these measures extend possibilistic opacity, we give algorithms to compute them for regular secrets and observations, and we apply these computations on several classical examples.

1 Introduction

1.1 Motivations

Opacity [1] 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 a passive attacker.

The first aspect regards the quantification of security properties. If executions leaking information are negligible with respect to the rest of executions,

⋆ Author partially supported by project DOTS (ANR-06-SETI-003) (French Government).
⋆⋆ Author partially supported by the NSERC of Canada under discovery grant No. 13321-2010.
⋆⋆⋆ Author partially supported by project CoChaT (DIGITEO-2009-27HD) (Région Île de France).

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 [2,3] 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 [4,5], it becomes important to extend specification frameworks in order to cope with it.

1.2 Contributions

In this paper we investigate several ways of extending opacity to a purely probabilistic framework. Opacity can be defined either as the capacity for an external observer to deduce that a predicate was true (asymmetrical opacity) or whether a predicate is true or false (symmetrical opacity). Both notions can model relevant security properties, hence deserve to be extended. On the other hand, two directions can be taken towards the quantification of opacity. The first one evaluates the degree of non-opacity of a system: how big is the security hole? It aims at assessing the probability for the system to yield *perfect* information. The second direction evaluates how opaque the system is: how robust is the security? The goal here is to measure how reliable is the information gained through observation. This yields up to four notions of quantitative opacity, displayed in Table 1, which are formally defined in this paper. In addition, an information-

	Asymmetric	Symmetric
Liberal (Security hole)	LPO (PO^A_ℓ)	LPSO (PO^S_ℓ)
Restrictive (Robustness)	RPO (PO^A_r)	RPSO (PO^S_r)

Table 1. The four probabilistic opacity measures

theoretic point of view allows to assess how much information is gained through observation, thus defining another measure of opacity.

Moreover, as opacity itself, all these measures can be instantiated into several probabilistic security properties such as probabilistic non-interference and anonymity. 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 [5].

Although the measures are defined in systems without nondeterminism, they can be extended to the case of systems scheduled by an adversary. Nevertheless, this case is so far computationally intractable.

1.3 Related Work

The notion of opacity was introduced recently with the aim to provide a uniform description for possibilistic security properties like non-interference and anonymity [2]. 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.

The authors of [6] 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 [7], 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 [8], 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.

In [9], the authors present a probabilistic version of anonymity, also using the tools of information theory. This approach was completed in [10] were anonymity is computed using regular expressions. More recently, in [11], the authors consider Interactive Information Hiding Systems that can be viewed as channels with feedback, where the input is the secret and the output is an observation.

In [12], 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 [12] lacks clear ties with the possibilistic notion of opacity.

1.4 Organization of the paper.

In Section 2, we recall the definitions of opacity and the probabilistic framework used throughout the paper. Section 3 and 4 present respectively the liberal and the restrictive version of probabilistic opacity, both for the asymmetrical and symmetrical case. In Section 5, we investigate measures of symmetrical opacity from the point of view of information theory. We present in Section 6 how to compute these measures automatically if the predicate and observations are regular. Section 7 compares the different measures and what they allow to detect about the security of the system, through abstract examples and a case study of the Crowds protocol. In Section 8, we present the framework of probabilistic systems dealing with nondeterminism, and open problems that arise in this setting.

2 Preliminaries

In this section, we recall the notions of opacity, entropy, and probabilistic automata.

2.1 Possibilistic opacity

The original definition of opacity was given in [2] 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.

Definition 1 (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* opaque *on Π for $\mathcal{O}$ if, for any $o \in Obs$, the following holds:*

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

However, detecting whether an event *did not* occur can give 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 [3]: 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 2 (Symmetrical opacity). *A predicate $\varphi \subseteq Run(\Pi)$ is* symmetrically opaque *on system Π for observation function $\mathcal{O}$ if, for any $o \in Obs$, the following holds:*

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

The symmetrical opacity is a stronger security requirement. Security goals can be expressed as either symmetrical or asymmetrical opacity, depending on the property at stake. Usually, when the predicate breaks the symmetry of a model, the asymmetric definition is more suited. Nonetheless, a noisy channel

with binary input can be seen as a system Π on which the input is the truth value of φ and the output is the observation $o \in Obs$. If φ is symmetrically opaque on Π with respect to $\mathcal{O}$, then this channel is not perfect: there would always be a possibility of noise.

2.2 Probabilities and information theory

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 $Z : \Omega \to \Gamma$ where $[Z = z]$ denotes the event $\{\omega \in \Omega \mid Z(\omega) = z\}$.

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 log is the base 2 logarithm.

For two random variables Z and Z' on Ω, the *conditional entropy* of Z *given the event* $[Z' = z']$ such that $\mu(Z' = z') \neq 0$ is defined by:

$$H(Z|Z' = z') = -\sum_{z} (\mu(Z = z|Z' = z') \cdot \log(\mu(Z = z|Z' = z')))$$

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:

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

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:

$$I(Z; Z') = H(Z) - H(Z|Z') = H(Z') - H(Z'|Z) = I(Z'; Z)$$

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 [13] for further properties of entropy and mutual information.

The *vulnerability* of a random variable Z, defined as:

$$V(Z) = \max_{z} \mu(Z = z)$$

gives the probability of the likeliest event of a random variable. Vulnerability evaluates the probability of a correct guess in one attempt.

Vulnerability can also be used as a measure of information, as [8,10] argue, by defining the *min-entropy* and the *conditional min-entropy* by (respectively):

$$H_\infty(Z) = -\log(V(Z)) \quad \text{and} \quad H_\infty(Z \mid Z') = -\log(V(Z \mid Z'))$$

where

$$V(Z \mid Z') = \sum_{z'} \mu(Z' = z') \cdot V(Z \mid Z' = z') = \sum_{z'} \mu(Z' = z') \cdot \max_z \mu(Z = z \mid Z' = z).$$

The corresponding mutual information is defined analogously:

$$I_\infty(Z; Z') = H_\infty(Z) - H_\infty(Z \mid Z')$$

2.3 Probabilistic models

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. Extensions to the non-deterministic setting are discussed in Section 8.

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 Fig. 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 3 (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:

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

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

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 $\mathbf{P}(K) = \mathbf{P}\left(\mathrm{tr}^{-1}(K)\right) = \sum_{\mathrm{tr}(\rho) \in K} \mathbf{P}(\rho)$.

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

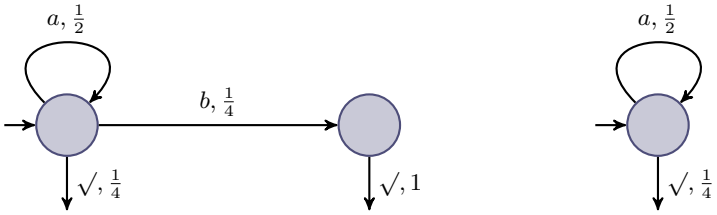
A non probabilistic version of any SA is obtained by forgetting any information about probabilities.

Definition 4. *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:
$unProb(\mathcal{O})(q_0 \xrightarrow{a_1} q_1 \cdots q_n) = \mathcal{O}(q_0 \xrightarrow{a_1} q_1 \cdots q_n\surd)$.



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

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

3 Measuring non-opacity

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 Definitions 1 and 2. Instead of wanting that *every* observation class should not be included in φ (resp. φ or $\overline{\varphi}$ for the symmetrical case), 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. Both the asymmetrical and the symmetrical notions of opacity can be generalized in this manner.

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

$$PO^A_\ell(\Pi, \varphi, \mathcal{O}) = \sum_{\substack{o \in Obs \\ \mathcal{O}^{-1}(o) \subseteq \varphi}} \mathbf{P}(\mathcal{O} = o).$$

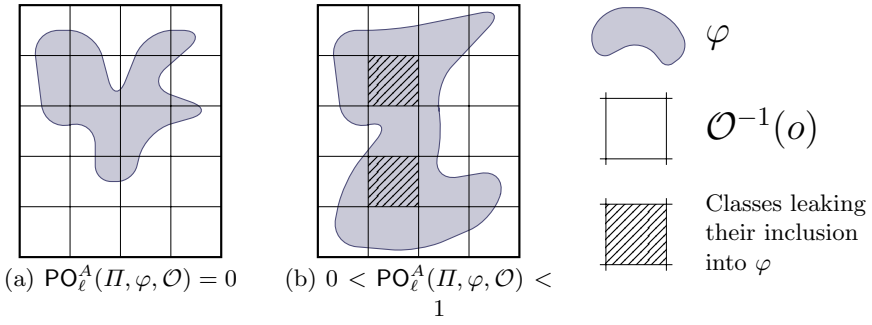
The liberal probabilistic symmetrical opacity *or LPSO is defined by:*

$$\begin{aligned} PO^S_\ell(\Pi, \varphi, \mathcal{O}) &= PO^A_\ell(\Pi, \varphi, \mathcal{O}) + PO^A_\ell(\Pi, \overline{\varphi}, \mathcal{O}) \\ &= \sum_{\substack{o \in Obs \\ \mathcal{O}^{-1}(o) \subseteq \varphi}} \mathbf{P}(\mathcal{O} = o) + \sum_{\substack{o \in Obs \\ \mathcal{O}^{-1}(o) \subseteq \overline{\varphi}}} \mathbf{P}(\mathcal{O} = o). \end{aligned}$$

This definition provides a measure of how insecure the system is. The following propositions shows that a null value for these measures coincides with (symmetrical) opacity for the system, which is then secure.

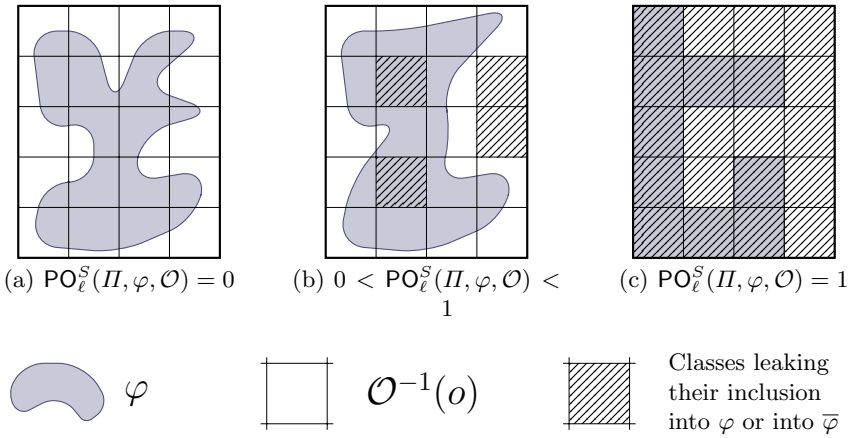
For LPO, it corresponds to classes either overlapping both φ and $\overline{\varphi}$ or included in $\overline{\varphi}$ as in Fig. 2(a). LPO measures only the classes that leak their inclusion in φ. So classes included in $\overline{\varphi}$ are not taken into account. On the other extremal point, $\mathsf{PO}^A_\ell(\Pi, \varphi, \mathcal{O}) = 1$ when φ is always true.

When LPSO is null, it means that each equivalence class $\mathcal{O}^{-1}(o)$ overlaps both φ and $\overline{\varphi}$ as in Fig. 3(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 Fig. 3(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 Fig. 3(b).



(a) $\mathsf{PO}^{A}_{\ell}(\Pi, \varphi, \mathcal{O}) = 0$ (b) $0 < \mathsf{PO}^{A}_{\ell}(\Pi, \varphi, \mathcal{O}) < 1$

Fig. 2. Liberal probabilistic opacity.



(a) $\mathsf{PO}^{S}_{\ell}(\Pi, \varphi, \mathcal{O}) = 0$ (b) $0 < \mathsf{PO}^{S}_{\ell}(\Pi, \varphi, \mathcal{O}) < 1$ (c) $\mathsf{PO}^{S}_{\ell}(\Pi, \varphi, \mathcal{O}) = 1$

Fig. 3. Liberal probabilistic symmetrical opacity.

Proposition 1.
(1) $0 \leq \mathsf{PO}^{A}_{\ell}(\Pi, \varphi, \mathcal{O}) \leq 1$
(2) $\mathsf{PO}^{A}_{\ell}(\Pi, \varphi, \mathcal{O}) = 0$ *if and only if* φ *is opaque on* $unProb(\Pi)$ *with respect to* $unProb(\mathcal{O})$.

(3) $\mathsf{PO}^A_\ell(\Pi, \varphi, \mathcal{O}) = 1$ *if and only if* $\varphi = Run(\Pi)$.

Proposition 2.
(1) $0 \leq \mathsf{PO}^S_\ell(\Pi, \varphi, \mathcal{O}) \leq 1$
(2) $\mathsf{PO}^S_\ell(\Pi, \varphi, \mathcal{O}) = 0$ *if and only if* φ *is symmetrically opaque on* $unProb(\Pi)$ *with respect to* $unProb(\mathcal{O})$.
(3) $\mathsf{PO}^S_\ell(\Pi, \varphi, \mathcal{O}) = 1$ *if and only if* $H(\mathbf{1}_\varphi|\mathcal{O}) = 0$.

Proof (Proof of Propositions 1 and 2.).
(1) The considered events are mutually exclusive, hence the sum of their probabilities never exceeds 1.
(2) First observe that a complete run $r_0 a \dots r_n \surd$ has a non null probability in Π iff $r_0 a \dots r_n$ is a run in $unProb(\Pi)$. Suppose $\mathsf{PO}^A_\ell(\Pi, \varphi, \mathcal{O}) = 0$. Then there is no observable o with non-null probability such that $\mathcal{O}^{-1}(o) \subseteq \varphi$. Hence for each observable o, $\mathcal{O}^{-1}(o) \not\subseteq \varphi$. Conversely, if φ is opaque on $unProb(\Pi)$, there is no observable $c \in Obs$ such that $\mathcal{O}^{-1}(c) \subseteq \varphi$, hence the null value for $\mathsf{PO}^A_\ell(\Pi, \varphi, \mathcal{O})$. The case of LPSO is similar, also taking into account the dual case of $\overline{\varphi}$ in the above.
(3) For LPO, this is straightforward from the definition. For LPSO, $H(\mathbf{1}_\varphi|\mathcal{O}) = 0$ 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 $\mathsf{PO}^S_\ell(\Pi, \varphi, \mathcal{O}) = 1$.

3.2 Example: Non-interference

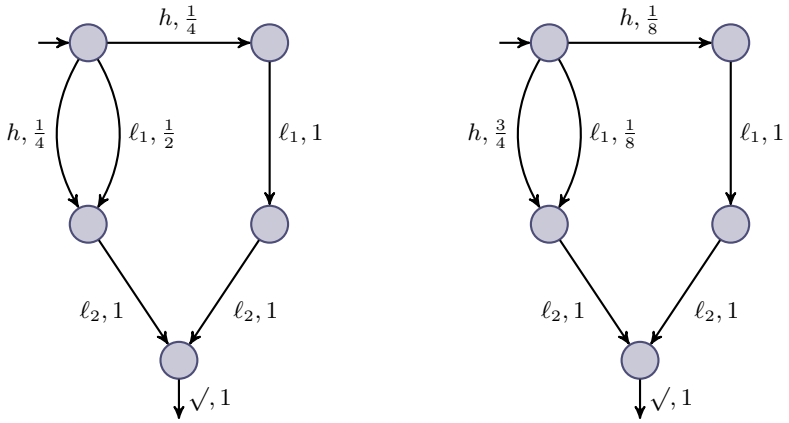
Consider the systems $\mathcal{A}_3$ and $\mathcal{A}_4$ of Fig. 4. On these systems we define the predicate φ_{NI} which is true if the trace of a run contains letter h. In both cases the observation function $\mathcal{O}_L$ returns the projection of the trace onto the alphabet $\{\ell_1, \ell_2\}$. Remark that this example is an interference property [14] 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 2. 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:

$$\mathsf{PO}^A_\ell(\mathcal{A}_3, \varphi_{NI}, \mathcal{O}_L) = \frac{1}{4} \qquad \mathsf{PO}^A_\ell(\mathcal{A}_4, \varphi_{NI}, \mathcal{O}_L) = \frac{3}{4}$$

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.

Note that in this example, LPO and LPSO coincide. This is not always the case. Indeed, in the unprobabilistic setting, both symmetrical and asymmetrical opacity of φ_{NI} with respect to $\mathcal{O}_L$ express the intuitive notion that "an external observer does not know whether an action happened or not". The asymmetrical notion corresponds to the definition of *strong nondeterministic non-interference* in [14] while the symmetrical one was defined as *perfect security property* in [3].



(a) FPA $\mathcal{A}_3$ (b) FPA $\mathcal{A}_4$

Fig. 4. Interferent FPAs $\mathcal{A}_3$ and $\mathcal{A}_4$.

$\mathrm{tr}(\rho)$	$\mathbf{P}_{\mathcal{A}_3}(\rho)$	$\mathbf{P}_{\mathcal{A}_4}(\rho)$	$\in \varphi_{NI}$?	$\mathcal{O}_L(\rho)$
$\mathrm{tr}(\rho_1) = \ell_1\ell_2\surd$	1/2	1/8	0	$\ell_1\ell_2$
$\mathrm{tr}(\rho_2) = h\ell_1\ell_2\surd$	1/4	1/8	1	$\ell_1\ell_2$
$\mathrm{tr}(\rho_3) = h\ell_2\surd$	1/4	3/4	1	ℓ_2

Table 2. Runs of $\mathcal{A}_3$ and $\mathcal{A}_4$.

4 Measuring the robustness of opacity

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 Fig. 3(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. Hence for each observation class, we will not ask if it is included in φ, but how likely φ is to be true inside this class with a probabilistic measure taking into account the likelihood of classes. This amounts to measuring, inside each observation class, $\overline{\varphi}$ in the case of asymmetrical opacity, and the balance between φ and $\overline{\varphi}$ in the case of symmetrical opacity.

4.1 Restrictive Probabilistic Opacity

In this section we extend the notion of (asymmetrical) opacity in order to measure how secure the system is.

Definition and properties. In this case, an observation class is more secure if φ is less likely to be true. That means that it is easy (as in "more likely") to find a run not in φ in the same observation class. Dually, a high probability for φ inside a class means that few (again probabilistically speaking) runs will be in the same class yet not in φ.

Restrictive probabilistic opacity is defined to measure this effect globally on all observation classes. It is tailored to fit the definition of opacity in the classical sense: indeed, if one class totally leaks its presence in φ, RPO will detect it.

Definition 6. *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*

$$\frac{1}{PO_r^A(\Pi,\varphi,\mathcal{O})} = \sum_{o\in Obs} \mathbf{P}(\mathcal{O}=o)\cdot\frac{1}{\mathbf{P}(\mathbf{1}_\varphi = 0 \mid \mathcal{O}=o)}$$

RPO is the harmonic means (weighted by the probabilities of observations) of the probability that φ is false in a given observation class. Hence it will be null if and only if there is one class that is contained in φ. On the other hand, it will be 1 only if φ is always false.

Proposition 3.
(1) $0 \leq PO_r^A(\Pi,\varphi,\mathcal{O}) \leq 1$
(2) $PO_r^A(\Pi,\varphi,\mathcal{O}) = 0$ *if and only if φ is not opaque on $unProb(\Pi)$ with respect to $unProb(\mathcal{O})$.*
(3) $PO_r^A(\Pi,\varphi,\mathcal{O}) = 1$ *if and only if* $\varphi = \emptyset$.

Example: Debit Card System. Consider a Debit Card system in a store. When a card is inserted, an amount of money x to be debited is entered, and the client enters his PIN number (all this being gathered as the action Buy(x)). The amount of the transaction is given probabilistically as an abstraction of the statistics of such transactions. Provided the PIN is correct, the system can either directly allow the transaction, or interrogate the client's bank for solvency. In

order to balance the cost associated with this verification (bandwidth, server computation, etc.) with the loss induced if an insolvent client was debited, the decision to interrogate the bank's servers is taken probabilistically according to the amount of the transaction. When interrogated, the bank can reject the transaction with a certain probability[3] or accept it. This system is represented by the FPA $\mathcal{A}_{\text{card}}$ of Fig. 5.

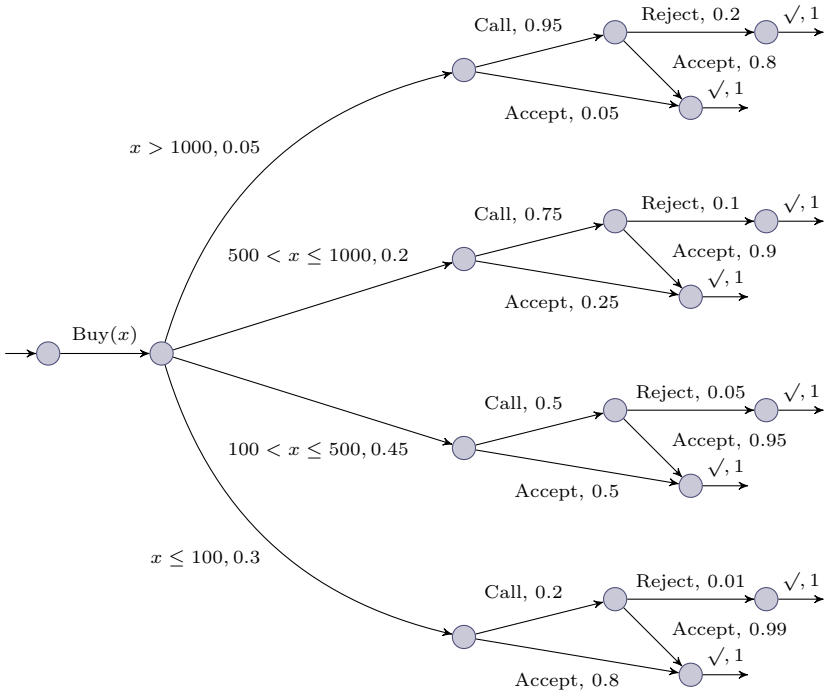


Fig. 5. The Debit Card system $\mathcal{A}_{\text{card}}$.

Now assume an external observer can only observe if there has been a call or not to the bank server. In practice, this can be achieved, for example, by measuring the time taken for the transaction to be accepted (it takes longer when the bank is called), or by spying on the telephone line linking the store to the bank's servers (detecting activity on the network or idleness). Suppose what the external observer wants to know is whether the transaction was worth more than 500€. By using RPO, one can assess how this knowledge can be derived from observation.

Formally, in this case the observables are $\{\varepsilon, \text{Call}\}$, the observation function $\mathcal{O}_{\text{Call}}$ being the projection on $\{\text{Call}\}$. The predicate to be hidden to the user

[3] Although the banks process to allow or forbid the transaction is deterministic, the statistics of the result can be abstracted into probabilities.

is represented by the regular expression $\varphi_{>500} = \Sigma^*(\text{“}x > 1000\text{”} + \text{“}500 < x \leq 1000\text{”})\Sigma^*$ (where Σ is the whole alphabet). By definition of RPO:

$$\frac{1}{\mathsf{PO}_r^A(\mathcal{A}_{\text{card}}, \varphi_{>500}, \mathcal{O}_{\text{Call}})} = \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon) \cdot \frac{1}{\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \varepsilon)} + \mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call}) \cdot \frac{1}{\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \text{Call})}$$

For individual probabilities we have:

$$\begin{aligned}
\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon) &= \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon, x > 1000) + \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon, 500 < x \leq 1000) \\
&\quad + \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon, 100 < x \leq 500) + \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon, x \leq 100) \\
\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon) &= \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon | x > 1000) \cdot \mathbf{P}(x > 1000) \\
&\quad + \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon | 500 < x \leq 1000) \cdot \mathbf{P}(500 < x \leq 1000) \\
&\quad + \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon | 100 < x \leq 500) \cdot \mathbf{P}(100 < x \leq 500) \\
&\quad + \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon | x \leq 100) \cdot \mathbf{P}(x \leq 100) \\
\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon) &= 0.05 \cdot 0.05 + 0.25 \cdot 0.2 + 0.5 \cdot 0.45 + 0.8 \cdot 0.3 \;=\; 0.5175
\end{aligned}$$

$$\mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call}) = 1 - \mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon) \;=\; 0.4825$$

$$\begin{aligned}
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \varepsilon) &= \frac{\mathbf{P}(\neg\varphi_{>500}, \mathcal{O}_{\text{Call}} = \varepsilon)}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon)} \\
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \varepsilon) &= \frac{\mathbf{P}(x \leq 100, \mathcal{O}_{\text{Call}} = \varepsilon) + \mathbf{P}(100 < x \leq 500, \mathcal{O}_{\text{Call}} = \varepsilon)}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon)} \\
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \varepsilon) &= \frac{\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon | x \leq 100) \cdot \mathbf{P}(x \leq 100)}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon)} \\
&\quad + \frac{\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon | 100 < x \leq 500) \cdot \mathbf{P}(100 < x \leq 500)}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \varepsilon)} \\
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \varepsilon) &= \frac{0.8 \cdot 0.3 + 0.5 \cdot 0.45}{0.5175} \;=\; \frac{0.465}{0.5175} \;\simeq\; 0.899
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \text{Call}) &= \frac{\mathbf{P}(\neg\varphi_{>500}, \mathcal{O}_{\text{Call}} = \text{Call})}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call})} \\
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \text{Call}) &= \frac{\mathbf{P}(x \leq 100, \mathcal{O}_{\text{Call}} = \text{Call}) + \mathbf{P}(100 < x \leq 500, \mathcal{O}_{\text{Call}} = \text{Call})}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call})} \\
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \text{Call}) &= \frac{\mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call} | x \leq 100) \cdot \mathbf{P}(x \leq 100)}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call})} \\
&\quad + \frac{\mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call} | 100 < x \leq 500) \cdot \mathbf{P}(100 < x \leq 500)}{\mathbf{P}(\mathcal{O}_{\text{Call}} = \text{Call})} \\
\mathbf{P}(\neg\varphi_{>500} | \mathcal{O}_{\text{Call}} = \text{Call}) &= \frac{0.2 \cdot 0.3 + 0.5 \cdot 0.45}{0.4825} \;=\; \frac{0.285}{0.4825} \;\simeq\; 0.591
\end{aligned}$$

Hence

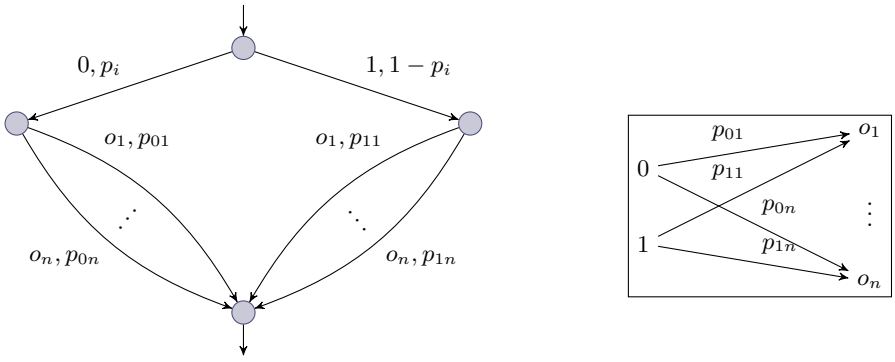
$$\frac{1}{\mathsf{PO}_r^A(\mathcal{A}_{\text{card}}, \varphi_{>500}, \mathcal{O}_{\text{Call}})} = 0.5175 \cdot \frac{0.5175}{0.465} + 0.4825 \cdot \frac{0.4825}{0.285}$$

$$\frac{1}{\mathsf{PO}_r^A(\mathcal{A}_{\text{card}}, \varphi_{>500}, \mathcal{O}_{\text{Call}})} = \frac{57 \cdot 0.5175 \cdot 69 + 31 \cdot 0.4825 \cdot 193}{3534} = \frac{4922.125}{3534}$$

$$\mathsf{PO}_r^A(\mathcal{A}_{\text{card}}, \varphi_{>500}, \mathcal{O}_{\text{Call}}) = \frac{3534}{4922.125} \simeq 0.718$$

4.2 Restricting symmetrical opacity

Symmetrical opacity offers a sound framework to analyze the secret of a binary value. For example, consider a binary canal with n outputs. It can be modeled by a tree-like system branching on 0 and 1 at the first level, then branching on observables $\{o_1, \ldots, o_n\}$, as in Fig. 6. If the system wishes to prevent communication, the secret of predicate "the input of the channel was 1" is as important as the secret of its negation; in this case "the input of the channel was 0". Such case is an example of *initial opacity* [2], since the secret appears only at the start of each run. Note that any system with initial opacity and a finite set of observables can be transformed into a channel [10].



(a) A system with initial secret

(b) Channel with binary input

Fig. 6. A system and its associated channel

The notion of asymmetrical opacity, however, fails to capture security in terms of opacity for both φ and $\overline{\varphi}$. And so does the RPO measure. Therefore we define a symmetric version of RPO.

Definition and properties. Symmetrical opacity ensures that for each observation class o (reached by a run), the probability of both $\mathbf{P}(\varphi \mid o)$ and $\mathbf{P}(\overline{\varphi} \mid o)$

is strictly above 0. That means that the lower of these probabilities should be above 0. In turn, the lowest of these probability is exactly the complement of the vulnerability (since $\mathbf{1}_\varphi$ can take only two values). That is, the security is measured with the probability of error in one guess (inside a given observation class). Hence, a system will be secure if, in each observation class, φ is *balanced* with $\overline{\varphi}$.

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

$$\mathsf{PO}_r^S(\Pi, \varphi, \mathcal{O}) = \frac{-1}{\sum_{o \in Obs} \mathbf{P}(\mathcal{O} = o) \cdot \log\left(1 - V(\mathbf{1}_\varphi \mid \mathcal{O} = o)\right)}$$

where $V(\mathbf{1}_\varphi \mid \mathcal{O} = o) = \max_{i \in \{0,1\}} \mathbf{P}(\mathbf{1}_\varphi = i \mid \mathcal{O} = o)$.

In this definition, taking $-\log(1 - V(\mathbf{1}_\varphi \mid \mathcal{O} = o))$ allows to give more weight to very imbalanced classes, up to infinity for classes completely included either in φ or in $\overline{\varphi}$. It also gives a measure in terms of bits instead of probabilities. These measures are then averaged with respect to the probability of each observation class. The final inversion is a normalization operation. The above motivations for the definition of RPSO yield some desired properties.

Proposition 4. *(1)* $0 \leq \mathsf{PO}_r^S(\Pi, \varphi, \mathcal{O}) \leq 1$

(2) $\mathsf{PO}_r^S(\Pi, \varphi, \mathcal{O}) = 0$ *if and only if φ is* not *symmetrically opaque on $unProb(\Pi)$ with respect to $unProb(\mathcal{O})$.*

(3) $\mathsf{PO}_r^S(\Pi, \varphi, \mathcal{O}) = 1$ *if and only if* $\forall o \in Obs,\ \mathbf{P}(\mathbf{1}_\varphi = 1 \mid \mathcal{O} = o) = \frac{1}{2}$.

Examples.

Sale protocol. We consider the sale protocol introduced in [11] depicted in Fig. 7. Two products can be put on sale, either a cheap or an expensive one, and two clients, either a rich or a poor one, may want to buy it. The products are put on sale according to a distribution (α and $\overline{\alpha} = 1 - \alpha$) while buyers behave probabilistically (through β and γ) although differently according to the price of the item on sale. The price of the item is public, but the identity of the buyer should remain secret. Hence the observation function *Price* yields *cheap* or *expensive*, while the secret is, without loss of symmetry, the set φ_{poor} of runs ending with *poor*. The bias introduced by the preference of, say, a cheap item by the poor client betrays the secret identity of the buyer. RPSO allows to measure this bias, and more importantly, to compare the bias obtained globally for different values of the parameters α, β, and γ.

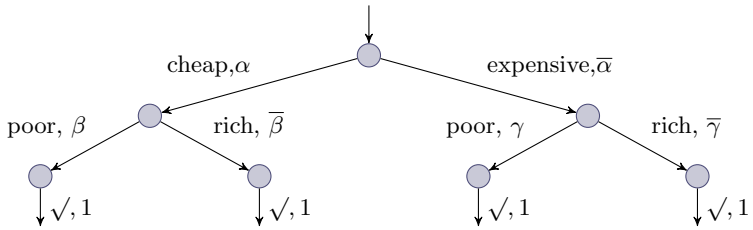


Fig. 7. A simple sale protocol represented as an FPA $\mathcal{S}ale$.

More formally, we have:

$$\mathbf{P}(Price = \text{cheap}) = \alpha \qquad \mathbf{P}(Price = \text{expensive}) = \overline{\alpha}$$

$$V(\mathbf{1}_{\varphi_{\text{poor}}} = 1 \mid \mathcal{O} = \text{cheap}) = \max(\beta, \overline{\beta})$$

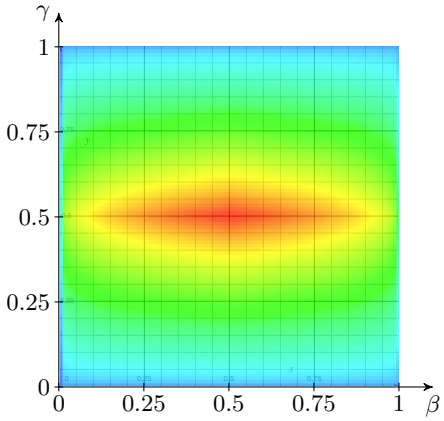
$$V(\mathbf{1}_{\varphi_{\text{poor}}} = 1 \mid \mathcal{O} = \text{expensive}) = \max(\gamma, \overline{\gamma})$$

$$\mathsf{PO}_r^S(\mathcal{S}ale, \varphi_{\text{poor}}, Price) = \frac{-1}{\alpha \cdot \log(\min(\beta, \overline{\beta})) + \overline{\alpha} \cdot \log(\min(\gamma, \overline{\gamma}))}$$

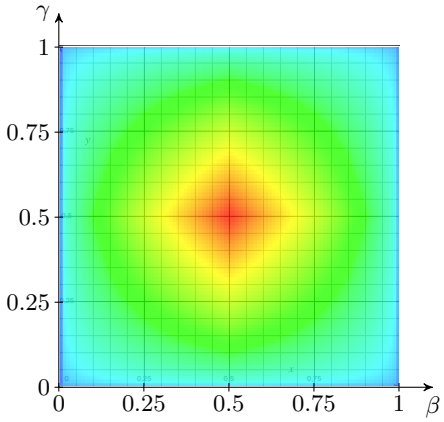
which is depicted for several values of α in Fig. 8, red meaning higher value for RPSO.

Dining Cryptographers Protocol. Introduced in [4], 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". Note that predicate φ_2 is well suited for analysis of symmetrical opacity, since detecting that φ_2 is false gives information on who paid (here C_3). 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.



(a) $\mathsf{PO}_r^S(\mathcal{S}ale, \varphi_{\text{poor}}, Price)$ when $\alpha = \frac{1}{8}$



(b) $\mathsf{PO}_r^S(\mathcal{S}ale, \varphi_{\text{poor}}, Price)$ when $\alpha = \frac{1}{2}$

Fig. 8. RPSO for the sale protocol.

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 Fig. 9. 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.

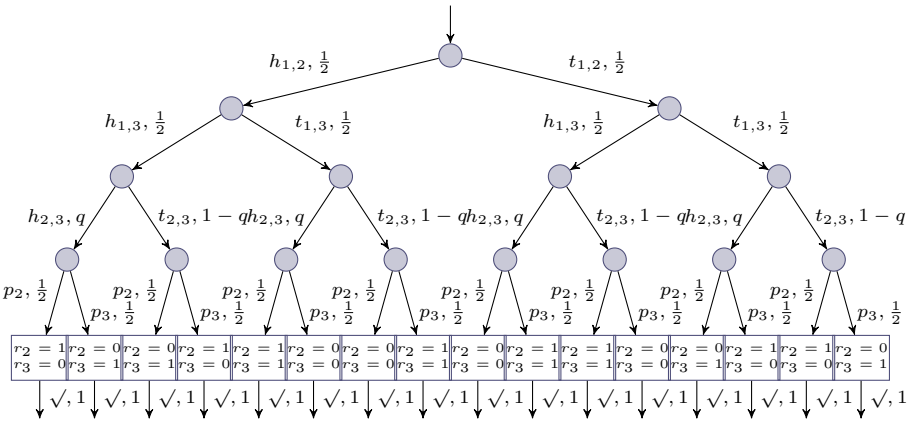


Fig. 9. The FPA corresponding to the Dining Cryptographers protocol.

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

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

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 $1 - q$. RPSO can measure this advantage globally.

For each observation class, the vulnerability of φ_2 is $\max(q, 1-q)$. Hence the RPSO will be

$$\mathsf{PO}_r^S(\mathcal{D}, \varphi_2, \mathcal{O}_1) = \frac{-1}{\log(\max(q, 1-q))}$$

The variations of the RPSO when changing the bias on q are depicted in Fig. 10. Analysis of RPSO according to the variation of q yields that the system is perfectly secure if there is no bias on the coin, and insecure if $q = 0$ or $q = 1$.

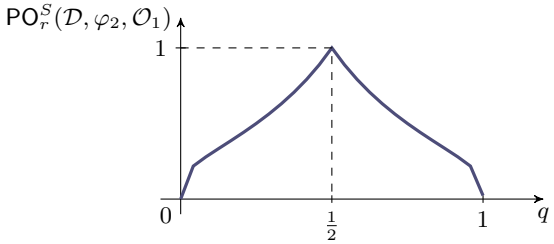


Fig. 10. Evolution of the restrictive probabilistic symmetric opacity of the Dining Cryptographers protocol when changing the bias on the coin.

5 Tightening opacity through information theory

5.1 Information based opacity

We now adopt another point of view on how to measure the *balance* between φ and $\overline{\varphi}$ in each observation class. Namely, the balance inside each class is compared to the global balance, thus allowing to measure not the security per se but what advantage is given to the attacker through observation. Since the balance between φ and $\overline{\varphi}$ is measured, they are considered evenly; thus this measure relates only to the symmetrical opacity.

Definition 8 (Information-based probabilistic opacity). *Let φ be a predicate on the complete runs of an FPA Π and $\mathcal{O}$ an observation function. The* information based probabilistic opacity *(IPSO) of φ on Π, with respect to $\mathcal{O}$, is defined by*

$$\mathsf{PO}_i^S(\Pi, \varphi, \mathcal{O}) = 1 - I(\mathbf{1}_\varphi; \mathcal{O})$$

Proposition 5.
(1) $0 \leq \mathsf{PO}_i^S(\Pi, \varphi, \mathcal{O}) \leq 1$
(2) If $\mathsf{PO}_i^S(\Pi, \varphi, \mathcal{O}) = 0$*, 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 2 case (3), $\mathsf{PO}_\ell^S(\Pi, \varphi, \mathcal{O}) = 1 > 0$, then φ is not opaque on $unProb(\Pi)$ with respect to $unProb(\mathcal{O})$.

A measure analogous to IPSO, but using the min-entropy instead of entropy [8] can also be defined:

$$\mathsf{PO}^S_{\min}(\Pi, \varphi, \mathcal{O}) = 1 - I_\infty(\mathbf{1}_\varphi; \mathcal{O}) = 1 - H_\infty(\mathbf{1}_\varphi) + H_\infty(\mathbf{1}_\varphi \mid \mathcal{O})$$

It has roughly the same properties as IPSO, since in our context $\mathbf{1}_\varphi$ can only take two values.

5.2 Example: the Dining Cryptographers Protocol

The Dining Cryptographers Protocol, as explained in Section 4.2 can be evaluated in terms of information leak. For the next IPSO computation, we write $\mathbf{1}_\varphi$ instead of $\mathbf{1}_{\varphi_2}$ and $\mathcal{O}$ instead of $\mathcal{O}_1$.

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

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 IPSO, parametrized by q:

$$\mathsf{PO}^S_i(\mathcal{D}, \varphi_2, \mathcal{O}_1) = -(q \cdot \log(q) + (1 - q) \cdot \log(1 - q))$$

On this expression we can see that $\mathsf{PO}^S_i(\mathcal{D}, \varphi_2, \mathcal{O}_1) = 1$ if $q = \frac{1}{2}$, and $\mathsf{PO}^S_i(\mathcal{D}, \varphi_2, \mathcal{O}_1) = 0$ if $q = 0$ or $q = 1$. The variations of the IPSO when changing the bias on q are depicted in Fig. 11. They bear the same features as for RPSO at extremal points ($q = 0, \frac{1}{2}, 1$), although the shape of the curve is altered.

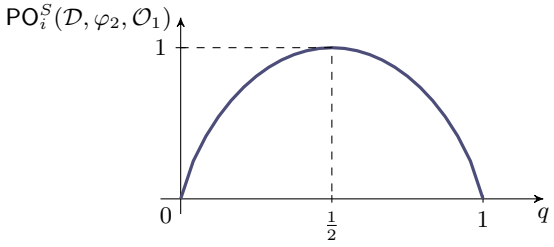


Fig. 11. Evolution of the restrictive probabilistic opacity of the Dining Cryptographers protocol when changing the bias on the coin.

6 Computing opacity measures automatically

We now show how all measures defined above 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 [15]. 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 a system of equations associated with this product. The computation of all measures results in applications of this operation with several automata.

6.1 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 [15,16,17] for probabilistic verification.

Definition 9 (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 [15,17], 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 $\mathcal{S}_\Pi$.*

6.2 Computing the probability of a regular language

In order to compute the probability of a language inside a system, we build a substochastic automaton that corresponds to the intersection of the system and the language, then compute the probability as above.

Definition 10 (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 $(q_2, r_2) \in Q \times Q_K$,*

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

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

In this synchronized product, the behaviors are constrained by the finite automaton. Actions not allowed by the automaton are trimmed, and states can accept only if they correspond to a valid behavior of the DFA. Note that this product is defined on SA in order to allow several intersections. The correspondence between the probability of a language in a system and the probability of the synchronized product is laid out in the following lemma.

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*

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

6.3 Computing all opacity measures

All measures defined previously can be computed as long as, for $i \in \{0, 1\}$ and $o \in Obs$, all probabilities

$$\mathbf{P}(\mathbf{1}_\varphi = i) \qquad \mathbf{P}(\mathcal{O} = o) \qquad \mathbf{P}(\mathbf{1}_\varphi = i, \mathcal{O} = o)$$

can be computed. Indeed, even deciding whether $\mathcal{O}^{-1}(o) \subseteq \varphi$ can be done by testing $\mathbf{P}(\mathcal{O} = o) > 0 \wedge \mathbf{P}(\mathbf{1}_\varphi = 0, \mathcal{O} = o) = 0$.

Now suppose Obs is a finite set, φ and all $\mathcal{O}^{-1}(o)$ are regular sets. Then one can build deterministic finite automata $\mathcal{A}_\varphi$, $\mathcal{A}_{\overline{\varphi}}$, $\mathcal{A}_o$ for $o \in Obs$ that accept respectively φ, $\overline{\varphi}$, and $\mathcal{O}^{-1}(o)$.

Synchronizing automaton $\mathcal{A}_\varphi$ with Π and pruning it yields a substochastic automaton $\Pi || \mathcal{A}_\varphi$. By Lemma 2, the probability $\mathbf{P}(\mathbf{1}_\varphi = 1)$ is then computed by solving the linear system associated with $\Pi || \mathcal{A}_\varphi$. Similarly, one obtain $\mathbf{P}(\mathbf{1}_\varphi = 0)$ (with $\mathcal{A}_{\overline{\varphi}}$), $\mathbf{P}(\mathcal{O} = o)$ (with $\mathcal{A}_o$), $\mathbf{P}(\mathbf{1}_\varphi = 1 \mathcal{O} = o)$ (synchronizing $\Pi || \mathcal{A}_\varphi$ with $\mathcal{A}_o$), and $\mathbf{P}(\mathbf{1}_\varphi = 0 \mathcal{O} = o)$ (synchronizing $\Pi || \mathcal{A}_{\overline{\varphi}}$ with $\mathcal{A}_o$).

Theorem 1. *Let Π be an FPA. If Obs is a finite set, φ is a regular set and for $o \in Obs$, $\mathcal{O}^{-1}(o)$ is a regular set, then for $PO^* \in \{PO^A_\ell, PO^S_\ell, PO^A_r, PO^S_r, PO^S_i, PO^S_{min}\}$, $PO^*(\Pi, \varphi, \mathcal{O})$ can be computed.*

The computation of opacity measures is done in polynomial time in the size of Obs and DFAs $\mathcal{A}_\varphi$, $\mathcal{A}_{\overline{\varphi}}$, $\mathcal{A}_o$.

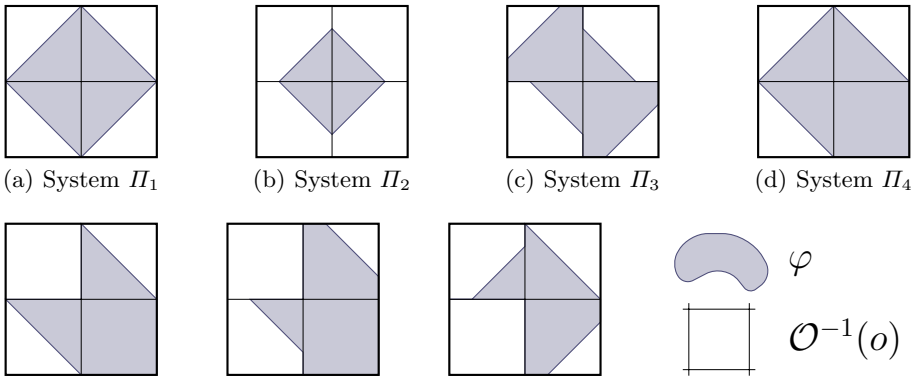
A prototype tool implementing this algorithm was implemented in Java, yielding numerical values for measures of opacity.

7 Comparison of the measures of opacity

In this section we compare the discriminating power of the different measures discussed above.

7.1 Abstract examples

The values of these metrics are first compared for pathologic cases of Fig. 12. These values are displayed in Table 3.



(a) System Π_1 (b) System Π_2 (c) System Π_3 (d) System Π_4

(e) System Π_5 (f) System Π_6 (g) System Π_7

Fig. 12. Example of repartition of probabilities of $\mathbf{1}_\varphi$ and $\mathcal{O}$ in 7 pathological cases.

System	PO^A_ℓ	PO^S_ℓ	PO^A_r	PO^S_r	PO^S_i	$\mathsf{PO}^S_{\min}$
(a) Π_1	0	0	$\frac{1}{2}$	1	1	1
(b) Π_2	0	0	$\frac{3}{4}$	$\frac{1}{2}$	1	1
(c) Π_3	0	0	$\frac{3}{8}$	$\frac{1}{2}$	$2-\frac{3}{4}\log 3 \simeq 0.81$	$2-\log 3 \simeq 0.42$
(d) Π_4	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{\log 15-10}{8} \simeq 0.80$	1
(e) Π_5	$\frac{1}{4}$	$\frac{1}{2}$	0	0	$\frac{1}{2}$	$2-\log 3 \simeq 0.42$
(f) Π_6	$\frac{1}{4}$	$\frac{1}{2}$	0	0	$1-\frac{3}{8}\log 3 \simeq 0.41$	$3-\log 7 \simeq 0.19$
(g) Π_7	0	$\frac{1}{4}$	$\frac{12}{25}$	0	$\frac{5\log 5-6}{8} \simeq 0.70$	$\log\frac{5}{3} \simeq 0.74$

Table 3. Values of the different opacity measures for systems of Fig. 12(a)-(g).

First, the system Π_1 of Fig. 12(a) is intuitively very secure since, with or without observation, an attacker has no information whether φ was true or not. This security is reflected in all symmetrical measures, with highest scores

possibles in all cases. It is nonetheless deemed more insecure for RPO, since opacity is perfect when φ is always false.

The case of Π_2 of Fig. 12(b) differs only from Π_1 by the global repartition of φ in $Run(\Pi)$. The information an attacker gets comes not from the observation, but from φ itself. Therefore both IPSO and minPO also evaluate Π_2 as very secure. But RPSO, which does not remove the information available before observation, evaluates this system as less secure than Π_1. On the other hand, RPO finds Π_2 more secure than Π_1: φ is verified less often. Note that the complement would no change the value for symmetrical measures, while being insecure for RPO (with $\mathsf{PO}_r^A = \frac{1}{4}$).

However, since each observation class is considered individually, RPSO does not discriminate Π_2 and Π_3 of Fig. 12(c). Here, the information is the same in each observation class as fo Π_2, but the repartition of φ gives no advantage at all to an attacker without observation. Hence both IPSO and minPO measure this difference well, since they are defined to measure the information *gained* through observation.

System Π_4 of Fig. 12(d) shows the limitation of minPO. Designed to accentuate imbalance inside each observation class, they are however averaged out proportionally to each $\mathbf{P}(o)$. In this case, the strong imbalance of the bottom-right class is averaged by the balance of the three others, up to the same imbalance of φ globally. Remark that $\mathsf{PO}_r^A = \mathsf{PO}_r^S = 0$ and both $\mathsf{PO}_\ell^A > 0$ and $\mathsf{PO}_\ell^S > 0$, since Π_4 is not opaque for the classical definitions.

When the system is not opaque (resp. symmetrically opaque), RPO (resp. RPSO) cannot discriminate them, and LPO (resp. LPSO) becomes relevant. For example, Π_5 of Fig. 12(e) has a greater PO_ℓ^S than Π_4. However, LPO is unchanged since the class completely out of φ is not taken into account. Both IPSO and minPO are also affected by the increase of (very) unbalanced observation classes, but do not reach a minimum nonetheless. This allows to measure further variations of imbalance, and to compare Π_5 to Π_6 of Fig. 12(f). Similarly, the difference of balance between Π_4 and Π_7 of Fig. 12(g) can also be captured by IPSO and minPO. Remark that system Π_7 is opaque but not symmetrically opaque, hence the relevant measures are PO_ℓ^S and PO_r^A.

7.2 Concrete examples

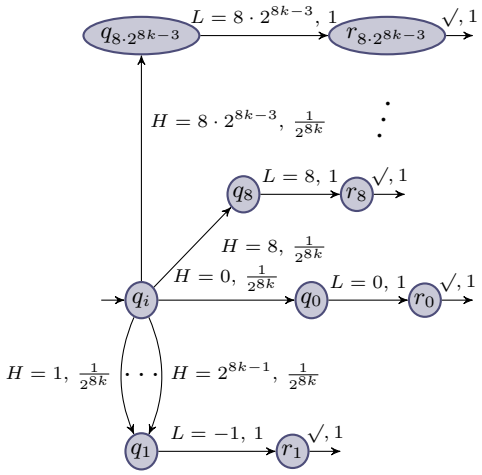
We shall now study more concrete examples. The first ones are inspired from [8]. We then use the previous study of the crowds protocol.

Consider the following programs P_1 and P_2, where k is a given parameter, `random` select uniformly an integer value (in binary) between its two arguments and & is the bitwise *and*:

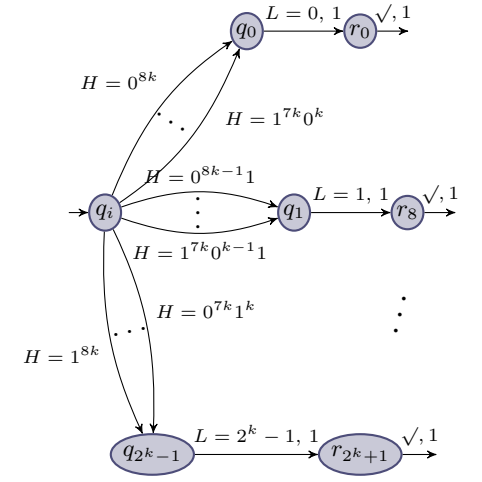
P_1: $H := \texttt{random}(0, 2^{8k} - 1)$;
if $H \mod 8 = 0$ then
  $L := H$
else
  $L := -1$
fi

P_2: $H := \texttt{random}(0, 2^{8k} - 1)$;
$L := H \ \&\ 0^{7k}1^k$

In these cases, the value of H, an integer over $8k$ bits, is supposed to remain secret. Intuitively, P_1 divulges the exact value of H with probability $\frac{1}{8}$. On the other hand, P_2 leaks the value of one eighth of its bits (the least significant ones) at every execution. These programs can be translated into FPAs $\mathcal{A}_{P_1}$ and $\mathcal{A}_{P_2}$ of Fig. 13. The observation is the "$L = \ldots$" action. In order to have a boolean predicate, the secret is not the value of variable H, but whether $H = L$: $\varphi_= = \big\{(H = x)(L = x) \mid x \in \{0, \ldots, 2^{8k} - 1\}\big\}$. First remark that $\varphi_=$ is not



(a) FPA $\mathcal{A}_{P_1}$



(b) FPA $\mathcal{A}_{P_2}$

Fig. 13. FPAs for programs P_1 and P_2.

opaque on P_1 in the classical sense (both symmetrically or not). Hence both RPO and RPSO are null. On the other hand, $\varphi_=$ is opaque on P_2, hence LPO and LPSO are null. However, in both cases IPSO and minPO are meaningful. The values for all measures are gathered in Table 4. Note that only restrictive

Program	PO^A_ℓ	PO^S_ℓ	PO^A_r	PO^S_r	PO^S_i	$\mathsf{PO}^S_{\min}$
P_1	$\frac{1}{8}$	1	0	0	$\frac{7}{8} \cdot \log 7 - 2 \simeq 0.46$	$\log 7 - 2 \simeq 0.81$
P_2	0	0	$1 - \frac{1}{2^{7k}}$	$\frac{1}{7k}$	1	1

Table 4. Opacity measures for programs P_1 and P_2.

opacity for P_2 depend on k. This comes from the fact that in all other cases, both $\varphi_=$ and the equivalence classes scale at the same rate with k. In the case of P_2, adding length to the secret variable H *dilutes* $\varphi_=$ inside each class. Hence the greater k is, the hardest it is for an attacker to know that $\varphi_=$ is true, thus to crack asymmetrical opacity. Indeed, it will tend to get false in most cases, thus providing an easy guess, and a low value for symmetrical opacity.

7.3 Case study: the Crowds protocol

The anonymity protocol known as *crowds* was introduced in [5] and recently studied in the probabilistic framework in [9,10]. 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, since no user is more likely than another to be the actual initiator; indeed its RPO is very low. 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 Fig. 14. 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.

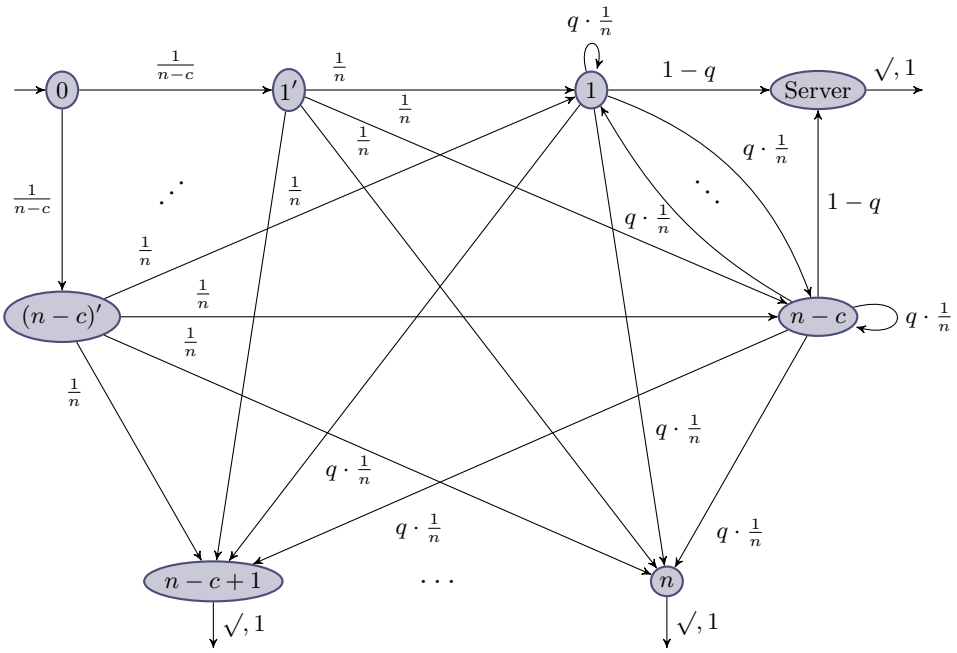


Fig. 14. FPA $\mathcal{C}_n^c$ for Crowds protocol with n users, among whom c are corrupted.

Computation of the probabilities. All probabilities $\mathbf{P}(i \rightsquigarrow j)$ can be automatically computed using the method described in Section 6. 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 Fig. 15. The associated system is represented in Table 5 where L_S corresponds to the "Server" state. Resolving it yields, $L_i = \frac{q}{n}$ for all $i \in \{1, \ldots, n-c-1\}$, $L_{n-c} = 1 - \frac{q \cdot (n-c-1)}{n}$, $L_{1'} = \frac{1}{n}$, 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 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

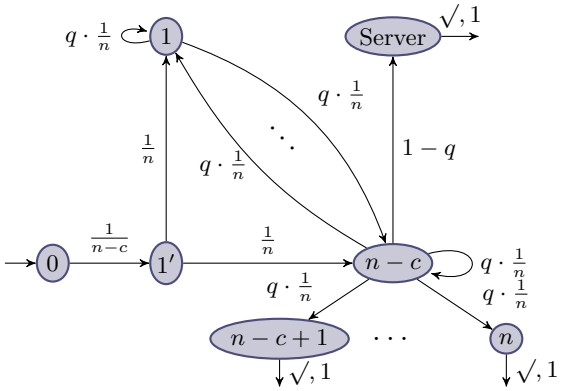


Fig. 15. SA $\mathcal{C}_n^c || \mathcal{A}_{1 \rightsquigarrow (n-c)}$ corresponding to runs where user 1 initiates the protocol and user $(n-c)$ is detected

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

Table 5. Linear system associated to SA $\mathcal{C}_n^c || \mathcal{A}_{1 \rightsquigarrow (n-c)}$ of Fig. 15

(the internal route between honest users before j is irrelevant). Therefore

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

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

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

Conditional probabilities thus follow:

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

Interestingly, these probabilities do not depend on q[4].

Computation of RPO. From the probabilities above, one can compute an analytical value for $\mathsf{PO}_r^A(\mathcal{C}_n^c, \mathbf{1}_{\varphi_i}, \mathcal{O})$.

$$\begin{aligned}
\frac{1}{\mathsf{PO}_r^A(\mathcal{C}_n^c \varphi_i, \mathcal{O})} &= \sum_{j=1}^{n-c} \mathbf{P}(\rightsquigarrow j) \frac{1}{\mathbf{P}(\neg i \rightsquigarrow \mid \rightsquigarrow j)} \\
&= (n-c-1) \cdot \frac{1}{n-c} \cdot \frac{n}{n-1} + \frac{1}{n-c} \cdot \frac{n}{n-c-1} \\
&= \frac{n}{n-c} \left(\frac{n-c-1}{n-1} + \frac{1}{n-c-1} \right) \\
\frac{1}{\mathsf{PO}_r^A(\mathcal{C}_n^c, \varphi_i, \mathcal{O})} &= \frac{n \cdot (n^2 + c^2 - 2nc - n + 2c)}{(n-c) \cdot (n-1) \cdot (n-c-1)}
\end{aligned}$$

Hence

$$\mathsf{PO}_r^A(\mathcal{C}_n^c, \varphi_i, \mathcal{O}) = \frac{(n-c) \cdot (n-1) \cdot (n-c-1)}{n \cdot (n^2 + c^2 - 2nc - n + 2c)}$$

which tends to 1 as n increases to $+\infty$ (for a fixed number of corrupted users). The evolution of RPO is represented in Fig. 16(a) where blue means low and red means high. If the proportion of corrupted users is fixed, say $n = 4c$, we obtain

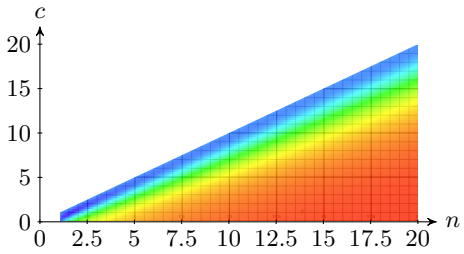
$$\mathsf{PO}_r^A(\mathcal{C}_{4c}^c, \varphi_i, \mathcal{O}) = \frac{(4c-1) \cdot (9c-3)}{4c \cdot (9c-2)}$$

[4] This stems from the fact that the original models had either the server or the corrupt users as attackers, not both at the same time.

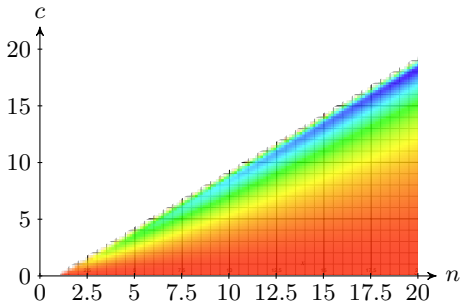
which also tends to 1 as the crowds size increases. When there are no corrupted users,

$$\mathsf{PO}_r^A(\mathcal{C}_n^0, \varphi_i, \mathcal{O}) = \frac{n-1}{n},$$

which is close to 1, but never exactly, since φ_i is not always false, although of decreasing proportion as the crowds grows. This result has to be put in parallel with the one from [5], which states that crowds is secure since each user is "beyond suspicion" of being the initiator, but "absolute privacy" is not achieved.



(a) Evolution of $\mathsf{PO}_r^A(\mathcal{C}_n^c, \varphi_i, \mathcal{O})$ with n and c.



(b) Evolution of $\mathsf{PO}_i^S(\mathcal{C}_n^c, \varphi_i, \mathcal{O})$ with n and c.

Fig. 16.

Computation of RPSO. From the probabilities computed above, we obtain that if $i \neq j$,

$$V(i \rightsquigarrow \mid \rightsquigarrow j) = \max\left(\frac{1}{n}, \frac{n-1}{n}\right) = \frac{1}{n}\max(1, n-1).$$

Except in the case of $n = 1$ (when the system is non-opaque, hence $\mathsf{PO}_r^S(\mathcal{C}_1^0, \varphi_1, \mathcal{O}) = 0$), $V(i \rightsquigarrow \mid \rightsquigarrow j) = \frac{n-1}{n}$.

In the case when $i = j$

$$V(i \rightsquigarrow \mid \rightsquigarrow i) = \max\left(\frac{c+1}{n}, \frac{n-c-1}{n}\right) = \frac{1}{n}\max(c+1, n-c-1).$$

That means the vulnerability for the observation class corresponding to the case when i is actually detected depends on the proportion of corrupted users in the crowd. Indeed, $V(i \rightsquigarrow \mid \rightsquigarrow i) = \frac{c+1}{n}$ if and only if $c \geq \frac{n}{2}$. The two cases shall be separated.

When $c \geq \frac{n}{2}$. There are more corrupted than honest users:

$$\begin{aligned}\sum_{j=1}^{n-c} \mathbf{P}(\rightsquigarrow j) \cdot \log(V(i \rightsquigarrow \mid \rightsquigarrow j)) &= \frac{1}{n-c} \cdot \left((n-c-1) \cdot \log\left(\frac{1}{n}\right) + \log\left(\frac{n-c-1}{n}\right)\right) \\ &= \frac{1}{n-c} \cdot (\log(n-c-1) - (n-c) \cdot \log(n)) \\ &= \frac{\log(n-c-1)}{n-c} - \log(n)\end{aligned}$$

$$\text{Hence} \quad \mathsf{PO}_r^S(\mathcal{C}_n^c, \varphi_i, \mathcal{O}) \;=\; \frac{1}{\log(n) - \frac{\log(n-c-1)}{n-c}}$$

When $c < \frac{n}{2}$. There are more honest than corrupted users:

$$\begin{aligned}\sum_{j=1}^{n-c} \mathbf{P}(\rightsquigarrow j) \cdot \log(V(i \rightsquigarrow \mid \rightsquigarrow j)) &= \frac{1}{n-c} \cdot \left((n-c-1) \cdot \log\left(\frac{1}{n}\right) + \log\left(\frac{c+1}{n}\right)\right) \\ &= \frac{1}{n-c} \cdot (\log(c+1) - (n-c) \cdot \log(n)) \\ &= \frac{\log(c+1)}{n-c} - \log(n)\end{aligned}$$

$$\text{Hence} \quad \mathsf{PO}_r^S(\mathcal{C}_n^c, \varphi_i, \mathcal{O}) \;=\; \frac{1}{\log(n) - \frac{\log(c+1)}{n-c}}$$

The evolution of RPSO for $c = 5$ is depicted in Fig. 17.

One can see that actually the RPSO decreases when n increases. That is because when there are more users in the crowd, user i is less likely to be the initiator. Hence the predicate chosen does not model anonymity as specified in [5] but a stronger property since RPSO is based on the definition of symmetrical opacity. Therefore it is meaningful in terms of security properties only when both the predicate and its negation are meaningful.

Computation of IPSO. We compute IPSO (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

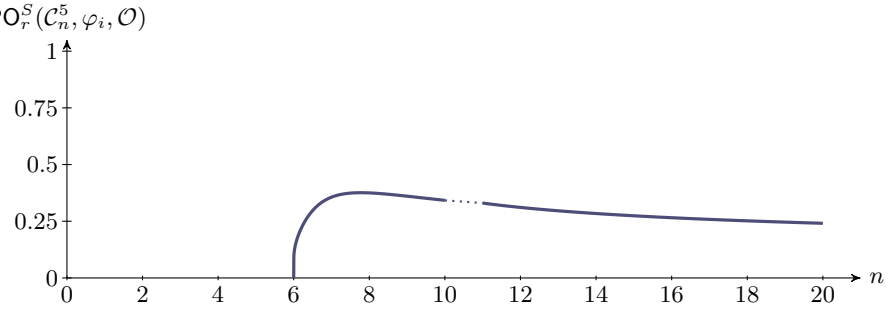


Fig. 17. Evolution of $\mathsf{PO}_r^S(\mathcal{C}_n^c, \varphi_i, \mathcal{O})$ with n when $c = 5$.

observation function of the penultimate state of the run:

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

On the other hand

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

Hence

$$\begin{aligned}
\mathsf{PO}_i^S(\mathcal{C}_n^c, \varphi_i, \mathcal{O}) &= 1 + \log(n) - \log(n-c) + \frac{n-c-1}{n-c} \cdot \log(n-c-1) \\
&\quad - \frac{1}{n-c} \cdot \Bigg(\frac{(n-c-1)\cdot(n-1)}{n} \cdot \log(n-1) \\
&\qquad + \frac{n-c-1}{n} \cdot \log(n-c-1) + \frac{c+1}{n} \cdot \log(c+1) \Bigg)
\end{aligned}$$

Remark that in the case where there is no corrupt user (*i.e.* when $c = 0$), we obtain $\mathsf{PO}_i^S(\mathcal{C}_n^0, \varphi_i, \mathcal{O}) = 1$, thus re-confirming in passing the result from [5] 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. The evolution of IPSO according to n and c for this protocol is shown on Fig. 16(b), where blue means low (near 0) and red means high (near 1).

7.4 Discussion

As a result, LPO and RPO (and their symmetrical counterparts) provide measures tightly attached to the notions of opacity. The measure IPSO, however, provides a measure of how much is leaked through the observation. Hence it is a good measure to assess the power of observation in terms of security. It does not however provide information as to whether the secrecy of predicate φ is protected even without observation. Indeed, the information φ leaks by itself, as measured by $H(\mathbf{1}_\varphi)$, is leveled out in IPSO. In the extreme case where $\varphi = Run(\Pi)$, φ is not secret (since it is always true), but no information is gained through observation.

Such imbalance in the global repartition of φ can be measured by minPO. Vulnerability, as discussed in [8], evaluates the probability of a correct guess in one attempt. It can also be used to assess how much the observation renders the system more vulnerable. However, any comparison of measure before and after observation will suffer from the same drawbacks when φ is initially more likely to be true than false, especially if this imbalance is respected by each observation class.

8 Dealing with nondeterminism

The measures presented above were all defined in the case of fully probabilistic automata. However, some systems present nondeterminism that cannot reasonably be abstracted away. For example, consider the case of a system, in which a malicious user Alice can control certain actions. The goal of Alice is to establish a covert communication channel with an external observer Bob. Hence she will try to influence the system in order to render communication easier. Therefore, the actual security of the system as observed by Bob should be measured against the best possible actions for Alice. Formally, Alice is a scheduler who, when facing several possible output distributions $\{\mu_1, \ldots, \mu_n\}$, can choose whichever distribution ν on $\{1, \ldots, n\}$ as weights for the μ_is. The security as measured by opacity is the minimal security of all possible successive choices.

8.1 The nondeterministic framework

Here we enlarge the setting of probabilistic automata considered before by allowing nondeterminism. Instead of having a single outgoing distribution from a given state, the model allows several.

Definition 11 (Nondeterministic probabilistic automaton). *A* nondeterministic probabilistic automaton *(NPA) 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,*
- $\Delta : Q \to \mathcal{P}(\mathcal{D}((\Sigma \times Q) \uplus \{\surd\}))$ *is a nondeterministic probabilistic transition function.*

– q_0 *is the initial state.*

The choice over the several possible distributions is made by the *scheduler*. It does not, however, selects one distribution to be used, but can give weight to the possible distributions.

Definition 12 (Scheduler). *A scheduler on* $\Pi = \langle \Sigma, Q, \Delta, q_0 \rangle$ *is a function*

$$\sigma : Run(\Pi) \to \mathcal{D}(\mathcal{D}((\Sigma \times Q) \uplus \{\surd\}))$$

such that $\sigma(\rho)(\nu) > 0 \Rightarrow \nu \in \Delta(lst(\rho))$.

The set of all schedulers is for Π *is denoted Sched (the dependence on* Π *is implicit).*

Observe that the choice made by a scheduler can depend on the (arbitrary long) history of the execution. A scheduler σ is *memoryless* if there exists a function $\sigma' : Q \to \mathcal{D}(\mathcal{D}((\Sigma \times Q) \uplus \{\surd\}))$ such that $\sigma(\rho) = \sigma'(\mathrm{lst}(\rho))$. Hence a memoryless scheduler takes only into account the current state.

Definition 13 (Scheduled NPA). *NPA* $\Pi = \langle \Sigma, Q, \Delta, q_0 \rangle$ *scheduled by* σ *is the (infinite) FPA* $\Pi_{/\sigma} = \langle \Sigma, Run(\Pi), \Delta', \varepsilon \rangle$ *where*

$$\Delta'(\rho)(a, \rho') = \sum_{\mu \in \Delta} \sigma(\rho)(\mu) \cdot \mu(a, q) \quad \text{if } \rho' = \rho \xrightarrow{a} q \quad \text{and} \quad \Delta'(\rho)(a, \rho') = 0 \quad \text{otherwise.}$$

A scheduled NPA behaves as an FPA, where the outgoing distribution is the set of all possible distributions weighted by the scheduler.

All measures defined in this paper on fully probabilistic automata can be extended to non-deterministic probabilistic automata. First note that all measures can be defined on infinite systems, although they cannot in general be computed automatically, even with proper restrictions on predicate and observables.

Definition 14. *Let* Π *be an NPA,* φ *a predicate, and* $\mathcal{O}$ *an observation function.*

$$\text{For } PO^* \in \{PO^A_\ell, PO^S_\ell\}, \quad \widehat{PO^*}(\Pi, \varphi, \mathcal{O}) = \max_{\sigma \in Sched} PO^*(\Pi_{/\sigma}, \varphi, \mathcal{O}).$$

$$\text{For } PO^* \in \{PO^A_r, PO^S_r, PO^S_i, PO^S_{min}\}, \quad \widehat{PO^*}(\Pi, \varphi, \mathcal{O}) = \min_{\sigma \in Sched} PO^*(\Pi_{/\sigma}, \varphi, \mathcal{O}).$$

8.2 The expressive power of schedulers

In general, schedulers have arbitrary power, as stated in Definition 12. An infinite memory yields untractable models. However, state-based schedulers are not sufficiently expressive, as shown by the following counterexample.

Consider the NPA $\mathcal{B}$ of Fig. 18. Transitions on a and b going to state q_1 (along with the westbound $\surd$) are part of the same probabilistic transition indicated by the arc linking the outgoing edges, and so are the a and b going to state q_2 (along with the eastbound $\surd$). Let φ be the (regular) predicate consisting of runs

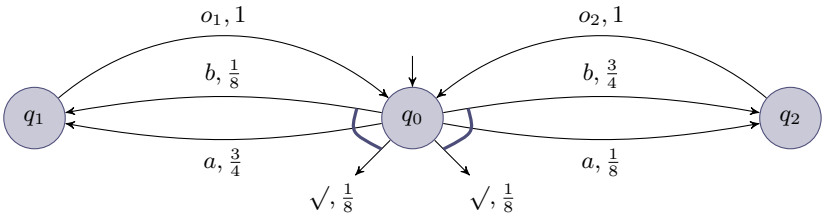


Fig. 18. A nondeterministic probabilistic automaton $\mathcal{B}$

whose trace projected onto $\{a, b\}$ is in $(ab)^+ + (ab)^*a$. Let $\mathcal{O}$ be the observation function that keeps the last o_i of the run. Hence there are only three observables, ε, o_1, and o_2. Intuitively, a scheduler can introduce a bias in the next letter read from state q_0.

Let us first consider a memoryless scheduler σ_p. This scheduler can only choose once what weight will be affected to each transition. This choice is parametrized by probability p that represents the weight of probability of the q_1 transition. The scheduled NPA is $\mathcal{B}_{/\sigma_p}$ is depicted on Fig. 19. The probabilities can be computed using the technique laid out in Section 6. One obtain the following probabilities:

$$\mathbf{P}(\varepsilon) = \frac{1}{8} \qquad \mathbf{P}(o_1) = \frac{7}{8} \cdot p \qquad \mathbf{P}(o_2) = \frac{7}{8} \cdot (1-p) \qquad \mathbf{P}(\varphi, \varepsilon) = 0$$

$$\mathbf{P}(\varphi, o_1) = \frac{p}{25p^2 - 25p + 58} \cdot \frac{5p + 49}{8} \qquad \mathbf{P}(\varphi, o_2) = \frac{1-p}{25p^2 - 25p + 58} \cdot \frac{5p + 7}{4}$$

Which yields

$$\frac{1}{\mathsf{PO}_r^A(\mathcal{B}_{/\sigma_p}, \varphi, \mathcal{O})} = \frac{1}{8} + \frac{49}{8} \cdot f(p) \cdot \left(\frac{p}{7f(p) - 5p - 49} + \frac{1-p}{7f(p) - 10p - 14} \right)$$

with $f(p) = 25p^2 - 25p + 58$. It can be shown that regardless of p, $\mathsf{PO}_r^A(\mathcal{B}_{/\sigma_p}, \varphi, \mathcal{O})$ never falls behind $\frac{1408}{1597} \simeq 0.88$.

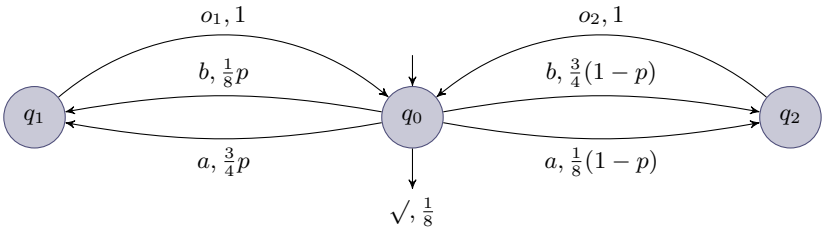


Fig. 19. Fully probabilistic automaton $\mathcal{B}_{/\sigma_p}$

Now consider a scheduler σ_m with memory who will try to maximize the realization of φ. In order to achieve that, it introduces a bias towards taking the letter which will fulfill φ: first an a, then a b, etc. Hence on the even positions, it will choose only transition to q_1 (with probability 1) while it will choose the transition to q_2 on odd positions. The resulting FPA is depicted on Fig. 20. In this case, the probabilities of interest are:

$$\mathbf{P}(\varepsilon) = \frac{1}{8} \qquad \mathbf{P}(o_1) = \frac{7}{15} \qquad \mathbf{P}(o_2) = \frac{7}{8} \cdot \frac{7}{15}$$

$$\mathbf{P}(\varphi, \varepsilon) = 0 \qquad \mathbf{P}(\varphi, o_1) = \frac{3}{10} \qquad \mathbf{P}(\varphi, o_2) = \frac{3}{4} \cdot \frac{3}{10}$$

Therefore $\mathsf{PO}_r^A(\mathcal{B}_{/\sigma_m}, \varphi, \mathcal{O}) = \frac{4400}{10301} \simeq 0.42$.

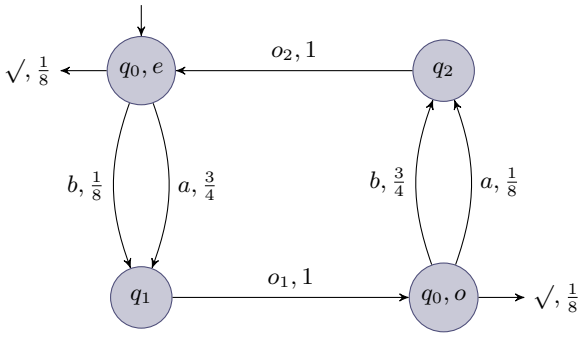


Fig. 20. Fully probabilistic automaton $\mathcal{B}_{/\sigma_m}$

Hence a lower security is achieved by a scheduler provided it has (a finite amount of) memory. Note that this example used RPO, but a similar argument could be adapted for the other measures.

8.3 Restricted schedulers

What made a scheduler with memory more powerful than the one without in the counterexample of Section 8.2 was the knowledge of the truth value of φ and exactly what was observed. More precisely, if the predicate and the observables are regular languages represented by finite deterministic and complete automata (FDCA), schedulers can be restricted to choices according to the current state of these automata and the state of the system. We conjecture that this knowledge is sufficient to any scheduler to compromise security at the best of its capabilities.

Let $\varphi \subseteq Crun(\Pi)$ be a regular predicate represented by an FDCA $\mathcal{A}_\varphi$. Let $\mathcal{O} : CRun(\Pi) \to \{o_1, \ldots, o_n\}$ be an observation function such that for $1 \leq i \leq n$, $\mathcal{O}^{-1}(o_i)$ is a regular set represented by an FDCA $\mathcal{A}_{o_i}$. Consider the synchronized

product $\mathcal{A}_{\varphi,\mathcal{O}} = \mathcal{A}_\varphi || \mathcal{A}_{o_1} || \dots || \mathcal{A}_{o_n}$, which is also an FDCA, and denote by $Q_{\varphi,\mathcal{O}}$ its set of states. Let $\mathcal{A}_{\varphi,\mathcal{O}}(\rho)$ be the state of $\mathcal{A}_{\varphi,\mathcal{O}}$ reached after reading ρ.

Definition 15 (Restricted $(\varphi,\mathcal{O})$-scheduler). *A scheduler σ for Π is said $(\varphi,\mathcal{O})$-restricted if there exist a function $\sigma' : (Q_{\varphi,\mathcal{O}} \times Q) \to \mathcal{D}(\mathcal{D}((\Sigma \times Q) \uplus \{\surd\}))$ such that for any run $\rho \in Run(\Pi)$, $\sigma(\rho) = \sigma'(\mathcal{A}_{\varphi,\mathcal{O}}(\rho), lst(\rho))$.*

The set of $(\varphi,\mathcal{O})$-restricted schedulers is denoted $Sched_{\varphi,\mathcal{O}}$.

Remark that memoryless schedulers are always $(\varphi,\mathcal{O})$-restricted.

Proposition 6. *If σ is $(\varphi,\mathcal{O})$-restricted, then $\Pi_{/\sigma}$ is isomorphic to a finite FPA.*

Conjecture 1. Let Π be an NPA, φ a predicate, and $\mathcal{O}$ an observation function.

$$\text{For } \mathsf{PO}^* \in \{\mathsf{PO}^A_\ell, \mathsf{PO}^S_\ell\}, \quad \arg\max_{\sigma \in Sched} \mathsf{PO}^*(\Pi_{/\sigma}, \varphi, \mathcal{O}) \in Sched_{\varphi,\mathcal{O}}.$$

$$\text{For } \mathsf{PO}^* \in \{\mathsf{PO}^A_r, \mathsf{PO}^S_r, \mathsf{PO}^S_i, \mathsf{PO}^S_{\min}\}, \quad \arg\min_{\sigma \in Sched} \mathsf{PO}^*(\Pi_{/\sigma}, \varphi, \mathcal{O}) \in Sched_{\varphi,\mathcal{O}}.$$

9 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. These notions yield measures that generalize either the case of asymmetrical or symmetrical opacity, thus providing four measures. Finally, we define measures of opacity linked with information theory, that measure how much is given away through observation, although these are less tightly tied to the classical notion of symmetrical opacity.

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. A prototype tool was implemented in Java, and allows numerical computation of the values of opacity.

In future work we plan to 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.

References

1. Mazaré, L.: 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., Springer (2005) 71–84

2. Bryans, J.W., Koutny, M., Mazaré, L., Ryan, P.Y.A.: Opacity generalised to transition systems. Intl. Jour. of Information Security **7**(6) (2008) 421–435
3. Alur, R., Černý, P., Zdancewic, S.: Preserving secrecy under refinement. In: Proc. of the 33rd Intl. Colloquium on Automata, Languages and Programming (ICALP'06). Volume 4052 of LNCS., Springer (2006) 107–118
4. Chaum, D.: The dining cryptographers problem: unconditional sender and recipient untraceability. Journal of Cryptology **1** (1988) 65–75
5. Reiter, M.K., Rubin, A.D.: Crowds: anonymity for web transactions. ACM Transactions on Information and System Security **1**(1) (1998) 66–92
6. Aldini, A., Bravetti, M., Gorrieri, R.: A process-algebraic approach for the analysis of probabilistic noninterference. Journal of Computer Security **12**(2) (2004) 191–245
7. Boreale, M.: Quantifying information leakage in process calculi. Information and Computation **207**(6) (2009) 699–725
8. Smith, G.: On the foundations of quantitative information flow. In: Proc. 12th Intl. Conf. on Foundations of Software Science and Computational Structures (FOSSACS '09), Springer-Verlag (2009) 288–302
9. Chatzikokolakis, K., Palamidessi, C., Panangaden, P.: Anonymity protocols as noisy channels. Information and Computation **206**(2-4) (February 2008) 378–401
10. Andrés, M.E., Palamidessi, C., van Rossum, P., Smith, G.: 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., Springer (March 2010) 373–389
11. Alvim, M.S., Andrés, M.E., Palamidessi, C.: Information flow in interactive systems. In Gastin, P., Laroussinie, F., eds.: Proceedings of the 21th International Conference on Concurrency Theory (CONCUR'10). Volume 6269 of LNCS., Springer (aug 2010) 102–116
12. Lakhnech, Y., Mazaré, L.: Probabilistic opacity for a passive adversary and its application to Chaum's voting scheme. Technical Report 4, Verimag (2 2005)
13. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience (July 2006)
14. Goguen, J.A., Meseguer, J.: Security policy and security models. In: Proc. of IEEE Symposium on Security and Privacy, IEEE Computer Society Press (1982) 11–20
15. Courcoubetis, C., Yannakakis, M.: Markov Decision Processes and Regular Events. IEEE Transactions on Automatc Control **43**(10) (1998) 1399–1418
16. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing **6**(5) (1994) 512–535
17. Bianco, A., de Alfaro, L.: 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., Springer (1995) 499–513