Probabilistic Opacity for Markov Decision Processes

Béatrice Bérard[a,b], Krishnendu Chatterjee[c], Nathalie Sznajder[a,b]

[a] *Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France*
[b] *CNRS, UMR 7606, LIP6, F-75005, Paris, France*
[c] *IST Austria (Institute of Science and Technology, Austria)*

Abstract

Opacity is a generic security property, that has been defined on (non probabilistic) transition systems and later on Markov chains with labels. For a secret predicate, given as a subset of runs, and a function describing the view of an external observer, the value of interest for opacity is a measure of the set of runs disclosing the secret. We extend this definition to the richer framework of Markov decision processes, where non deterministic choice is combined with probabilistic transitions, and we study related decidability problems with partial or complete observation hypotheses for the schedulers. We prove that all questions are decidable with complete observation and ω-regular secrets. With partial observation, we prove that all quantitative questions are undecidable but the question whether a system is almost surely non opaque becomes decidable for a restricted class of ω-regular secrets, as well as for all ω-regular secrets under finite-memory schedulers.

1. Introduction

Due to the tremendous increase in network communications in the last thirty years, a large amount of work was devoted to the study of security properties, to ensure the preservation of secret data during these communications. *Information flow* characterizes the (possibly illegal and indirect) transmission of such data from a high level user to a low level one. Already in the eighties, a basic version of non-interference was defined in [20], stating that a system is secure if high level actions cannot be detected by low level observations. Among all the subsequent studies, opacity was introduced in [24, 7] as a general framework where a wide range of security properties can be specified, for a system interacting with a passive attacker. For a system $\mathcal{S}$, opacity is parameterized by a secret predicate φ described as a subset of executions and an observation function over executions. The system is opaque if, for any secret run in φ, there is another run not in φ with the same observation. When this property is satisfied, the passive attacker cannot learn from the observation if the execution is secret. Ensuring opacity by controller synthesis was further studied in [18, 9] while relations with two-player games were established in [23].

Deciding opacity, however, only provides a yes/no answer, but no evaluation of the amount of information gained by a passive attacker. Since more and more security protocols make use of randomization to reach some security objectives [16, 29], it becomes important to extend specification frameworks in order to handle measures of information leaks. For this reason, quantitative approaches for security properties were already advocated in [25, 34], mostly based on information theory. From this point on, numerous studies were devoted to the computation of (covert) channel capacity in various cases (see e.g. [22]) or more generally information leakage.

To provide quantitative measures of opacity, several definitions have been proposed in a probabilistic setting [21, 2, 5, 8, 3, 31]. They were, however, restricted to purely probabilistic models, based on Markov chains equipped with labels, to permit observations on runs. We show here how to extend some measures of [3] to Markov decision processes (MDPs) with infinite runs. The simplest one computes what we call here the *probabilistic disclosure*, providing a probabilistic measure for the set of runs whose observation reveals that a secret run has been executed. With the richer model of MDPs, where non determinism is combined with probabilities, a scheduler can cooperate with the passive external observer to break the system opacity. We focus

on ω-regular secrets and morphisms for the observation functions, and prove that the probabilistic disclosure can be computed when the scheduler can distinguish the states of the model. The class of ω-regular languages provides a robust specification language [32], extending classical regular languages from finite words to infinite words. Such ω-regular languages are often needed to express opacity in the non probabilistic as well as the probabilistic setting. With partial observation for the schedulers, the question whether a system is almost surely non opaque remains decidable for a restricted class of ω-regular secrets, as well as for all ω-regular secrets under finite-memory schedulers, whereas all quantitative problems become undecidable. Moreover, for all decidable results we present optimal complexity results: for complete observation (where the scheduler can distinguish states of the model) we present polynomial-time results with respect to the size of the model, whereas for partial observation, for all decidable results we show EXPTIME-completeness.

We recall some definitions for probabilistic models in Section 2. Opacity and disclosure are defined for Markov decision processes in Section 3 and proofs for the (un)decidability results are given in Section 4. We conclude in Section 5.

2. Preliminaries

For a finite alphabet Z, we denote by Z^* the set of finite words over Z, by Z^ω the set of infinite words over Z, with $Z^\infty = Z^* \cup Z^\omega$.

We first recall some classical notions on automata.

2.1. Automata

Definition 1. A (deterministic) automaton is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an input alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function, $q_0 \in Q$ is the initial state, and F is either a subset of Q, or a mapping from Q to a finite subset of natural numbers.

Accepting conditions defined from F will be described hereafter.

A run of the automaton $\mathcal{A}$ on a word $w = a_1 a_2 \cdots \in \Sigma^\omega$ is an infinite sequence $\rho = q_0 q_1 \cdots$ such that for all $i \geq 0$, $q_{i+1} = \delta(q_i, a_{i+1})$. The accepting runs of an automaton are defined according to the acceptance condition. In the sequel, we consider Büchi, co-Büchi and parity acceptance conditions.

For a run $\rho = q_0 q_1 \cdots \in Q^\omega$, we let $\mathrm{Inf}(\rho)$ be the set of states appearing infinitely often in the sequence. When $F \subseteq Q$, we note $\text{Büchi}(F) = \{\rho \in Q^\omega \mid \mathrm{Inf}(\rho) \cap F \neq \emptyset\}$ and $\text{co-Büchi}(F) = \{\rho \in Q^\omega \mid \mathrm{Inf}(\rho) \cap F = \emptyset\}$. When $F : Q \to \{1, \ldots, k\}$, with $k \in \mathbb{N}$, the acceptance condition is a parity condition. We note $\text{Parity}(F) = \{\rho \in Q^\omega \mid \min\{F(q) \mid q \in \mathrm{Inf}(\rho)\} \text{ is even}\}$. For an acceptance condition $Acc \in \{\text{Büchi}(F), \text{co-Büchi}(F), \text{Parity}(F)\}$, we say that a run ρ over a word w is accepting if it is in Acc. The word w is then said to be accepted by ρ.

We denote respectively by $L_B(\mathcal{A})$, $L_C(\mathcal{A})$ and $L_P(\mathcal{A})$ the set of words accepted by the runs of $\mathcal{A}$ in Büchi(F), co-Büchi(F) and Parity(F). A subset L of Σ^ω is ω-regular if there is an automaton $\mathcal{A}$ such that $L = L_P(\mathcal{A})$.

In the sequel, we write DBA for deterministic Büchi automata, DCA for deterministic co-Büchi automata and DPA for deterministic parity automata, according to the choice of acceptance condition.

2.2. Probabilistic systems

We consider systems modeled by Markov decision processes, that generalize Markov chains by combining non deterministic actions with probabilistic transitions. To define opacity measures on Markov chains, the probabilistic transitions are equipped with labels that may be used to define an observation function on runs. In the setting of Markov decision processes, labels are also added on the probabilistic transitions. They may be observed by a passive attacker while non deterministic actions are chosen by a scheduler, as explained below.

Given a countable set S, a discrete distribution is a mapping $\mu : S \to [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$. The set of all discrete distributions on S is denoted by $\mathcal{D}(S)$.

Definition 2 (Markov Decision Process). A Markov decision process (MDP) is a tuple $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$ where:

- Q is a finite set of states,
- Act is a finite set of actions,
- Σ is a finite alphabet for the labeling of transitions,
- $\Delta : Q \times Act \to \mathcal{D}(\Sigma \times Q)$ is a (partial) transition function that associates with a state and an action from Act a probability distribution

over the possible transition labels and successor states,

- q_0 is the initial state.

Figure 1 shows an MDP with four actions. Actions α_1 and α_2 bear two different distributions for labels a and b. They start either from state q_0 or from state q_0', and lead to either q_1 or q_2. Actions β_1 and β_2 start from q_1 and q_2 respectively and return to q_0 or q_0' with probability $\frac{1}{2}$.

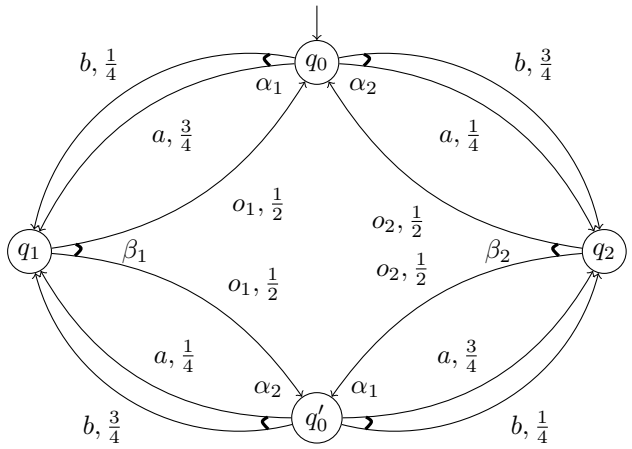

Figure 1: A Markov Decision process.

The definition could be extended with an initial distribution instead of an initial state, but we restrict to this one for the sake of simplicity. When $\Delta(q, \alpha)$ is defined, α is said to be *enabled* in state q. Intuitively, in an execution of an MDP, from a given state q, an action $\alpha \in Act$ enabled in q is chosen non deterministically, and then the next label in Σ and the next state are chosen according to the probability distribution $\Delta(q, \alpha)$. Formally, a (finite or infinite) run of an MDP is a sequence $\rho = q_0 \cdot (\alpha_0, a_0) \cdot q_1 \cdot (\alpha_1, a_1) \cdot q_2 \cdot \ldots \in Q \cdot ((Act \times \Sigma) \cdot Q)^\infty$, also written $q_0 \xrightarrow{\alpha_0, a_0} q_1 \xrightarrow{\alpha_1, a_1} q_2 \ldots$ such that, for all $i \geq 0$, α_i is enabled in q_i and $\Delta(q_i, \alpha_i)(a_i, q_{i+1}) > 0$. The trace of ρ is the word $(\alpha_0, a_0)(\alpha_1, a_1) \ldots$ over $Act \times \Sigma$ labelling the run, obtained by projecting away the visited states. The length of ρ, denoted by $|\rho|$, is the length of its trace in $\mathbb{N} \cup \{\infty\}$. The set of infinite (resp. finite) runs of an MDP $\mathcal{A}$ is denoted by $Runs(\mathcal{A})$ (resp. $Runs_f(\mathcal{A})$). The set of traces of infinite runs of $\mathcal{A}$ is denoted by $T(\mathcal{A})$ and $tr : Runs(\mathcal{A}) \to T(\mathcal{A})$ is the mapping that associates with each run its trace. For a run ρ, and $i < |\rho|$, we denote by ρ_i the finite run consisting of its first i transitions, and we say that ρ_i is a prefix of ρ.

The non determinism of MDPs is resolved by a scheduler, that gives a probability distribution over the different actions in Act along each finite run.

Definition 3 (Scheduler). A *scheduler* on $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$ is a function $\sigma : Runs_f(\mathcal{A}) \to \mathcal{D}(Act)$ such that, for any finite run $\rho = q_0 \xrightarrow{\alpha_0, a_0} \ldots \xrightarrow{\alpha_{n-1}, a_{n-1}} q_n$ of $\mathcal{A}$, for all $\alpha \in Act$, if $\sigma(\rho)(\alpha) > 0$ then α is enabled in q_n.

A scheduler is *deterministic* if $\sigma : Runs_f(\mathcal{A}) \to Act$. We say that a scheduler has *finite memory* if its decision only depends on a finite set of so-called *memory states*. Similarly, a scheduler is *memoryless* if its decision depends only on the last state of the run. Formally, they are defined as follows.

Definition 4 (Finite-Memory Schedulers). A *finite-memory scheduler* on $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$ is given by a tuple $(M, m_0, \sigma, \sigma_{up})$ where M is a finite set of memory states, m_0 is the initial memory state, $\sigma : M \times Q \to \mathcal{D}(Act)$ is a mapping such that, for all $m \in M$, for all $q \in Q$, and for all $\alpha \in Act$, if $\sigma(m, q)(\alpha) > 0$ then α is enabled in q. Finally, $\sigma_{up} : M \times Act \times \Sigma \times Q \to \mathcal{D}(M)$ is the memory update function.

If $|M| = 1$, then the scheduler, reduced to $\sigma : Q \to \mathcal{D}(Act)$ is *memoryless*.

In some systems, the underlying state is only partially observable. Those are modeled by Partially Observable MDPs, in which some sets of states are undistinguishable for external observers (including the scheduler):

Definition 5. A partially observable Markov decision process (POMDP) is an MDP $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$ equipped with an equivalence relation $\sim$ over Q such that if $p \sim q$ then the set of actions from Act enabled in p and q are the same.

In that case, given two sequences of states $p_0 \cdots p_n$ and $q_0 \cdots q_n$, we say that $p_0 \cdots p_n \sim q_0 \cdots q_n$ if and only if $p_i \sim q_i$ for all $0 \leq i \leq n$. In a POMDP, the scheduler cannot distinguish between equivalent states. The scheduler definition is then modified:

Definition 6. Let $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$ be a POMDP with equivalence relation $\sim \subseteq Q \times Q$. An observation-based scheduler (or $\sim$-scheduler) is a scheduler $\sigma : Runs_f(\mathcal{A}) \to \mathcal{D}(Act)$ such that for any

two finite runs $\rho = q_0 \xrightarrow{\alpha_0,a_0} \dots \xrightarrow{\alpha_{n-1},a_{n-1}} q_n$ and $\rho' = p_0 \xrightarrow{\alpha_0,b_0} \dots \xrightarrow{\alpha_{n-1},b_{n-1}} p_n$ with same length, if $p_0 \cdots p_n \sim q_0 \cdots q_n$, then $\sigma(\rho) = \sigma(\rho')$.

For instance, associating with the MDP of Figure 1 the three equivalence classes $\{q_0, q_0'\}$, $\{q_1\}$ and $\{q_2\}$, produces a POMDP. In this case, the scheduler cannot know if it is in q_0 or in q_0' when it chooses action α_1 or α_2.

Recall that, given a POMDP $\mathcal{A}$ and a scheduler σ, a probability measure $\mathbf{P}_\sigma$ can be defined on $Runs(\mathcal{A})$[4]: first it is defined on *cones*, where the cone C_ρ associated with a finite run ρ is the subset of infinite runs in $Runs(\mathcal{A})$ having ρ as prefix; and then it is extended to measurable sets of infinite runs. If $\rho = q_0 \xrightarrow{\alpha_0,a_0} \dots \xrightarrow{\alpha_{n-1},a_{n-1}} q_n$, the probability of C_ρ is defined by:

$$\mathbf{P}_\sigma(C_\rho) = \sigma(\rho_0)(\alpha_0) \times \Delta(q_0, \alpha_0)(a_0, q_1) \times \dots \times \sigma(\rho_{n-1})(\alpha_{n-1}) \times \Delta(q_{n-1}, \alpha_{n-1})(a_{n-1}, q_n)$$

3. Opacity and disclosure

The notion of opacity was originally defined in [7] for a (non probabilistic) transition system, with respect to some external observation function and some predicate (the secret) on the runs of the system. It extends trivially to probabilistic transition systems. In this case, given an MDP $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$, we consider a predicate $\varphi \subseteq Runs(\mathcal{A})$, given as an ω-regular language (the secret). The observation the attacker has of the runs of the MDP is defined by a morphism $\mathcal{O} : Runs(\mathcal{A}) \to \Gamma^\infty$ obtained from a given application $\pi : Q \cup (Act \times \Sigma) \to \Gamma \cup \{\varepsilon\}$, where Γ is a finite alphabet. The morphism $\mathcal{O}$ is the observation function, and the elements of $Obs = \mathcal{O}(Runs(\mathcal{A}))$ are the observables. For a given run ρ, every run in $\mathcal{O}^{-1}(\mathcal{O}(\rho))$ – its observation class – is undistinguishable from ρ. The predicate is *opaque* in $\mathcal{A}$ for $\mathcal{O}$ if each time a run satisfies the predicate, another run in the same observation class does not. Formally, we let $\overline{\varphi} = Runs(\mathcal{A}) \setminus \varphi$, and define opacity as follows.

Definition 7 (Opacity). Let $\mathcal{A}$ be an MDP, with observation function $\mathcal{O} : Runs(\mathcal{A}) \to Obs$. A predicate $\varphi \subseteq Runs(\mathcal{A})$ is said to be *opaque* in $\mathcal{A}$ for $\mathcal{O}$ if $\varphi \subseteq \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$.

Variants of opacity have been defined, by modifying the observation function or the predicate, or by requiring symmetry: the predicate φ is *symmetrically opaque* in $\mathcal{A}$ for $\mathcal{O}$ if both φ and $\overline{\varphi}$ are opaque.

Note that φ is opaque if and only if for any $o \in Obs$, $\mathcal{O}^{-1}(o) \not\subseteq \varphi$. By extension, we say that an observation class $\mathcal{O}^{-1}(o)$, for $o \in Obs$, is itself *opaque* if $\mathcal{O}^{-1}(o) \not\subseteq \varphi$, and we define Obs_{opaque} as the set of corresponding observations, with $Obs_{leak} = Obs \setminus Obs_{opaque} = \{o \in Obs \mid \mathcal{O}^{-1}(o) \subseteq \varphi\}$.

For instance, for the POMDP in Figure 1 above, we can define:

- an observation function $\mathcal{O}$ from the projection π such that $\pi(q) = \varepsilon$ for any $q \in Q$, $\pi(\alpha, o_1) = o_1$, $\pi(\alpha, o_2) = o_2$ and $\pi(\alpha, a) = \pi(\alpha, b) = \varepsilon$, for any $\alpha \in Act$,
- a predicate φ as the set of all runs with trace in $(ab)^\omega$, where the labels as and bs strictly alternate.

When a probabilistic system is non opaque, we are interested in quantifying the security hole. One of the measures proposed in [3] for Markov chains, is the probability of the set of runs violating opacity. With this measure of non opacity, called here *Probabilistic Disclosure* and extended to MDPs with infinite runs, it becomes possible to compare non opaque systems. The measure, computed in a worst case scenario, corresponds to the maximal probability of disclosure over all possible schedulers. More precisely:

Definition 8 (Probabilistic Disclosure). Let $\mathcal{A}$ be an MDP, with observation function $\mathcal{O} : Runs(\mathcal{A}) \to Obs$, let $\varphi \subseteq Runs(\mathcal{A})$ be a predicate and let σ be a scheduler. The probabilistic disclosure of φ in $\mathcal{A}$ scheduled by σ is:

$$\begin{aligned} \mathrm{PD}_\sigma(\varphi, \mathcal{A}, \mathcal{O}) &= \mathbf{P}_\sigma(\varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))) \\ &= \sum_{o \in Obs_{leak}} \mathbf{P}_\sigma(\mathcal{O}^{-1}(o)). \end{aligned}$$

The probabilistic disclosure of φ in $\mathcal{A}$ is $\mathrm{PD}(\varphi, \mathcal{A}, \mathcal{O}) = \sup_\sigma\{\mathrm{PD}_\sigma(\varphi, \mathcal{A}, \mathcal{O})\}$.

Remark 9. Note that the probabilistic disclosure is well defined, since, when φ is ω-regular, and $\mathcal{O}$ is a morphism as assumed above, the set of runs $\varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$ is measurable. Indeed, the class of ω-regular languages is closed by complement, intersection, morphism and inverse morphism. Hence, the set $\varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$ is ω-regular, thus measurable [33].

Questions we aim to address are the following:

1. The value problem: What is the value of the probabilistic disclosure of the system?
2. The general disclosure problem:
   Is the value of the probabilistic disclosure of the system greater than some given threshold (*i.e.* for $\delta \in [0,1]$, $\mathrm{PD}(\varphi, \mathcal{A}, \mathcal{O}) > \delta$)?
3. The almost-sure opacity problem:
   Is the system almost surely opaque (i.e. $\mathrm{PD}(\varphi, \mathcal{A}, \mathcal{O}) = 0$)?
4. The limit disclosure problem:
   Is $\mathrm{PD}(\varphi, \mathcal{A}, \mathcal{O}) = 1$?
5. The almost-sure disclosure problem: Does there exist a scheduler σ such that $\mathrm{PD}_\sigma(\varphi, \mathcal{A}, \mathcal{O}) = 1$?

All these problems can be considered with a restriction to finite-memory schedulers. The last three questions refer to qualitative aspects of the problem, while the two first ones concern quantitative properties. In the next section, we show that recent results on MDPs (with partial or perfect observation) allow us to answer such questions on probabilistic disclosure of the systems. More precisely, we prove that all these questions are decidable under perfect observation, while they are undecidable under partial observation. However, we identify restrictions that allow to decide the last problem.

4. Results

4.1. MDPs and Schedulers with Perfect Observation

Theorem 10. *Given an MDP $\mathcal{A}$, an ω-regular secret φ given as a DPA (deterministic parity automaton), and observation function $\mathcal{O}$ as a morphism, the value is computable, in polynomial time in the size of $\mathcal{A}$, and exponential in the size of φ.*

Proof. Immediate, since $\varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$ is ω-regular and can be described as a DPA, and from the results of [17, 15, 14] for solving MDPs with parity conditions. □

From this theorem, it follows that:

Corollary 11. *The general disclosure, the limit disclosure problem, and the almost-sure opacity problem are decidable.*

Moreover, since it is sufficient to consider memoryless deterministic schedulers for MDPs with parity conditions [15], $\sup_\sigma\{\mathrm{PD}_\sigma(\varphi, \mathcal{A}, \mathcal{O})\} = 1$ if and only if there exists a memoryless scheduler σ such that $\mathrm{PD}_\sigma(\varphi, \mathcal{A}, \mathcal{O}) = 1$. The following result is then obtained.

Corollary 12. *The almost-sure disclosure problem is decidable.*

Note that this result can be applied to symmetrical opacity. It can also be extended to the case considered in [3] with an observation function $\mathcal{O}$ (not necessarily a morphism) producing a finite number of observation classes such that for each $o \in Obs$, $\mathcal{O}^{-1}(o)$ is ω-regular.

4.2. POMDPs and Observation-based Schedulers

Theorem 13. *Given a POMDP $\mathcal{A}$, and a morphism $\mathcal{O}$ for the observation function,*

1. *the almost-sure disclosure problem is undecidable for secrets given as DCA, DPA.*
2. *the almost-sure opacity problem is undecidable for secrets given as DBA, DPA.*
3. *the limit disclosure problem, the general disclosure problem, and the value problem are undecidable, for secrets given as DBA, DCA, DPA, both with general and finite memory schedulers.*

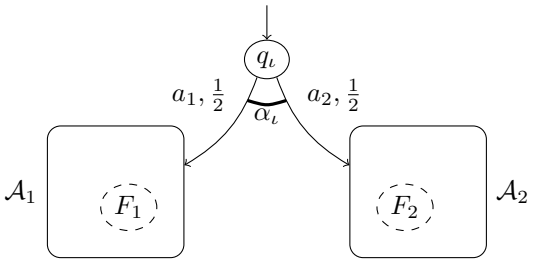

Figure 2: MDP $\mathcal{A}'$ from two copies $\mathcal{A}_1$ and $\mathcal{A}_2$ of $\mathcal{A}$.

Proof. We describe a reduction from qualitative problems on POMDP to the opacity problems addressed in this paper. Let $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$ be a POMDP, with equivalence relation $\sim$ on states. Given a set of accepting states $F \subseteq Q$, we let $Acc(F)$ be either Büchi(F), or co-Büchi(F) (for the underlying non probabilistic runs of $\mathcal{A}$). We build a POMDP $\mathcal{A}' = (Q', Act', \Sigma', \Delta', q_\iota)$, observation function $\mathcal{O} : Runs(\mathcal{A}') \to Obs$, and an ω-regular

secret φ such that schedulers for $\mathcal{A}$ and $\mathcal{A}'$ are in correspondance (explained in more details below).

The POMDP $\mathcal{A}'$ is obtained as follows: we consider two copies $\mathcal{A}_1$ and $\mathcal{A}_2$ of $\mathcal{A}$ with the same alphabets Act and Σ, denoting their disjoint set of states by Q_1 and Q_2, their initial states by q_0^1 and q_0^2 and their target states by F_1 and F_2, respectively. We add a new state q_ι not in $Q_1 \cup Q_2$, a new action α_ι not in Act and two new letters a_1 and a_2 not in Σ, for which the transition function is defined by $\Delta'(q_\iota, \alpha_\iota)(a_1, q_0^1) = \Delta'(q_\iota, \alpha_\iota)(a_2, q_0^2) = 1/2$, as depicted in Figure 2. The equivalence relation on states $\sim'$ is given by $q \sim' q'$ if $q, q' \in Q_i$ and $q \sim_i q'$ for $i = 1, 2$, or $q \in Q_1, q' \in Q_2$ are the copies of the same state in Q. The secret φ is the union of two sets of runs: those starting with $q_\iota \xrightarrow{\alpha_\iota, a_1} q_0^1$ meeting the acceptance condition $Acc(F_1)$ (through $\mathcal{A}_1$) and all the runs starting with $q_\iota \xrightarrow{\alpha_\iota, a_2} q_0^2$ (going into $\mathcal{A}_2$). Formally:

$$\varphi = \big(q_\iota \cdot (\alpha_\iota, a_1) \cdot Acc(F_1)\big) \cup \big(q_\iota \cdot (\alpha_\iota, a_2) \cdot Runs(\mathcal{A}_2)\big)$$

Then, φ can be easily given by an automaton whose acceptance condition depends on $Acc(F_1)$.

Finally, we define the observation function as follows: for $i = 1, 2$, for all $\rho_i \in Runs(\mathcal{A}_i)$,

$$\mathcal{O}(q_\iota \cdot (\alpha_\iota, a_i) \cdot \rho_i) = \rho,$$

where ρ is the corresponding run in $\mathcal{A}$.

Given a $\sim'$-scheduler $\sigma' : Runs_f(\mathcal{A}') \to \mathcal{D}(Act)$, the probabilistic disclosure is thus:

$$\mathrm{PD}_{\sigma'}(\varphi, \mathcal{A}', \mathcal{O}) = \mathbf{P}_{\sigma'}(\varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))) =$$
$$\mathbf{P}_{\sigma'}\Big(\big(q_\iota.(\alpha_\iota, a_1).Acc(F_1)\big) \cup \big(q_\iota.(\alpha_\iota, a_2).Acc(F_2)\big)\Big).$$

Since σ' is a $\sim'$-scheduler, it is easy to see that $\mathbf{P}_{\sigma'}\big(q_\iota \cdot (\alpha_\iota, a_1) \cdot Acc(F_1)\big) = \mathbf{P}_{\sigma'}\big(q_\iota \cdot (\alpha_\iota, a_2) \cdot Acc(F_2)\big)$. Hence we get that $\mathrm{PD}_{\sigma'}(\varphi, \mathcal{A}', \mathcal{O}) = 2 \cdot \mathbf{P}_{\sigma'}\big(q_\iota \cdot (\alpha_\iota, a_1) \cdot Acc(F_1)\big)$

We build the $\sim$-scheduler σ for $\mathcal{A}$ as follows: for each $\rho \in Runs_f(\mathcal{A})$, we let $\overline{\rho} = q_\iota \cdot (\alpha_\iota, a_1) \cdot \rho_1$ and we define $\sigma(\rho) = \sigma'(\overline{\rho})$. Then for the corresponding cones, we have: $\mathbf{P}_\sigma(C_\rho) = 2 \cdot \mathbf{P}_{\sigma'}(C_{\overline{\rho}})$. We deduce that $\mathbf{P}_\sigma(Acc(F)) = 2 \cdot \mathbf{P}_{\sigma'}(q_\iota \cdot (\alpha_\iota, a_1) \cdot Acc(F_1)) = \mathrm{PD}_{\sigma'}(\varphi, \mathcal{A}', \mathcal{O})$.

Conversely, given a $\sim$-scheduler $\sigma : Runs_f(\mathcal{A}) \to \mathcal{D}(Act)$, we define a $\sim'$-scheduler $\sigma' : Runs_f(\mathcal{A}') \to \mathcal{D}(Act)$ as follows:

$$\sigma'(q_\iota) = (\alpha_\iota \mapsto 1)$$

and, for all runs $q_\iota \cdot (\alpha_\iota, a_i) \cdot \rho_i \in Runs_f(\mathcal{A}')$, for $i = 1, 2$,

$$\sigma'(q_\iota \cdot (\alpha_\iota, a_i) \cdot \rho_i) = \sigma(\rho).$$

Since σ is a $\sim$-scheduler, σ' is a $\sim'$-scheduler, and for $i = 1, 2$, we obtain that $\mathbf{P}_{\sigma'}(q_\iota \cdot (\alpha_\iota, a_i) \cdot Acc(F_i)) = \frac{1}{2}\mathbf{P}_\sigma(Acc(F))$. Hence, $\mathrm{PD}_{\sigma'}(\varphi, \mathcal{A}', \mathcal{O}) = \mathbf{P}_\sigma(Acc(F))$.

Then, there exists a $\sim$-scheduler σ for $\mathcal{A}$ such that $\mathbf{P}_\sigma(Acc(F)) > 0$ if and only if there exists a $\sim'$-scheduler σ' for $\mathcal{A}'$ such that $\mathrm{PD}_{\sigma'}(\varphi, \mathcal{A}', \mathcal{O}) > 0$. Also, there exists a $\sim$-scheduler σ for $\mathcal{A}$ such that $\mathbf{P}_\sigma(Acc(F)) = 1$ if and only if there exists a $\sim'$-scheduler σ' for $\mathcal{A}'$ such that $\mathrm{PD}_{\sigma'}(\varphi, \mathcal{A}', \mathcal{O}) = 1$. Moreover, $\sup\{\mathbf{P}_\sigma(Acc(F)), \sigma \sim\text{-scheduler for } \mathcal{A}\} = 1$ if and only if $\mathrm{PD}(\varphi, \mathcal{A}', \mathcal{O}) = 1$.

By [1, 11], we obtain that the almost-sure disclosure problem is undecidable for DCA (and thus for DPA), and that the almost sure opacity is undecidable for DBA, and limit disclosure problem is undecidable for DBA, DCA, hence for DPA that are more expressive. From this result, we get undecidability for the general disclosure problem and the value problems for DBA, DCA and DPA. Note that in the case of limit disclosure, general disclosure and value problems, the undecidability holds also when restricted to finite-memory strategies. Indeed, undecidability results for quantitative questions about probabilistic finite automata [28, 27] and for value 1 problem [19] carry over POMDPs restricted to finite-memory schedulers. □

We now show that, under some natural restrictions, one can recover decidability for the almost-sure disclosure and almost-sure opacity problems. First, if the secret is given as a Deterministic Büchi Automaton (DBA), then the almost-sure disclosure problem is decidable. Although deterministic Büchi automata are strictly less expressive than non deterministic ones, they can still be used to describe realistic predicates. For instance, a secret which is always recognized after a finite run would correspond to a set of runs that reach some sink state and remain there forever. The corresponding set of traces would be of the form $L\Sigma'^\omega$ for some language L of finite words and a subset Σ' of the alphabet Σ.

Theorem 14. *Given a DBA $\mathcal{A}_\varphi$ describing the secret, the almost-sure disclosure problem for POMDP is EXPTIME-complete.*

PROOF. Let $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0, \sim)$ be the POMDP modeling the system, and $\mathcal{A}_\varphi$ be the (complete) deterministic Büchi automaton over $Q \cup (Act \times \Sigma)$ that recognizes the runs of $\mathcal{A}$ that are in φ. We show how to obtain a deterministic automaton $\mathcal{A}_{\text{discl}} = (Q', Q \cup (Act \times \Sigma), \delta, q'_0, F)$ such that $L_B(\mathcal{A}_{\text{discl}}) = \varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$.
Indeed, with a co-Büchi acceptance condition for $\mathcal{A}_\varphi$, we get that $L_C(\mathcal{A}_\varphi) = \overline{L_B(\mathcal{A}_\varphi)}$. Then, it is possible to obtain a deterministic co-Büchi automaton $\mathcal{B}$ such that $L_C(\mathcal{B}) = \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$ (recall that non-deterministic co-Büchi automata are as expressive as deterministic co-Büchi automata [26]). Then $L_B(\mathcal{B}) = \overline{L_C(\mathcal{B})}$, and $\mathcal{A}_{\text{discl}}$ is the (complete) Büchi automaton obtained by intersecting the two deterministic Büchi automata $\mathcal{A}_\varphi$ and $\mathcal{B}$.

We build a new POMDP that will jointly simulate $\mathcal{A}$ and $\mathcal{A}_{\text{discl}}$. Since the automaton $\mathcal{A}_{\text{discl}}$ runs over runs of $\mathcal{A}$, we have to make explicit the transitions of $\mathcal{A}_{\text{discl}}$ on states of $\mathcal{A}$. For that we introduce a copy of each state of $\mathcal{A}$ in the product POMDP, from which we will allow $\mathcal{A}_{\text{discl}}$ to take the corresponding transition. Formally, we consider the product POMDP $\mathcal{A} \otimes \mathcal{A}_{\text{discl}} = (\overline{Q} \times Q', \Sigma \cup \{\alpha_\iota\}, Act \cup \{a_0\}, \Delta', (q_0?, q'_0), \sim')$ where $\overline{Q} = Q \cup \{q? \mid q \in Q\}$ is the set of states of $\mathcal{A}$ augmented with a copy of these states, α_ι and a_0 are new symbols, and Δ' is defined as follows: for all $q_1, q_2 \in Q$, $q'_1, q'_2 \in Q'$, $\alpha \in Act$ and $a \in \Sigma$,

$$\Delta'((q_1, q'_1), \alpha)(a, (q_2?, q'_2)) = \begin{cases} \Delta(q_1, \alpha)(a, q_2) \\ \text{ if } q'_2 = \delta(q'_1, (\alpha, a)) \\ 0 \text{ otherwise.} \end{cases}$$

$$\Delta'((q_1?, q'_1), \alpha_\iota)(a_0, (q_1, q'_2)) = \begin{cases} 1 & \text{if } q'_2 = \delta(q'_1, q_1) \\ 0 & \text{otherwise.} \end{cases}$$

The new equivalence $\sim'$ is defined by: $(q_1, q'_1) \sim' (q_2, q'_2)$ and $(q_1?, q'_1) \sim' (q_2?, q'_2)$ if $q_1 \sim q_2$. Let ρ' be a run of $\mathcal{A} \otimes \mathcal{A}_{discl}$. To define the projection of ρ' on $\mathcal{A}$, we use the following mapping $\Pi_{\mathcal{A}}$: for all $q \in Q$, $q' \in Q'$, $\alpha \in Act$, $a \in \Sigma$,

$$\Pi_{\mathcal{A}}((q, q')) = q$$
$$\Pi_{\mathcal{A}}((\alpha, a)) = (\alpha, a)$$
$$\Pi_{\mathcal{A}}((\alpha_\iota, a_0)) = \Pi_{\mathcal{A}}((q?, q')) = \varepsilon$$

which is extended to finite or infinite runs of $\mathcal{A} \otimes \mathcal{A}_{\text{discl}}$ in the natural way.
Similarly, the projection of ρ' onto $\mathcal{A}_{\text{discl}}$ uses the following mapping:

$$\Pi_{\text{discl}} : \mathit{Runs}_f(\mathcal{A} \otimes \mathcal{A}_{\text{discl}}) \to \mathit{Runs}_f(\mathcal{A}_{\text{discl}})$$

defined by induction on the length of ρ': For all $q'_1 \in Q'$, we let $\Pi_{\text{discl}}((q_0?, q'_0)(\alpha_\iota, a_0)(q_0, q'_1)) = q'_0 \cdot q_0 \cdot q'_1$. Then, for all $\rho' \in \mathit{Runs}_f(\mathcal{A} \otimes \mathcal{A}_{\text{discl}})$, for all $q_1 \in Q$, $q'_1, q'_2 \in Q'$, $\alpha \in Act$, $a \in \Sigma$, we define:

$$\Pi_{\text{discl}}(\rho' \cdot (\alpha, a) \cdot (q_1?, q'_1) \cdot (\alpha_\iota, a_0) \cdot (q_1, q'_2)) = \Pi_{\text{discl}}(\rho') \cdot (\alpha, a) \cdot q'_1 \cdot q_2 \cdot q'_2$$

The mapping Π_{discl} is increasing, hence for ρ' an infinite run of $\mathcal{A} \otimes \mathcal{A}_{\text{discl}}$, we can define $\Pi_{\text{discl}}(\rho') = \bigsqcup_{r \text{ finite prefix of } \rho'} \Pi_{\text{discl}}(r)$.

It is easy to see that $\rho = \Pi_{\mathcal{A}}(\rho')$ is a run of $\mathcal{A}$, and that $\Pi_{\text{discl}}(\rho')$ is a run of $\mathcal{A}_{\text{discl}}$ over ρ. Then, $\rho \in \varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$ if and only if $\rho \in L_B(\mathcal{A}_{\text{discl}})$, if and only if $\Pi_{\text{discl}}(\rho') \in \text{Büchi}(F)$ if and only if $\rho' \in \text{Büchi}(\overline{Q} \times F)$.

Let σ' be a $\sim'$-scheduler of $\mathcal{A} \otimes \mathcal{A}_{\text{discl}}$, and let ρ be a finite run of $\mathcal{A}$. Observe that there is a unique run $\rho' \in \mathit{Runs}_f(\mathcal{A} \otimes \mathcal{A}_{\text{discl}})$ such that $\Pi_{\mathcal{A}}(\rho') = \rho$. We then let $\sigma(\rho) = \sigma'(\rho')$, which is clearly a $\sim$-scheduler for $\mathcal{A}$. Moreover, for all finite runs ρ of $\mathcal{A}$, we have $\mathbf{P}_\sigma(C_\rho) = \mathbf{P}_{\sigma'}(C_{\rho'})$. Hence $\mathbf{P}_\sigma(\varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))) = \mathbf{P}_{\sigma'}(\text{Büchi}(\overline{Q} \times F))$.
Conversely, let σ be a $\sim$-scheduler of $\mathcal{A}$. We define a $\sim'$-scheduler σ' as follows. For $\rho' \in \mathit{Runs}_f(\mathcal{A} \otimes \mathcal{A}_{\text{discl}})$, for all $q \in Q$, $q' \in Q'$,

$$\sigma'(\rho' \cdot (q?, q)) = (\alpha_\iota \mapsto 1)$$
$$\sigma'(\rho' \cdot (q, q')) = \sigma(\Pi_{\mathcal{A}}(\rho' \cdot (q, q')))$$

In that case again, $\mathbf{P}_\sigma(\Pi_{\mathcal{A}}(\rho')) = \mathbf{P}_{\sigma'}(\rho')$, so $\mathbf{P}_\sigma(\varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))) = \mathbf{P}_{\sigma'}(\text{Büchi}(\overline{Q} \times F))$.

Now, the almost-sure disclosure problem is equivalent to deciding whether there is a $\sim'$-scheduler σ' for $\mathcal{A} \otimes \mathcal{A}_{\text{discl}}$ such that $\mathbf{P}_{\sigma'}(\text{Büchi}(\overline{Q} \times F)) = 1$. From [1, 13, 12], this last problem is in EXPTIME. To solve the problem on a given POMDP, one builds an MDP in which each state is enriched with the belief of the scheduler at this point, hence with a size exponentially larger than the original model. A naive application of this algorithm to the POMDP $\mathcal{A} \otimes \mathcal{A}_{\text{discl}}$ would yield a POMDP of size exponentially larger than the original $\mathcal{A}$ and $\mathcal{A}_\varphi$, hence would provide an algorithm in 2-EXPTIME. We then need a more careful and less costly construction: it consists in computing the belief only on the POMDP $\mathcal{A}$ part, and not on the component coming from $\mathcal{A}_{\text{discl}}$, which is simply a deterministic automaton. Hence, the obtained MDP is only exponential in the size of $\mathcal{A}$ and $\mathcal{A}_\varphi$, and the overall algorithm is in EXPTIME.

Moreover, proof of Theorem 13 provides a reduction from qualitative problems on POMDP

to almost-sure opacity and almost sure disclosure problems. Given a run ρ, we let $\mathrm{Appear}(\rho)$ be the set of states appearing (at least once) in the run, and consider the acceptance condition $\mathrm{Reach}(F)$ defined, for $F \subseteq Q$, by $\mathrm{Reach}(F) = \{\rho \in Q^\omega \mid \mathrm{Appear}(\rho) \cap F \neq \emptyset\}$. Then, we have shown that given a POMDP $\mathcal{A}$, and a set of states F, one can build a POMDP $\mathcal{A}'$ (which is the POMDP of Figure 2, in which the set F_1 is made absorbing), an observation function $\mathcal{O}$, and a secret φ given by a DBA, such that there exists a $\sim$-scheduler for $\mathcal{A}$ such that $\mathbf{P}_\sigma(\mathrm{Reach}(F)) = 1$ if and only if there exists a $\sim'$-scheduler σ' for $\mathcal{A}'$ such that $\mathrm{PD}_{\sigma'}(\mathcal{A}', \mathcal{O}, \varphi) = 1$. The EXPTIME-hardness for our problem follows from the EXPTIME-hardness of the almost-sure problem for POMDP with reachability conditions [12]. □

Finally, we show that if we restrict to finite-memory schedulers, then both the almost-sure disclosure and almost-sure opacity problems become decidable for secrets given as DPA. Since finite-memory schedulers are the only schedulers of practical interest, and DPA allow to describe any ω-regular predicate, this restriction is of great interest.

Theorem 15. *Given a POMDP $\mathcal{A}$, a morphism $\mathcal{O}$ as observation function, and a secret given as a DPA, the finite-memory almost-sure opacity problem and the finite-memory almost-sure disclosure problem are EXPTIME-complete.*

Proof. The proof follows the same lines than the proof of Theorem 14. Given a POMDP $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0, \sim)$ modeling the system and a DPA $\mathcal{A}_\varphi$ describing the secret φ, one can obtained a DPA $\mathcal{A}_{\mathrm{discl}} = (Q', Q \cup (Act \times \Sigma), \delta, q'_0, F)$ such that $L_P(\mathcal{A}_{\mathrm{discl}}) = \varphi \setminus \mathcal{O}^{-1}(\mathcal{O}(\overline{\varphi}))$, since this language is ω-regular.

As in the previous proof, we build a new POMDP as a product of $\mathcal{A}$ and $\mathcal{A}_{\mathrm{discl}}$, $\mathcal{A} \otimes \mathcal{A}_{\mathrm{discl}} = (\overline{Q} \times Q', \Sigma \cup \{\alpha_\iota\}, \Delta', (q_0?, q'_0), \sim')$. If $F : Q' \to \{1, \cdots, k\}$, we let $F' : \overline{Q} \times Q' \to \{1, \ldots, k\}$, where, for all $q \in \overline{Q}, q' \in Q'$, $F'(q, q') = F(q')$. Then, the finite-memory almost-sure disclosure problem is equivalent to deciding whether there is a finite-memory $\sim'$-scheduler σ' for $\mathcal{A} \otimes \mathcal{A}_{\mathrm{discl}}$ such that $\mathbf{P}_{\sigma'}(\mathrm{Parity}(F')) = 1$, and the finite-memory almost-sure opacity problem is equivalent to deciding whether there is a finite-memory scheduler σ' for $\mathcal{A} \otimes \mathcal{A}_{\mathrm{discl}}$ such that $\mathbf{P}_{\sigma'}(\mathrm{Parity}(F')) = 0$. From [10], when restricting to finite-memory schedulers, these two problems are in EXPTIME. As in the proof of Theorem 14, to maintain the procedure within exponential time, the powerset construction on the POMDP will only be made on the $\mathcal{A}$ component of the product.

Also, the proof of EXPTIME-hardness follows the same lines than the proof of Theorem 14. □

5. Conclusion

Extending the definition of probabilistic opacity to MDPs (with infinite runs), we solve decidability questions raised in [3]. More elaborate measures could be studied in this context, and are left for future work. Another interesting issue would be to investigate *disclosure before some given delay*, either as a number of steps in the spirit of [30] for discrete event systems, or for probabilistic timed system with an explicit time bound. In the latter case, decidability results could be obtained by combining our results with recent ones like [6].

Acknowledgements. We thank anonymous referees for their comments and suggestions. The research was partly supported by Austrian Science Fund (FWF) Grant No P 23499- N23, FWF NFN Grant No S11407-N23, ERC Start grant (279307: Graph Games), Microsoft faculty fellows award, Coopération France-Québec, Service Coopération et Action Culturelle 2012/26/SCAC, and project ImpRo ANR-2010-BLAN-0317.

References

[1] Christel Baier, Marcus Größer, and Nathalie Bertrand. Probabilistic ω-automata. *J. ACM*, 59(1):1, 2012.

[2] Béatrice Bérard, John Mullins, and Mathieu Sassolas. Quantifying opacity. In Gianfranco Ciardo and Roberto Segala, editors, *Proceedings of the 7th International Conference on Quantitative Evaluation of Systems (QEST'10)*, pages 263–272. IEEE Computer Society, September 2010.

[3] Béatrice Bérard, John Mullins, and Mathieu Sassolas. Quantifying opacity. *CoRR*, abs/1301.6799, 2013. extended version.

[4] Patrick Billingsley. *Probability and Measure*. Wiley, New York, NY, 3rd edition, 1995.

[5] Michele Boreale, Francesca Pampaloni, and Michela Paolini. Quantitative information flow, with a view. In Vijay Atluri and Claudia Díaz, editors, *Proc. of 16th European Symposium on Research in Computer Security (ESORICS 2011)*, volume 6879 of *Lecture Notes in Computer Science*, pages 588–606. Springer, 2011.

[6] Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joël Ouaknine, Jean-François Raskin, and James Worrell. Time-bounded reachability for monotonic hybrid automata: Complexity and fixed points. In Dang Van Hung and Mizuhito Ogawa, editors, *Proc. of 11th International Symposium on Automated Technology for*

Verification and Analysis, ATVA 2013, volume 8172 of *Lecture Notes in Computer Science*, pages 55–70. Springer, 2013.

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

[8] Jeremy W. Bryans, Maciej Koutny, and Chunyan Mu. Towards quantitative analysis of opacity. In Catuscia Palamidessi and Mark Dermot Ryan, editors, *Proc. 7th Int. Symp. on Trustworthy Global Computing (TGC'12), Revised Selected Papers*, volume 8191 of *Lecture Notes in Computer Science*, pages 145–163. Springer, 2013.

[9] Franck Cassez, Jeremy Dubreil, and Hervé Marchand. Synthesis of opaque systems with static and dynamic masks. *Formal Methods in System design*, 40(1):88 – 115, 2012.

[10] Krishnendu Chatterjee, Martin Chmelik, and Mathieu Tracol. What is decidable about partially observable Markov decision processes with omega-regular objectives. In *CSL*, pages 165–180, 2013.

[11] Krishnendu Chatterjee, Laurent Doyen, Hugo Gimbert, and Thomas A. Henzinger. Randomness for free. In *Proceedings of MFCS 2010: Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 6281, pages 246–257. Springer-Verlag, 2010.

[12] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Qualitative analysis of partially-observable Markov decision processes. In Petr Hliněný and Antonín Kučera, editors, *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science (MFCS'10)*, volume 6281 of *Lecture Notes in Computer Science*, pages 258–269, Brno, Czech Republic, August 2010. Springer.

[13] Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, 3(3), 2007.

[14] Krishnendu Chatterjee and Monika Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *SODA*, pages 1318–1336, 2011.

[15] Krishnendu Chatterjee, Marcin Jurdzinski, and Thomas A. Henzinger. Quantitative stochastic parity games. In *SODA*, pages 121–130, 2004.

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

[17] Costas Courcoubetis and Mihalis Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[18] Jeremy Dubreil, Philippe Darondeau, and Hervé Marchand. Supervisory Control for Opacity. *IEEE Transactions on Automatic Control*, 55(5):1089 –1100, may 2010.

[19] Hugo Gimbert and Youssouf Oualhadj. Probabilistic automata on finite words: Decidability and undecidability results. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Proceedings of ICALP 2010*, volume 6199 of *Lecture Notes in Computer Science*, pages 527–538. Springer, 2010.

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

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

[22] Heiko Mantel and Henning Sudbrock. Information-theoretic modeling and analysis of interrupt-related covert channels. In P. Degano, J. Guttman, and F. Martinelli, editors, *Proceedings of the Workshop on Formal Aspects in Security and Trust, FAST 2008*, Springer, LNCS 5491, pages 67–81, 2009.

[23] Bastien Maubert, Sophie Pinchinat, and Laura Bozzelli. Opacity issues in games with imperfect information. In *2nd Int. Symp. on Games, Automata, Logics and Formal Verification (GandALF'11)*, pages 87–101, 2011.

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

[25] Jonathan K. Millen. Covert Channel Capacity. In *Proc. of IEEE Symposium on Research in Computer Security and Privacy*, pages 144–161, 1987.

[26] Satoru Miyano and Takeshi Hayashi. Alternating finite automata on omega-words. *Theor. Comput. Sci.*, 32:321–330, 1984.

[27] A. Paz. *Introduction to probabilistic automata (Computer science and applied mathematics)*. Academic Press, 1971.

[28] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

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

[30] Anooshiravan Saboori and Christoforos N. Hadjicostis. Verification of k-step opacity and analysis of its complexity. *IEEE T. Automation Science and Engineering*, 8(3):549–559, 2011.

[31] Anooshiravan Saboori and Christoforos N. Hadjicostis. Current-state opacity formulations in probabilistic finite automata. *IEEE Trans. Automat. Contr.*, 59(1):120–133, 2014.

[32] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1997.

[33] Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of 26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 327–338. IEEE Computer Society, 1985.

[34] John T. Wittbold and Dale M. Johnson. Information flow in nondeterministic systems. In *Proc. of IEEE Symposium on Research in Computer Security and Privacy*, pages 144–161, 1990.