Enforcing Opacity of Regular Predicates on Modal Transition Systems

Philippe Darondeau

Inria, Centre Rennes-Bretagne Atlantique, Campus de Beaulieu, F35042 Rennes-Cedex (e-mail: Philippe.Darondeau@inria.fr).

Abstract: Given a labelled transition system LTS partially observed by an attacker, and a regular predicate Sec over the runs of LTS, enforcing opacity of the secret Sec in LTS means computing a supervisory controller K such that an attacker who observes a run of K/LTS cannot ascertain that the trace of this run belongs to Sec based on the knowledge of LTS and K. We lift the problem from a single labelled transition system LTS to the class of all labelled transition systems specified by a modal transition system MTS. The lifted problem is to compute the maximally permissive controller K such that Sec is opaque in K/LTS for every labelled transition systems LTS which is a model of MTS. The situations of the attacker and of the controller are dissymmetric: at run time, the attacker may fully know LTS and K whereas the controller knows only MTS and the sequence of actions executed so far by the unknown LTS. We address the problem in two cases. Let Σ_a denote the set of actions that can be observed by the attacker, and let Σ_c and Σ_o denote the sets of actions that can be controlled and observed by the controller, respectively. We provide optimal and regular controllers that enforce the opacity of regular secrets when $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_a = \Sigma$. We provide optimal and regular controllers that enforce the opacity of regular upper-closed secrets ($Sec = Sec.\Sigma^*$) when $\Sigma_a \subseteq \Sigma_c \subseteq \Sigma_o = \Sigma$.

Keywords: Partial Observation, Opacity, Modal Automata, Supervisory Control.

1. INTRODUCTION

The concept of opacity, first introduced in the context of sessions of security protocols [12, 13], was extended later on to transition systems [3]. A predicate over the runs of a transition system is opaque w.r.t. an observation function if every observation produced by a run that satisfies the predicate is also produced by some run that does not satisfy the predicate. The concept of opacity is very flexible as it depends both on the class of predicates and on the observation function. By adjusting these two parameters, many common security properties such as confidentiality, anonymity and so on, can be rephrased in terms of opacity [3, 10]. Opacity is in general undecidable but this property may be checked effectively when it is applied to regular predicates on runs of finite transition systems and with observation functions induced by projection operators. Algorithms for checking opacity in Discrete Event Systems are presented together with applications in [17, 19, 10].

An active and hot topic at the frontier of the theories of Security and Discrete Event Systems is the search for Supervisory Controllers enforcing the opacity of a predicate on a given transition system. As written in [8], long term motivation for such work may be found in the need to protect SCADA systems and networks of sensors and actuators from interferences with malicious agents through TCP/IP. However, work done till now has born upon finite transition systems exclusively. Approaches differ by considering either initial-state opacity [17, 18], or current-state opacity [8], or language opacity [1, 2, 4, 5, 10, 19, 20]. With state opacity, the secret predicate bears either upon the initial state, or upon the current state, or upon the set of all states that have been gone through from the beginning of a run. With language opacity, the secret predicate is a set of sequences of actions that label transitions. Language opacity and current-state opacity are mutually reducible. Approaches also differ upon whether they provide synthesis algorithms or closed formulas or both for maximally permissive controllers enforcing opacity. Closed formulas are proposed for instance in [2, 20, 19]. *In fine*, all approaches rely on Ramadge and Wonham's basic theory of supervisory control for DES [14, 15, 16]. Significant adaptations must however be brought to the basic theory, because opacity objectives do not reduce to safety and liveness. In fact, opacity objectives are not concerned with individual runs but with sets of indiscernible runs from the perspective of the attacker. Classes of indiscernible runs may be captured by estimators as usually done for the purpose diagnosis.

In this paper, we lift the opacity enforcing control problem from finite transition systems to families of finite transition systems specified by modal transition systems. Modal transition systems were introduced in [9] as tuples $(S, \Sigma, \rightarrow_\Box, \rightarrow_\Diamond, s_0)$ with two modal transition relations $\rightarrow_\Box$ (the *strong* or *must* transition relation) and $\rightarrow_\Diamond$ (the *weak* or *may* transition relation), both included in $S \times \Sigma \times S$ and subject to the inclusion constraint $\rightarrow_\Box \subseteq \rightarrow_\Diamond$. A modal transition system MTS should be understood as a logical formula, with labelled transition systems as models. Modal transition systems are indeed a well-identified fragment of the modal μ-calculus [7]. Intuitively, a labelled transition system LTS is a model of a modal transition system MTS if there exists a relation $\models$ between their respective sets of states Q and S such that $q_0 \models s_0$ holds for the initial states and whenever $q \models s$, all *must* transitions from s are simulated by transitions from q, all transitions from q are simulated by *may* transitions from s and $\models$ is preserved under simulation of transitions in both directions.

Example 1. (adapted from [6]). The modal transition system MTS depicted in Figure 1, where the relations $\rightarrow_\square$ and $\rightarrow_\diamond$ are represented with plain arrows and dashed arrows, respectively, expresses the fact that the presence of the first transition a is mandatory in any model LTS, while the second transition a is optional, and that after any a, a model LTS should be able to trigger a b (after the execution of a single a, the execution of this b transition is not mandatory, since LTS may alternatively trigger a second a). The presence of a second transition b (returning to the initial state of MTS) is optional in LTS. The two LTS on the right hand side of Figure 1 are models of MTS, whereas the LTS depicted in Figure 2 is not. Indeed, after the sequence aa, MTS requires a transition labelled by b, which in not present in this LTS. $\diamond$

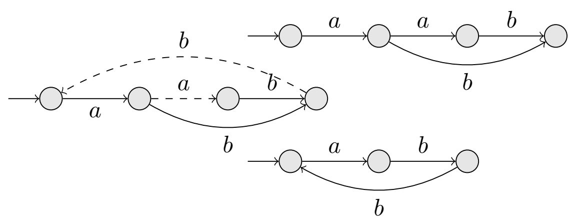

Fig. 1. A modal specification MTS and some LTS models

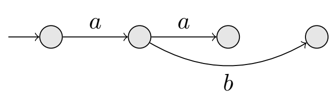

Fig. 2. An LTS which is not a model of MTS

In everyday life, one uses frequently systems without an exact knowledge of their behaviour. This is generally the case when the system belongs to a range of products with many versions, such as mobile phones or software, and all the more for software with automatic updates. This is also the case when the system is a web service or an orchestration, selected on request by a broker so as to match operating guidelines specified in the request [11]. In such situations, modal transition systems may serve to represent the partial knowledge of the user on the possible behaviours of the system (modal transition systems with final states, introduced in [6], are in fact a restricted form of the operating guidelines of [11]). Enforcing opacity of regular predicates on modal transition systems may then serve to prevent user confidential information to be leaked by the partially unknown system which they actually use.

The purpose of this paper differs from the purpose of our earlier paper [6]. In [6], the goal was to enforce specifications of service, expressed by modal transition systems, on service providers, modelled by LTS. Here the goal is to enforce the opacity of a secret predicate on all models LTS of a modal transition system MTS.

The rest of the paper is organized as follows. First, we recall briefly the background of Modal Transition Systems and Supervisory Control for Opacity, and we state the opacity enforcement problem for modal transition systems. The parameters of the problem are the secret predicate, the subset of actions Σ_a that the attacker can observe, and the subsets of actions Σ_o and Σ_c that the controller can observe and control, respectively. Then, we address the opacity enforcement problem for regular secrets in the case $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_a$ and for upper-closed regular secrets in the case $\Sigma_a \subseteq \Sigma_c \subseteq \Sigma_o = \Sigma$. Possible extensions of this work are considered in a short conclusion.

2. BACKGROUND OF TRANSITION SYSTEMS

We recall in this section the background of labelled transition systems and modal transition systems.

2.1 Labelled Transition Systems

A deterministic *labelled transition system* (or LTS) over Σ is a 4-tuple $LTS = (Q, \Sigma, \delta, q_0)$ where Q is a finite set of states, $q_0 \in Q$ is an *initial state*, and δ is a partial map from $Q \times \Sigma$ to Q, called the *labelled transition map*. This map is extended inductively to $\delta : Q \times \Sigma^* \rightarrow Q$ by letting $\delta(q, \varepsilon) = q$ (where ε is the empty word) and $\delta(q, w.\sigma) = \delta(\delta(q, w), \sigma)$ for all $q \in Q$, $w \in \Sigma^*$ and $\sigma \in \Sigma$ ($w.\sigma$ denotes the word got by appending σ to w, and similarly, $w.w'$ and $w.L'$ denote the concatenation of two words and the prefixing of a language by a word, i.e., $w.L' = \{w.w' \mid w' \in L'\}$). A state $q \in Q$ is *reachable* (from q_0) if $\delta(q_0, w) = q$ for some word $w \in \Sigma^*$. An LTS is *finite* if Q and Σ are finite; it is *reduced* if all states in Q are reachable and every event $\sigma \in \Sigma$ is enabled at some state q, i.e., $\delta(q, \sigma)$ is defined for the considered state q. In the sequel, we always consider finite and reduced labelled transition systems. The *language* of LTS is the set of words $\mathcal{L}(LTS) = \{w \in \Sigma^* \mid \delta(q_0, w) \text{ defined}\}$. More generally, for $q \in Q$, we let $\mathcal{L}(LTS, q) = \{w \in \Sigma^* \mid \delta(q, w) \text{ defined}\}$.

Given two labelled transition systems $LTS = (Q, \Sigma, \delta, q_0)$ and $LTS' = (Q', \Sigma, \delta', q'_0)$ over the same alphabet Σ, their *product* is the (reachable restriction of the) labelled transition system $LTS \times LTS' = (Q \times Q', \Sigma, \delta \times \delta', (q_0, q'_0))$ where $(\delta \times \delta')((q, q'), \sigma) = (\delta(q, \sigma), \delta'(q', \sigma))$.

2.2 Modal Transition Systems [9]

A deterministic *modal transition system* (or MTS) over Σ is a 5-tuple $MTS = (S, \Sigma, \delta^\square, \delta^\diamond, s_0)$ where S is a finite set of *logical states*, and $\delta^\square : S \times \Sigma \rightarrow S$ and $\delta^\diamond : S \times \Sigma \rightarrow S$ are two partial maps, called the *strong* and the *weak* labelled transition maps, respectively, subject to the constraint $\delta^\square \subseteq \delta^\diamond$. The maps $\delta^\square$ and $\delta^\diamond$ are extended inductively to words like the transition maps of labelled transition systems. For any modal transition system MTS, we let $\mathcal{L}(MTS) = \mathcal{L}(\overline{MTS})$ where $\overline{MTS} = (S, \Sigma, \delta^\diamond, s_0)$, thus $\overline{MTS}$ denotes the LTS whose transition map is the weak transition map of MTS. Similarly, $\underline{MTS} = (S, \Sigma, \delta^\square, s_0)$ denotes the LTS whose transition map is the strong transition map of MTS.

A modal transition system MTS determines a family of labelled transition systems LTS called its *models* (notation: $LTS \models MTS$). A labelled transition system $LTS = (Q, \Sigma, \delta, q_0)$ is a model of $MTS = (S, \Sigma, \delta^\square, \delta^\diamond, s_0)$ if there exists a relation $\models \subseteq Q \times S$ such that $q_o \models s_0$ and for all $q \in Q$ and $s \in S$, $q \models s$ entails the following for all $\sigma \in \Sigma$:

- if $\delta(q,\sigma)$ is defined then $\delta^\diamond(s,\sigma)$ is defined and $\delta(q,\sigma) \models \delta^\diamond(s,\sigma)$,
- if $\delta^\square(s,\sigma)$ is defined then $\delta(q,\sigma)$ is defined and $\delta(q,\sigma) \models \delta^\square(s,\sigma)$.

With these definitions, $\underline{MTS} \models MTS$, $\overline{MTS} \models MTS$, and $LTS \models MTS \Rightarrow \mathcal{L}(\underline{MTS}) \subseteq \mathcal{L}(LTS) \subseteq \mathcal{L}(\overline{MTS})$. Therefore, $\mathcal{L}(\underline{MTS}) = \bigcap\{\mathcal{L}(LTS) \mid LTS \models MTS\}$ and $\mathcal{L}(\overline{MTS}) = \bigcup\{\mathcal{L}(LTS) \mid LTS \models MTS\}$. However, $\underline{MTS}$ and $\overline{MTS}$ are not the unique models of MTS with minimal or maximal language, respectively, since there may exist other LTS with the same language. A central property of modal transition systems is stated by the following relation: $LTS_1 \models MTS \wedge LTS_2 \models MTS \Rightarrow LTS_1 \times LTS_2 \models MTS$. We refer the reader to [7] for more information.

In addition to these reminders, we introduce a specific construction used in later proofs. Given a modal transition system MTS, we want to construct for each word $w \in \mathcal{L}(MTS)$ a labelled transition system $w \circ MTS$ such that $w \circ MTS \models MTS$, $w \in \mathcal{L}(w \circ MTS)$, and $\mathcal{L}(w \circ MTS)$ is the infimum of $\mathcal{L}(LTS)$ for all labelled transition systems LTS satisfying $LTS \models MTS$ and $w \in \mathcal{L}(LTS)$.

Definition 1. Given $w = \sigma_1 \ldots \sigma_n \in \mathcal{L}(MTS)$ where $MTS = (S, \Sigma, \delta^\square, \delta^\diamond, s_0)$, let $w \circ MTS$ denote the LTS produced by the following procedure, where $s_i = \delta^\diamond(s_0, \sigma_1 \ldots \sigma_i)$ for $1 \leq i \leq n$:

- make $n+1$ separate copies of the set of states S with elements (s,i), $s \in S$ and $0 \leq i \leq n$,
- for $1 \leq i \leq n$, let $\delta((s_{i-1}, i-1), \sigma_i) = (s_i, i)$,
- for $0 \leq i \leq n$ and for all pairs $(s,\sigma) \in S \times \Sigma$ such that $s \neq s_i$ or $\sigma \neq \sigma_{i+1}$, let $\delta((s,i),\sigma) = (\delta^\square(s,\sigma), i)$,
- let $(s_0, 0)$ be the initial state and δ be the partial transition map,
- remove all unreachable states. ◇

For $w = \varepsilon$ (the empty word), i.e., for $n = 0$, $\varepsilon \circ MTS$ is isomorphic to $\underline{MTS}$. For any other word $w.\sigma \in \mathcal{L}(MTS)$ with $w \in \Sigma^*$, $\sigma \in \Sigma$ and $\delta^\diamond(s_0, w.\sigma) = s$, the language of the labelled transition system $(w.\sigma) \circ MTS$ is equal to $\mathcal{L}(w \circ MTS) \cup w.\sigma.\, \mathcal{L}(\underline{MTS}, s)$.

Proposition 1. $w \circ MTS \models MTS$, $w \in \mathcal{L}(w \circ MTS)$, and $\mathcal{L}(w \circ MTS) = \bigcap\{\mathcal{L}(LTS) \mid LTS \models MTS \wedge w \in \mathcal{L}(LTS)\}$.

Proof. The first two statements are obvious. We prove $LTS \models MTS \wedge w \in \mathcal{L}(LTS) \Rightarrow \mathcal{L}(w \circ MTS) \subseteq \mathcal{L}(LTS)$ by induction on w. For $w = \varepsilon$, this holds since $\mathcal{L}(\varepsilon \circ MTS) = \mathcal{L}(\underline{MTS})$. For any other word $w.\sigma \in \mathcal{L}(MTS)$ with $\sigma \in \Sigma$, $\mathcal{L}(w.\sigma) \circ MTS = \mathcal{L}(w \circ MTS) \cup w.\sigma.\, \mathcal{L}(\underline{MTS}, s)$. By induction, $\mathcal{L}(w \circ MTS) = \bigcap\{\mathcal{L}(LTS) \mid LTS \models MTS \wedge w \in \mathcal{L}(LTS)\} \subseteq \bigcap\{\mathcal{L}(LTS) \mid LTS \models MTS \wedge w.\sigma \in \mathcal{L}(LTS)\}$. By definition of the relation $\models$, $LTS \models MTS \wedge w.\sigma \in \mathcal{L}(LTS) \Rightarrow w.\sigma.\, \mathcal{L}(\underline{MTS}, s) \subseteq \mathcal{L}(LTS)$ for any labelled transition system LTS. □

3. BACKGROUND OF OPACITY AND SUPERVISORY CONTROL FOR OPACITY

Given $LTS = (Q, \Sigma, \delta, q_0)$, let $Sec \subseteq \Sigma^*$ be a regular predicate called the *secret*, and let $\Sigma_a \subseteq \Sigma$ be the set of actions that the *attacker* can observe. The secret predicate Sec is said to be *opaque* in LTS w.r.t. Σ_a if, for any word $w \in \mathcal{L}(LTS) \cap Sec$, there exists some word $w' \in \mathcal{L}(LTS) \setminus Sec$ with an identical projection on Σ_a, i.e., $\pi_a(w) = \pi_a(w')$ where $\pi_a(w)$ is the *natural projection* of w on Σ_a defined inductively by:

- $\pi_a(\varepsilon) = \varepsilon$ (the empty word),
- $\pi_a(v.\sigma) = \pi_a(v).\sigma$ for $v \in \Sigma^*$ and $\sigma \in \Sigma_a$,
- $\pi_a(v.\sigma) = \pi_a(v)$ for $v \in \Sigma^*$ and $\sigma \notin \Sigma_a$.

Enforcing the opacity of the secret Sec w.r.t. Σ_a in LTS means computing a *supervisory controller* K such that Sec is opaque w.r.t. Σ_a in the product $LTS \times K$, usually written K/LTS. In Ramadge and Wonham's setting for supervisory control [14, 15, 16], an *admissible* controller K may be seen as an LTS $K = (X, \Sigma, \delta_K, x_0)$, subject to constraints parametric on two subsets of actions Σ_c and Σ_o. The first set Σ_c is comprised of the actions that the controller can block or *control*. For any uncontrollable action $\sigma \notin \Sigma_c$ and for any word w, if $\delta_K(x_0, w) = x$ and $w\sigma \in \mathcal{L}(LTS)$ then $\delta_K(x,\sigma)$ must be defined. The second set Σ_o is comprised of the actions that the controller can *observe*. For any action $\sigma \notin \Sigma_o$ and for any state x in which $\delta_K(x,\sigma)$ is defined, it is required that $\delta_K(x,\sigma) = x$. Moreover, if $x = \delta_K(x_0, w)$ and $x' = \delta_K(x_0, w')$ for two words w and w' with equal projections on Σ_0, then $\delta_K(x,\sigma)$ and $\delta_K(x',\sigma)$ should be both defined or both undefined for any controllable action $\sigma \in \Sigma_c$,

$K^\dagger$ is said to be *maximal permissive* among the controllers that enforce the opacity of Sec in LTS w.r.t. Σ_a if $\mathcal{L}(K/LTS) \subseteq \mathcal{L}(K^\dagger/LTS)$ for all such controllers K. In [4], it was shown that there exists a maximal permissive and regular controller $K^\dagger$ in all cases where $\Sigma_c \subseteq \Sigma_o$ and Σ_a compares with Σ_c and Σ_o. In [5], more elaborate constructions were presented for computing $K^\dagger$ in the case where $\Sigma_c \subseteq \Sigma_o$ and $\Sigma_a \subseteq \Sigma_o$. In this paper, we make a first step to extend these results to modal transition systems.

4. ENFORCING OPACITY IN MODAL TRANSITION SYSTEMS

From now on, $MTS = (S, \Sigma, \delta^\square, \delta^\diamond, s_0)$ is a fixed modal transition system, and Sec is a fixed regular subset of Σ^*, called the secret. Let Σ_a be the subset of actions in Σ that can be observed by the attacker. Let Σ_o and Σ_c be the subsets of actions in Σ that may be observed or blocked by the controller, respectively. We want to construct controllers $K = (X, \Sigma, \delta_K, x_0)$ such that, for every labelled transition system LTS over Σ, if $LTS \models MTS$, then K is an admissible controller of LTS (w.r.t. Σ_o and Σ_c) and the secret Sec is opaque in K/LTS (w.r.t. Σ_a). In this case, we say that K enforces the opacity of Sec in MTS w.r.t. Σ_a. As regards permissivity, it would not make any sense to require that $K^\dagger$ be maximal permissive for every model LTS of MTS (among the controllers K that enforce the opacity of Sec in LTS w.r.t. Σ_a). In the framework of opacity control for modal transition systems, we will say that $K^\dagger$ is *maximal permissive* if $\mathcal{L}(K/LTS) \subseteq \mathcal{L}(K^\dagger/LTS)$ for every controller K that enforces the opacity of Sec in MTS (w.r.t. Σ_a) and for every model LTS of MTS. In the following sections, we address two cases in which a maximal permissive and regular controller $K^\dagger$ can be constructed.

5. COMPUTING $K^\dagger$ WHEN THE ATTACKER HAS FULL OBSERVATION

In this section, Sec is an arbitrary regular subset of Σ^* and we assume that $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_a$. Under these assumptions, $\pi_a(w) \neq \pi_a(w')$ for any two distinct words $w, w' \in \Sigma^*$, i.e., the attacker has full observation. In order that a controller K enforces the opacity of the secret Sec in MTS, it is necessary and sufficient that $\mathcal{L}(K/LTS) \subseteq \Sigma^* \setminus Sec$ for every model LTS of MTS, where $\Sigma^* \setminus Sec$ is the complement of the predicate Sec (hence it is a regular subset of Σ^*).

Now $\mathcal{L}(K/LTS) = \mathcal{L}(K) \cap \mathcal{L}(LTS)$ and $\mathcal{L}(\overline{MTS})$ is the supremum of $\mathcal{L}(LTS)$ for all $LTS \models MTS$. Therefore, $\mathcal{L}(K/LTS) \subseteq \Sigma^* \setminus Sec$ for all models LTS of MTS if and only if $\mathcal{L}(K/\overline{MTS}) \subseteq \Sigma^* \setminus Sec$, and K is an admissible controller for all models LTS of MTS if and only if it is an admissible controller of $\overline{MTS}$ (w.r.t. Σ_c and Σ_o).

As $\Sigma_c \subseteq \Sigma_o$ (entailing that K is observable if and only if it is normal), the maximally permissive controller $K^\dagger$ enforcing the opacity of the secret Sec in MTS (w.r.t. Σ_a) is the maximally permissive solution $K^\dagger$ of the basic supervisory control problem for $\overline{MTS}$ and the safe behaviour $\mathcal{L}(\overline{MTS}) \setminus Sec$. This $K^\dagger$ is a finite state controller, and it may be computed by applying Ramadge and Wonham's theory and algorithms [14, 15, 16].

6. COMPUTING $K^\dagger$ FOR REGULAR UPPER-CLOSED SECRETS

In this section, we assume that the secret Sec is upper-closed w.r.t. the prefix-order on words, i.e., $Sec = Sec.\Sigma^*$. This working assumption, also made in [1], implies that the goal of the opacity game is to avoid that the attacker may ascertain that some *prefix* of the partially observed run of the LTS *was* in the secret.

We moreover assume that $\Sigma_a \subseteq \Sigma_c \subseteq \Sigma_o = \Sigma$. Under these assumptions, the attacker has partial observation, whereas the controller has full observation and can block all actions observed by the attacker. This gives a strong advantage to the controller over the attacker, but remember that in counterpart the controller ignores which LTS among all models of MTS is executing, whereas the attacker may know precisely which LTS is executing.

Integrating the secret into the modal transition system

As a first step towards computing controllers, we combine the modal transition system $MTS = (S, \Sigma, \delta^\Box, \delta^\Diamond, s_0)$ and the secret Sec into a modal transition system $MTS_\#$, with distinguished logical states representing the intersection of $\mathcal{L}(MTS)$ and the complement of Sec. First, one constructs a *complete* deterministic automaton $A = (Y, \Sigma, \delta, y_0, Y_F)$ recognizing Sec, with initial state y_0, final states Y_F, and labelled transition map δ_A. Note that $y \in Y_F \Rightarrow \delta_A(y, \sigma) \in Y_F$ if the latter is defined, because Sec is upper-closed w.r.t. the prefix-order on words. Next, one computes the product $MTS_\#$ of MTS and A. The initial state of $MTS_\#$ is the pair (s_0, y_0). The set of states $S_\#$ of $MTS_\#$ and the weak transition map $\delta^\Diamond_\#$ are jointly and inductively defined by setting $\delta^\Diamond_\#((s, y), \sigma) = (s', y')$ and $(s', y') \in S_\#$ when $\delta^\Diamond(s, \sigma) = s'$ and $\delta_A(y, \sigma) = y'$. The strong transition map $\delta^\Box_\#$ is defined similarly, but replacing $\delta^\Diamond(s, \sigma)$ with $\delta^\Box(s, \sigma)$. The distinguished logical states $S^F_\#$ of $MTS_\#$ are the pairs $(s, y) \in S_\#$ such that $y \in Y_F$, i.e., for all $w \in \mathcal{L}(MTS_\#)$, $w \in Sec$ if and only if $\delta^\Diamond_\#(s_0, w) \in S^F_\#$.

As the automaton A has been chosen complete, $\mathcal{L}(MTS) = \mathcal{L}(MTS_\#)$ and $LTS \models MTS \Leftrightarrow LTS \models MTS_\#$ for all LTS (over Σ). From now on, we assume w.l.o.g. that $MTS = MTS_\#$, and we let $S^F = S^F_\#$. As Sec is upper-closed, $s \in S^F \Rightarrow \delta^\Diamond(s, \sigma) \in S^F$ if the latter is defined.

The general schema

As $\Sigma_o = \Sigma$ and $\mathcal{L}(MTS) = \mathcal{L}(\overline{MTS})$ is the supremum of $\mathcal{L}(LTS)$ for all labelled transition systems $LTS \models MTS$, in order that a controller K may be admissible for *every* model LTS of MTS, it is necessary and sufficient that $w.\sigma \in \mathcal{L}(MTS) \Rightarrow w.\sigma \in \mathcal{L}(K)$ for any word $w \in \mathcal{L}(MTS) \cap \mathcal{L}(K)$ and for any uncontrollable action $\sigma \in \Sigma \setminus \Sigma_c$. When this condition is satisfied, we say that K is an *admissible* controller of MTS (w.r.t. Σ_c and $\Sigma_o = \Sigma$).

Among the admissible controllers of MTS, we should search for controllers K such that the following condition holds for *every* labelled transition system $LTS \models MTS$ (recall that K/LTS denotes the product of LTS and K):
$\forall w \in \mathcal{L}(K/LTS)\ \exists w' \in \mathcal{L}(K/LTS)$
$\pi_a(w) = \pi_a(w') \ \wedge\ \delta^\Diamond(s_0, w') \notin S^F$.
We want to compute the maximal permissive controller K satisfying this condition.

We proceed in two steps. In a first step, we derive from MTS an LTS H with the language $\mathcal{L}(H) = \mathcal{L}(MTS)$ and with a set of states included in $S \times \mathcal{P}(S)$. The intended meaning of these states is as follows. If $\delta^\Diamond(s_0, w) = s$ in MTS, then w should lead in H to the state (s, E) where $E = \{s' \in S | \forall LTS : LTS \models MTS \wedge w \in \mathcal{L}(LTS) \Rightarrow \exists w' \in \mathcal{L}(LTS) : \pi_a(w) = \pi_a(w') \wedge \delta^\Diamond(s_0, w') = s'\}$ (thus $s \in E$). As $\mathcal{L}(w \circ MTS)$ is the infimum of $\mathcal{L}(LTS)$ for all LTS such that $LTS \models MTS$ and $w \in \mathcal{L}(LTS)$, $E = \{s' \in S | \exists w' \in \mathcal{L}(w \circ MTS) : \pi_a(w) = \pi_a(w') \wedge \delta^\Diamond(s_0, w') = s'\}$.

Given a model LTS of MTS, let us say that w *discloses* the secret Sec in LTS if $w \in \mathcal{L}(LTS) \cap Sec$ and $w' \in Sec$ for any other word $w' \in \mathcal{L}(LTS)$ such that $\pi_a(w) = \pi_a(w')$. As $\mathcal{L}(w \circ MTS)$ is the infimum of $\mathcal{L}(LTS)$ for all LTS such that $LTS \models MTS$ and $w \in \mathcal{L}(LTS)$, w discloses the secret Sec in some model LTS of MTS if and only if w discloses this secret in $w \circ MTS$. So, if w leads to state (s, E) in H, then w discloses the secret Sec in some model LTS of MTS if and only if $E \subseteq S^F$. Therefore, in a second step, we trim down H according to Ramadge and Wonham's procedure for avoiding to reach any state (s, E) with $E \subseteq S^F$. We will show that the labelled transition system $K^\dagger$ obtained in this way is the maximal permissive controller that enforces the opacity of Sec in MTS.

A preliminary construction

In the sequel, $\Sigma_{ua} = \Sigma \setminus \Sigma_a$ denotes the set of actions which are unobservable from the perspective of the attacker. For all transition maps δ and for all sets E and $L \subseteq \Sigma^*$, we let $\delta(E, \sigma) = \{\delta(s, \sigma) \,|\, s \in E\}$, $\delta(s, L) = \{\delta(s, w) \,|\, w \in L\}$, and $\delta(E, L) = \{\delta(s, w) \,|\, s \in E \wedge w \in L\}$.

Definition 2. Let $H = (\Theta, \Sigma, \delta_H, \theta_0)$ be the LTS with the set of states $\Theta \subseteq S \times \mathcal{P}(S)$ and the labelled transition map δ_H jointly and inductively defined as follows:

- let $\theta_0 = (s_0, \delta^\Box(s_0, \Sigma_{ua}^*))$ and $\theta_0 \in \Theta$,

- inductively, for each state $(s, E) \in \Theta$ and for each action $\sigma \in \Sigma$ such that $\delta^\Diamond(s, \sigma)$ is defined, let $\delta_H((s, E), \sigma) = (s', E')$ and $(s', E') \in \Theta$ where $s' = \delta^\Diamond(s, \sigma)$ and the set of states E' is given according to the case by:
  - $\sigma \notin \Sigma_a$: $E' = E \cup \delta^\Box(s', \Sigma_{ua}^*)$,
  - $\sigma \in \Sigma_a$: $E' = \delta^\Box(E, \sigma.\Sigma_{ua}^*) \cup \delta^\Box(s', \Sigma_{ua}^*)$. ⋄

Obviously, $\mathcal{L}(H) = \mathcal{L}(MTS)$. The following lemma, which is a bit technical, shows that the above construction achieves the announced goals.

Lemma 2. For any $w \in \mathcal{L}(MTS)$, $\delta_H(\theta_0, w) = (s, E) \Rightarrow s = \delta^\Diamond(s_0, w)$ and $E = \{s' \in S \mid \exists w' \in \mathcal{L}(w \circ MTS) : \pi_a(w') = \pi_a(w) \wedge \delta^\Diamond(s_0, w') = s'\}$.

Proof. The proof is by induction on w. The base of the induction is given by the case $w = \varepsilon$. Then $\delta_H(\theta_0, \varepsilon) = \theta_0 = (s_0, \delta^\Box(s_0, \Sigma_{ua}^*))$ by Def. 2. Clearly, $s_0 = \delta^\Diamond(s_0, \varepsilon)$. For $w' \in \Sigma^*$, $w' \in \Sigma_{ua}^* \Leftrightarrow \pi_a(w') = \pi_a(\varepsilon)$, and $\delta^\Box(s_0, w')$ is defined if and only if $w' \in \mathcal{L}(\underline{MTS}) = \mathcal{L}(\varepsilon \circ MTS)$ (see the observations after Def. 1). As $\delta^\Diamond(s_0, w') = \delta^\Box(s_0, w')$ if the latter is defined, the lemma holds for $w = \varepsilon$.

Assume now that the lemma holds for $w = \sigma_1 \dots \sigma_{n-1}$ (by convention, $n = 1$ means $w = \varepsilon$), and consider $w.\sigma_n \in \mathcal{L}(MTS)$ with $\sigma_n \in \Sigma$. Let $\delta_H(\theta_0, \sigma_1 \dots \sigma_i) = (s_i, E_i)$ for $1 \leq i \leq n$. As $s_n = \delta^\Diamond(s_{n-1}, \sigma_n)$ (by Def. 2) and $s_{n-1} = \delta^\Diamond(s_0, w)$ (by the induction hypothesis), $s_n = \delta^\Diamond(s_0, w.\sigma_n)$. To simplify the notation, let $\sigma = \sigma_n$ and $s = s_n$. We prove $E_n = \{s' \in S \mid \exists w' \in \mathcal{L}((w.\sigma) \circ MTS) : \pi_a(w') = \pi_a(w.\sigma) \wedge \delta^\Diamond(s_0, w') = s'\}$ by case analysis.

Case $\sigma \notin \Sigma_a$. By Def. 2, $E_n = E_{n-1} \cup \delta^\Box(s, \Sigma_{ua}^*)$, and by induction, $E_{n-1} = \{s' \in S \mid \exists w' \in \mathcal{L}(w \circ MTS) : \pi_a(w') = \pi_a(w) \wedge \delta^\Diamond(s_0, w') = s'\}$. As $\pi_a(w) = \pi_a(w.\sigma)$ and $\mathcal{L}((w.\sigma) \circ MTS) = \mathcal{L}(w \circ MTS) \cup w.\sigma.\, \mathcal{L}(\underline{MTS}, s)$ (see the observations after Def. 1), it suffices to prove $\delta^\Box(s, \Sigma_{ua}^*) = \{s' \in S \mid \exists w' \in w.\sigma.\, \mathcal{L}(\underline{MTS}, s) : \pi_a(w') = \pi_a(w.\sigma) \wedge \delta^\Diamond(s_0, w') = s'\}$. Now $w' \in w.\sigma.\, \mathcal{L}(\underline{MTS}, s) \wedge \pi_a(w') = \pi_a(w.\sigma)$ if and only if $w' = w.\sigma.v'$ and $v' \in \mathcal{L}(\underline{MTS}, s) \cap \Sigma_{ua}^*$. For $v' \in \Sigma_{ua}^*$, $v' \in \mathcal{L}(\underline{MTS}, s)$ if and only if $\delta^\Box(s, v')$ is defined, and then $\delta^\Diamond(s_0, w.\sigma.v') = \delta^\Box(s, v')$. Therefore, the lemma holds in this case.

Case $\sigma \in \Sigma_a$. By Def. 2, $E_n = \delta^\Box(E_{n-1}, \sigma.\Sigma_{ua}^*) \cup \delta^\Box(s, \Sigma_{ua}^*)$ and by induction, $E_{n-1} = \{s' \in S \mid \exists w' \in \mathcal{L}(w \circ MTS) : \pi_a(w') = \pi_a(w) \wedge \delta^\Diamond(s_0, w') = s'\}$. Accordingly, $\delta^\Box(E_{n-1}, \sigma.\Sigma_{ua}^*) = \{s" \in S \mid \exists s' \in S\, \exists w' \in \mathcal{L}(w \circ MTS)\, \exists v' \in \Sigma_{ua}^* : \pi_a(w') = \pi_a(w) \wedge \delta^\Diamond(s_0, w') = s' \wedge \delta^\Box(s', \sigma.v') = s"\}$. For s', w' and v' as above, let $w" = w'.\sigma.v'$. As $w' \in \mathcal{L}(w \circ MTS)$ and $\delta^\Diamond(s_0, w') = s'$, $\delta^\Box(s', \sigma.v')$ is defined if and only if $w" \in \mathcal{L}(w \circ MTS)$, and then $\delta^\Diamond(s_0, w") = \delta^\Box(s', \sigma.v')$. Moreover, $\pi_a(w") = \pi_a(w'.\sigma) = \pi_a(w.\sigma)$. The above relation simplifies therefore to $\delta^\Box(E_{n-1}, \sigma.\Sigma_{ua}^*) = \{s" \in S \mid \exists w" \in \mathcal{L}(w \circ MTS) : \pi_a(w") = \pi_a(w.\sigma) \wedge \delta^\Diamond(s_0, w") = s"\}$. As $\mathcal{L}((w.\sigma) \circ MTS) = \mathcal{L}(w \circ MTS) \cup w.\sigma.\, \mathcal{L}(\underline{MTS}, s)$ (see the observations after Def. 1), in order to complete the proof, it suffices to show $\delta^\Box(s, \Sigma_{ua}^*) = \{s" \in S \mid \exists v' \in \mathcal{L}(\underline{MTS}, s) : \pi_a(w.\sigma.v') = \pi_a(w.\sigma) \wedge \delta^\Diamond(s_0, w.\sigma.v') = s"\}$. This follows easily because $\pi_a(w.\sigma.v') = \pi_a(w.\sigma)$ if and only if $v' \in \Sigma_{ua}^*$ and $\delta^\Diamond(s_0, w.\sigma.v') = \delta^\Box(s, v')$ if the latter is defined. □

The construction of $K^\dagger$

As $\Theta \subseteq S \times \mathcal{P}(S)$ where S is the set of logical states of the modal transition system MTS, $H = (\Theta, \Sigma, \delta_H, \theta_0)$ is a finite LTS, with the language $\mathcal{L}(H) = \mathcal{L}(MTS) = \cup\{\mathcal{L}(LTS) \mid LTS \models MTS\}$. Our goal is now to produce $K^\dagger$ from H by removing all words $w \in \mathcal{L}(H)$ that disclose the secret Sec in some model of MTS.

As $\mathcal{L}(w \circ MTS)$ is the infimum of $\mathcal{L}(LTS)$ for all LTS such that $LTS \models MTS$ and $w \in \mathcal{L}(LTS)$, a word $w \in \mathcal{L}(H)$ discloses the secret Sec in some model of MTS if and only if it discloses the secret Sec in $w \circ MTS$. By Lemma 2, a word $w \in \mathcal{L}(H)$ discloses the secret Sec in $w \circ MTS$ if and only if $\delta_H(\theta_0, w) \in Bad$ where we let $Bad = \{(s, E) \in \Theta \mid E \subseteq S^F\}$. Enforcing the opacity of the secret Sec in all models of MTS amounts therefore to barring access to Bad states of H.

As $\mathcal{L}(H) = \cup\{\mathcal{L}(LTS) \mid LTS \models MTS\}$, a controller K is an admissible controller of all models LTS of MTS if and only if it is an admissible controller of H.

So, in order that $K = (X, \Sigma, \delta_K, x_0)$ may, for *every* model LTS of MTS, enforce the opacity of the secret in LTS (w.r.t. Σ_a) and be an admissible controller of LTS (w.r.t. Σ_c), it is necessary that the following two conditions C1 and C2 hold:

- no state (θ, x) with $\theta \in Bad$ can be reached from (θ_0, x_0) in K/H,
- K is an admissible controller of H (w.r.t. Σ_c).

According to Ramadge and Wonham's theory of state-based supervision, the maximal permissive controller $K^\dagger$ for which both conditions hold is obtained by pruning H iteratively according to the following method. Throughout the iteration, one maintains a partition $\{Good, Bad\}$ of the set of states $X = \Theta$ and a partial transition map $\delta_X : \Theta \times \Sigma \to \Theta$. Initially, Bad is the set of Bad states of H, and $\delta_X = \delta_H$. At each step in the iteration, one picks some pair of arguments $\theta \in Good$ and $\sigma \in \Sigma$ such that $\delta_X(\theta, \sigma) \in Bad$, and one removes (θ, σ) from the domain of definition of δ_X. Moreover, if σ is uncontrollable ($\sigma \notin \Sigma_c$), then one moves the considered state θ from the set $Good$ to the set Bad (which may cause the set of $Good$ states to be disconnected). The iteration stops when $\delta_X(\theta, \sigma) \in Bad$ for no pair of arguments $\theta \in Good$ and $\sigma \in \Sigma$. At this stage, let $K^\dagger$ be the induced restriction of the LTS $(Good, \Sigma, \delta_X, \theta_0)$ on states reachable from θ_0. If $\theta_0 \notin Good$,then no controller can prevent Bad states from being reached (hence no controller can enforce the opacity of the secret in all models of MTS). If $\theta_0 \in Good$, then $K^\dagger$ is the maximal permissive controller preventing Bad states from being reached in H. However, this does not entail directly that $K^\dagger$ enforces the opacity of the secret in all models of MTS, since C1 and C2 were only necessary conditions for achieving this goal. The following lemma is crucial to prove that $K^\dagger$ enforces indeed the opacity of the secret in all models of MTS.

Lemma 3. When the iterative procedure defined above is applied to the LTS H specified by Def. 2 and to the set $Bad = \{(s, E) \in \Theta \mid E \subseteq S^F\}$, the partition $\{Good, Bad\}$ of Θ stays unchanged throughout the iteration.

Proof. Assume for the sake of contradiction that (s, E) is the first element of Θ moved from $Good$ to Bad. By definition of the iterative procedure, $\delta_H((s, E), \sigma) = (s', E')$ and $(s', E') \in Bad$ for some $\sigma \in \Sigma \setminus \Sigma_c$. As $\Sigma_a \subseteq \Sigma_c$, $\sigma \notin \Sigma_a$. Therefore, by Def. 2, $E \subseteq E'$. As (s', E') was already in Bad at the initialization of the procedure, $E' \subseteq S^F$, hence $E \subseteq S^F$, contradicting the assumption that (s, E) was $Good$. □

Remark 4. If $\delta_H((s, E), \sigma) = (s', E')$ for $\sigma \in \Sigma_c \setminus \Sigma_a$ then $E \subseteq E'$ by Def. 2, hence $(s', E') \in Bad \Rightarrow (s, E) \in Bad$. Therefore, actions σ in $\Sigma_c \setminus \Sigma_a$ are in never blocked by $K^\dagger$.

Proposition 1. $K^\dagger$ enforces the opacity of the secret in all models of MTS.

Proof. Let $LTS \models MTS$ and $w \in \mathcal{L}(K^\dagger/LTS)$. We must show that there exists $w' \in \mathcal{L}(K^\dagger/LTS)$ such that $\pi_a(w) = \pi_a(w')$ and $\delta^\diamond(s_0, w') \notin S^F$ (in MTS). By Lemma 2 and the definition of the set Bad, $w \in \mathcal{L}(K^\dagger)$ entails that $\pi_a(w) = \pi_a(w')$ and $\delta^\diamond(s_0, w') \notin S^F$ for some $w' \in \mathcal{L}(w \circ MTS)$. As $\mathcal{L}(w \circ MTS)$ is the infimum of $\mathcal{L}(LTS')$ for all LTS' such that $LTS' \models MTS$ and $w \in \mathcal{L}(LTS')$, necessarily $w' \in \mathcal{L}(LTS)$. As the secret Sec is an upper-closed set, $\delta^\diamond(s_0, w') \notin S^F$ entails $\delta^\diamond(s_0, v') \notin S^F$ for all prefixes v' of w', hence $\delta_H(s_0, v')$ is not in Bad for any prefix v' of w'. As a consequence, $w' \in \mathcal{L}(K^\dagger)$ and therefore, $w' \in \mathcal{L}(K^\dagger/LTS)$. □

Theorem 5. $K^\dagger$ is maximal permissive among all admissible controllers enforcing the opacity of the secret in all models of MTS.

Proof. $K^\dagger$ is maximal permissive among the controllers of H that satisfy the two necessary conditions C1 and C2. Prop. 1 completes the proof of the theorem.

7. CONCLUSION

In a first attempt to extend opacity enforcing supervisory control to classes of transition systems loosely described by modal transition systems, we have dealt with two cases where either the attacker or the controller has full observation. The next case we have begun to investigate is when $\Sigma_c \subseteq \Sigma_a \subseteq \Sigma_o$. In this case, $Good$ states may turn to Bad, and the construction which we have presented for controller K must be iterated, i.e., the opacity control problem should be solved recursively for K/LTS. The difficulty is to show that the iteration stops.

REFERENCES

[1] E. Badouel, M.A. Bednarczyk, A.M. Borzyszkowski, B. Caillaud, P. Darondeau: Concurrent Secrets. Proc. 8th Int. Workshop on Discrete Event Systems, WODES, 2006, 51-57, and *Discrete Event Dynamic Systems*, vol. 17 no 4, 2007, 425-446.

[2] M. Ben-Kalefa, F. Lin: Supervisory Control for Opacity of Discrete Event Systems. Proc. 45th Annual Allerton Conference on Communication, Control, and Computing, Allerton House, IL, 2011, 1113-1119.

[3] J.W. Bryans, M. Koutny, L. Mazaré, P.Y.A. Ryan: Opacity Generalized to Transition Systems. *Int. Journal of Computer Security*, vol. 7 no 6, 2008, 421-435.

[4] J. Dubreil, P. Darondeau, H. Marchand: Opacity Enforcing Control Synthesis. Proc. 9th Int. Workshop on Discrete Event Systems, WODES, 2008, 28-35.

[5] J. Dubreil, P. Darondeau, H. Marchand: Supervisory Control for Opacity. *IEEE Trans. Automatic Control*, vol 55 no 5, 2010, 1089-1100.

[6] J. Dubreil, P. Darondeau, H. Marchand: Supervisory Control for Modal Specifications of Services. Proc. 10th Int. Workshop on Discrete Event Systems, WODES, 2010, 428-435.

[7] G. Feuillade, S. Pinchinat: Modal Specifications for the Control Theory of Discrete Event Systems. *Discrete Event Dynamic Systems*, vol. 17, 2007, 211-232.

[8] C.N. Hadjicostis: Supervisory Control Strategies for Enhancing System Security and Privacy. Proc. 48th Annual Allerton Conference on Communication, Control, and Computing, Allerton House, IL, 2010, 1622-1627.

[9] K.G. Larsen: Modal Specifications. in: *Automatic Verification Methods for Finite State Systems*, Springer-Verlag, LNCS vol. 407, 1990, 232-246.

[10] F. Lin: Opacity of Discrete Event Systems and its Applications. *Automatica*, vol. 47 no 3, 2011, 496-503.

[11] N. Lohmann, P. Massuthe, K. Wolf: Operating Guidelines for Finite-State Services. Proc. ICATPN, Springer-Verlag, LNCS vol. 4546, 2007, 321-341.

[12] L. Mazaré: Using Unification for Opacity Properties. Proc. 4th IFIP WG 1.7 Workshop on Issues in the Theory of Security (WITS'04), Barcelona (Spain) 2004, 165-176.

[13] L. Mazaré: Decidability of Opacity with Non-Atomic Keys. Proc. FAST'04, 2004, 71-84.

[14] P.J. Ramadge, W.M. Wonham: Supervisory Control of a Class of Discrete Event Processes. *SIAM J. of Control and Optimization*, vol. 25, 1987, 206-230.

[15] P.J. Ramadge, W.M. Wonham: On the Supremal Controllable Language of a Given Language. *SIAM J. of Control and Optimization*, vol. 25, 1987, 637-659.

[16] P.J. Ramadge, W.M. Wonham: The Control of Discrete Event Systems. *Proc. of the IEEE, Special Issue on Dynamics of Discrete Event Systems*, vol. 77, 1989, 81-98.

[17] A. Saboori, C.N. Hadjicostis: Verification of initial-state opacity in security applications of DES. Proc. of the 9th Int. Workshop on Discrete Event Systems, WODES, 2008, 328-333.

[18] A. Saboori and C.N. Hadjicostis: Opacity-Enforcing Supervisory Strategies for Secure Discrete Event Systems. Proc. 47th IEEE Conference on Decision and Control, Cancun, Mexico, 2008, 889-894.

[19] S. Takai, R. Kumar: Verification and Synthesis for Secrecy in Discrete-event Systems. Proc. American Control Conference, 2009, 4741-4746.

[20] S. Takai, Y. Oka: A formula for the supremal controllable and opaque sublanguage arising in supervisory control. *SICE Journal of Control, Measurement, and System Integration* vol. 1 no 4, 2008, 307-312.