Opacity Enforcing Supervisory Control using Non-deterministic Supervisors ⋆

Yifan Xie * **Xiang Yin** * **Shaoyuan Li** *

* *Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China. (E-mail: {xyfan1234,yinxiang,syli}@sjtu.edu.cn)*

Abstract: In this paper, we investigate the enforcement of opacity via supervisory control in the context of discrete-event systems. A system is said to be opaque if the intruder, which is modeled as a passive observer, can never infer confidentially that the system is at a secret state. The design objective is to synthesize a supervisor such that the closed-loop system is opaque even though the control policy is publicly known. We propose to use *non-deterministic supervisors*, provides a set of control decisions at each instant, to enforce opacity. Such a non-deterministic control mechanism can enhance the plausible deniability of the controlled system as the online control decision cannot be implicitly inferred from the control policy. We provide an effective approach to synthesize a non-deterministic opacity-enforcing supervisor. We show that non-deterministic supervisors are strictly more powerful than deterministic supervisors.

Keywords: Opacity, Supervisory Control, Discrete Event Systems.

1. INTRODUCTION

We investigate an information-flow security property called *opacity* in the context of Discrete-Event Systems. Opacity is a confidential property capturing whether or not the system can always *plausibly deny* the execution of a secret behavior. Due to the increasing demands for security certification in safety-critical systems, opacity has been extensively studied in the past years; see, e.g., Yin and Lafortune [2017], Lafortune et al. [2018], Yin et al. [2019]. Given a system that is verified to be non-opaque, one important problem is to *enforce* opacity via some enforcement mechanisms. For example, Cassez et al. [2012] consider the enforcement of opacity via dynamic masks that change the output information dynamically. Similarly, Ji et al. [2019] uses insertion functions to enforce opacity by adding factitious events.

One of the most widely investigated opacity enforcement mechanism is via the *supervisory control theory* Badouel et al. [2007], Saboori and Hadjicostis [2011]. In this framework, a supervisor is used to restrict the system such that the closed-loop system is opaque. For example, in Dubreil et al. [2010], the authors solve the opacity control problem by assuming that all controllable events are observable and the observation of the intruder is included in the observation of the supervisor. In Yin and Lafortune [2016], a uniform approach is provided to solve opacity-enforcing control problem without the assumption that controllable events are observable by assuming the intruder and the supervisor have the same observation. Recently in Tong et al. [2018], the authors provide an algorithm for synthesizing an opacity enforcing control without any assumption on event sets. However, it needs to assume that the control policy is not publicly known.

All existing works on opacity-enforcing supervisory control considers deterministic supervisors. Such a deterministic mechanism, however, may decrease the plausible deniability of the system. In this paper, we propose to use *non-deterministic supervisors* to enforce opacity. Unlike a deterministic supervisor that issues a specific control decision at each instant, a non-deterministic supervisor provides *a set of* control decisions and the specific control decision applied is chosen randomly via a "coin toss" manner. In other words, even if the intruder knows the control policy, it still does not know the specific control decision applied as it is decided randomly on-the-fly. Compared with the deterministic mechanism, the non-deterministic mechanism can significantly enhance the plausible deniability of the system, and, therefore, is more likely to enforce opacity.

To synthesize a non-deterministic supervisor that enforces opacity, we propose a new information-state that not only contains the state-estimate from the supervisor's point of view, but also contains the estimate of the supervisor's estimate from the intruder's point of view. In other words, the control decision should be made not only based on what the supervisor thinks about the plant, but also based on what the intruder thinks about the supervisor. Based on this information-state, we provide an effective approach that synthesizes a non-deterministic supervisor that enforces opacity. In particular, we show that using non-deterministic supervisors is strictly more powerful than using deterministic supervisors, in the sense that, there may exist a non-deterministic opacity-enforcing supervisor even when deterministic supervisors cannot enforce opacity. To the best of our knowledge, non-deterministic supervisors have only been applied to the standard supervisory control problem for safety and non-blockingness Inan [1994],

⋆ This work was supported by the National Natural Science Foundation of China (61803259, 61833012) and Shanghai Jiao Tong University Scientific and Technological Innovation Funds..

Kumar et al. [2005]; it has not yet been applied to the opacity-enforcement problem.

2. PRELIMINARIES

A DES is modeled as a deterministic finite-state automaton $G = (X, \Sigma, \delta, x_0)$, where X is the finite set of states, Σ is the finite set of events, $\delta : X \times \Sigma \to X$ is the partial transition function, where $\delta(x, \sigma) = y$ means that there is a transition labeled by event σ from state x to y, and $x_0 \in X$ is the initial state. For simplicity, we write $\delta(x, s)$ as $\delta(s)$ when $x = x_0$. The language generated by G is denote $\mathcal{L}(G)$.

When the system is partially observed, Σ is partitioned into two disjoint sets: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, where Σ_o is the set of observable events and Σ_{uo} is the set of unobservable events. We denote by $P : \Sigma^* \to \Sigma_o^*$ the natural projection from Σ to Σ_o. The natural projection is also extended to $P : 2^{\Sigma^*} \to 2^{\Sigma_o^*}$.

In the framework of supervisory control, we assume that the events set is further partitioned as $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$, where Σ_c is the set of controllable events and Σ_{uc} is the set of uncontrollable events. A control decision $\gamma \in 2^\Sigma$ is a set of events such that $\Sigma_{uc} \subseteq \gamma$, namely uncontrollable events can never be disabled. We define $\Gamma = \{\gamma \in 2^\Sigma : \Sigma_{uc} \subseteq \gamma\}$ as the set of control decisions. Then a *deterministic supervisor* is a function $S : P(\mathcal{L}(G)) \to \Gamma$. The language generated by the controlled system, denoted by $\mathcal{L}(S/G)$, is defined recursively in the usual manner; see, e.g, Cassandras and Lafortune [2008] (page 139).

We assume that system G has a "secret", which is modeled as a set of secret states $X_S \subseteq X$. Furthermore, we consider a passive *intruder* having the following capabilities:

- A1 The intruder knows the system model;
- A2 The intruder can observe the occurrences of observable events.

Such an intruder is essentially an "eavesdropper" and we say that system G is *opaque* w.r.t. X_S and Σ_o if

$$(\forall s \in \mathcal{L}(G) : \delta(s) \in X_S)(\exists t \in \mathcal{L}(G) : \delta(t) \notin X_S)[P(s) = P(t)].$$

That is, the intruder cannot infer for sure that the system is in a secret state based on the information flow.

When the original system is not opaque, one approach is to design a supervisor S such that the closed-loop system S/G is opaque; this is referred to as the *opacity-enforcing control problem*. In this setting, however, the implementation of such a supervisor may become a public information. To capture this severe scenario, we assume:

- A3 The intruder knows the functionality of the supervisor, i.e., the control policy.

Note that this knowledge together with the assumption that the intruder and the observer both observe Σ_o imply that the intruder knows precisely the control decision applied at each instant. Formally, we say that a deterministic supervisor $S : P(\mathcal{L}(G)) \to \Gamma$ enforces opacity on G if for any string $s \in \mathcal{L}(S/G)$ such that $\delta(s) \in X_S$, there exists a string $t \in \mathcal{L}(S/G)$ such that $\delta(s) \notin X_S$ and $P(s) = P(t)$.

Finally, we introduce some operators that will be used in this paper. Given a set of states $m \in 2^X$, we denote

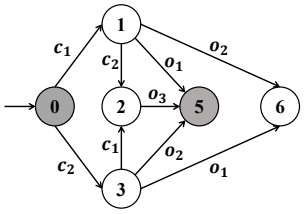

Fig. 1. System G with $\Sigma_c = \{c_1, c_2\}$, $\Sigma_o = \{o_1, o_2, o_3\}$, $X_S = \{0, 5\}$.

by $UR_\gamma(m)$ the *unobservable reach* of m under control decision $\gamma \in \Gamma$, i.e.,

$$UR_\gamma(m) = \{\delta(x, w) \in X : x \in m, w \in (\Sigma_{uo} \cap \gamma)^*\}. \quad (1)$$

We also denote by $NX_\sigma(m)$ the *observable reach* of m upon the occurrence of an observable event $\sigma \in \Sigma_o$, i.e.,

$$NX_\sigma(m) = \{\delta(x, \sigma) \in X : x \in m\}. \quad (2)$$

3. ENFORCING OPACITY USING NON-DETERMINISTIC SUPERVISORS

3.1 Motivating Example

Example 1. Let us consider system G shown in Fig. 1 with $\Sigma_o = \Sigma_{uc} = \{o_1, o_2, o_3\}$ and $X_S = \{0, 5\}$. This system is not opaque since upon the occurrence of o_3, the intruder knows for sure that the system is at secret state 5.

For this system, we cannot even synthesize a deterministic supervisor to enforce opacity. To see this clearly, let us evaluate what the supervisor can do initially. We have $\Gamma = \{\emptyset, \{c_1\}, \{c_2\}, \{c_1, c_2\}\}$. [1] Clearly, the supervisor cannot choose $\emptyset$ as the initial control decision; otherwise secret state 0 will be the only reachable state. Also, the supervisor cannot make $\{c_1\}$ initially. This is because, under this control decision and by observing event o_1, the intruder knows for sure that the system is at state 5 which is reached via $1 \xrightarrow{o_1} 5$. Note that transition $3 \xrightarrow{o_1} 6$ cannot provide the plausible deniability as state 3 is not reachable under $\{c_1\}$ as c_2 is disabled initially. For the same reason, making $\{c_2\}$ initially will also reveal the secret. Finally, decision $\{c_1, c_2\}$ is also problematic initially as it makes state 2 reachable from which transition $2 \xrightarrow{o_3} 5$ will also reveal the secret. Therefore, we cannot enforce opacity for this system using a deterministic supervisor.

However, one can enforce opacity using the following control mechanism. Initially, the supervisor randomly chooses either it decides to enable c_1 or to enable c_2, but not both. In other words, the control policy initially is a set $\{\{c_1\}, \{c_2\}\}$ and the specific choice is made randomly on-the-fly. Therefore, upon the occurrence of o_1 or o_2, the intruder does not know whether this event is from state 1 or from state 3 since it does not know whether or not the initial control decision is $\{c_1\}$ or $\{c_2\}$. On the other hand, since c_1 and c_2 will not be enabled simultaneously, state 2 is not reachable; hence, event o_3, which reveals the secret, will also not occur. □

The above example shows that using non-deterministic control mechanism is more powerful than the deterministic

[1] For the sake of simplicity, uncontrollable events are omitted in each control decision, i.e., $\emptyset$ standards for $\{o_1, o_2, o_3\}$ in this example.

one for the purpose of enforcing opacity. Using a non-deterministic decision framework will, on the one hand, enhance the plausible deniability of the secret behavior of the system and, on the other hand, decrease the confidentiality of the intruder's knowledge about the system.

3.2 Non-deterministic Supervisor

Compared with a deterministic supervisor that issues a specific control decision at each instant, a non-deterministic supervisor works as follows. At each instant, the non-deterministic supervisor provides *a set of possible control decisions*. Then it non-deterministically picks a specific control decision from this set in a "coin-toss" manner. In other words, the control policy only determines a set of allowed decisions, but the specific control decision chosen is unknown *a priori*, which is a *realization* under the policy. Therefore, the supervisor makes decision not only based on observable events, but also depends on the specific control decisions chosen along the trajectory.

To define the "history" of the supervisor, we introduce the notion of the *extended string* which is an alternating sequence of control decisions and events in the form of

$$\rho = \gamma_1\sigma_1\gamma_2\sigma_2\dots\gamma_n\sigma_n \in (\Gamma\Sigma)^*,$$

where $\gamma_i \in \Gamma$ is the ith selected control decision and $\sigma_i \in \gamma_i$ is the ith event. Since some events are unobservable for the supervisor, we define a mapping

$$\mathcal{O} : (\Gamma\Sigma)^* \to (\Gamma\Sigma_o)^*,$$

such that, for any extended string, it erases

- each unobservable event together with its successor control decision; and
- the successor control decision of the last observable event.

Formally, for any extended string

$$\rho = \gamma_1\sigma_1\gamma_2\sigma_2\dots\gamma_n\sigma_n,$$

let $1 \le i_1 < i_2 < \dots < i_k \le n$ be all indices such that $\sigma_{i_j} \in \Sigma_o$, then we have

$$\mathcal{O}(\rho) = \begin{cases} \epsilon & \text{if } \{i_1,\dots,i_k\} = \emptyset \\ \gamma_1(\sigma_{i_1}\gamma_{i_1+1})\cdots(\sigma_{i_{k-1}}\gamma_{i_{k-1}+1})\sigma_{i_k} & \text{if } \{i_1,\dots,i_k\} \neq \emptyset \end{cases} \tag{3}$$

Such an projected extended string is also referred to as a *decision history*. We force a projected extended string to also end up with an (observable) event as this is the instant where the supervisor needs to make a new decision.

The non-deterministic supervisor is defined as a function

$$S_N : (\Gamma\Sigma_o)^* \to 2^\Gamma$$

that maps a decision history $\mathcal{O}(\rho) \in (\Gamma\Sigma_o)^*$ to a set of possible control decisions. This definition essentially says that, although the control policy is non-deterministic, the supervisor knows the *realization history*, i.e., which specific decision was picked at each previous instant.

Definition 1. Let S_N be a non-deterministic supervisor. The set of extended strings generated by the closed-loop system, denoted by $\mathcal{L}_e(S_N/G)$, is defined recursively by:

- $\epsilon \in \mathcal{L}_e(S_N/G)$;
- $\gamma_1\sigma_1 \in \mathcal{L}_e(S_N/G)$ if $\gamma_1 \in S_N(\epsilon)$ and $\sigma_1 \in \gamma_1$;
- For any $\rho = \gamma_1\sigma_1\dots\gamma_n\sigma_n\gamma_{n+1}\sigma_{n+1} \in (\Gamma\Sigma)^*$, $n \ge 1$, we have $\rho \in \mathcal{L}_e(S_N/G)$, if and only if
  - $\gamma_1\sigma_1\dots\gamma_n\sigma_n \in \mathcal{L}_e(S_N/G)$; and
  - $\sigma_1\dots\sigma_n\sigma_{n+1} \in \mathcal{L}(G)$; and
  - $\sigma_{n+1} \in \gamma_{n+1}$; and
  - $\gamma_{n+1} \in \begin{cases} \{\gamma_n\} & \text{if } \sigma_n \in \Sigma_{uo} \\ S_N(\mathcal{O}(\gamma_1\sigma_1\dots\gamma_n\sigma_n)) & \text{if } \sigma_n \in \Sigma_o \end{cases}$

The intuition of the above definition is as follows. At each instant, if no observable event occurs, then the applied control decision should not change. On the other hand, if a new observable event occurs, then the supervisor may choose a specific control decision from the set of all possible control decisions provided by S_N.

For any extended string $\rho = \gamma_1\sigma_1\dots\gamma_n\sigma_n \in (\Gamma\Sigma)^*$, we denote by $\rho|_\Sigma$ the projection to Σ^*, i.e., $\rho|_\Sigma = \sigma_1\dots\sigma_n$. A string $s \in \Sigma^*$ is said to be generated by S_N/G if there exists an extended string $\rho \in \mathcal{L}_e(S_N/G)$ such that $\rho|_\Sigma = s$. We define $\mathcal{L}(S_N/G) = \{\rho|_\Sigma \in \Sigma^* : \rho \in \mathcal{L}_e(S_N/G)\}$ as the language generated by the closed-loop system.

Therefore, when an extended string $\rho \in \mathcal{L}_e(S_N/G)$ is generated, the supervisor observes $\mathcal{O}(\rho)$. Then for any observable extended string $\rho \in \mathcal{O}(\mathcal{L}_e(S_N/G))$, we define

$$\hat{\mathcal{E}}_{S_N}(\rho) = \left\{ \delta(\rho'|_\Sigma) : \begin{array}{c} \exists \rho' \in \mathcal{L}_e(S_N/G) \cap (\{\epsilon\} \cup (\Gamma\Sigma)^*(\Gamma\Sigma_o)) \\ \text{s.t. } \mathcal{O}(\rho') = \rho \end{array} \right\} \tag{4}$$

$$\mathcal{E}_{S_N}(\rho) = \{\delta(\rho'|_\Sigma) : \exists \rho' \in \mathcal{L}_e(S_N/G) \text{ s.t. } \mathcal{O}(\rho') = \rho\}. \tag{5}$$

That is, $\hat{\mathcal{E}}_{S_N}(\rho)$ is the state-estimate of the supervisor immediately after observing an observable event, while $\mathcal{E}_{S_N}(\rho)$ is the state-estimate of the supervisor with the observable tile included. These state estimates can be computed recursively as follows Yin and Lafortune [2016]:

- $\hat{\mathcal{E}}_{S_N}(\epsilon) = \{x_0\}$;
- For any $\rho = \rho'\gamma\sigma \in \mathcal{O}(\mathcal{L}_e(S_N/G))$, we have
  - $\mathcal{E}_{S_N}(\rho') = UR_\gamma(\hat{\mathcal{E}}_{S_N}(\rho'))$; and
  - $\hat{\mathcal{E}}_{S_N}(\rho) = NX_\sigma(\mathcal{E}_{S_N}(\rho'))$.

3.3 Opacity of Non-deterministic Control Systems

In the definition of opacity of control systems, the intruder model has been specified by A1-A3. Here, we still consider the same intruder model, but we explain A3 more clearly in the non-deterministic setting.

A3' The intruder knows the functionality of the supervisor. This means that it knows the set of all possible control decisions the supervisor may pick according to the control policy. However, it does not know which specific control decision the supervisor picks online.

This assumption is reasonable in many applications as long as the communication channel between supervisor and the actuator is reliable. Then under this setting, when the supervisor observes $\rho \in \mathcal{O}(\mathcal{L}_e(S_N/G))$, the intruder can only observes $\rho|_\Sigma \in P(\mathcal{L}(S_N/G))$. Therefore, the state estimate of the intruder essentially is more uncertainty, which needs to estimate all possible realizations consistent with the control policy and the observation. Then for any observable string $s \in P(\mathcal{L}(S_N/G))$, we define $X_I(s)$ as the state estimate of the intruder, i.e.,

$$X_I(s) = \{\delta(s') : \exists s' \in \mathcal{L}(S_N/G) \text{ s.t. } P(s') = s\}. \tag{6}$$

Then opacity of control systems under non-deterministic supervisors is defined as follows.

Definition 2. Let $S_N : (\Gamma\Sigma_o)^* \to 2^\Gamma$ be a non-deterministic supervisor. We say the closed-loop system S_N/G is opaque (w.r.t. Σ_o and X_S) if $\forall s \in P(\mathcal{L}(S_N/G))$:$X_I(s) \not\subseteq X_S$.

Then the opacity-enforcing control synthesis problem is formulated as follows.

Problem 1. Given system G and secret states $X_S \subseteq X$, synthesize a nondeterministic supervisor $S_N : (\Gamma\Sigma_o)^* \to 2^\Gamma$ such that S_N/G is opaque.

The estimate of the supervisor and the estimate of the intruder can be related as follows. Since the intruder observes strictly less than the supervisor, its estimate of the system is essentially the union of its estimate of all possible supervisor's knowledge about the system. To see this more clearly, for any observable string $s \in P(\mathcal{L}(S_N/G))$, we also define

$$\hat{\mathcal{E}}_I(s) = \{\hat{\mathcal{E}}_{S_N}(\rho) \in 2^X : \rho \in \mathcal{O}(\mathcal{L}_e(S_N/G)) \text{ s.t. } \rho|_\Sigma = s\} \tag{7}$$

$$\mathcal{E}_I(s) = \{\mathcal{E}_{S_N}(\rho) \in 2^X : \rho \in \mathcal{O}(\mathcal{L}_e(S_N/G)) \text{ s.t. } \rho|_\Sigma = s\} \tag{8}$$

as the intruder's estimate of the state-estimations of the supervisor. Then we have the following result.

Proposition 1. For any $s \in P(\mathcal{L}(S_N/G))$, we have

$$X_I(s) = \cup\mathcal{E}_I(s).$$

4. INFORMATION STATE AND ITS FLOW

4.1 Proposed Information Structure

In the deterministic control problem, it is known that 2^X is sufficient to realize an opacity-enforcing supervisor. That is, a supervisor can be represented as $S : 2^X \to \Gamma$ which recursively estimates the state of the system and makes decision based on the estimate.

In the non-deterministic control problem, the supervisor and the intruder observe different information. Hence, the supervisor needs to make decision based on both the state estimates of itself and that of the intruder. To separate the observation of the supervisor and the intruder, we propose the following information-state space

$$I := 2^X \times 2^{2^X}.$$

Intuitively, the first component aims to represent the state estimate of supervisor, while the second component aims to represent intruder's knowledge of the supervisor.

Formally, given a non-deterministic supervisor S_N and let $\rho \in \mathcal{O}(\mathcal{L}_e(S_N/G))$ be a decision history observed by the supervisor. We define

$$\mathcal{I}(\rho) = (\hat{\mathcal{E}}_{S_N}(\rho), \hat{\mathcal{E}}_I(\rho|_\Sigma)) \in 2^X \times 2^{2^X}$$

as the information-state reached by ρ under S_N. Clearly, we have $\hat{\mathcal{E}}_{S_N}(\rho) \in \hat{\mathcal{E}}_I(\rho|_\Sigma)$ by definition.

Definition 3. We say that a non-deterministic supervisor $S_N : (\Gamma\Sigma_o)^* \to 2^\Gamma$ is *information-state-based* (IS-based) if

$$\forall \rho, \rho' \in \mathcal{O}(\mathcal{L}_e(S_N/G)) : \mathcal{I}(\rho) = \mathcal{I}(\rho') \Rightarrow S_N(\rho) = S_N(\rho').$$

In other words, an IS-based non-deterministic supervisor can be represented as

$$S_N : I \to 2^\Gamma$$

which makes control decision based on the proposed information state. Hereafter, we will restrict our attention to IS-based supervisors.

As we discussed earlier, the first component can be computed recursively based on ρ. However, the question is how to compute the second component. To this end, we should not only know the control decision for ρ, but should also know the control decisions for those ρ' such that $\rho|_\Sigma = \rho'|_\Sigma$. In the remaining part of this section, we will elaborate on how $\hat{\mathcal{E}}_I(\rho|_\Sigma)$ can be computed recursively and by what information.

4.2 Micro/Macro States and Decisions

Before we proceed further, we define some necessary concepts. First, we introduce the notion of micro-state, which is used to represent the knowledge of supervisor.

Definition 4. (Micro-State) A *micro-state* $m \in 2^X$ is a set of states and we define $M = 2^X$ as the set of micro-states. An *augmented micro-state* $m^+ = (m, \gamma) \in 2^X \times \Gamma$ is a micro-state augmented with a control decision and we define $M^+ = 2^X \times \Gamma$ as the set of augmented micro-states.

Then, we define the notion of macro-state, which is used to represent the knowledge of intruder about the supervisor.

Definition 5. (Macro-State) A *macro-state* $\mathbf{m} = \{m_1, m_2, \ldots, m_n\} \subseteq 2^X$ is a set of micro-states and and we define $\mathbb{M} = 2^{2^X}$ as the set of macro-states. An *augmented macro-state* $\mathbf{m}^+ = \{(m_1, \gamma_1), (m_2, \gamma_2), \ldots, (m_n, \gamma_n)\} \subseteq 2^X \times \Gamma$ is a set of augmented micro-states and we define $\mathbb{M}^+ = 2^{2^X \times \Gamma}$ as the set of augmented macro-states.

In order to estimate the knowledge of the intruder, we should not only know the decision of the supervisor at a specific micro-state, but also should know the decisions at other micro-states in the same macro-state, which means that these micro-states are indistinguishable from the intruder's point of view. This leads to the notion of macro-control-decision.

Definition 6. (Macro-Control-Decision) A *macro-control-decision* is a set in the form of

$$d = \{(m_1, \Gamma_1), (m_2, \Gamma_2), \ldots, (m_n, \Gamma_n)\} \subseteq 2^X \times 2^\Gamma,$$

where each (m_i, Γ_i) is a pair of micro-state and a non-deterministic control decision (a set of control decisions). We denote by $D = 2^{2^X \times 2^\Gamma}$ the set of macro-control-decisions.

Let $\mathbf{m} = \{m_1, m_2, , \ldots, m_n\} \in \mathbb{M}$ be a macro-state and $d \in D$ be a macro-control-decision. We say that d is *compatible* with $\mathbf{m}$ if it is in the form of

$$d = \{(m_1, \Gamma_1), (m_2, \Gamma_2), \ldots, (m_n, \Gamma_n)\} \subseteq 2^X \times 2^\Gamma,$$

i.e., d essentially assigns each micro-state $m_i \in \mathbf{m}$ a non-deterministic control decision $\Gamma_i \in 2^\Gamma$.

The unobservable reach of a macro-control-decision $d \in D$ is defined by

$$\odot(d) = \{(m', \gamma) : \exists (m, \Gamma) \in d, \gamma \in \Gamma \text{ s.t. } m' = UR_\gamma(m)\}.$$

Let $\mathbf{m}^+$ be an augmented macro-state and $\sigma \in \Sigma_o$ be an observable event. Then the observable reach of $\mathbf{m}^+$ upon the occurrence of σ is defined as

$$\widehat{NX}_\sigma(\mathbf{m}^+) = \{m' : \exists (m, \gamma) \in \mathbf{m}^+ \text{ s.t. } m' = NX_\sigma(m) \wedge \sigma \in \gamma\}.$$

4.3 Information-Flow Analysis

Now, suppose that an IS-based non-deterministic supervisor $S_N : I \to 2^\Gamma$ is given and let $\mathbf{m} = \{m_1, \ldots, m_k\}$ be a macro-state representing the intruder's estimate of the supervisor's knowledge. We define

$$d_{S_N}(\mathbf{m}) = \{(m_1, S_N(m_1, \mathbf{m})), \ldots, (m_k, S_N(m_k, \mathbf{m}))\}$$

as the macro-control-decision made by IS-based supervisor S_N at macro-state $\mathbf{m}$.

Initially, the state-estimate of the supervisor is $m_0 = \{x_0\}$ and the intruder believes that this is the unique estimate of the system with estimate $\mathbf{m}_0 = \{m_0\}$.

Then the supervisor allows non-deterministic control decision $\Gamma_0 = S_N(\epsilon) = S_N(m_0, \mathbf{m}_0)$. Note that, we have pre-specified that the supervisor is IS-based. Therefore, we denote the control decision information at this instant by a macro-control-decision $d_{S_N}(\mathbf{m}_0) = \{(m_0, S_N(m_0, \mathbf{m}_0))\}$, which means that "*the supervisor will make control decision if its state-estimate is* m_0". Note that, at this instant, $d_{S_N}(\mathbf{m}_0)$ is a singleton as the intruder does not yet have ambiguity about the supervisor, i.e., $\mathbf{m}_0 = \{m_0\}$.

Once the allowed decision set Γ_0 is specified, the supervisor will pick a concrete control decision in it. The intruder does not know which decision is chosen while the supervisor knows. Suppose that $\Gamma_0 = \{\gamma_0^1, \ldots, \gamma_0^k\}$ contains k control decisions. Then the intruder's knowledge about the supervisor is

$$\begin{aligned}\mathbf{m}_0^+ &= \odot\left(d_{S_N}(\mathbf{m}_0)\right) \qquad (9)\\ &= \{(UR_{\gamma_0^1}(m_0), \gamma_0^1), \ldots, (UR_{\gamma_0^k}(m_0), \gamma_0^k)\}\\ &= \{(m_0^1, \gamma_0^1), \ldots, (m_0^k, \gamma_0^k)\},\end{aligned}$$

which means that the supervisor's estimate (with the unobservable tile) is possibly $UR_{\gamma_0^i}(m_0)$ and the control decision applied is γ_0^i. Note that, the supervisor knows which augmented micro-state (m_0^i, γ_0^i) it is at precisely.

Then when a new observable event $\sigma \in \Sigma_o$ occurs, and the intruder updates its knowledge to

$$\mathbf{m}_1 = \widehat{NX}_\sigma(\mathbf{m}_0^+) = \{m_1^1, \ldots, m_2^p\}. \qquad (10)$$

Now, let us assume that, after some steps, the intruder's knowledge about the supervisor (immediately after the occurrence of an observable event) is

$$\mathbf{m}_n = \{m_n^1, \ldots, m_n^k\}, \qquad (11)$$

Note that the supervisor knows the exact state estimate, i.e., $m_n^i \in \mathbf{m}_n$, and for each m_n^i, it allows control decisions $\Gamma_i = S_N(m_n^i, \mathbf{m}_n)$. Therefore, the corresponding marco-control-decision is

$$d_{S_N}(\mathbf{m}_n) = \{(m_n^1, S_N(m_n^1, \mathbf{m}_n)), \ldots, (m_n^k, S_N(m_n^k, \mathbf{m}_n))\}. \qquad (12)$$

Then the intruder's knowledge about the supervisor by adding this control information becomes

$$\mathbf{m}_n^+ = \odot(d_{S_N}(\mathbf{m}_n)), \qquad (13)$$

which is an augmented marco-state containing at most $\sum_{i=1}^k |S(m_n^i, \mathbf{m}_n)|$ augmented micro-states.

Based on the above discussion, suppose that the intruder observes $\sigma_1 \ldots \sigma_n \in P(\mathcal{L}(S_N/G))$ and by knowing the fact that S_N is an IS-based supervisor, it induces the following sequence

$$\mathbf{m}_0 \xrightarrow{d_0} \mathbf{m}_0^+ \xrightarrow{\sigma_1} \mathbf{m}_1 \xrightarrow{d_1} \ldots \xrightarrow{\sigma_n} \mathbf{m}_n \xrightarrow{d_n} \mathbf{m}_n^+, \qquad (14)$$

where $\mathbf{m}_0 = \{\{x_0\}\}$, $d_i = d_{S_N}(\mathbf{m}_i)$, $\mathbf{m}_i^+ = \odot(d_i)$ and $\mathbf{m}_{i+1} = \widehat{NX}_{\sigma_{i+1}}(\mathbf{m}_i^+)$.

For any augmented macro-state $\mathbf{m}^+$, we define $\Xi(\mathbf{m}^+) = \{m \in M : (m, \gamma) \in \mathbf{m}^+\}$ as the macro-state obtained by removing the control decision components from $\mathbf{m}^+$. Then the following theorem reveals that the above defined states $\mathbf{m}_n$ and $\Xi(\mathbf{m}_n^+)$ are indeed $\hat{\mathcal{E}}_I(\sigma_1 \ldots \sigma_n)$ and $\mathcal{E}_I(\sigma_1 \ldots \sigma_n)$, respectively.

Theorem 1. Let S_N be an IS-based non-deterministic supervisor and $\sigma_1 \ldots \sigma_n \in P(\mathcal{L}(S_N/G))$ be an observable string available to the intruder. Let $\mathbf{m}_n$ and $\mathbf{m}_n^+$ be states induced by $\sigma_1 \ldots \sigma_n$ and S_N according to Equation (14). Then we have (i) $\mathbf{m}_n = \hat{\mathcal{E}}_I(s)$; and (ii) $\Xi(\mathbf{m}_n^+) = \mathcal{E}_I(s)$.

Example 2. Let us consider system G in Figure 1. We consider a non-deterministic supervisor S_N defined by $\forall \rho \in (\Gamma\Sigma_o)^* : S_N(\rho) = \{\{c_1\}, \{c_2\}\}$. Clearly. S_N is IS-based. Initially, the supervisor's estimate is $m_0 = \{0\}$ and the intruder's estimate of the supervisor is $\mathbf{m}_0 = \{\{0\}\}$. The first macro-control decision is $d_{S_N}(\mathbf{m}_0) = \{(\{0\}, \{\{c_1\}, \{c_2\}\})\}$. Then the intruder's knowledge is updated to $\mathbf{m}_0^+ = \{(\{0, 1\}, \{c_1\}), (\{0, 3\}, \{c_2\})\}$ according to Eq. (9). If event o_1 is observed, then the intruder update its knowledge to $\mathbf{m}_1 = \{\{5\}, \{6\}\}$ according to Eq. (10), which means that the intruder guesses that the supervisor's state-estimate is either $\{5\}$ or $\{6\}$ based on the information available.

5. SUPERVISOR SYNTHESIS PROCEDURE

5.1 Bipartite Transition System

By the analysis in the previous section, we see that the update of the intruder's knowledge consists of two steps: one is when the supervisor picks a macro-control-decision and the other is when a new observable event occurs. To separate these two steps, we adopt the idea of the bipartite transition system proposed in Yin and Lafortune [2016].

Definition 7. A generalized bipartite transition system (G-BTS) T w.r.t. G is a 7-tuple

$$T = (Q_Y, Q_Z, h_{YZ}, h_{ZY}, \Sigma_o, D, y_0).$$

where

- $Q_Y \subseteq \mathbb{M}$ is a set of macro-states;
- $Q_Z \subseteq \mathbb{M}^+$ is the set of augmented macro-states;
- $h_{YZ} : Q_Y \times D \to Q_Z$ is the transition function from Y-states to Z-states satisfying: for any $h_{YZ}(\mathbf{m}, d) = \mathbf{m}^+$, we have
  - d is *compatible* with $\mathbf{m}$; and
  - $\mathbf{m}^+ = \odot(d)$.
- $h_{ZY} : Q_Z \times \Sigma_o \to Q_Y$ is the transition function from Z-states to Y-states satisfying: for any $h_{ZY}(\mathbf{m}^+, \sigma) = \mathbf{m}$, $\sigma \in \Sigma_o$, we have
  - $\mathbf{m} = \widehat{NX}_\sigma(\mathbf{m}^+)$.
- D is the set of macro-control-decisions;
- Σ_o is the set of observable events of system G;
- $y_0 = \{\{x_0\}\} \in Q_Y$ is the initial Y-state.

The G-BTS essentially captures the information-flow analyzed in Section 4. Specifically, at each Y-state, the IS-based supervisor makes a macro-control-decision d and

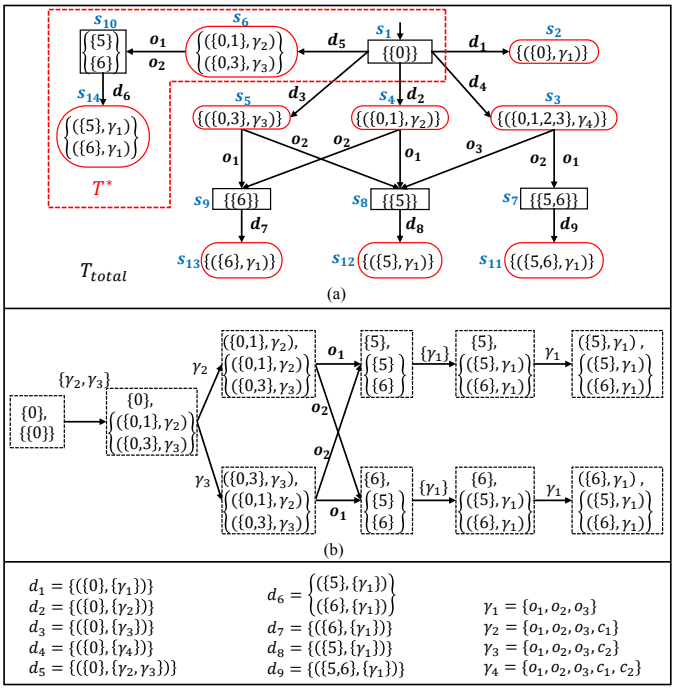

Fig. 2. (a) An Example of the G-BTS, where rectangular states represent Y-states and oval states represent Z-states. (b) Decision diagram of the synthesized non-deterministic supervisor.

then moves to a Z-state by updating of the intruder knowledge via unobservable reaches under the issued macro-control-decision d. When a new observable event $\sigma \in \Sigma_o$ occurs at a Z-state, we move to a Y-state by computing the observable reach, and so forth.

Example 3. Again, we consider system G in Figure 1. An example of the G-BTS is shown in Figure 2(a), in which rectangular states represent Y-states and oval states represent Z-states. States are named by $s_1, \ldots, s_{14}$. The initial Y-state is $s_1 = \{\{0\}\}$, from which macro-control-decisions $d_1, \ldots, d_5$ that are compatible with s_1 can be made. For example, if the macro-control-decision made is $d_5 = \{(\{0\}, \{\{c_1\}, \{c_2\}\})\}$, then we move to Z-state $s_6 = \odot(d_5)$. From this state, observable events o_1 and o_2 can occur, and both lead to the same Y-state s_{10}.

5.2 Synthesis of Nondeterministic Supervisors

Now, we present how to synthesize non-deterministic opacity-enforcing supervisors. Given a G-BTS T, for any Y-state $y \in Q_Y$, we define

$$C_T(y) := \{d \in D : h_{YZ}(y, d)!\}$$

as the set of macro-control-decisions defined at y in T. Also, we say that a Y-state y is *consistent* if $C_T(y) \neq \emptyset$; and a Z-state z is *consistent* if, for any $\sigma \in \Sigma_o$, we have

$$h_{ZY}(z, \sigma)! \Leftrightarrow (\exists (m, \gamma) \in z)[NX_\sigma(m) \neq \emptyset \wedge \sigma \in \gamma\}].$$

Intuitively, a Y-state is consistent if at least one macro-control-decision is defined and a Z-state is consistent if all feasible events are defined. Consistency is required for the purpose of control as the supervisor should be able to provide a control decision for any observation. We denote by Q^T_{const} the set of consistent states in T and we call T consistent if all reachable states are consistent.

Then given an IS-based non-deterministic supervisor S_N, we say that S_N is *included* in a consistent G-BTS T if

$$\forall \rho \in P(\mathcal{L}(S_N/G)) : d_{S_N}(\mathcal{I}(\rho)) \in C_T(\mathcal{I}(\rho))$$

As discussed earlier, we restrict our attention to IS-based supervisors. Our approach for synthesizing non-deterministic opacity-enforcing supervisors is to (i) first construct a G-BTS that includes as many opacity-enforcing supervisor as possible; and (ii) then extract one IS-based supervisor from it.

More specifically, if an IS-based supervisor is included in a G-BTS T, then by Theorem 1, we know that the Z-state z reached is essentially the set of all possible state-estimates of the supervisor. Moreover, by Proposition 1, we know that $\cup\Xi(z) = X_I(s)$, where s is the observation leading to the Z-state. Therefore, to make sure that the closed-loop system S_N/G is opaque, it suffices to guarantee that, for any Z-state z reached, we have

$$\cup\Xi(z) \nsubseteq X_S.$$

To this end, we define

$$Q_{reveal} = \{z \in \mathbb{M}^+ : \cup\Xi(z) \subseteq X_s\}$$

as the set of *secret-revealing* Z-states.

In order to synthesize an IS-based supervisor, first, we construct the largest G-BTS w.r.t G that enumerates all the feasible transitions satisfying the constraints of h_{YZ} and h_{ZY}. We denote such a largest G-BTS by T_{total}. Then, we need to delete all secret-revealing Z-states in T_{total} and obtain a new G-BTS

$$T_0 = T_{total}\restriction_{(Q_Y \cup Q_Z) \setminus Q_{reveal}},$$

where $T\restriction_Q$ denotes the G-BTS obtained by restricting the state-space of T to $Q \subseteq Q_Y \cup Q_Z$.

However, by deleting secret-revealing states, the resulting G-BTS may become inconsistent. Hence, we also need to delete inconsistent states recursively. Specifically, we define an operator F that maps a G-BTS to a new G-BTS by:

$$F : T \mapsto T\restriction_{Q^T_{const}},$$

and we define $T^* = \lim_{k\to\infty} F^k(T_0)$ as the largest consistent G-BTS in which there is no secret-revealing state. The correctness of the iteration is similar to the deterministic case in Yin and Lafortune [2016]. We illustrate the procedure by the following example.

Example 4. Consider again system G in Figure 1. First, we construct the largest G-BTS T_{total} which is in fact the structure shown in Figure 2(a). For sake of simplicity, some macro-control-decisions with redundant information are omitted in T_{total}. For example, $d = \{(\{0\}, \{\gamma_1, \gamma_2\})\}$ is not listed at state s_1, since $\gamma_1 \subset \gamma_2$ and macro-control-decision d_2 is sufficient enough to carry this information. Note that Z-states s_2 and s_{12} are secret-revealing states since $\cup\Xi(s_2) = \{0\} \subseteq X_S$ and $\cup\Xi(s_{12}) = \{5\} \subseteq X_S$. Therefore, we need to deleted states s_2 and s_{12}. However, this creates inconsistent states s_8, which needs to be deleted. Again, this further creates inconsistent states s_3, s_4 and s_5. So we also need to delete them and obtain the final structure T^* containing states s_1,s_6,s_{10} and s_{14}, which is the largest G-BTS having no secret-revealing state.

Based on T^*, we synthesize an IS-based non-deterministic supervisor as follows. First, we construct a sub-system of T^* such that at each Y-state, there is only one macro-

control decision defined. This can be done by a depth-first-search or a breath-first-search starting from the initial Y-state. Since we do not consider any other design objective in this work, e.g., maximally-permissiveness or non-blockingness, we can just arbitrarily pick a macro-control decision for each Y-state encountered, until we traverse the entire reachable space. We denote by T_{solu} the resulting G-BTS, which includes a unique IS-based supervisor. Such an included supervisor can be "encoded" as follows. The supervisor tracks its state estimate $\mathcal{E}(\cdot)$ and $\hat{\mathcal{E}}(\cdot)$ as well as Y- and Z-states in T_{solu}. At each decision making instant, suppose the state estimate is $m \in M$ and the Y-state is $\mathbf{m}$, where we have $m \in \mathbf{m}$. Then the supervisor allows a a non-deterministic control decision Γ' such that $(m, \Gamma') \in d$, where d is the unique macro-control decision defined at $\mathbf{m}$ in T_{solu}. Then supervisor will randomly pick a control decision $\gamma \in \Gamma'$ to apply and wait for the next observation. The state tracked in T then moved to the unique successor state and the estimate of the supervisor is updated more precisely based on the specific control decision γ applied. This decoding procedure is formally summarized by Algorithm 1.

Algorithm 1 ENCODE-ND-SUP-BTS(T_{solu})

1: $\rho \leftarrow \epsilon, \hat{\mathcal{E}}(\rho) \leftarrow \{x_0\}$ and $y \leftarrow \{\{x_0\}\}$.
2: Define d be the macro-control-decision such that $C_{T_{solu}}(y) = \{d\}$
3: Define $\Gamma' \in 2^{\Gamma}$ be the non-deterministic control decision such that $(\hat{\mathcal{E}}(\rho), \Gamma') \in d$
4: Randomly pick $\gamma \in \Gamma'$ and apply this control decision
5: $\mathcal{E}(\rho) \leftarrow UR_{\gamma}(\hat{\mathcal{E}}(\rho))$ and $z \leftarrow h_{YZ}(y, d)$
6: **while** A new event $\sigma \in (\Sigma_o \cap \gamma)$ is observed **do**
7: $\quad \hat{\mathcal{E}}(\rho) \leftarrow NX_{\gamma}(\mathcal{E}(\rho))$ and $y \leftarrow h_{ZY}(z, \sigma)$
8: $\quad \rho \leftarrow \rho\gamma\sigma$
9: $\quad$ Define d be the macro-control-decision such that $C_{T_{solu}}(y) = \{d\}$
10: $\quad$ Define $\Gamma' \in 2^{\Gamma}$ be the non-deterministic control decision such that $(\hat{\mathcal{E}}(\rho), \Gamma') \in d$
11: $\quad$ Randomly pick $\gamma \in \Gamma'$ and apply this control decision
12: $\quad \mathcal{E}(\rho) \leftarrow UR_{\gamma}(\hat{\mathcal{E}}(\rho))$ and $z \leftarrow h_{YZ}(y, d)$
13: **end while**

Example 5. In our running example, we have $T^* = T_{solu}$ since the macro-control-decision at each Y-state in T^* is already unique. Then we decode a non-deterministic supervisor from T_{solu} by Algorithm 1 as follows. Initially, we have $\hat{\mathcal{E}}(\epsilon) = \{0\}$, $y = \{\{0\}\}$, and $d = d_5 = \{(\{0\}, \{\gamma_2, \gamma_3\})\}$. Therefore, we have $\Gamma' = \{\gamma_2, \gamma_3\}$, i.e., the supervisor can randomly choose to apply γ_2 or γ_3. Then, first, we update the state in T_{solu} to $s_6 = h_{YZ}(s_0, d_5) = \{(\{0, 1\}, \gamma_2), (\{0, 3\}, \gamma_3)\}$. If γ_2 is applied, then the state-estimate is to $\mathcal{E}(\epsilon) = UR_{\gamma_2}(\hat{\mathcal{E}}(\epsilon)) = \{0, 1\}$; otherwise, it is updated to $\mathcal{E}(\epsilon) = UR_{\gamma_3}(\hat{\mathcal{E}}(\epsilon)) = \{0, 3\}$. Suppose that decision γ_2 is chosen. Then by observing o_1, first, we update the state-estimate to $\hat{\mathcal{E}}(\gamma_2 o_1) = NX_{o_1}(\mathcal{E}(\epsilon)) = \{5\}$. Also, we update the state in T_{solu} to $s_{10} = h_{ZY}(s_6, o_1) = \{\{5\}, \{6\}\}$. Again, we need to decode the control decision for each information-state based on the unique macro-control-decision d_6 defined at s_{10}. Here, note that for both $(5, \{\gamma_1\}), (6, \{\gamma_1\}) \in d_6$, $\{\gamma_1\}$ is a singleton. This means that we will apply deterministic control decision γ_1 for both information-states $(\{5\}, \{\{5\}, \{6\}\})$ and $(\{6\}, \{\{5\}, \{6\}\})$. The overall decision diagram of the supervisor included in T_{solu} is shown in Figure 2(b).

6. CONCLUSION

In this paper, we proposed to use non-deterministic control mechanism to enforce opacity. We showed that non-deterministic supervisors may successfully enforce opacity even when deterministic supervisors fail to do so. Effective approach was provided to synthesize a non-deterministic opacity-enforcing supervisor based on both the information of the supervisor and the information of the intruder. This work makes the first step towards the enforcement of opacity using non-deterministic supervisors. Note that, in this paper, we restrict *a priori* the structure of the solution to IS-based supervisors. We conjecture that such a restriction is without loss of generality, but it still requires further investigation.

REFERENCES

Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., and Darondeau, P. (2007). Concurrent secrets. *Discrete Event Dynamic Systems*, 17(4), 425–446.

Cassandras, C. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer, 2nd edition.

Cassez, F., Dubreil, J., and Marchand, H. (2012). Synthesis of opaque systems with static and dynamic masks. *Formal Methods in System Design*, 40(1), 88–115.

Dubreil, J., Darondeau, P., and Marchand, H. (2010). Supervisory control for opacity. *IEEE Trans. Automatic Control*, 55(5), 1089–1100.

Inan, K. (1994). Nondeterministic supervision under partial observations. In *11th Int. Conf. Analysis and Optim. Systems Discrete Event Systems*, 39–48.

Ji, Y., Yin, X., and Lafortune, S. (2019). Enforcing opacity by insertion functions under multiple energy constraints. *Automatica*, 108, 108476.

Kumar, R., Jiang, S., Zhou, C., and Qiu, W. (2005). Polynomial synthesis of supervisor for partially observed discrete-event systems by allowing nondeterminism in control. *IEEE Trans. Auto. Control*, 50(4), 463–475.

Lafortune, S., Lin, F., and Hadjicostis, C. (2018). On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, 45, 257–266.

Saboori, A. and Hadjicostis, C. (2011). Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Trans. Automatic Control*, 57(5), 1155–1165.

Tong, Y., Li, Z., Seatzu, C., and Giua, A. (2018). Current-state opacity enforcement in discrete event systems under incomparable observations. *Discrete Event Dynamic Systems*, 28(2), 161–182.

Yin, X. and Lafortune, S. (2016). A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Trans. Automatic Control*, 61(8), 2140–2154.

Yin, X. and Lafortune, S. (2017). A new approach for the verification of infinite-step and k-step opacity using two-way observers. *Automatica*, 80, 162–171.

Yin, X., Li, Z., Wang, W., and Li, S. (2019). Infinite-step opacity and K-step opacity of stochastic discrete-event systems. *Automatica*, 99, 266–274.