2017 11th Asian Control Conference (ASCC)
Gold Coast Convention Centre, Australia
December 17-20, 2017

# Infinite-Step Opacity of Stochastic Discrete-Event Systems

Xiang Yin, Zhaojian Li, Weilin Wang and Shaoyuan Li

*Abstract*— **Opacity is an important information-flow property that arises in security and privacy analysis of cyber-physical systems. It captures the plausible deniability of the system's "secret" in the presence of a malicious intruder modeled as a passive observer. As a specific type of opacity, infinite-step opacity requires that the intruder can never determine unambiguously that the system was at a secret system for any specific instant in the past. Existing works on the analysis of infinite-step opacity only provide a binary characterization, i.e., a system is either opaque or non-opaque. However, a non-infinite-step-opaque system may only have a small probability of violation; this may be still tolerable in many applications. To analyze infinite-step opacity more quantitatively, in this paper, we investigate the verification of infinite-step opacity in the context of stochastic discrete-event systems. A new notion of opacity, called almost infinite-step opacity, is proposed to capture whether or not the probability of violating infinite-step opacity is smaller than a given threshold. This notion is weaker than its purely logical counter-part as it takes the transition probability of the system into account. We also provide an effective algorithm for the verification of almost infinite-step opacity.**

## I. Introduction

Opacity is an information-flow property that arises in security and privacy analysis of cyber-physical systems. In this paper, we investigate the verification of opacity for Discrete-Event Systems (DES), an important class of man-made cyber-physical systems with discrete state-spaces and event-triggered dynamics. In this problem, we assume that the system is monitored by a (potentially malicious) intruder that is modeled as a passive observer. We say that the system is opaque if the intruder can never determine the system's "secret" unambiguously based on its limited observation. In the past years, opacity has drawn considerable attention and has been extensively studied in the DES literature; see, e.g., [1], [4], [5], [8], [9], [11], [12], [16], [17], [19]–[21], [23]–[26], [28]–[33] and a recent survey [13] for more references.

In the context of DES, different notions of opacity have been proposed to capture different types of security requirements, e.g., current-state opacity [26], initial-state opacity [21], $K$-step opacity [19], [32] and infinite-step opacity [20], [32]. Particularly, in [20], the notion of *infinite-step opacity* was proposed in order to capture whether or not the intruder may know that the system was/is at a secret state for some specific instant. This definition is stronger than the notion of current-state opacity as it allows the intruder to use future information to *infer* the system's state in the past. More recently, an improved approach for the verification of infinite-step opacity was proposed by using a structure called the two-way observer [32].

The definition of infinite-step opacity in [20], [32] only provides a binary characterization, i.e., a system is either opaque or not. However, a non-infinite-step-opaque system may only have a small probability of violation; this may be still tolerable in many applications. Recently, many works have considered how to quantitatively evaluate opacity by using probabilistic models (stochastic DES), e.g., [2], [3], [6], [10], [14], [15], [22], [27]. By precisely capturing the transition probability of the system, one is able to evaluate the possibility of being not secure, rather than simply providing a binary answer. However, all these works on opacity analysis of stochastic DES only consider *current-state-type* opacity. (Note that initial-state opacity can also be transformed to current-state opacity by considering its reversed automaton [26].) For current-step-type opacity, we only need to consider all information available so far in order to determine whether the system is opaque or not. However, for infinite-step opacity, we also need to consider how future information can affect our knowledge about the current status of the system, which is much more challenging. To the best of our knowledge, how to evaluate infinite-step opacity in the context of stochastic DES has still not yet been investigated.

In this paper, we investigate the analysis of infinite-step opacity in the context of stochastic DES. The main contributions of this paper are as follows. First, we define the notion of *almost infinite-step opacity* to capture whether or not the cumulated probability of violating infinite-step opacity is smaller than a given threshold. Then we propose an effective approach for the verification of almost infinite-step opacity. Our definition of almost infinite-step opacity is motivated by the definitions of almost current-step opacity [22] and almost initial-state opacity [14]. However, the proposed verification algorithm is quite different from those in [14], [22]. As we mentioned earlier, both current-state opacity and initial-state opacity fail into the current-state-type category, where no *delayed* information is involved. In order to handle the delayed information in infinite-step opacity, in this paper, we propose a new information structure that precisely captures all possible delayed state estimates along the trajectory. This new structure is different from the current (respectively, initial)-state estimator used in [22] (respectively, [14]) for the

X. Yin and S. Li is with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: xiangyin@umich.edu,syli@sjtu.edu. X. Yin is also with the Department of EECS, University of Michigan, Ann Arbor, MI 48109, USA. Z. Li is with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA. E-mail: zhaojli@umich.edu. W. Wang is with the Faculty of Engineering, Monash University, Clayton, VIC 3800, Australia E-mail: weilin.wang@monash.edu. This work is supported by the National Nature Science Foundation of China (61590924, 61233004).

verification of stochastic current (respectively, initial) state opacity. It is also different from the two-way observer [32] that is used for the verification of infinite-step opacity in the context of logical DES.

The rest of this paper is organized as follows. Section II provides some necessary preliminaries. In Section III, we propose the notion of almost infinite-step opacity for stochastic DES; an effective approach for the verification of this notion is provided in Section IV. Finally, we conclude the paper in Section V. Due to space constraints, all proofs in the paper are omitted.

## II. Preliminaries

### A. System Model

Let $\Sigma$ be a set of events. A string over $\Sigma$ is a finite sequence of events $s = \sigma_1 \dots \sigma_n, \sigma_i \in \Sigma$. We denote by $\Sigma^*$ the set of all strings over $\Sigma$ including the empty string $\epsilon$. For any string $s \in \Sigma^*$, we denote by $|s|$ its length with $|\epsilon| = 0$. A language $L \subseteq \Sigma^*$ is a set of strings; we denote by $\overline{L}$ the prefix-closure of $L$, i.e., $\overline{L} := \{s \in \Sigma^* : \exists t \in \Sigma^* \text{ s.t. } st \in L\}$. For any string $s \in \Sigma^*$, we denote by $t \leq s$ if $t \in \overline{\{s\}}$ and denote by $t < s$ if $t \in \overline{\{s\}} \setminus \{s\}$.

We consider a DES modeled as a deterministic finite-state automaton (DFA) $G = (X, \Sigma, \delta, x_0)$, where $X$ is the finite set of states, $\Sigma$ is the finite set of events, $\delta : X \times \Sigma \to X$ is a (partial) deterministic transition function and $x_0$ is the unique initial state. The transition function $\delta$ is also extended to $X \times \Sigma^* \to X$ in the usual manner; see, e.g., [7]. For the sake of simplicity, we denote $\delta(x, s)$ by $\delta(s)$ if $x = x_0$. We denote by $\mathcal{L}(G) = \{s \in \Sigma^* : \delta(s)!\}$ the language generated by $G$, where "!" means "is defined".

A stochastic discrete-event system is modeled as a probabilistic finite-state automaton (PFA) $(G, p)$, where $G = (X, \Sigma, \delta, x_0)$ is a DFA and $p : X \times \Sigma \to [0, 1]$ is the transition probability function. Specifically, for any $x \in X, \sigma \in \Sigma$, we write $p(\sigma \mid x)$ the probability that event $\sigma$ occurs from state $x$. We assume that (i) $\forall x \in X : \sum_{\sigma \in \Sigma} p(\sigma \mid x) = 1$; an (ii) $\forall x \in X, \sigma \in \Sigma : p(\sigma \mid x) > 0 \Leftrightarrow \delta(x, \sigma)!$. For any string $s \in \mathcal{L}(G)$, we denote by $Pr(s)$ the probability that $s$ occurs, i.e., $Pr(\epsilon) = 1$ and$Pr(s\sigma) = Pr(s)p(\sigma \mid \delta(s))$.

### B. Intruder Model

In opacity analysis of DES, we assume that the intruder is modeled as a *passive observer* that can observe partial behavior of the system and then *infer* the secret of the system based on its imperfect information. To this end, we assume that the event set $\Sigma$ is partitioned into two disjoint sets: $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, where $\Sigma_o$ is the set of observable events and $\Sigma_{uo}$ is the set of unobservable events. The natural projection $P : \Sigma^* \to \Sigma_o^*$ is defined recursively by

$$P(\epsilon) = \epsilon \quad \text{and} \quad P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{if } \sigma \in \Sigma_{uo} \end{cases} \tag{1}$$

The natural projection is also extended to $P : 2^{\Sigma^*} \to 2^{\Sigma_o^*}$ by $P(L) = \{P(s) \in \Sigma_o^* : s \in L\}$.

Based on its observation, the intruder can *infer* which state of the system could be in at some specific instant. Formally, let $\alpha \in P(\mathcal{L}(G))$ be an observable string. Then the *current state estimate* upon the occurrence of $\alpha$ is defined by

$$\hat{X}_G(\alpha) = \{x \in X : \exists s \in \mathcal{L}(G) \text{ s.t. } P(s) = \alpha \wedge x = \delta(s)\} \tag{2}$$

The current state estimate can be computed by building the observer automaton (current state estimator); see, e.g., [7].

In some situations, the intruder is also interested in knowing which states the system could be in for some previous instant. Suppose that $\alpha \in P(\mathcal{L}(G))$ is observed and we are interested in the state estimate of the system for the instant when only $\beta \leq \alpha$ was executed. Then we define the *delayed state estimate* for $\beta$ given $\alpha$ been observed by

$$\hat{X}_G(\beta \mid \alpha) = \tag{3}$$
$$\{x \in X : \exists st \in \mathcal{L}(G) \text{ s.t. } P(s) = \beta \wedge P(st) = \alpha \wedge x = \delta(s)\}$$

Intuitively, $\hat{X}_G(\beta \mid \alpha)$ estimates the state of the system $(|\alpha| - |\beta|)$-steps earlier when $\alpha$ is observed. Clearly, we have that $\hat{X}_G(\alpha \mid \alpha) = \hat{X}_G(\alpha)$. Also, since more information is used to infer the state of the system for the instant when $\beta$ is observed, we know that $\hat{X}_G(\beta \mid \alpha) \subseteq \hat{X}_G(\beta)$, i.e, $\hat{X}_G(\beta \mid \alpha)$ has less uncertainty than $\hat{X}_G(\beta)$.

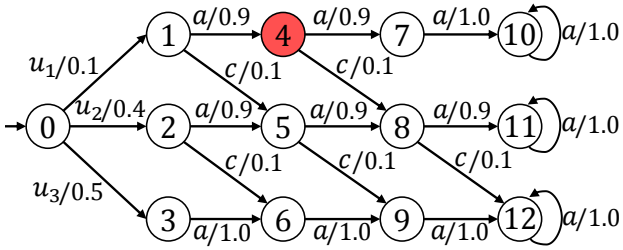


Fig. 1. System $(G_1, p_1)$ with $\Sigma_o = \{a, c\}$

*Example 1:* Let us consider system $G_1$ shown in Figure 1, where $\Sigma_o = \{a, c\}$. If string $a \in P(\mathcal{L}(G))$ is observed, then we know that $\hat{X}_{G_1}(a) = \{4, 5, 6\}$. If string $ac \in P(\mathcal{L}(G))$ is observed, then we know that $\hat{X}_{G_1}(a \mid ac) = \{4, 5\}$. If string $acc \in P(\mathcal{L}(G))$ is observed, then we know that $\hat{X}_{G_1}(a \mid acc) = \{4\}$. Clearly, we see that $\hat{X}_{G_1}(a \mid acc) \subseteq \hat{X}_{G_1}(a \mid ac) \subseteq \hat{X}_{G_1}(a)$. That is, our knowledge about the system's state for the instant when $a$ is observed is improved when more future events are observed.

Finally, we define the following operators that will be used later. Let $r \in 2^X$ be a set of states and $\sigma \in \Sigma_o$ be an observable event. The unobservable reach is defined by:

$$UR(r) = \{x \in X : \exists x' \in r, \exists s \in \Sigma_{uo}^* \text{ s.t. } \delta(x', s) = x\}$$

The observable transition is defined by:

$$Next_\sigma(r) = \{x \in X : \exists x' \in r \text{ s.t. } \delta(x', \sigma) = x\}$$

Let $\alpha \in \Sigma_o^*$ be an observable string. Operator $\Xi : \Sigma_o^* \to 2^{X \times X}$ is defined by:

$$\Xi(\alpha) = \{(x, x') \in X \times X : \exists s \in \Sigma^* \text{ s.t. } P(s) = \alpha \wedge x' = \delta(x, s)\}$$

Let $\tilde{r}_1, \tilde{r}_2 \in 2^{X\times X}$ be two sets of state pairs. Operator $\circ : 2^{X\times X} \times 2^{X\times X} \to 2^{X\times X}$ is defined by:

$$\tilde{r}_1 \circ \tilde{r}_2 = \{(x_1, x_3) \in X \times X : \exists x_2 \in X \text{ s.t. } (x_1, x_2) \in \tilde{r}_1 \wedge (x_2, x_3) \in \tilde{r}_2\}$$

Let $r \in 2^X$ be a set of states. Operator $\odot : 2^X \to 2^{X\times X}$ is defined by:

$$\odot(r) = \{(x, x') \in X \times X : \exists x \in r, \exists s \in \Sigma_{uo}^* \text{ s.t. } \delta(x, s) = x'\}$$

## III. Infinite-Step Opacity in Stochastic DES

In this section, we first review the definition of infinite-step opacity in logical DES. Then we propose the notion of almost infinite-step opacity for stochastic DES.

### A. Infinite-Step Opacity in Logical DES

In opacity analysis, we assume that the system has a "secret". Specifically, we model the "secret" of the system as a set of states $X_S \subseteq X$. For example, in location-based services, a secret state may represent a secret location corresponding to a hospital or a bank. Then, within this setting, infinite-step opacity requires that the intruder can never determine for sure that the system was (or is) at a secret state for any previous (or current) instant. This notion is formally defined as follows.

*Definition 1:* (Infinite-Step Opacity [20]) Let $G$ be a DFA, $\Sigma_o \subseteq \Sigma$ be a set of observable events and $X_S \subseteq X$ be a set of secret states. We say that $G$ is infinite-step opaque (w.r.t. $\Sigma_o$ and $X_S$) if $\forall \alpha\beta \in P(\mathcal{L}(G)) : \hat{X}_G(\alpha \mid \alpha\beta) \not\subseteq X_S$.

To test infinite-step opacity, one approach is to construct a structure called the two-way observer; the reader is referred to [32] for details on its verification.

*Example 2:* Let us return to system $G_1$ shown in Figure 1. Suppose that state 4 is the unique secret state, i.e., $X_S = \{4\}$. Then we have $\{4\} = \hat{X}_{G_1}(a \mid acc) \subseteq X_S$, i.e., the intruder knows for sure that the system was in a secret state 2-steps ago when $acc$ is observed. Therefore, the secret can be revealed and the system is not infinite-step opaque.

Note that infinite-step opacity in Definition 1 only provides a binary characterization, i.e., a system is either opaque or non-opaque. This notion does not consider the system's transition probability into account. For example, in Example 2, one can verify that the only observable string that leads to the violation of infinite-step opacity is $u_1acc$. Then the probability of violating infinite-step opacity is $Pr(u_1acc) = 0.0009$, which is very small and this may still be tolerable in many applications. Therefore, to quantitatively evaluate infinite-step opacity, it may also be useful to consider the transition probability of the system into account. This motivates the definition of almost infinite-step opacity that will be presented next.

### B. Almost Infinite-Step Opacity

First, we define the following language:

$$L_{\mathcal{IF}} = \{s \in \mathcal{L}(G) : \exists \alpha \leq P(s) \text{ s.t. } \hat{X}(\alpha \mid P(s)) \subseteq X_S\} \quad (4)$$

That is, $L_{\mathcal{IF}}$ is the set of strings whose occurrences violate infinite-step opacity for some instant. In order to consider the cumulated probability of the violation of infinite-step opacity, we only need to consider those strings violating infinite-step opacity for the first time. Formally, we define the following language:

$$L_{\mathcal{IF}}^{\mathcal{P}} = \{s \in L_{\mathcal{IF}} : \forall t < s \text{ s.t. } t \notin L_{\mathcal{IF}}\} \quad (5)$$

Now, we are already to define the notion of almost infinite-step opacity.

*Definition 2:* (Almost Infinite-Step Opacity) Let $(G, p)$ be a PFA, $\Sigma_o \subseteq \Sigma$ be the set of observable events, $X_S \subseteq X$ be a set of secret state and $\theta < 1$ be a threshold value. We say that $(G, p)$ is almost infinite-step opaque (w.r.t. $\Sigma_o, X_S$ and $\theta$) if $\sum_{s\in L_{\mathcal{IF}}^{\mathcal{P}}} Pr(s) < \theta$.

Essentially, almost infinite-step opacity requires that the cumulated probability of strings that violate infinite-step opacity in the logic sense is smaller than a given threshold $\theta$. The reason why we consider language $L_{\mathcal{IF}}^{\mathcal{P}}$ rather than language $L_{\mathcal{IF}}$ is that, once the secret of the system is revealed by some string, any of its continuation will also reveal the secret. Therefore, we only need to consider strings in $L_{\mathcal{IF}}^{\mathcal{P}}$ to avoid counting the probability of violation duplicately.

Let us illustrate almost infinite-step opacity by the following example.

*Example 3:* Let us consider system $(G_2, p_2)$ shown in Figure 2, where $\Sigma_o = \{a, b, c\}$ and $X_S = \{3\}$. Then we have $L_{\mathcal{IF}} = \{uab, aca\}\Sigma^* \cap \mathcal{L}(G_2)$ and $L_{\mathcal{IF}}^{\mathcal{P}} = \{uab, aca\}$. Since $\sum_{s\in L_{\mathcal{IF}}^{\mathcal{P}}} Pr(s) = 0.01 + 0.09 = 0.1$, we know that this system is almost infinite-step opaque for any threshold $\theta > 0.1$.

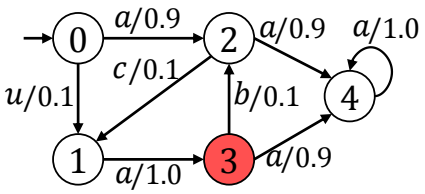


Fig. 2. System $(G_2, p_2)$ with $\Sigma_o = \{a, b, c\}$ and $X_S = \{5\}$.

## IV. Verification of Almost Infinite-Step Opacity

In this section, we show how to formally verify almost infinite-step opacity.

To verify almost infinite-step opacity, the main idea is to construct a new automaton that (i) recognizes $L_{\mathcal{IF}}^{\mathcal{P}}$; and (ii) tracks the original transition probability of $(G, p)$. When we consider current-state opacity (or initial-state opacity), these requirements can be simply fulfilled by taking the product composition of $G$ and its current state estimator (or initial-state estimator). However, this task is much more challenging for infinite-step opacity as $L_{\mathcal{IF}}^{\mathcal{P}}$ involves delayed information. Therefore, we need a new information structure that recognizes $L_{\mathcal{IF}}^{\mathcal{P}}$. This is detailed next.

Let $G$ be a DFA. We define a new automaton

$$V_G = (Q, \Sigma, f, q_0) \quad (6)$$

where

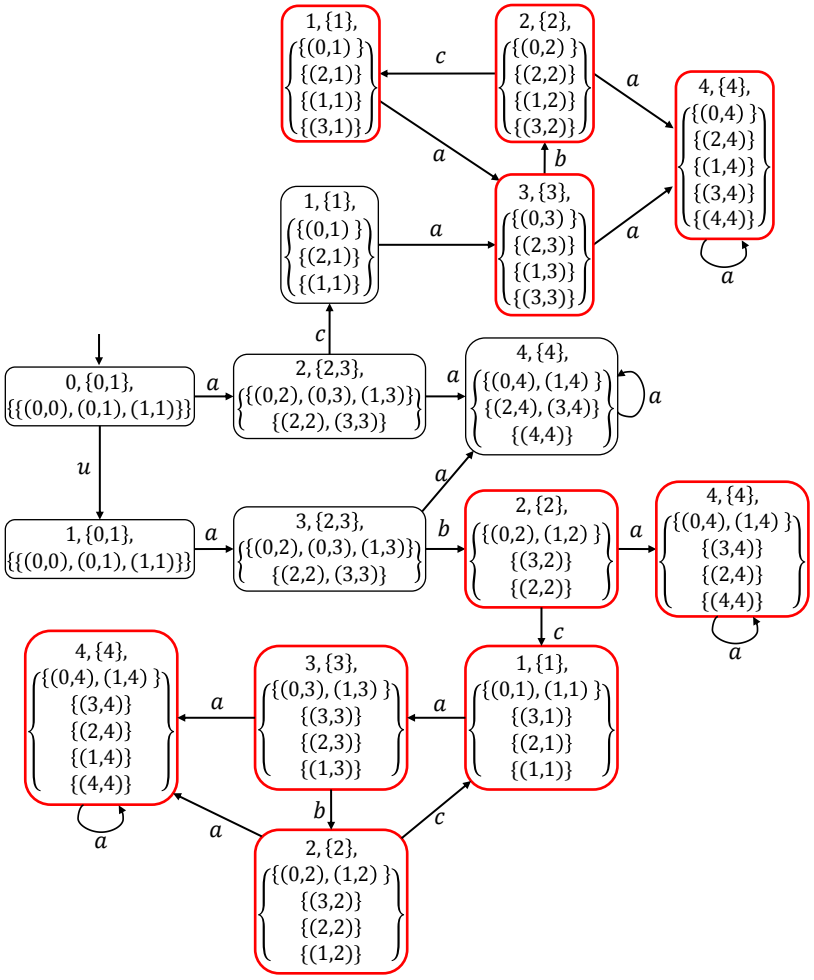


Fig. 3. Automaton $V_{G_2}$ for the system in Figure 2.

- $Q \subseteq X \times 2^X \times 2^{2^{X\times X}}$ is the set of states;
- $\Sigma$ is the set of events;
- $f : Q \times \Sigma \to Q$ is the transition function defined by: for any $q = (x, r, R) \in X \times 2^X \times 2^{2^{X\times X}}$ and $\sigma \in \Sigma$, we have

$$f(q,\sigma) = \begin{cases} (\delta(x,\sigma), r, R) & \text{if } \sigma \in \Sigma_{uo} \\ (\delta(x,\sigma), r', R') & \text{if } \sigma \in \Sigma_o \end{cases} \tag{7}$$

where

$$r' = UR(Next_\sigma(r)) \tag{8}$$

$$R' = \{\{\tilde{r} \circ \Xi(\sigma) \in X \times X : \tilde{r} \in \rho\} \in 2^{X\times X} : \rho \in R\} \cup \{\odot(r')\} \tag{9}$$

- $q_0 = (x_0, r_0, R_0)$ is the unique initial state, where $r_0 = UR(\{x_0\})$ and $R_0 = \{\odot(r_0)\}$.

Let us explain the intuition of the above construction. Note that each state $(x, r, R)$ in $V_G$ consists of three components, which are used as follows. The first component $x$ simply tracks the current state in the original system $G$. Hence, the transition of $f$ is consistent with $\delta$ for this component and we have that $\mathcal{L}(V_G) = \mathcal{L}(G)$. The second component $r$ tracks the current state estimate of the original system. That is, $r = \hat{X}_G(P(s))$, where $s$ is a string leading to $(x, r, R)$. The transition function of this component is essentially the same as the transition function of the standard observer automaton, i.e., $UR(Next_\sigma(\cdot))$. The third component $R$ is used to track the following information. Note that, $R$ is a set of sets of state pairs. Each $\rho \in R$ essentially represents the delayed state estimate of the system for some (current or previous) instant. Specifically, for any $(x, x') \in \rho$, $x$ is a state the system could be in at that instant and $x'$ is a state the system could be in currently from $x'$, and $\rho$ consists of all such pairs for that specific instant. Although the number of previous instant may be infinite, $R$ is a finite set as there are only finite such configurations for all possible delayed state estimates. Therefore, upon the occurrence of a new observable event, say $\sigma$, we need to update the delayed state estimate for all previous instant, i.e., $\{\{\tilde{r} \circ \Xi(\sigma) : \tilde{r} \in \rho\} : \rho \in R\}$, and, at the same time, remember the current state estimate $\{\odot(r')\}$.

Let us explain the construction of $V_G$ by the following example.

*Example 4:* Again, let us consider system $G_2$ shown in Figure 2. Its corresponding automaton $V_{G_2}$ is shown in Figure 3. First, $V_{G_2}$ starts with the initial state $(x_0, r_0, R_0) = (0, \{0, 1\}, \{\{(0, 0), (0, 1), (1, 1)\}\})$. The

second component means that the current state estimate is $\hat{X}_{G_2}(\epsilon) = \{0, 1\}$ and the third component means that, within the initial step, the system may start from state 0 and end up with state 0, or start from state 0 and end up with state 1, or start from state 1 and end up with state 1. When observable event $a$ occurs, we move to a new state $(x_1, r_1, R_1) = (2, \{2, 3\}, \{\{(0, 2), (0, 3), (1, 3)\}, \{(2, 2), (3, 3)\}\})$, where $\{2, 3\}$ is the current state estimate of the system, and $\{(0, 0), (0, 1), (1, 1)\} \in R_0$ is updated to $\{(0, 2), (0, 3), (1, 3)\}$, which is the updated knowledge about the system for the instant one step ago, i.e., the delayed state estimate. At the same time, we need to add $\{\odot(\{2, 3\})\}$ to $R_2$ in order to remember the new state estimate for the current instant.

Next, we formally summarize the properties of $V_G$. First, for any $\rho \in 2^{X \times X}$, we define

$$I_1(\rho) = \{x \in X : \exists x' \in X \text{ s.t. } (x, x') \in \rho\} \tag{10}$$

Also, for any $R \in 2^{2^{X \times X}}$, we also define

$$I_1(R) = \{I_1(\rho) \in 2^X : \rho \in R\} \tag{11}$$

Then we have the following result, which essentially states the intuition of $V_G$ explained above.

*Lemma 1:* For any string $s \in \mathcal{L}(V_G) = \mathcal{L}(G)$, let $f(q_0, s) = (x_s, r_s, R_s)$ be the state reached in $V_G$ via $s$ from the initial state, then we have:

(i) $r_s = \hat{X}_G(P(s))$;
(ii) $I_1(R_s) = \{\hat{X}_G(\alpha \mid P(s)) \in 2^X : \alpha \leq P(s)\}$.

By Lemma 1, we know that, for any string $s \in \mathcal{L}(V_G)$ such that $f(q_0, s) = (x, r, R)$, $I_1(R)$ is actually the set of all delayed state estimates for all previous instant. Therefore, to determine whether or not $s \in L_{\mathcal{IF}}$, it suffices to determine whether or not there exists $\rho \in R$ such that $I_1(\rho) \subseteq X_S$. More specifically, let

$$Q_{\mathcal{IF}} = \{(x, r, R) \in Q : \exists \rho \in R \text{ s.t. } I_1(\rho) \subseteq X_S\} \tag{12}$$

Then we have the following result.

*Lemma 2:* For any string $s \in \mathcal{L}(V_G) = \mathcal{L}(G)$, let $(x, r, R) = f(q_0, s)$ be the state reached in $V_G$ via string $s$, then we have

$$L^{\mathcal{P}}_{\mathcal{IF}} = \{s \in \mathcal{L}(G) : [f(q_0, s) \in Q_{\mathcal{IF}}] \wedge [\forall t < s : f(q_0, s) \notin Q_{\mathcal{IF}}]\}$$

By Lemma 2, it is clear that, to compute $\sum_{s \in L^{\mathcal{P}}_{\mathcal{IF}}} Pr(s)$, it suffices to compute the probability of hinting a state in $Q_{\mathcal{IF}}$ in a Markov chain associated to $V_G$ with transition probability reflecting the original system $(G, p)$. This is formalizes as follows. First, we denote by $\tilde{V}_G = (\tilde{Q}, \Sigma, \tilde{f}, q_0)$ the accessible part of the automaton obtained by removing all outgoing transitions from states in $Q_{\mathcal{IF}}$. Then we define a Markov Chain (MC) $\mathcal{M} = (\tilde{Q}, p_{\mathcal{M}}, \pi_0)$, where the state space of the MC is the same as the state space of $\tilde{V}_G$ and the transition probability function $p_{\mathcal{M}} : \tilde{Q} \times \tilde{Q} \to [0, 1]$ is defined by: for any $q = (x, r, R), q' = (x', r', R') \in \tilde{Q}$,

$$p_{\mathcal{M}}(q' \mid q) = \begin{cases} \sum_{\sigma \in \Sigma : f(q, \sigma) = q'} p(\sigma \mid x) & \text{if } q \notin Q_{\mathcal{IF}} \\ 1 & \text{if } q = q' \in Q_{\mathcal{IF}} \\ 0 & \text{otherwise} \end{cases}$$

and $\pi_0 : \tilde{Q} \to [0, 1]$ is the initial state distribution defined by $\pi_0(q_0) = 1$.[1] Therefore, $\mathcal{M}$ is constructed such that all states in $Q_{\mathcal{IF}} \cap \tilde{Q}$ are absorbing, i.e., once we reach a state in $Q_{\mathcal{IF}} \cap \tilde{Q}$, we will stay in it forever. This absorbing probability, denoted by $p^{abs}_{\mathcal{M}}$, can be computed by [18]:

$$p^{abs}_{\mathcal{M}} = \sum_{q \in \tilde{Q}} \pi_0(q) \mathbb{P}(q) \tag{13}$$

where $\mathbb{P} : \tilde{Q} \to \mathbb{R}^+_0$ is the vector of minimal non-negative solution to the following equation

$$\mathbb{P}(q) = \begin{cases} \sum_{q' \in \tilde{Q}} p_{\mathcal{M}}(q' \mid q) \mathbb{P}(q') & \text{if } q \notin Q_{\mathcal{IF}} \\ 1 & \text{if } q \in Q_{\mathcal{IF}} \end{cases} \tag{14}$$

Note that, since $\mathcal{L}(V_G) = \mathcal{L}(G)$ and $\mathcal{M}$ is constructed by tracking the transition probability of the original system, we know that $p^{abs}_{\mathcal{M}} = \sum_{s \in L^{\mathcal{P}}_{\mathcal{IF}}} Pr(s)$. Hence, we have the following main theorem.

*Theorem 1:* Let $(G, p)$ be a PFA and $\mathcal{M}$ be its associated MC constructed above. Then $(G, p)$ is infinite-step opacity w.r.t. threshold $\theta$ if and only if $p^{abs}_{\mathcal{M}} < \theta$.

Let us illustrate how to verify almost infinite-step opacity by the following example.

*Example 5:* Still, let us consider system $(G_2, p_2)$ shown in Figure 2 and $V_{G_2}$ is shown in Figure 3. States in $Q_{\mathcal{IF}}$ are marked by red lines. For example, we know that $(2, \{2\}, \{\{(0, 2), (1, 2)\}, \{(3, 2)\}, \{(2, 2)\}\}) \in Q_{\mathcal{IF}}$, since $I_1(\{(3, 2)\}) = \{3\} \subseteq X_S$. Then its associated MC $\mathcal{M}$ is shown in Figure 4. For the sake of simplicity, each state in $\mathcal{M}$ is renamed from $M_1$ to $M_8$. To compute the absorbing probability in $Q_{\mathcal{IF}} \cap \tilde{Q}$, we need to solve the following equation

$$\begin{cases} \mathbb{P}(M_1) = 0.1 \times \mathbb{P}(M_2) + 0.9 \times \mathbb{P}(M_3) \\ \mathbb{P}(M_2) = \mathbb{P}(M_4) \\ \mathbb{P}(M_3) = 0.1 \times \mathbb{P}(M_5) + 0.9 \times \mathbb{P}(M_8) \\ \mathbb{P}(M_4) = 0.1 \times \mathbb{P}(M_6) + 0.9 \times \mathbb{P}(M_8) \\ \mathbb{P}(M_8) = \mathbb{P}(M_8) \\ \mathbb{P}(M_5) = \mathbb{P}(M_6) = \mathbb{P}(M_7) = 1 \end{cases} \tag{15}$$

Note that, the solution to Equation (15) is not unique as $\mathbb{P}(M_8)$ is a free term. To obtain the minimal solution, we need to set $\mathbb{P}(M_8) = 0$ and we have $\mathbb{P} = [0.1 \ \ 0.1 \ \ 0.1 \ \ 0.1 \ \ 1 \ \ 1 \ \ 1 \ \ 0]$. Therefore, we know that $\sum_{s \in L^{\mathcal{P}}_{\mathcal{IF}}} Pr(s) = p^{abs}_{\mathcal{M}} = \pi_0(M_1) \times \mathbb{P}(M_1) = 0.1$, i.e., the system is almost infinite-step opaque for any $\theta > 0.1$, and this is consistent with our result in Example 3.

[1] We assume w.l.o.g. that $q_0 \in \tilde{Q}$; otherwise it implies that the system is not infinite-step opacity even for threshold $\theta = 1$.

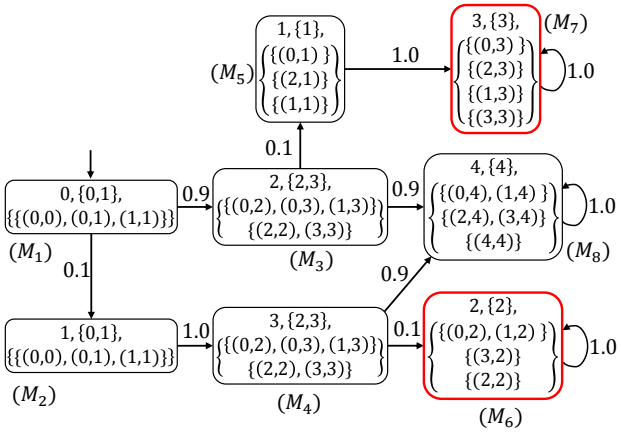


Fig. 4. Markov chain $\mathcal{M}$ for the system in Figure 2.

## V. Conclusion

We investigated the analysis of infinite-step opacity in the context of stochastic DES. A new notion called almost infinite-step opacity was proposed to quantitatively evaluate the probability that infinite-step opacity is violated. An effective approach was also provided for the verification of almost infinite-step opacity. The complexity of the verification algorithm is doubly-exponential in the size of the original system, which is higher than the complexity for the verification of almost current-state opacity. Intuitively, this higher complexity comes from the fact that, in the analysis of almost infinite-step opacity, one not only needs to track the current state estimate of the system, but also needs to track all possible delayed state estimates for previous instants. Investigating more efficient approach for the verification of almost infinite-step opacity and establishing precise complexity for this problem are interesting future directions.

## References


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

[2] B. Bérard, K. Chatterjee, and N. Sznajder. Probabilistic opacity for markov decision processes. *Information Processing Letters*, 115(1):52–59, 2015.

[3] B. Bérard, J. Mullins, and M. Sassolas. Quantifying opacity. *Math. Structures in Computer Science*, 25(2):361–403, 2015.

[4] A. Bourouis, K. Klai, N. Ben Hadj-Alouane, and Y. El Touati. On the verification of opacity in web services and their composition. *IEEE Transactions on Services Computing*, 10(1):66–79, 2017.

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

[6] J.W. Bryans, M. Koutny, and C. Mu. Towards quantitative analysis of opacity. In *Int. Symp. Trustworthy Global Comp.*, pages 145–163, 2012.

[7] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, 2nd edition, 2008.

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

[9] S. Chédor, C. Morvan, S. Pinchinat, and H. Marchand. Diagnosis and opacity problems for infinite state systems modeled by recursive tile systems. *Discrete Event Dynamic Systems: Theory & Applications*, 25(1-2):271–294, 2015.

[10] J. Chen, M. Ibrahim, and R. Kumar. Quantification of secrecy in partially observed stochastic discrete event systems. *IEEE Transactions on Automation Sci. Eng.*, 14(1):185–195, 2017.

[11] J. Dubreil, P. Darondeau, and H. Marchand. Supervisory control for opacity. *IEEE Transactions on Automatic Control*, 55(5):1089–1100, 2010.

[12] Y. Falcone and H. Marchand. Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems: Theory & Applications*, pages 1–40, 2014.

[13] R. Jacob, J.-J. Lesage, and J.-M. Faure. Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, 41:135–146, 2016.

[14] C. Keroglou and C.N. Hadjicostis. Initial state opacity in stochastic des. In *18th IEEE Conference on ETFA*, pages 1–8, 2013.

[15] C. Keroglou and C.N. Hadjicostis. Probabilistic system opacity in discrete event systems. In *13th International Workshop on Discrete Event Systems*, pages 379–384, 2016.

[16] F. Lin. Opacity of discrete event systems and its applications. *Automatica*, 47(3):496–503, 2011.

[17] J. Mullins and M. Yeddes. Opacity with orwellian observers and intransitive non-interference. In *12th Int. Workshop on Discrete Event Systems*, pages 344–349, 2014.

[18] J.R. Norris. *Markov chains*. Number 2. Cambridge University Press, 1998.

[19] A. Saboori and C.N. Hadjicostis. Verification of $K$-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, 8(3):549–559, 2011.

[20] A. Saboori and C.N. Hadjicostis. Verification of infinite-step opacity and complexity considerations. *IEEE Transactions on Automatic Control*, 57(5):1265–1269, 2012.

[21] A. Saboori and C.N. Hadjicostis. Verification of initial-state opacity in security applications of discrete event systems. *Information Sciences*, 246:115–132, 2013.

[22] A. Saboori and C.N. Hadjicostis. Current-state opacity formulations in probabilistic finite automata. *IEEE Transactions on Automatic Control*, 59(1):120–133, 2014.

[23] S. Takai and Y. Oka. A formula for the supremal controllable and opaque sublanguage arising in supervisory control. *SICE J. Control, Measu. & Syst. Integration*, 1(4):307–311, 2008.

[24] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Decidability of opacity verification problems in labeled Petri net systems. *Automatica*, 80:48–53, 2017.

[25] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Verification of state-based opacity using Petri nets. *IEEE Transactions on Automatic Control*, 62(6):2823–2837, 2017.

[26] Y.-C. Wu and S. Lafortune. Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems: Theory & Applications*, 23(3):307–339, 2013.

[27] Y.-C. Wu, G. Lederman, and S. Lafortune. Enhancing opacity of stochastic discrete event systems using insertion functions. In *American Control Conference*, pages 2053–2060, 2016.

[28] X. Yin and S. Lafortune. A general approach for solving dynamic sensor activation problems for a class of properties. In *54th IEEE Conference on Decision and Control*, pages 3610–3615, 2015.

[29] X. Yin and S. Lafortune. A new approach for enforcing opacity via supervisory control for partially-observed discrete-event systems. In *American Control Conference*, pages 377–383, 2015.

[30] X. Yin and S. Lafortune. On two-way observer and its application to the verification of infinite-step and K-step opacity. In *13th International Workshop on Discrete Event Systems*, pages 361–366, 2016.

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

[32] X. Yin and S. Lafortune. A new approach for the verification of infinite-step and $K$-step opacity using two-way observers. *Automatica*, 80:162–171, 2017.

[33] B. Zhang, S. Shu, and F. Lin. Maximum information release while ensuring opacity in discrete event systems. *IEEE Transactions on Automation Science and Engineering*, 12(4):1067–1079, 2015.