2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Verification of Opacity in Networked Supervisory Control Systems with Insecure Control Channels

Xiang Yin and Shaoyuan Li

Abstract— **We investigate security and privacy issues in networked discrete-event systems, where supervisory controllers are connected with actuators and sensors via communication networks. In this paper, we consider the case where the control channel between the supervisor and the actuators may not be secure in the sense that the control decisions sent by the supervisor can be "listened" by an intruder. We adopt the concept of an information-flow property called opacity to capture whether or not the networked supervisory control system is secure. Specifically, we say that the supervisory control system is opaque with insecure control channel if the intruder can never determine for sure that the system is in a secret state based on the control decisions sent by the supervisor. Based on different control decision transmission mechanisms, two notions of opacity are defined. Effective algorithms are also provided to verify different notions of opacity for networked supervisory control systems.**

I. Introduction

Supervisory control theory is a widely used formal methods to enforce closed-loop properties for Discrete-Event Systems (DES). In this framework, the system is controlled by a *supervisor* that disables events dynamically based on its observations so that the closed-loop behavior under control meets some design specifications. Supervisory control of DES has been extensively developed since the seminal work of Ramadge and Wonham in the late 1980s [19].

In many modern applications, controllers are implemented in networked environments where system components are connected with each other via communication networks [11]. Control systems with such networked information structures are referred to as the *networked control systems* (NCSs). In the context of DES, the development of NCSs also motivates a new active research area called *networked discrete-event systems*; see, e.g., [10], [17]. As depicted in Figure 1, in networked supervisory control systems, there are generally two communication channels: (i) the sensors send observable events to the supervisor via the *observation channel*; and (ii) the supervisor sends control decisions to the actuators via the *control channel*. Compared with classical supervisory control systems, networked supervisory control systems have several advantages. In particular, the networked information structure allows us to use outside computational devices to control large-scale systems remotely.

This work is supported by the National Natural Science Foundation of China (61803259, 61833012).

Xiang Yin and Shaoyuan Li are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. {yinxiang,syli}@sjtu.edu.cn.

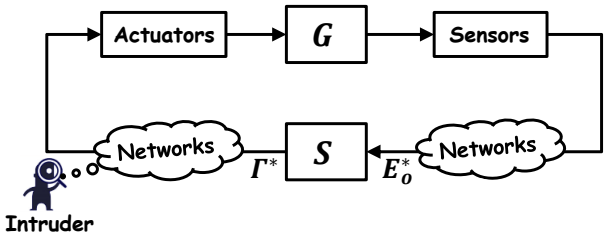


Fig. 1. A networked supervisory control system with insecure control channel.

Although networked control systems have many advantages compared with classical control systems, security issue becomes one of the main challenges in NCSs due to large communications. In particular, the control channel and the observation channel may not be secure in the sense that the information sent in the channels may be "listened" by *intruders*. Therefore, some "secret" of the system may be revealed to the intruder due to the information leak. In the context of networked DES, unfortunately, most existing works in the literature focus on the issues of communication delays and losses; see, e.g., [4], [13], [14], [16], [17], [20], [25]. The security issue, which is becoming progressively more important, has not been systematically studied in the context of networked DES.

In this paper, we propose an approach to analyze security for networked supervisory control systems with insecure control channels. Specifically, we consider a DES controlled by a networked supervisor as shown in Figure 1. We assume that the observation channel is secure, i.e., the intruder does not know the observation sequence sent by the sensors. However, the control channel between the supervisor and the actuators is assumed to be insecure in the sense that the control decisions sent by the supervisor can be accessed by an *intruder* that is potentially malicious. Such a scenario arises when the supervisor is located close to the sensors but far way from the actuators, and hence, the supervisor needs to control the actuators remotely via communication networks. The goal is to analyze, in a formal manner, whether or not the supervisory control system is secure under such an insecure control channel.

Our approach is to adopt the concept of *opacity* to characterize whether or not the networked supervisory control system is secure. Opacity is an information-flow property that captures the plausible deniability of the system's "secret" under possible information leak. A system is said to be (current-state) opaque if an outside intruder (potentially malicious) cannot infer unambiguously, based on its limited information about the system, that the system is at a secret state. The

property of opacity has drawn considerable attention in the last few years in the DES literature; see, e.g., [1]–[3], [6], [8], [9], [12], [15], [18], [22]–[24] and a recent survey [7] for more references. In our context, since we assume that the control channel is not secure, we say that the supervisory control system is opaque if the intruder cannot infer unambiguously that the system is at a secret state based on the control decisions sent by the supervisor.

In the standard supervisory control theory, the supervisor updates its control decision and then sends it to the actuators once a new observable event occurs. We call such decision transmission mechanism *event-based*. However, communication is usually costly in the networked environment. To reduce the communication burden, the networked supervisor may not need to resend a control decision if it is the same as the previous one. Instead, the supervisor can just communicate only when the control decision needs to be changed. We call such a decision transmission mechanism *decision-based*, which will not change the closed-loop behavior of the system since the actuators will still disable the same events when the current control decision is the same as the previous one. In the paper, both the event-based and the decision-based transmission mechanisms are investiagted. For each case, we provide the definition of opacity and propose a corresponding verification algorithm.

II. Preliminary

A. System Model

Let E be a finite set of events. A string is a finite sequence of events. We denote by E^* the set of all strings over E including the empty string ϵ. A language $L \subseteq E^*$ is a set of strings. A discrete-event system is modeled as a finite-state automaton (FSA)

$$G = (X, E, \delta, x_0)$$

where X is a finite set of states, E is a finite set of events, $\delta : X \times E \to X$ is the (partial) transition function and $x_0 \in X$ is the initial state. The transition function δ is also extended to $X \times E^* \to X$ in the usual manner; see, e.g., [5]. For the sake of simplicity, we write $\delta(x, s)$ as $\delta(s)$ if $x = x_0$. The language generated by G is $\mathcal{L}(G) = \{s \in E^* : \delta(s)!\}$, where "!" means "is defined". For each state $x \in X$, we denote by $\Sigma_G(x) = \{\sigma \in E : \delta(x, \sigma)!\}$ the set of events defined at x.

B. Supervisory Control Theory

In many cases, the open-loop system G may not satisfy some design specifications. Hence, the *supervisory control theory* has been widely adopted to enforce desired properties for DES. In the supervisory control framework, the event set is assumed to be partitioned as $E = E_c \dot\cup E_{uc} = E_o \dot\cup E_{uo}$, where E_c is the set of controllable events, E_{uc} is the set of uncontrollable events, E_o is the set of observable events and E_{uo} is the set of unobservable events. In general, there is no relationship between E_c and E_o, i.e., a controllable event could be either observable or unobservable. We define $\Gamma = \{\gamma \in 2^E : E_{uc} \subseteq \gamma\}$ as the set of control decisions (or control patterns), i.e., uncontrollable events are always enabled. The natural projection $P : E^* \to E_o^*$ is defined recursively by: for any $s \in E^*, \sigma \in E$, we have

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

The natural projection is also extended to $P : 2^{E^*} \to 2^{E^*}$ by: for any $L \subseteq E^*$, $P(L) = \{P(s) \in E_o^* : s \in L\}$.

A supervisor $\mathbb{S} : P(\mathcal{L}(G)) \to \Gamma$ is a function that makes control decisions dynamically based on its observations. Specifically, supervisor $\mathbb{S}$ works as follows. Initially, it issues an initial control decision $\mathbb{S}(\epsilon)$, i.e., only events in $\mathbb{S}(\epsilon)$ are *enabled*. The control decision is then sent to each actuator associated with each controllable event via the control channel. Once the first observable event σ_1 occurs, the supervisor changes its control decision to $\mathbb{S}(\sigma_1)$ and sends it. Similarly, when the second observable event σ_2 occurs, the supervisor again changes its control decision to $\mathbb{S}(\sigma_1\sigma_2)$ and sends it, and so forth. We denote by $\mathcal{L}(\mathbb{S}/G)$ the language generated by the closed-loop system, which is defined recursively by: $\epsilon \in \mathcal{L}(\mathbb{S}/G)$; and $s\sigma \in \mathcal{L}(\mathbb{S}/G)$ iff $s \in \mathcal{L}(\mathbb{S}/G) \wedge s\sigma \in \mathcal{L}(G) \wedge \sigma \in \mathbb{S}(P(s))$.

C. State-Space Recognizers of Supervisors

In practice, supervisor $\mathbb{S}$ is usually *recognized* as a FSA[1]

$$S = (X_S, E, \delta_S, x_{0,S})$$

such that the following conditions hold:

- $\forall x \in X_S, \forall \sigma \in E_{uo} : \delta_S(x, \sigma)! \Rightarrow \delta_S(x, \sigma) = x$; and
- $\forall s \in \mathcal{L}(S) : \Sigma_S(\delta_S(s)) = \mathbb{S}(P(s))$.

Essentially, the first condition says that only observable events can update control decisions (by updating states in G). The second condition says that S indeed works the same as $\mathbb{S}$, i.e., the set of events defined at each state encodes the current control decision. This also implies, implicitly, that uncontrollable events are always defined at each state in S. We denote by

$$G \times S = (X_{G\times S}, E, \delta_{G\times S}, x_{0,G\times S})$$

the product (c.f. [5] pp. 78) of G and S and we have $\mathcal{L}(G \times S) = \mathcal{L}(\mathbb{S}/G)$. In this paper, we assume that a supervisor always has its FSA recognizer (this essentially requires that the supervisor can only have a finite memory). Hereafter, we will only consider the recognizer FSA S and will use $\mathcal{L}(G \times S)$ instead of $\mathcal{L}(S/G)$. For the sake of simplicity, for any $s \in \mathcal{L}(G \times S)$, we also write the control decision upon the occurrence of s as $S(s)$, i.e., $S(s) := \Sigma_S(\delta_S(s)) = \Sigma_S(\delta_S(P(s))) = \mathbb{S}(P(s))$.

III. Opacity in Networked Supervisory Control Systems

A. Opacity under Event-Based Transmission

In networked supervisory control systems, control decisions made by the supervisor need to be transmitted to

[1]The FSA representation of a supervisor usually comes directly, by construction, from supervisor synthesis procedures.

the actuators via the control channel. More specifically, let $s \in \mathcal{L}(G \times S)$ be a string generated by the closed-loop system and assume that $P(s) = \sigma_1 \dots \sigma_n, \sigma \in E_o$. Then upon the occurrence of string s, supervisor S issues *a sequence of control decisions* defined by

$$D_S(s) = S(\epsilon)S(\sigma_1)\cdots S(\sigma_1 \dots \sigma_n) \in \Gamma^*$$

We also refer to $D_S(s)$ as the *decision history* of string s. We call the control decision transmission mechanism *event-based* if $D_S(s)$ is entire information history the supervisor sends to the actuators, i.e., the supervisor will always send $S(\sigma_1 \dots \sigma_n)$ when $\sigma_1 \dots \sigma_n$ is observed. We denote by

$$\mathcal{D}_S = \{D_S(s) \in \Gamma^* : s \in \mathcal{L}(G \times S)\}$$

the set of all possible decision histories in the supervisory control system.

As depicted in Figure 1, however, the information transmission between the supervisor and the actuators may not be secure and there may exist an *intruder* that can access the decision history. The question then arises as whether or not this information leak will reveal some "secret" of the system. Formally, we consider an intruder having the following capabilities:

- A1 The intruder knows the system model and the functionality of the supervisor; and
- A2 The control channel is not secure so that the intruder knows all information the supervisor sends to the actuators.

In order to characterize whether or not the supervisory control system is secure, motivated by recent works on information-flow analysis of DES, we propose to use the notion of *opacity* in this problem. Specifically, we assume that the system has a "secret" modeled as a set of secret states $X_{secret} \subset X$. We say that the supervisory control system is *opaque* under event-based transmission if the intruder can never infer unambiguously that the system is at a secret state based on the decision history. This leads to the following definition.

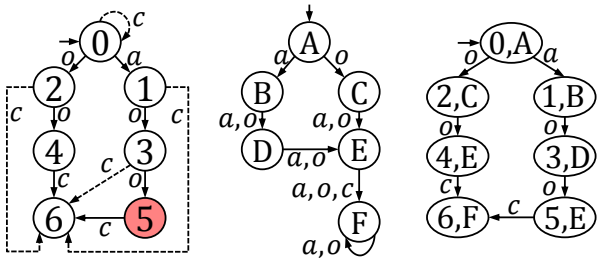
Definition 1: Supervisory control system (G, S) is said to be *opaque under event-based transmission* if

$$(\forall s \in \mathcal{L}(G \times S) : \delta(s) \in X_{Secret}) \quad (2)$$
$$(\exists t \in \mathcal{L}(G \times S) : \delta(t) \notin X_{Secret})[D_S(s) = D_S(t)]$$

Intuitively, opacity requires that for any string that goes to a secret state, there exists a string that goes to a non-secret state such that the supervisor will produce the same decision history for these two different strings. Therefore, the system can guarantee its plausible deniability for visiting a secret state even when the control channel is "listened" by the intruder.

B. Opacity under Decision-Based Transmission

In the event-based transmission mechanism, we assume that the supervisor will send the latest control decision whenever a new event is observed. However, in networked control systems, sending control decisions requires communication, which is costly in general. Therefore, the supervisor may not need to resend the newly computed control decision if it is the same as the previous one. Instead, it suffices to send a new control decision only when it is different from the previously one. We call such a control decision transmission mechanism *decision-based*. The decision-based transmission mechanism will reduce communication burden in the control channel without affecting the behavior of the closed-loop system, since the supervisor will still use the previous control decision when it is the same as the current one.



(a) G (b) S (c) $G \times S$

Fig. 2. A supervisory control system (G, S)

.

Formally, for any control decision sequence $\alpha \in \Gamma^*$, we denote by $\text{LAST}(\alpha)$ the last control decision in α with $\text{LAST}(\epsilon) := \epsilon$. Then we define the *event filtering function*

$$\mathcal{F} : \Gamma^* \to \Gamma^*$$

recursively by: for any $\alpha \in \Gamma^*, \gamma \in \Gamma$, we have

$$\mathcal{F}(\epsilon) = \epsilon \quad \text{and} \quad \mathcal{F}(\alpha\gamma) = \begin{cases} \mathcal{F}(\alpha)\gamma & \text{if } \gamma \neq \text{LAST}(\alpha) \\ \mathcal{F}(\alpha) & \text{if } \gamma = \text{LAST}(\alpha) \end{cases}$$

Function $\mathcal{F}$ is also extended to $\mathcal{F} : 2^{\Gamma^*} \to 2^{\Gamma^*}$. Therefore, in the decision-based transmission mechanism, upon the occurrence of $s \in \mathcal{L}(G \times S)$, the intruder will observe decision string $\mathcal{F}(D_S(s)) \in \Gamma^*$ in the control channel. This also leads to the following definition of opacity.

Definition 2: Supervisory control system (G, S) is said to be *opaque under decision-based transmission* if

$$(\forall s \in \mathcal{L}(G \times S) : \delta(s) \in X_{Secret}) \quad (3)$$
$$(\exists t \in \mathcal{L}(G \times S) : \delta(t) \notin X_{Secret})[\mathcal{F}(D_S(s)) = \mathcal{F}(D_S(t))].$$

In Sections III and IV, we will show formally how to verify opacity under event-based transmission and opacity under decision-based transmission, respectively. First, we illustrate Definitions 1 and 2 by the following example.

Example 1: Let us consider system G shown in Figure 2(a) with $\Sigma_c = \{c\}$ and $\Sigma_o = \{o, a, c\}$, i.e., all events are observable and the supervisor can only decide to enable or disable event c at each instant. Suppose that the control objective is to prevent the dashed transitions from happening. This objective can be achieved by supervisor S shown in Figure 2(b) and the closed-loop language is generated by $G \times S$ shown in Figure 2(c). Suppose that $X_{Secret} = \{5\}$, i.e., the intruder should never know for sure that the system is at the unique secret state 5.

This system is not opaque under event-based transmission. To see this, let us consider string $aoo \in \mathcal{L}(G \times S)$,

where $\delta(aoo) = 5 \in X_{Secret}$. The decision history along this trajectory is $D_S(aoo) = S(\epsilon)S(a)S(ao)S(aoo) = \{o,a\}\{o,a\}\{o,a\}\{o,a,c\}$. Since events o and a are uncontrollable, which are always enabled in the control decision, for the sake of simplicity, we will omit uncontrollable events in each control decision and write the above decision history as $D_S(aoo) = \{\}\{\}\{\}\{c\}$. However, aoo is the unique string that can induce decision history $\{\}\{\}\{\}\{c\}$. Therefore, by observing $\{\}\{\}\{\}\{c\}$ in the control channel, the intruder knows unambiguously that the system is at secret state 5.

However, this system is opaque under decision-based transmission. To see this, we still consider the unique string leading to secret state 5, i.e., aoo. Under the decision-based transmission mechanism, only filtered decision history $\mathcal{F}(D_S(aoo)) = \mathcal{F}(\{\}\{\}\{\}\{c\}) = \{\}\{c\}$ is sent in the control channel upon the occurrence of aoo. However, we can find another string oo leading to non-secret state 4 such that $\mathcal{F}(D_S(oo)) = \mathcal{F}(\{\}\{\}\{c\}) = \{\}\{c\} = \mathcal{F}(D_S(aoo))$. Therefore, the intruder does not know whether the system is at secret state 5 or at non-secret state 4 by observing $\{\}\{c\}$ in the control channel. ■

IV. Verification of Opacity under Event-Based Transmission

In this section, we show how to verify opacity under event-based transmission. First, we define

$$\mathcal{E}(\alpha) := \{x \in X : \exists s \in \mathcal{L}(G \times S) \text{ s.t. } x = \delta(s) \wedge D_S(s) = \alpha\}$$

as the *state estimate* when decision history α is generated. According to Definition 1, the system is opaque under event-based transmission if and only if $\forall \alpha \in \mathcal{D}_S : \mathcal{E}(\alpha) \not\subseteq X_{Secret}$. Therefore, the key to the opacity verification problem is to effectively compute $\mathcal{E}(\alpha)$ for all possible control decision history. However, this cannot be done by the standard observer automaton [5] since there are two levels of inferences here: we need to first use the decision history to infer all possible observable strings and then use each possible observable string to infer all possible actual strings generated by the system. To solve this state estimation problem, we propose the following *Decision-to-State Observer* (D-observer) under event-based transmission.

Definition 3: Let $G = (X, E, \delta, x_0)$ be a system and $S = (X_S, E, \delta_S, x_{0,S})$ be a supervisor. The *decision-to-state observer under event-based transmission* is a FSA

$$Obs(G, S) = (Q, \Gamma, f, q_0) \tag{4}$$

where

- $Q \subseteq 2^{X_{G\times S}}$ is the set of states;
- Γ is the set of control decisions, i.e., each event in $Obs(G, S)$ is a control decision;
- $f : Q \times \Gamma \to Q$ is the transition function defined by: for any $q, q' \in Q, \gamma \in \Gamma$, $f(q, \gamma) = q'$ if

$$q' = \left\{ (x', x'_s) \in X_{G\times S} : \begin{array}{r} \exists (x, x_s) \in q, \sigma \in E_o, w \in E^*_{uo} \\ \text{s.t. } (x', x'_s) = \delta_{G\times S}((x, x_s), \sigma w) \\ \text{and } \Sigma_S(x'_s) = \gamma \end{array} \right\}$$

- The initial state is defined by

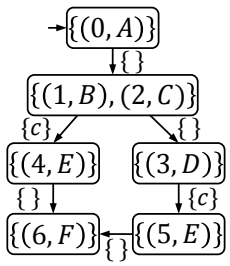


Fig. 3. The D-observer under event-based transmission $Obs(G, S)$.

$$q_0 = \{(x, x_{0,S}) \in X_{G\times S} : \exists w \in E^*_{uo} \text{ s.t. } (x, x_{0,S}) = \delta_{G\times S}(w)\}$$

We only consider the reachable part of $Obs(G, S)$.

Note that, for any decision history $\alpha \in \mathcal{D}_S$, it can be written in the form of $\alpha = S(\epsilon)\alpha'$. In other words, the initial control decision $S(\epsilon)$ does not carry additional information for the intruder since any decision history starts with $S(\epsilon)$. Next, we present the main properties of $Obs(G, S)$.

First, we show that $Obs(G, S)$ can track all possible states consistent with the decision history.

Proposition 1: For any $\alpha = S(\epsilon)\alpha' \in \mathcal{D}_S$, we have $\alpha' \in \mathcal{L}(Obs(G, S))$. Moreover,

$$q = \left\{ (x, x_s) \in X_{G\times S} : \begin{array}{c} \exists s \in \mathcal{L}(G \times S) \text{ s.t. } D_S(s) = \alpha \\ \text{and } (x, x_s) = \delta_{G\times S}(s) \end{array} \right\} \tag{5}$$

where $q = f(q_0, \alpha')$ is the state reached by α' in $Obs(G, S)$.

Next, we show that, in fact, $Obs(G, S)$ exactly generates all possible decision histories.

Proposition 2: $\{S(\epsilon)\}\mathcal{L}(Obs(G, S)) = \mathcal{D}_S$.

For any $q \in 2^{X\times X_S}$, we define $q|_X := \{x \in X : \exists x_s \in X_S \text{ s.t. } (x, x_s) \in q\}$ as the restriction to its first component. By Proposition 1, we know that, for any $\alpha = S(\epsilon)\alpha' \in \mathcal{D}_S$, $\mathcal{E}(\alpha) = f(\alpha')|_X$. Therefore, by Proposition 2, we have the following theorem that shows how to use $Obs(G, S)$ to verify opacity under event-based transmission.

Theorem 1: Supervisory control system (G, S) is opaque under event-based transmission w.r.t. X_{secret}, if and only if, $\forall q \in Q : q|_X \not\subseteq X_{Secret}$.

We illustrate the decision-to-state observer under event-based transmission by the following example.

Example 2: Again, let us consider system G and supervisor S shown in Figures 2(a) and 2(b), respectively, with $\Sigma_c = \{c\}, \Sigma_o = \{o, a, c\}$ and $X_{Secret} = \{5\}$. Here, we use Theorem 1 to verify whether or not the system is opaque under event-based transmission. To this end, we build the the decision-to-state observer under event-based transmission $Obs(G, S)$, which is shown in Figure 3. Since all events are observable, the initial state is $\{(0, A)\}$. By observing the first control decision $\{\}$, we move to state $\{(1, B), (2, C)\}$. In particular, the occurrences of events a and c lead to states $(1, B)$ and $(2, C)$, respectively, where $\Sigma_S(B) = \Sigma_S(C) = \{\}$. Since state $\{(5, E)\} \in Q$ and $\{(5, E)\}|_X = \{5\} \subseteq X_{Secret}$, we know that (G, S) is not opaque under event-based transmission, which is consistent with our result in Example 1. ■

V. Verification of Opacity under Decision-Based Transmission

In this section, we show how to verify opacity under decision-based transmission. Similarly, we define

$$\tilde{\mathcal{E}}(\alpha) := \{x \in X : \exists s \in \mathcal{L}(G \times S) \text{ s.t. } x = \delta(s) \wedge \mathcal{F}(D_S(s)) = \alpha\}$$

as the state estimate when filtered decision history α is observed under the decision-based transmission mechanism. In order to effectively compute $\tilde{\mathcal{E}}(\alpha)$ for each possible filtered decision history $\alpha \in \mathcal{F}(\mathcal{D}_S)$, we propose the following decision-to-state observer under decision-based transmission.

Definition 4: Let $G = (X, E, \delta, x_0)$ be a system and $S = (X_S, E, \delta_S, x_{0,S})$ be a supervisor. Then the *decision-to-state observer under event-based transmission* is a FSA

$$\tilde{Obs}(G, S) = (\tilde{Q}, \Gamma, \tilde{f}, \tilde{q}_0) \tag{6}$$

where

- $\tilde{Q} \subseteq 2^{X_{G \times S}}$ is the set of states;
- Γ is the set of control decisions, i.e., each event in $\tilde{Obs}(G, S)$ is a control decision;
- $\tilde{f} : \tilde{Q} \times \Gamma \to \tilde{Q}$ is the transition function defined by: for any $q, q' \in \tilde{Q}, \gamma \in \Gamma$, $\tilde{f}(q, \gamma) = q'$ if

$$\forall (x, x_s) \in q : \Sigma_S(x_s) \neq \gamma$$

and

$$q' = \left\{ \begin{matrix} (x', x'_s) \\ \in X_{G \times S} \end{matrix} : \begin{matrix} \exists (x, x_s) \in q, s \in E^* \\ \text{s.t. } (x', x'_s) = \delta_{G \times S}((x, x_s), s) \text{ and} \\ [\forall s' \in \overline{\{s\}} \setminus \{\epsilon\} : \Sigma_S(\delta_S(x_s, s')) = \gamma] \end{matrix} \right\} \tag{7}$$

- The initial state is defined by

$$q_0 = \left\{ \begin{matrix} (x, x_s) \\ \in X_{G \times S} \end{matrix} : \begin{matrix} \exists s \in E^* \text{ s.t. } (x, x_s) = \delta_{G \times S}(s) \\ \text{and } [\forall s' \in \overline{\{s\}} : \Sigma_S(\delta_S(s')) = S(\epsilon)] \end{matrix} \right\} \tag{8}$$

We only consider the reachable part of $\tilde{Obs}(G, S)$.

Still, for any filtered decision history $\alpha \in \mathcal{F}(\mathcal{D}_S)$, it can be written as $\alpha = S(\epsilon)\alpha'$. The following result states that the D-observer under decision-based transmission $\tilde{Obs}(S, G)$ tracks all possible states consistent with the filtered decision history.

Proposition 3: For any $\alpha = S(\epsilon)\alpha' \in \mathcal{F}(\mathcal{D}_S)$, we have $\alpha' \in \mathcal{L}(\tilde{Obs}(G, S))$. Moreover,

$$q = \left\{ (x, x_s) \in X_{G \times S} : \begin{matrix} \exists s \in \mathcal{L}(G \times S) \text{ s.t. } \mathcal{F}(D_S(s)) = \alpha \\ \text{and } (x, x_s) = \delta_{G \times S}(s) \end{matrix} \right\} \tag{9}$$

where $q = \tilde{f}(\tilde{q}_0, \alpha')$ is the state reached by α' in $\tilde{Obs}(G, S)$.

Next, we show that $\tilde{Obs}(G, S)$ exactly generates all possible filtered decision histories.

Proposition 4: $\{S(\epsilon)\}\mathcal{L}(\tilde{Obs}(G, S)) = \mathcal{F}(\mathcal{D}_S)$.

According to Definition 2, we know that the system is opaque under decision-based transmission if and only if $\forall \alpha \in \mathcal{F}(\mathcal{D}_S) : \tilde{\mathcal{E}}(\alpha) \not\subseteq X_{Secret}$. Moreover, by Proposition 3, we have $\tilde{\mathcal{E}}(\alpha) = \tilde{f}(\alpha')|_X$, where $\alpha = S(\epsilon)\alpha'$. Therefore, by Proposition 4, we have the following theorem that shows how to use $\tilde{Obs}(G, S)$ to verify opacity under decision-based transmission.

Theorem 2: Supervisory control system (G, S) is opaque under decision-based transmission w.r.t. X_{secret}, if and only if, $\forall q \in \tilde{Q} : q|_X \not\subseteq X_{secret}$.

In order to construct the D-observer under decision-based transmission (either online or off-line), we need to compute the initial state and each successor state based on Equations (8) and (7), respectively, which are rather involved. In Algorithm 1, we show specifically how the D-observer under decision-based transmission can be constructed. This algorithm involves two recursive procedures DoDFS and Search. Procedure DoDFS mainly aims to traverse the state space of $\tilde{Q}$ by a depth-first search from the initial state. At each state q encountered, it computes the successor state for each control decision γ that is different from the current control decision in q, i.e., $\forall (x, x_s) \in q : \Sigma_S(x_s) \neq \gamma$. The computation of the successor state q' is implemented by lines 8-10. First, we set q' as the empty set and then, we add states to q' by another depth-first search procedure Search. Specifically, for each state (x, x_s) and control decision γ, procedure Search considers all possible events whose occurrence will maintain control decision γ and repeat this by a recursive call until all such states have been explored. The initial state $\tilde{q}_0$ is computed in the same manner by procedure Search, where the depth-first search starts from $(x_0, x_{0,S})$ with $\gamma = S(\epsilon)$.

We illustrate the construction of $\tilde{Obs}(G, S)$ and show how to verify opacity under decision-based transmission using Theorem 2 by the following example.

Example 3: Still, let us consider system G and supervisor S shown in Figures 2(a) and 2(b), respectively, with $\Sigma_c = \{c\}, \Sigma_o = \{o, a, c\}$ and $X_{Secret} = \{5\}$. We have shown in Example 1 that this system is opaque under decision-based transmission. Here, we verify this result by Theorem 2.

The D-observer under decision-based transmission for (G, S) is shown in Figure 4. The initial state is $\tilde{q}_0 = \{(0, A), (1, B), (2, C), (3, D)\}$; this can be computed by procedure Search in line 2. Specifically, we start from $(0, A)$ and search all reachable states whose control decision is $S(\epsilon) = \{\}$. Then we expend the D-observer by procedure DoDFS from the initial state $\tilde{q}_0$. At $\tilde{q}_0$, only control decision $\{c\}$ satisfies the condition in line 7 since it is only control decision different from $\{\}$. Then procedure Search needs to consider each state in $\tilde{q}_0$. In particular, Search$((0, A), \{c\})$ and Search$((1, B), \{c\})$ do not add state to q' since no state whose control decision is $\{c\}$ can be reached. However, Search$((2, C), \{c\})$ and Search$((3, D), \{c\})$ will add states $(4, E)$ and $(5, E)$ to q', respectively. Therefore, the successor state of $\tilde{q}_0$ upon the occurrence of $\{c\}$ is $\{(4, E), (5, E)\}$. Then we consider control decision $\{\}$ from $\{(4, E), (5, E)\}$ and we reach successor state $\{(6, F)\}$. This completes the construction of $\tilde{Obs}(G, S)$. For this D-observer, we have $\{(0, A), (1, B), (2, C), (3, D)\}|_X = \{0, 1, 2, 3\}$, $\{(4, E), (5, E)\}|_X = \{4, 5\}$ and $\{(6, F)\}|_X = \{6\}$; none of them is a subset of X_{Secret}. Therefore, by Theorem 2, we conclude that the system is opaque under decision-based transmission. ■

Remark 1: We conclude this section by analyzing the

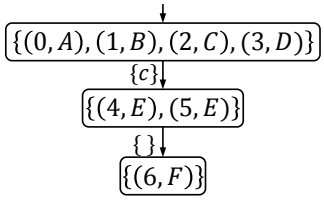


Fig. 4. The D-observer under decision-based transmission $\tilde{Obs}(G, S)$.

Algorithm 1: Compute D-Observer $\tilde{Obs}(G, S)$

input : G and S
output: $\tilde{Obs}(G, S)$

$q' \leftarrow \emptyset$;
SEARCH($(x_0, x_{0,S}), S(\epsilon)$);
$\tilde{q}_0 \leftarrow q', \tilde{Q} \leftarrow \{\tilde{q}_0\}$;
DODFS($\tilde{Obs}(G, S), \tilde{q}_0$);
return $\tilde{Obs}(G, S) = (\tilde{Q}, \Gamma, \tilde{f}, \tilde{q}_0)$;

procedure DODFS($\tilde{Obs}(G, S), q$);
  for $\gamma \in \Gamma$ **do**
    if $\forall (x, x_s) \in q : \Sigma_S(x_s) \neq \gamma$ **then**
      $q' \leftarrow \emptyset$;
      for $(x, x_s) \in q$ **do**
        SEARCH($(x, x_s), \gamma$);
      if $q' \notin \tilde{Q}$ **then**
        $\tilde{Q} \leftarrow \tilde{Q} \cup \{q'\}$;
        Add transition $q \xrightarrow{\gamma} q'$ to $\tilde{f}$;
        DODFS($\tilde{Obs}(G, S), q'$);

procedure SEARCH($(x, x_s), \gamma$);
  for $\sigma \in E$ **do**
    if $\delta_{G \times S}((x, x_s), \sigma)!$ **then**
      $(x', x'_s) := \delta_{G \times S}((x, x_s), \sigma)$;
      if $\Sigma_S(x'_s) = \gamma \wedge (x', x'_s) \notin q'$ **then**
        $q' \leftarrow q' \cup \{(x', x'_s)\}$;
        SEARCH($(x', x'_s), \gamma$);

complexity of the proposed algorithms for checking opacity. In the worst case, both $Obs(G, S)$ and $\tilde{Obs}(G, S)$ contain at most $2^{|X| \times |X_S|}$ states and $|2^{|E_c|}| \times 2^{|X| \times |X_S|}$ transitions. Therefore, the complexity of verifying opacity under event-based transmission and the complexity of verifying opacity under decision-based transmission are both exponential in the number of states and the number of events in G and S.

VI. Conclusion

In the paper, we presented a framework for analyzing security for networked supervisory control systems with insecure control channels. We adopted the notion of opacity to characterize whether or not the system's secret can be revealed to an intruder that can access all control decisions in the control channel. Two types of opacity were defined and effective algorithms were also provided to verify different notions of opacity. An interesting future direction is to consider the synthesis of a maximally-permissive supervisor [21] that is provably opaque under insecure control channel.

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, J. Mullins, and M. Sassolas. Quantifying opacity. *Mathematical Structures in Computer Science*, 25(2):361–403, 2015.

[3] 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.

[4] L.K. Carvalho, J.C. Basilio, and M.V. Moreira. Robust diagnosis of discrete event systems against intermittent loss of observations. *Automatica*, 48(9):2068–2078, 2012.

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

[6] 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.

[7] 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.

[8] C. Keroglou and S. Lafortune. Verification and synthesis of embedded insertion functions for opacity enforcement. In *56th IEEE Conference on Decision and Control*, pages 4217–4223, 2017.

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

[10] F. Lin. Control of networked discrete event systems: dealing with communication delays and losses. *SIAM Journal on Control and Optimization*, 52(2):1276–1298, 2014.

[11] J. Lunze. *Control Theory of Digitally Networked Dynamic Systems*. Springer, 2014.

[12] 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.

[13] C. Nunes, M.V. Moreira, M. Alves, L.K. Carvalho, and J.C. Basilio. Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation. *Discrete Event Dynamic Systems*, 2018.

[14] S.-J. Park and K.-H. Cho. Delay-robust supervisory control of discrete-event systems with bounded communication delays. *IEEE Transactions on Automatic Control*, 51(5):911–915, 2006.

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

[16] S. Shu and F. Lin. Deterministic networked control of discrete event systems with nondeterministic communication delays. *IEEE Transactions on Automatic Control*, 62(1):190–205, 2017.

[17] S. Shu and F. Lin. Predictive networked control of discrete event systems. *IEEE TAC*, 62(9):4698–4705, 2017.

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

[19] W.M. Wonham, K. Cai, and K. Rudie. Supervisory control of discrete-event systems: A brief history–1980-2015. *IFAC-PapersOnLine*, 50(1):1791–1797, 2017.

[20] X. Yin. Supervisor synthesis for mealy automata with output functions: A model transformation approach. *IEEE Transactions on Automatic Control*, 62(5):2576–2581, 2017.

[21] X. Yin and S. Lafortune. Synthesis of maximally permissive supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 61(5):1239–1254, 2016.

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

[23] 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.

[24] X. Yin, Z. Li, W. Wang, and S. Li. Infinite-step opacity and K-step opacity of stochastic discrete-event systems. *Automatica*, 2019.

[25] R. Zhang, K. Cai, Y. Gan, and W.M. Wonham. Distributed supervisory control of discrete-event systems with communication delay. *Discrete Event Dynamic Systems*, 26(2):263–293, 2016.