Proceedings of the SICE Annual Conference 2018
September 11-14, 2018, Nara, Japan

A Framework for Opacity in Networked Supervisory Control Systems

Xiang Yin[1†]

[1]Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China.
(Tel: +86-13636434613; E-mail: yinxiang@sjtu.edu.cn)

Abstract: In this extended abstract, we discuss a framework for opacity in networked supervisory control systems. In networked control systems, the supervisors send control decisions to actuators by control channels implemented by communication networks. Security and privacy then become important issues in such systems since communications may not be secure. We consider networked supervisory control systems where the control channels may not be secure in the sense that the control decisions sent by the supervisors can be "listened" by intruders. We present a framework, by adopting the notion of opacity, to capture whether or not a supervisory control system is secure. We provide two definitions of opacity and discuss some relevant problems in this framework.

Keywords: Discrete Event Systems, Security, Supervisory Control, Networked Control Systems

1. INTRODUCTION

Supervisory control theory is a widely used formal methods to enforce closed-loop behaviors for Discrete-Event Systems (DES) [7]. In many modern applications, controllers are implemented in networked environments where system components are connected via communication networks. In the context of DES, *supervisory control of networked DES* has drawn considerable attention in the past years; see, e.g., [4, 5]. The basic diagram of a networked supervisory control system is shown in Figure 1, where sensors send observable events to the supervisor via the observation channel and the supervisor sends control decisions to actuators via the control channel. The main advantage of using networked control architecture is that it allows us to use outside devices to control large-scale systems remotely.

Most existing works on supervisory control of networked DES focus on handling communication delays and losses in the control and observation channels; see, e.g., [4–6, 9]. However, security issue has not been fully investigated in the context of networked supervisory control systems. This issue is particularly important since communication networks may not be secure, which may lead to privacy and security leakage of the control system.

2. OPACITY IN NETWORKED SUPERVISORY CONTROL SYSTEMS

In this extended abstract, we discuss a framework for analyzing security of networked supervisory control systems shown in Figure 1 with *insecure control channels*.

To formalize this framework, we use standard notations in DES; the reader is referred to [1] for details. We consider a DES modeled as a finite-state automaton $G = (X, E, \delta, x_0)$. The event set is partitioned by $E = E_c \dot\cup E_{uc} = E_o \dot\cup E_{uo}$, 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. We define $\Gamma = \{\gamma \in 2^{E_c} : E_{uc} \subseteq \gamma\}$ as the set of control decisions and define $P : E^* \to E_o^*$ as the natural projection. A supervisor $S : P(\mathcal{L}(G)) \to \Gamma$ is a function that makes control decisions dynamically based on its observations. We denote by $\mathcal{L}(S/G)$ the language generated by the closed-loop system under control.

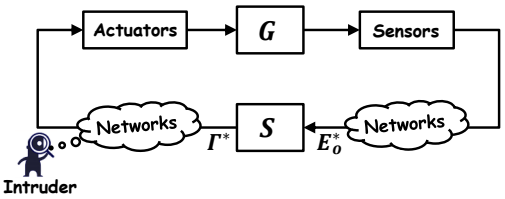

Fig. 1 A networked supervisory control system with insecure control channel.

In networked supervisory control systems, control decisions made by the supervisors are transmitted to actuators via control channels. More specifically, let $s \in \mathcal{L}(S/G)$ 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 generates a *decision history* defined by

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

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

In many applications, 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. Motivated by recent works on information-flow analysis of DES, we propose to use the notion of *opacity* to capture the security of the system. The reader is referred to a recent survey [3] for extensive references on opacity of DES. Specifically, we assume that the system has a "secret" modeled as a set of secret states $X_s \subset X$. We say that (G, S) 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 in the control channel.

† Xiang Yin is the presenter of this paper.

Definition 1: Supervisory control system (G, S) is said to be *opaque under event-based transmission* if for any string $s \in \mathcal{L}(S/G) : \delta(s) \in X_S$, there exists a string $t \in \mathcal{L}(S/G) : \delta(t) \notin X_S$ such that $D_S(s) = D_S(t)$.

Opacity requires that for any string leading to a secret state, there exists a string leading to a non-secret state such that the supervisor will produce the same decision history for these two strings. Hence, the intruder cannot infer unambiguously that the system is at a secret state.

In the event-based transmission mechanism, the supervisor will send the latest control decision whenever a new event is observed. However, in networked control systems, sending control decisions 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. This will reduce communication burden in the control channel and will not affect the behavior of the closed-loop system, since the supervisor will still use the previous control decision which is the same as the current one. We call such decision transmission mechanism *decision-based*.

Formally, for any control decision sequence $\alpha \in \Gamma^*$, we denote by $\textsc{Last}(\alpha)$ the last control decision in α with $\textsc{Last}(\epsilon) := \epsilon$. Then we define the *event filtering function* $\mathcal{F} : \Gamma^* \to \Gamma^*$ recursively by: $\forall \alpha \in \Gamma^*, \gamma \in \Gamma$, we have

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

Therefore, in decision-based transmission, upon the occurrence of $s \in \mathcal{L}(G)$, the intruder will observe decision string $\mathcal{F}(D_S(s)) \in \Gamma^*$ in the control channel.

Definition 2: Supervisory control system (G, S) is said to be *opaque under decision-based transmission* if $\forall s \in \mathcal{L}(S/G) : \delta(s) \in X_S$, there exists a string $t \in \mathcal{L}(S/G) : \delta(t) \notin X_S$ such that $\mathcal{F}(D_S(s)) = \mathcal{F}(D_S(t))$.

3. RELEVANT PROBLEMS

In this section, we discuss some relevant problems in our framework. The first relevant question is how to check opacity.

Problem 1: (**Opacity Verification Problem**).
Given: a supervisory control system (G, S).
Decide: whether or not (G, S) is opaque under event (or decision)-based transmission.

Suppose that (G, S) is verified to be non-opaque. Then another relevant problem is how to re-design the control system, without changing the closed-loop behavior, such that the new system is opaque.

Problem 2: (**Opacity Redesign Problem**).
Given: a supervisory control system (G, S) that is not opaque under event (or decision)-based transmission.
Synthesize: a new supervisor S' such that (G, S') is opaque under event (or decision)-based transmission and $\mathcal{L}(S'/G) = \mathcal{L}(S/G)$.

Problem 2 essentially asks us to re-design the control decisions sent by the supervisor, by adding or removing infeasible events, such that the closed-loop language remains the same and the new system is opaque. An alternative way to guarantee opacity is to consider opacity as a requirement at the design stage of the supervisory control system. This leads to the following synthesis problem.

Problem 3: (**Opacity Synthesis Problem**).
Given: a DES G and secret states X_S.
Synthesize: a maximally-permissive safe and nonblocking supervisor S such that (G, S) is opaque under event-based (or decision-based) transmission.

Note that, Problem 3 is different from the problem of supervisory control for opacity [2]. In the opacity control problem, it is assumed that the intruder can monitor observable events directly, whereas in Problem 3, we assume that the observation channel is secure and the intruder can only "listen" the control channel. Problem 3 is very challenging since the intruder's information depends on what control decisions the supervisor decides to send.

In a companion paper [8], we solve Problem 1 by providing two effective algorithms for verifying opacity under event-based transmission and opacity under event-based transmission, respectively. The basic idea is to construct a structure called *decision-to-state observer* that estimates the current state of the system based on the decision history. However, Problems 2 and 3 are still open problems. How to solve them are important future works in this framework.

REFERENCES

[1] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, 2008.

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

[3] 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.

[4] F. Lin. Control of networked discrete event systems: dealing with communication delays and losses. *SIAM J. Contr. Optim.*, 52(2):1276–1298, 2014.

[5] S. Shu and F. Lin. Predictive networked control of discrete event systems. *IEEE Transactions on Automatic Control*, 62(9):4698–4705, 2017.

[6] T. Ushio and S. Takai. Nonblocking supervisory control of discrete event systems modeled by mealy automata with nondeterministic output functions. *IEEE TAC*, 61(3):799–804, 2016.

[7] W.M. Wonham, K. Cai, and K. Rudie. Supervisory control of discrete-event systems: A brief history–1980-2015. In *20th IFAC World Congress*, pages 1791–1797, 2017.

[8] X. Yin and S. Li. Verification of opacity in networked supervisory control systems with insecure control channels.

[9] 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.