A Polynomial Approach to Verifying the Existence of A Threatening Sensor Attacker*

Yin Tong, *Member, IEEE*, and Yucheng Wang, and Alessandro, Giua, *Fellow, IEEE*

Abstract

The development of cyber-physical systems (CPS) has brought much attention of researchers to cyber-attack and cyber-security. A sensor attacker targeting on supervised discrete event system can modify a set of sensor readings and cause the closed-loop system to reach undesirable states. In this paper, we propose a new attack detection mechanism under which the supervisor only needs to keep track of the last observable event received. Given a plant and a supervisor enforcing a state specification, we define a sensor attacker *threatening* if it may cause the closed-loop system to enter a forbidden state. Our goal is to verify whether there exists such a threatening sensor attacker for a given controlled system. A new structure, called All Sensor Attack (ASA), is proposed to capture all possible sensor attacks launched by the attacker. Based on the ASA automaton, a necessary and sufficient condition for the existence of a stealthy threatening sensor attacker is presented. Finally, we show that the condition can be verified in polynomial time.

Index Terms

Discrete event systems, cyber-physical systems, sensor attack

Published as:

Y. Tong, Y. Wang and A. Giua, "A Polynomial Approach to Verifying the Existence of a Threatening Sensor Attacker" in IEEE Control Systems Letters, vol. 6, pp. 2930-2935, 2022, doi: 10.1109/LCSYS.2022.3163467.

---

*This work was supported by the National Natural Science Foundation of China under Grant No. 61803317.

Y. Tong (Corresponding Author) and Y. C. Wang are both with the School of Information Science and Technology, Southwest Jiaotong University, 611756 Chengdu, China yintong@swjtu.edu.cn; yuchengwang@my.swjtu.edu.cn

A. Giua is with with the Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy giua@unica.it

I. INTRODUCTION

CYBER-PHYSICAL systems (CPS) are engineered systems that are built from, and depend upon, the seamless integration of computation and physical components. In this paper, we consider the CPS that has been modeled with a supervisory controlled discrete event system (DES). Many works in recent years have been devoted to problems of cyber-security of discrete event systems [1].

In [2], the problem of attack detection and controllability is addressed. Four types of attacks: actuator enablement attack (AE-attack), actuator disablement attack (AD-attack), sensor erasure attack (SE-attack), and sensor insertion attack (SI-attack), are considered in the controlled DES. Actuator attack can enable/disable events that are disabled/enabled by the supervisor while sensor attack can change the observation of the supervisor by erasing or inserting the observation of some events. It is shown that diagnoser can be constructed to detect attacks. Polynomial algorithms are developed to verify if the controlled system can stay safe after attacks are detected. In [3], sensor attack is generalized to multiple attacks such that the observation of the system is modified based on a set of static or dynamic attack dictionary, and a diagnoser is constructed to detect the attack. Yin et al. [4] propose a new concept approximate opacity in systems with metrics output to quantitatively evaluate the security level.

Rather than passively detecting the existence of attacks or verifying if the system is safe under attack, a majority of works on cyber-security in DES focuses on synthesizing a resilient supervisor against attacks. In [5], given specific bounded sensor attacks, a method of synthesizing a supremal controllable and normal resilient supervisor against the attack is proposed. In [6], it is shown that the problems of testing the existence of resilient supervisor and synthesizing it can be solved using classical supervisory control theory without expanding the state space of the original supervisory control problem. In the presence of both actuator attack and sensor attack, algorithms for synthesizing a resilient supervisor are presented in [7]. In [8], without imposing normality condition, a game theoretical approach is proposed to obtain a supervisor such that for any sensor attack the supervisor is resilient.

Our perspective in this paper is different from the aforementioned works, as we aim at verifying whether there exists a sensor attacker for a given plant and supervisor such that under its attack the supervisor fails to keep the plant within the specification. Without considering supervisory controlled system, Zhang et al. [9] study the problem of state estimation of plant under sensor attack. A structure called joint estimator is proposed to describe all possible attacks and the corresponding state estimation. The joint estimator is applied to determining whether there exists a stealthy attacker that leads the plant to a forbidden state. In [10], a game-like automaton called insertion-deletion attack structure is proposed, from which the existence of stealthy attacker can be verified, and different stealthy attack strategies can also be extracted. The complexity of the methods in [10], [9] is exponential with respect to the number of states of the system. Lin and Su [11] consider both actuator and sensor attacker synthesis. They show that the attacker synthesis problem can be reduced to standard supervisor synthesis problem using a method with exponential complexity.

In this paper, we consider only sensor attackers that can launch SE-attacks and SI-attacks but no actuator attacks. In particular, the sets of events that can be inserted and erased by the attacker may be different while in [9], [10], [11] they are assumed to be the same. Therefore, the sensor attack considered in this paper is more general. We assume that the supervisor does not know the existence of the attacker at the beginning. However, if the supervisor receives an observable event that is not enabled at its current state, the attack is detected. Under this new attack detection mechanism, there is no need to compute the set of observations generated by the closed-loop system, which requires exponential complexity. If the attacker can cause the closed-loop system reaching a forbidden state, we call the attacker *threatening*. We aim at verifying the existence of threatening sensor attacker for a given closed-loop system. The proposed approach does not require the construction of the observer of the plant nor of the closed-loop system, whose complexity is exponential in the cardinality of the system state space. A deterministic model of the plant under attack is constructed and composed with the model of the supervisor under attack to obtain a structure called *All Sensor Attack* (ASA) model, which captures the behavior of the controlled system under all possible sensor attacks. The ASA model provides a necessary and sufficient condition for the existence of a threatening sensor attacker, and the condition can be verified with polynomial complexity.

II. PRELIMINARIES

In this section, some basics on automata and supervisory control are recalled.

A. Finite State Automata

In this paper, a plant is modeled as a deterministic finite automaton (DFA) $G = (X, E, f, x_0)$, where X is the finite set of *states*, E is the set of *events*, $f : X \times E \to X$ is the (partial) *transition function*, and $x_0 \in X$ is the *initial state*. The transition function can be extended to $f : X \times E^* \to X$ recursively: for all $x \in X$, $f(x, \varepsilon) = x$ and $f(x, \sigma e) = f(f(x, \sigma), e)$ for $\sigma \in E^*$ and $e \in E$. We denote by $f(x, \sigma)!$ the fact that σ is defined at x. The *generated language* of G is defined as $L(G) := \{\sigma \in E^* | f(x_0, \sigma)!\}$. The set of *active events* of G at state x is defined as $\Gamma_G(x) := \{e \in E | f(x, e)!\}$. A string $s \in E^*$ is said to be a *prefix* of $\sigma \in E^*$ if there exists $t \in E^*$ such that $st = \sigma$. The set of all the prefix of s is denoted as $\overline{s}$.

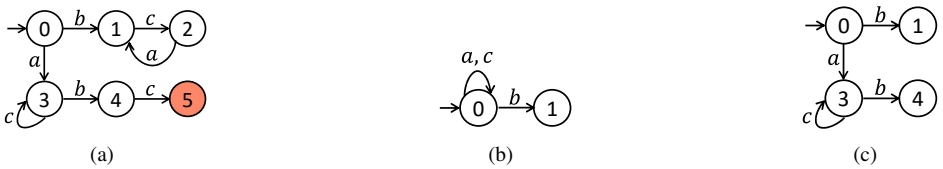


Fig. 1. (a) Plant $\boldsymbol{G}$, where $\boldsymbol{E_o = \{b, c\}}$, $\boldsymbol{E_c = \{a, c\}}$ and $\boldsymbol{X_f = \{5\}}$, and (b) supervisor $\boldsymbol{S}$ enforcing the specification, (c) the controlled system $\boldsymbol{S/G}$. Events subject to sensor attacks are $\boldsymbol{E_{si} = \{c\}}$ and $\boldsymbol{E_{se} = \{b\}}$.

Given two DFA $G_1 = (X_1, E_1, f_1, x_{01})$ and $G_2 = (X_2, E_2, f_2, x_{02})$, the *parallel composition* of G_1 and G_2 is $G_1||G_2 = Ac(X, E, f, (x_{01}, x_{02}))$, where $X \subseteq X_1 \times X_2$, $E = E_1 \cup E_2$, the transition function f is defined as, for $x = (x_1, x_2) \in X$, $e \in E$,

$$f(x,e) = \begin{cases} (x_1', x_2'), & \text{if } e \in \Gamma_{G1}(x_1) \cap \Gamma_{G2}(x_2), \\ (x_1', x_2), & \text{if } e \in \Gamma_{G1}(x_1) \setminus E_2, \\ (x_1, x_2'), & \text{if } e \in \Gamma_{G2}(x_2) \setminus E_1, \\ \text{Not defined}, & \text{Otherwise}, \end{cases}$$

with $x_1' = f_1(x_1, e)$ and $x_2' = f_2(x_2, e)$, and $Ac(\cdot)$ denotes the operation of taking the accessible part of an automaton.

B. Supervisory Control

A supervisor is a control agent that ensures that the plant under its control satisfies a given specification. In this paper, we assume that the specification is given as a set of *legal states*, denoted as $X_l \subseteq X$, such that the controlled system should not reach any state in $X_f = X \setminus X_l$, which is called the set of *forbidden states*. We define $X_w := \{x \in X | \exists \sigma \in E_{uc}^*, f(x, \sigma) \in X_f\}$ be the set of *weakly forbidden states*.

Due to limited sensing capacity and controllability of the supervisor, the set of events is partitioned into two subsets $E = E_o \cup E_{uo}$, where E_o is the set of *observable* events while $E_{uo} = E \setminus E_o$ is the set of *unobservable* events. From the perspective of controllability, the set of events E can be partitioned into $E = E_c \cup E_{uc}$, where E_c is the set of *controllable* events while $E_{uc} = E \setminus E_c$ is the set of *uncontrollable* events. The *natural projection* $P_o : E^* \to E_o^*$ is defined as i) $P_o(\varepsilon) := \varepsilon$; ii) for all $s\sigma \in E^*$ and $e \in E$, $P_o(\sigma e) := P_o(\sigma)e$ if $e \in E_o$, and $P_o(\sigma e) := P_o(\sigma)$, otherwise. We define $w = P_o(\sigma)$ the *observation* of σ. The set of all the observations generated by G is $P_o(L(G)) := \bigcup_{\sigma \in L(G)} P_o(\sigma)$. Note that no assumption on the relation between E_o and E_c is made in this paper.

Formally, a supervisor with partial observation is a function $\varphi : P_o(L(G)) \to 2^E$. In words, when observing a word $w \in P_o(L(G))$, the supervisor generates a corresponding control decision $\varphi(w) \subseteq E$ with $E_{uc} \subseteq \varphi(w)$ such that only events in φ are allowed to occur next. In this way, the closed-loop system is prevented from reaching any state in X_f.

A supervisor can be described by a DFA $S = (Y, E, f_S, y_0)$ (cf. Section 3.7.2 in [12]) such that control decisions are encoded in S. Suppose that a string $\sigma \in L(G)$ occurs in the plant. The supervisor observes $w \in P_o(\sigma)$ and generates the corresponding control decision $\varphi(w) = \Gamma_S(f_S(y_0, w))$, i.e., the set of events allowed to occur after σ. Note that $\forall y \in Y$ and $e \in E_{uo}$, if e is enabled by the supervisor (i.e., $e \in \Gamma_S(y)$), then $f_S(y, e) := y$.

Given a plant and a supervisor enforcing X_l, the controlled system is denoted as S/G, and can be obtained by the parallel composition of G and S.

Example 1: Consider the plant G in Fig. 1(a), where $E_c = \{a, c\}$, $E_o = \{b, c\}$, and $X_f = \{5\}$. A supervisor S that prevents G from reaching state 5 is shown in Fig. 1(b), which encodes the control function $\varphi(c^*) = \Gamma_S(0) = \{a, b, c\}$ and $\varphi(c^*b) = \Gamma_S(1) = \emptyset$. The controlled system S/G is shown in Fig. 1(c). $\diamond$

III. PROBLEM FORMALIZATION

In this paper, we focus on sensor attacks and our goal is to synthesize a sensor attacker that causes the controlled system violating the specification, and meanwhile remains stealthy.

A. Sensor Attacker

In an attack scenario, some sensors are deemed vulnerable and the attacker can change the sensor readings by either erasing a genuine sensor event and/or inserting a fictitious one [2]. Thus, we consider a sensor attacker that can launch two types of attacks:

- Sensor Erasure Attack (SE-attack): the attacker erases the last reading generated by the plant.
- Sensor Insertion Attack (SI-Attack): the attacker inserts false readings that have not occurred in the plant.

We denote $E_{se} \subseteq E_o$ (resp. $E_{si} \subseteq E_o$) the set of events subject to SE-attacks (resp. SI-attacks), i.e., the occurrence of event in E_{se} (resp. E_{si}) may be erased (resp. inserted) by the attacker. To make the problem more general, no relation is imposed

(a) (b)

Fig. 2. (a) A supervisory controlled system under sensor attacks, and (b) relations between the attack functions and projections.

between the sets E_{se} and E_{si}. After observing a new event, the sensor attacker can either insert a string of events in E_{si}, or erase the last event (if it is in E_{se}), or do nothing. Fig. 2(a) illustrates a controlled system under sensor attacks, where the red block denotes a sensor attacker. If the plant generates a word $\sigma \in E^*$, the attacker observes $P_o(\sigma)$ and then produces a corrupted observation that is denoted as $\hat{w} \in E_o^*$. The supervisor observes $\hat{w}$ and generates the corresponding control decision $\varphi(\hat{w})$.

We define $E_+ := \{e_+ | \forall e \in E_{si}\}$ the set of *inserted events*, and $E_- := \{e_- | \forall e \in E_{se}\}$ the set of *erased events*. The occurrence of event e_+ (resp. e_-) denotes the fact that the attacker inserts event $e \in E_{si}$ that has not occurred in the plant (resp. erases event $e \in E_{se}$ generated by the plant). We denote

$$E_a := E \cup E_+ \cup E_-$$

the *attack alphabet*. The attacker's action can be characterized in terms of a new word in E_a^* [9].

Definition 1: A sensor attacker is a function $\xi : P_o(L(G)) \to E_a^*$ satisfying the following constraints:

- $\xi(\varepsilon) \in E_+^*$,
- $\forall we \in P_o(L(G))$ with $e \in E_o$,

$$\begin{cases} \xi(we) \in \xi(w)\{e_-, e\}E_+^*, & \text{if } e \in E_{se}, \\ \xi(we) \in \xi(w)eE_+^*, & \text{otherwise.} \end{cases} \quad \diamond$$

In the definition, a sensor attacker may insert a string $w_+ \in E_+^*$ at the initial state before any observable event is generated by the system. If the latest observable event e belongs to E_{se}, the attacker may either erase it or not, and then insert w_+. Otherwise, the attacker may only insert w_+ after e. Given $w \in P_o(L(G))$, we call $w_a = \xi(w)$ the *attack word* corresponding to w, which describes the action of the attacker on the sensor reading generated by the plant. The set of all possible sensor attackers is denoted as Ξ.

We define the projection $\hat{P}_o : E_a^* \to E_o^*$ as $\forall w_a \in E_a^*, e' \in E_a$, i) $\hat{P}_o(\varepsilon) = \varepsilon$; ii) $\hat{P}_o(w_a e') = \hat{P}_o(w_a)e$, if $e' = e \in E_o$ or $e' = e_+ \in E_+$; iii) $\hat{P}_o(w_a e') = \hat{P}_o(w_a)$, if $e' \in E_{uo}$ or $e' = e_- \in E_-$. The projection describes the observation of the supervisor after a string is corrupted. Namely, given an attack word $w_a \in E_a^*$, the corresponding string observed by the supervisor is $\hat{w} = \hat{P}_o(w_a) \in E_o^*$, which is called *corrupted observation*. The internal structure of the attacker is visualized in Fig. 2(a).

It is assumed in this paper that the attacker has full knowledge of the plant G and the supervisor S, while the supervisor does not know the existence of the attacker at the beginning. Given $w \in P_o(L(G))$, if the corrupted observation satisfies $\hat{P}_o(\xi(w)) \in P_o(L(S))$, the attack on w is said to be *stealthy*. Namely, the attack is not detected by the supervisor. Given an attacker ξ, if $\forall w \in P_o(L(G))$, the attack on w is stealthy, we say the attacker ξ is stealthy.

Note that the attack detection mechanism in this paper is different from the one in [10], where the attacker is discovered when a word not in $P_o(L(S/G))$ is observed by the supervisor.

In general, $P_o(L(S/G)) \subseteq P_o(L(S))$. Therefore, the detection mechanism in this paper is less sensitive. However, it is more efficient since to determine if an observed word belongs to $P_o(L(S/G))$ the supervisor needs in general to construct the observer of the controlled system, which has exponential complexity. On the other hand, to check whether the observed word belongs to $P_o(L(S))$, it is sufficient for the supervisor to compare the new received observable event with the expected ones (i.e., checking if the received observable event is defined at the current state of the supervisor), which has linear complexity.

As soon as a word not in $P_o(L(S))$ is observed, the supervisor detects the presence of an attacker, and it disables all controllable events thereafter. The main reason why the supervisor chooses such a prevention mechanism is the following. If the current state of the plant belongs to X_w, such a control action guarantees that the specification will not be violated, while in the opposite case no suitable control action exists.

An example is provided to illustrate how sensor attacker affects the behavior of the controlled system S/G.

Example 2: Consider again the controlled system in Fig. 1. Assume that the sets of events subject to SE-attacks and SI-attacks are $E_{si} = \{c\}$ and $E_{se} = \{b\}$, respectively. Suppose that the sensor attacker is

$$\xi(w) = \begin{cases} \varepsilon, & \text{for } w = \varepsilon; \\ c^{n_1} b_- c^{n_2}, & \text{for } w = c^{n_1} b c^{n_2}, \text{ with } n_1, n_2 \in \mathbb{N}. \end{cases}$$

In words, the attacker erases event b after observing it. Suppose that $\sigma = acb$ occurs in $L(S/G)$. The plant reaches state 4. The attacker observes $P_o(acb) = cb$ and erases b (i.e., $\hat{P}_o(\xi(cb)) = c$). As a result, the supervisor observes c and reaches state 0, where event c is enabled. Therefore, the plant reaches the forbidden state 5 from state 4 with the occurrence of c. Since $cc \in P_o(L(S))$, the attack on $acbc$ is stealthy. Analogously, we can construct the controlled system under the attack of ξ, which is identical to G. ◇

Due to limited space, we refer the readers to [10] for the formalization of the intricate interaction between plant, supervisor and attacker. The controlled system under the action of the sensor attacker ξ is denoted as S_ξ/G.

B. Problem Statement

Example 2 shows that since the control decision is made by the supervisor based on observation of the corrupted observation, it may happen that under attack the controlled system no longer satisfies the specification.

Definition 2: Given a plant $G = (X, E, f, x_0)$, a supervisor S enforcing a state specification $X_l \subseteq X$, a sensor attacker ξ is said to be *threatening* for S/G, if there exist $\sigma \in L(S_\xi/G)$ and $\sigma_{uc} \in E_{uc}^*$ such that

1) $f(x_0, \sigma\sigma_{uc}) \in X_f$, and
2) $\forall \sigma' \in \overline{\sigma} \setminus \{\sigma\}$, $\hat{P}_o(\xi(w')) \in P_o(L(S))$, where $w' = P_o(\sigma')$. ◇

Condition 1) implies that when σ occurs the plant reaches a weakly forbidden state. Condition 2) implies that before the plant reaches the weakly forbidden state, the attacks remain stealthy. Once the plant reaches a weakly forbidden state, the attacker needs not be stealthy, i.e., $\hat{P}_o(\xi(P_o(\sigma)))$ may not belong to $P_o(L(S))$.

Finally, the problem addressed in this paper is formalized.

Attacker Existence Problem: given a plant G, a supervisor S enforcing specification X_l, and the sets E_{si} and E_{se} of events subject to sensor attacks, determine whether there exists a threatening sensor attacker for S/G.

IV. VERIFICATION OF ATTACKER EXISTENCE

A. Modeling of Plant and Supervisor under Attack

In this section, the effects of sensor attacks on the behavior of G and S are separately considered.

We first take the perspective of the plant. Given a plant G, a DFA $\hat{G} = (X, E_a, \hat{f}, x_0)$ is proposed to capture all possible sensor attacks on strings generated by G.

Definition 3: Given a plant $G = (X, E, f, x_0)$ with sets E_{si} and E_{se}, a DFA $\hat{G} = (X, E_a, \hat{f}, x_0)$ is 4-tuple, where the transition function $\hat{f}$ is defined as, $\forall x \in X$

$$\begin{cases} \hat{f}(x, e) := f(x, e), & \forall e \in E; \\ \hat{f}(x, e_+) := x, & \forall e \in E_{si}; \\ \hat{f}(x, e_-) := f(x, e), & \forall e \in E_{se} \cap \Gamma_G(x). \end{cases}$$

◇

DFA $\hat{G}$ has the same set of states with G. An event $e \in E$ corresponds to an event generated by the plant that is not corrupted by the attacker. Therefore, $\forall x \in X$, $e \in E$, $\hat{f}(x, e) = f(x, e)$. The attacker can insert any string in E_{si}^* between the occurrence of two events in G. The attacker can also erase the observation of an observable event $e \in E_{se}$ after it occurs in G, and the occurrence of event e and its erasure are represented by event e_-.

To characterize the properties of $\hat{G}$, one may extend functions $\xi : P_o(L(G)) \to E_a^*$ and $P_o : E^* \to E_o^*$ to new functions: $\xi' : L(G) \to E_a^*$ and $P_o' : E_a^* \to E_a^*$. A *string attack function* ξ' satisfies Definition 1 assuming $E_o = E$; we denote Ξ' the set of all these functions. Projection P_o' removes from a string in E_a^* all events in E_{uo}.

The relation among these functions is shown in Fig. 2(b), where the part in black corresponds to the scheme in Fig. 2(a) while the part in red is related to the newly defined functions. String $\sigma_a = \xi'(\sigma)$ is the *attack string*, which describes the action of the attacker on the word generated by the plant. Given a generated string σ it holds that $w_a = \xi(P_o(\sigma)) = P_o'(\xi'(\sigma))$. We have the following properties of $\hat{G}$.

Proposition 1: Let $\hat{G}$ be the DFA defined in Definition 3. It holds that

1) $\sigma_a \in L(\hat{G}) \Leftrightarrow (\exists \xi' \in \Xi', \exists \sigma \in L(G))\sigma_a = \xi'(\sigma)$;
2) for all $\xi' \in \Xi'$, $\sigma \in L(G)$: $\hat{f}(y_0, \xi'(\sigma)) = f(y_0, \sigma)$.

Proof: Statement 1). (proof of $\Rightarrow$) Consider a word $\sigma_a \in L(\hat{G})$ that reaches a state x and only contains events in E: it represents an original evolution that has not been attacked. At each state all events e_+ are in selfloop: this means that σ_a can be corrupted inserting at each step a string of fake events in E_{si}. In addition if the string σ_a is generated executing a transition $\hat{f}(x', e) = x''$ with $e \in E_{se}$, it is also possible to execute the "parallel" transition $\hat{f}(x', e_-) = x''$ corresponding to an attack that erases e. Therefore, there exist $\xi' \in \Xi', \sigma \in L(G)$ such that $\xi'(\sigma) = \sigma_a$.

(Proof of $\Leftarrow$) By the definition of $\hat{f}$, all possible attacks have been included in $\hat{G}$. Therefore, given a word $\sigma \in L(G)$ and $\xi' \in \Xi'$, there exists a corresponding $\sigma_a \in L(\hat{G})$ such that $\sigma_a = \xi'(\sigma)$.

Statement 2). Note that the original string and all the corrupted ones reach the same state x, and the result follows statement 1). ■

(a) (b)

Fig. 3. (a) Model $\hat{G}$ of the plant under attack, and (b) model $\hat{S}$ of the supervisor under attack.

According to statement 1), the language generated by $\hat{G}$ consists of all possible attack strings corresponding to evolutions of the plant. By statement 2), the state reached in the plant by an evolution σ and the state reached in $\hat{G}$ with an attack string $\sigma_a = \xi'(\sigma)$ coincide.

Now let us take the perspective of the supervisor. Given a supervisor $S = (Y, E, f_S, y_0)$, a DFA $\hat{S}$ is proposed to describe the control decisions generated by the supervisor under all possible sensor attacks.

Definition 4: Given a supervisor $S = (Y, E, f_S, y_0)$ with sets E_{si} and E_{se}, a DFA $\hat{S} = (Y \cup \{y_d\}, E_a, \hat{f}_S, y_0)$ is a 4-tuple, where the transition function $\hat{f}_S$ is defined as,

- for all $y \in Y$,

$$\begin{cases} \hat{f}_S(y,e) := f_S(y,e), \ \forall e \in E \cap \Gamma_S(y); & (1)\\ \hat{f}_S(y,e) := y, \ \forall e \in E_{uc} \cap E_{uo} \setminus \Gamma_S(y); & (2)\\ \hat{f}_S(y,e) := y_d, \ \forall e \in E_{uc} \cap E_o \setminus \Gamma_S(y); & (3)\\ \hat{f}_S(y,e_+) := f_S(y,e), \ \forall e \in E_{si} \cap \Gamma_S(y); & (4)\\ \hat{f}_S(y,e_+) := y_d, \ \forall e \in E_{si} \setminus \Gamma_S(y); & (5)\\ \hat{f}_S(y,e_-) := y, \ \forall e \in E_{se} \cap (\Gamma_S(y) \cup E_{uc}). & (6) \end{cases}$$

- for $y = y_d$,

$$\begin{cases} \hat{f}_S(y_d,e) := y_d, & \forall e \in E_{uc}; & (7)\\ \hat{f}_S(y_d,e_-) := y_d, & \forall e \in E_{uc} \cap E_{se}; & (8)\\ \hat{f}_S(y_d,e_+) := y_d, & \forall e \in E_{si}. & (9) \end{cases}$$

◇

For the uncorrupted observations, $\hat{S}$ and S have the same control decisions (Eq. (1)). The new state y_d is added in $\hat{S}$, and it is reachable when a corrupted observation $\hat{w} \notin P_o(L(S))$ is observed by the supervisor (Eqs. (3) and (5)). If the corrupted observation belongs to $P_o(L(S))$, the supervisor cannot detect the attack and update the control decision (Eq. (4)). If the attacker erases the observation of an event $e \in E_{se}$ that is enabled by the supervisor, the supervisor cannot observe the occurrence of e and thus $\hat{f}_S(y, e_-) = y$ (Eq. (6)). After attacks are detected, only uncontrollable events are enabled (Eq. (7)). Then the attacker may erase their occurrence, if possible (Eq. (8)), or insert any other events in E_{si} (Eq. (9)).

Based on the construction of $\hat{S}$, we have the following properties of $\hat{S}$.

Proposition 2: Let ξ be a sensor attacker. Let $w \in P_o(L(G))$ be an observation of the attacker, and $\hat{w} = \hat{P}_o(\xi(w))$ the corrupted observation received by the supervisor S.

1) The control decision corresponding to $\hat{w}$ is $\varphi(\hat{w}) = \Gamma_{\hat{S}}(y) \cap E$, where $y = \hat{f}_S(y_0, \xi(w))$;
2) $\hat{f}_S(y_0, \xi(w)) = y_d$ iff $\hat{w} \notin P_o(L(S))$.

Proof: Follows from the definition of the transition function $\hat{f}_S$. From Eqs. (1) to (6), one can verify that state y_d is reachable from a state $y \in Y$ if and only if an observable event $e \notin \Gamma_S(y)$ occurs.

The definition of $\hat{f}_S$ guarantees that, if $\hat{w} \in P_o(L(S))$, $f_S(y_0, \hat{w}) = \hat{f}_S(y_0, \hat{w}) = y$. Thus, $\varphi(\hat{w}) = \Gamma_S(y) = \Gamma_{\hat{S}}(y) \cap E$. By statement 2), if $\hat{w} \notin P_o(L(S))$, $\hat{f}_S(y_0, \hat{w}) = y_d$, which corresponds to the control decision $\varphi(\hat{w}) = E_{uc} = \Gamma_{\hat{S}}(y_d) \cap E$. Therefore, statement 1) holds. ■

Statement 1) implies that the DFA $\hat{S}$ describes the control decisions for all corrupted observations under all possible sensor attacks. Statement 2) implies that if the attack word $\xi(w)$ leads to the state y_d, the attack on w is not stealthy.

Example 3: Consider again the plant and the supervisor in Figs. 1(a) and 1(b). Their corresponding $\hat{G}$ and $\hat{S}$ are shown in Fig. 3. To make the construction more illustrative, additional transitions of events in E_+, E_-, and E_{uc} are colored in red, blue, and green respectively. ◇

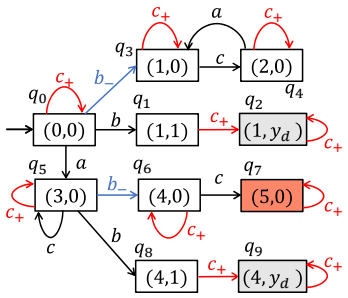
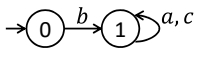
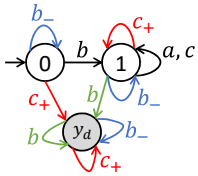


Fig. 4. The ASA model $\boldsymbol{H}$ of $\hat{\boldsymbol{G}}$ and $\hat{\boldsymbol{S}}$ in Fig. 3.



(a)



(b)

Fig. 5. (a) Another supervisor $\boldsymbol{S}$ enforcing the specification on $\boldsymbol{G}$ in Example 1, and (b) the corresponding model $\hat{\boldsymbol{S}}$ of the supervisor under attack.

B. All Sensor Attack Automaton

Definition 5: Given $G = (X, E, f, x_0)$, $S = (Y, E, f_S, y_0)$, and sets E_{si} and E_{se}, let $\hat{G}$ and $\hat{S}$ be their corresponding automata constructed by Definitions 3 and 4. The *All Sensor Attack* (ASA) model (for G, S, E_{si} and E_{se}) is a DFA $H = (Q, E_a, f_H, q_0) := \hat{G}||\hat{S}$. ◇

Each state of H is a pair $(x, y) \in X \times Y$, where x and y are respectively the state reached in the plant and the state of the supervisor reached through the observation of a corrupt or an original string.

Proposition 3: Let $H = (Q, E_a, f_H, q_0)$ be the ASA of G and S. It holds that:

$$\sigma_a \in L(H) \Leftrightarrow (\exists \xi' \in \Xi', \exists \sigma \in L(S_{\xi'}/G))\sigma_a = \xi'(\sigma),$$

where $S_{\xi'}/G$ is the controlled system under the attack of ξ' (cf. Section III-A).

Proof: By Proposition 1, $\hat{G}$ includes all event insertion and erasure attacks on string $\sigma \in L(G)$. By Proposition 2, $\hat{S}$ describes the control decisions corresponding to all possible corrupted observations. Thus, $H = \hat{G}||\hat{S}$ describes the behavior of the controlled system under all possible sensor attacks. ■

After an ASA model is constructed, two sets of states in H are defined:

- $Q_d := \{(x, y) \in Q | y = y_d\}$, the set of states whose second element equals y_d, and
- $Q_f := \{(x, y) \in Q | x \in X_f\}$, the set of states whose first element is a state in X_f.

Example 4: [continuation of Example 3]

The ASA model $H = \hat{G}||\hat{S}$ is shown in Fig. 4. We have $Q_d = \{q_2, q_9\}$ (in grey), and $Q_f = \{q_7\}$ (in red). ◇

Theorem 4: Given $G = (X, E, f, x_0)$, $S = (Y, E, f_S, y_0)$, and sets E_{si} and E_{se}, let $H = (Q, E_a, f_H, q_0)$ be the ASA model. There exists a threatening sensor attacker for S/G iff $Q_f \neq \emptyset$.

Proof: (If) Let $q = (x, y) \in Q_f$ with $f_H(q_0, \sigma_a) = q$. By Proposition 3, there exists $\xi' \in \Xi'$ and $\sigma \in L(G)$ such that $\sigma_a = \xi'(\sigma)$. By statement 2) in Proposition 1, $f(x_0, \sigma) = x \in X_f$. If $y = y_d$, there exists $\sigma'_a \in \overline{\sigma_a}$ such that $f_H(y_0, \sigma'_a) = (x', y_d) \in Q_d$ and $\forall \sigma''_a \in \overline{\sigma'_a} \setminus \{\sigma'_a\}$, $f_H(y_0, \sigma''_a) \notin Q_d$. By statement 2) of Proposition 2, the attack is detected by the supervisor with observing $\hat{P}_o(\sigma'_a)$. Since after the attacker is detected, only uncontrollable events can occur in the controlled system, there exists $\sigma_{uc} \in E^*_{uc}$ such that $f(x', \sigma_{uc}) = x \in X_f$. It implies that although the supervisor detects the attack, it cannot prevent the plant from reaching a state in X_f. By Definition 1, the attacker is threatening. On the other hand, if $y \neq y_d$, the attack on $P_o(\sigma)$ is stealthy. Therefore, the attacker is threatening.

Now we prove that such an attacker exists. By Definition 4, for any $\sigma' \in L(G)$ with $P_o(\sigma') = P_o(\sigma) = w$, the possible attack actions are the same (i.e., $\Gamma_{\hat{S}}(\hat{f}_S(y_0, w))$. For any other σ', the attacker can take the same attack actions as σ. For other strings that have different observation from $P_o(\sigma)$, the attacker just takes no action. Namely, from ξ' we can obtain one ξ, and thus the attacker exists.

(Only if) Assume that there exists a threatening sensor attacker ξ. The controlled system under its attack is S_ξ/G. By Definition 2, there exists $\sigma \in L(S_\xi/G)$ with $f(x_0, \sigma) = x$ such that $\exists \sigma_{uc} \in E^*_{uc}$, $f(x, \sigma_{uc}) \in X_f$, and the attack of ξ on $\overline{w} \setminus \{w\}$ is stealthy, where $w = P_o(\sigma)$. By Proposition 3, H contains all the possible sensor attacks. Therefore, there exists $\sigma_a \in L(H)$ such that $\xi'(\sigma) = \sigma_a$. Let $f_H(q_0, \sigma_a) = q$. By the construction of H and statement 2) in Proposition 1, $q = (x, \cdot) \in Q_f$, where the dot means that the second element is not specified. Therefore, $Q_f \neq \emptyset$. ■

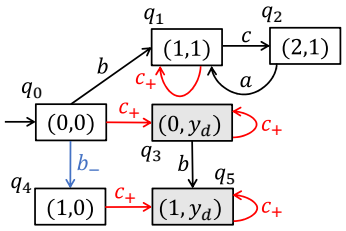


Fig. 6. The ASA model $\boldsymbol{H}$ of $\hat{\boldsymbol{G}}$ in Fig. 3(a) and $\hat{\boldsymbol{S}}$ in Fig. 5(b).

Example 5: [continuation of Example 4]

Since $Q_f = \{q_7\}$, by Theorem 4, for G and S in Fig. 1, $E_{si} = \{c\}$ and $E_{se} = \{b\}$, there exists a threatening sensor attacker. Indeed, the string ab^-c, for instance, leading to q_7 implies that for $\sigma = ab$ (i.e., after observing b), the attack erasing b misleads the supervisor to enabling c and causes the plant reaching 5. The string captures one attack action of the attacker ξ in Example 2.

The threatening attacker does not always exist. Suppose that the supervisor is replaced with the one in Fig. 5(a). To check whether there exists a stealthy threatening attacker, first we construct its corresponding $\hat{S}$ (as shown Fig. 5(b)), and then the ASA model H in Fig. 6. We have $Q_d = \{q_3, q_5\}$ and $Q_f = \emptyset$. By Theorem 4, there is no threatening sensor attacker with capability of $E_{si} = \{c\}$ and $E_{se} = \{b\}$ that can cause damage to the controlled system S/G. $\diamond$

C. Complexity Analysis

Finally, we analyze the complexity of the proposed approach. According to Definition 3, the worst-case time complexity of constructing $\hat{G}$ is $\mathcal{O}(|X||E|)$. According to Definition 4, the worst-case time complexity of constructing $\hat{S}$ is $\mathcal{O}(|Y||E|)$. Model H contains at most $|X|(|Y|+1)$ states. Thus, the worst-case time complexity of constructing $H = \hat{G}||\hat{S}$ is $\mathcal{O}(|E||X||Y|)$, and the complexity of verifying the conditions in Theorem 4 is polynomial with respect to the size of the plant, the supervisor and the set of events.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the stealthy sensor attacker existence problem in cyber-physical discrete event systems. It is assumed that every time when receiving an observation, the attacker can either erase the last event (if it belongs to set E_{se}) or not and then insert or not a string of events in set E_{si}. A new attack detection mechanism was proposed. The notion of *threatening attacker* was defined. Construction of the plant and the supervisor under attack are presented. Then we proposed a structure, called *All Sensor Attack* (ASA) automaton, to capture all the behavior of the controlled system under all possible sensor attacks. Based on the ASA, a necessary and sufficient condition for the existence of a threatening attacker was derived. It is shown that the attacker existence problem can be solved in polynomial complexity. Our future work will focus on the synthesis of a threatening sensor attacker using the ASA model.

REFERENCES

[1] J. C. Basilio, C. N. Hadjicostis, and R. Su, "Analysis and control for resilience of discrete event systems: Fault diagnosis, opacity and cyber security," *Foundations and Trends in Systems and Control*, vol. 8, no. 4, pp. 285–443, 2021.

[2] L. K. Carvalho, Y. C. Wu, K. Raymond, and S. Lafortune, "Detection and mitigation of classes of attacks in supervisory control systems," *Automatica*, vol. 97, pp. 121–133, 2018.

[3] C. Gao, C. Seatzu, Z. Li, and A. Giua, "Multiple attacks detection on discrete event systems," in *2019 IEEE International Conference on Systems, Man and Cybernetics*, (Bari, Italy), pp. 2352–2357, 2019.

[4] X. Yin, M. Zamani, and S. Liu, "On approximate opacity of cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1630–1645, 2021.

[5] S. Rong, "Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations," *Automatica*, vol. 94, pp. 35–44, 2018.

[6] M. Wakaiki, P. Tabuada, and J. Hespanha, "Supervisory control of discrete-event systems under attacks," *Dynamic Games and Applications*, vol. 9, pp. 965–983, 2019.

[7] Y. Wang and M. Pajic, "Supervisory control of discrete event systems in the presence of sensor and actuator attacks," in *2019 IEEE 58th Conference on Decision and Control*, (Nice, France), pp. 5350–5355, 2019.

[8] R. Meira-Goes, S. Lafortune, and H. Marchand, "Synthesis of supervisors robust against sensor deception attacks," *IEEE Transactions on Automatic Control*, vol. 66, no. 10, pp. 4990–4997, 2021.

[9] Q. Zhang, C. Seatzu, Z. Li, and A. Giua, "Joint state estimation under attack of discrete event systems," *IEEE Access*, vol. 9, pp. 168068–168079, 2021.

[10] R. Meira-Goes, E. Kang, R. H. Kwong, and S. Lafortune, "Synthesis of sensor deception attacks at the supervisory layer of cyber-physical systems," *Automatica*, vol. 121, p. 109172, 2020.

[11] L. Lin and R. Su, "Synthesis of covert actuator and sensor attackers," *Automatica*, vol. 130, p. 109714, 2021.

[12] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer, Cham, 3 ed., 2021.