Fault Diagnosis of Discrete Event Systems Under Attack

Tenglong Kang, Carla Seatzu, Zhiwu Li, and Alessandro Giua

July 2023

Abstract

In this paper, we study the problem of fault diagnosis under cyber attacks in the context of partially-observed discrete event systems. An operator monitors the evolution of a system through the received observations and computes its current diagnosis state. The observation is corrupted by an attacker which has the ability to edit a subset of sensor readings by inserting or erasing some events. In this sense, the attacker may induce the operator to draw incorrect diagnostic conclusions based on the corrupted observation regarding the fault occurrence. In particular, the attack is harmful if a fault can be detected by the operator when looking at an uncorrupted observation, while it is not detected when looking at the corresponding corrupted observation. In addition, the attacker must remain stealthy, i.e., its presence should not be discovered by the operator. To this end, we propose a special structure, called a stealthy joint diagnoser, which describes the set of all possible stealthy attacks. We show how to use the stealthy joint diagnoser to perform fault diagnosis under attack. Finally, such a structure also allows one to establish if a stealthy harmful attack may be implemented.

Published as:

Tenglong Kang, Carla Seatzu, Zhiwu Li, and Alessandro Giua. "Fault Diagnosis of Discrete Event Systems Under Attack," in *Proceedings of 62th IEEE Conference on Decision and Control*, December 12-15, 2023, Singapore. DOI: 10.1109/CDC49753.2023.10383814

Tenglong Kang is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with the Department of Electrical and Electronic Engineering, University of Cagliari, 09124 Cagliari, Italy. tlkang@stu.xidian.edu.cn.

Zhiwu Li (corresponding Author) is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with the Institute of Systems Engineering, Macau University of Science and Technology, Macau, China. zhwli@xidian.edu.cn.

Carla Seatzu and Alessandro Giua are with the Department of Electrical and Electronic Engineering, University of Cagliari, 09124 Cagliari, Italy. {carla.seatzu, giua}@unica.it

1 Introduction

Cyber-physical systems (CPS), such as intelligent transportation systems, process control systems and advanced communication networks, are characterized by the interaction of computational and physical components [1]. In this work, we use the formalism of discrete event systems to model the behavior of CPS. Discrete event systems are dynamic systems equipped with a discrete state space and an event-driven transition structure [2]. The undesired behavior of CPS arises from two possible phenomena, i.e., *component faults* and *cyber attacks*.

A component fault is an endogenous phenomenon that causes a deviation in the behavior of a system such that its performance or throughput is degraded. For this reason, fault diagnosis is a crucial task for an operator monitoring a CPS. In the context of discrete event systems, the problem of fault diagnosis is originally formalized by [3], where the fault is an unobservable event whose occurrence is usually necessary to be detected based on the observation. In the past years, robust fault diagnosis of discrete event systems has been a topic of great interest due to its importance; see, e.g., [4] and [5]. These works focus on the issue of the robust diagnosis subject to sensor failures, i.e., permanent and intermittent loss of communication between sensors and the diagnoser, but they do not consider the impact of an attacker on the physical parts of the system.

A cyber attack on the other hand is an exogenous phenomenon that causes damage to the cyber-security of CPS [6]. In the context of discrete event systems, cyber-security has become a topical subject of increasing attention in the last few years. Several aspects of cyber-security have been explored in the literature of discrete event systems. The problem of attack detection is focused on modeling the attacker as a fault behavior [7] and [8]. Some discrete frameworks have been developed to handle adversarial attacks for resilient supervisory control [9, 10, 11] and intelligent attack synthesis [12, 13, 14].

Rather than passively detecting the existence of attacks or deriving a supervisory control law that is robust to attacks, relatively few works focus on synthesizing a successful attack strategy satisfying the desired objective. Meira-Góes et al. [13] address the problem of synthesizing stealthy attacks that can induce the plant into a forbidden state without being detected by an existing supervisor. A bipartite discrete transition structure, called an Insertion-Deletion Attack structure (IDA) is proposed to capture a game-like relationship between the supervisor and the environment (the plant and the attacker). Based on the IDA, three different types of successful stealthy attack strategies can be derived. Zhang et al. [14] investigate the problem of state estimation in the presence of an attack. A novel discrete transition structure, called a joint estimator, is proposed to capture all possible attacks and the corresponding state estimation. The joint estimator is applied to show how the possible suitable choices of the attacker may affect the state estimate of the operator. In [13] and [14], they study the effect of an attacker on the state estimation of the supervisor and the operator, respectively, without considering the impact on the fault diagnosis

process.

The study of component faults and cyber attacks in isolation has been extensively researched in various fields. However, there is a crucial need to consider both phenomena together since they are not always independent of each other and can even interact to amplify their effects. In this context, we study the problem of *fault diagnosis under attack* in partially-observed discrete event systems. We utilize a similar attack model that has been used in [13] and [14]. We assume that an attacker with full knowledge of the plant may corrupt sensor readings available to an operator, by inserting certain fake events that do not occur or erasing some observations that have occurred. The goal of the attacker is to prevent the operator from making the correct diagnostic conclusions regarding the fault occurrence. The methodology developed to derive a suitable attack policy from the attacker's viewpoint is inspired by the work in [13] and [14]. As in these works, we propose a discrete structure to describe how an attacker may affect the operator's ability to perform the correct fault diagnosis. We call this structure a *stealthy joint diagnoser*. By construction, a stealthy joint diagnoser embeds all and only stealthy attacks. Once constructed, a stealthy joint diagnoser serves as the basis for solving the fault diagnosis problem under attack. The states of the stealthy joint diagnoser can be classified according to the corresponding diagnosis pairs. In particular, we define as *harmful states* of the stealthy joint diagnoser those states corresponding to the *harmful attacks* that can hide the occurrence of a fault during the system evolution. Finally, we notice that such a structure may also be used by the operator to establish if the diagnoser is robust to certain possible attacks.

2 Preliminaries

Let E be an *alphabet*. The set of all words over E is denoted by E^*. Given a word $\sigma \in E^*$, (i) the *length* of σ is denoted by $|\sigma|$; (ii) the number of occurrence of event $e \in E$ in σ is denoted by $|\sigma|_e$; (iii) the *support* of σ, denoted by $\|\sigma\| = \{e \in E | |\sigma|_e > 0\}$, is the set of events that appear at least once in the word.

Let $G = (X, E, \delta, x_0)$ denote a *deterministic finite state automaton*, where X is the finite set of states, E is the finite set of events, $\delta : X \times E \to X$ is the partial transition function, and x_0 is the initial state. The transition function can be extended to the domain $X \times E^*$, denoted by $\delta^* : X \times E^* \to X$, such that $\delta^*(x, \varepsilon) = x$, where ε denotes the empty word, and $\delta^*(x, \sigma e) = \delta(\delta^*(x, \sigma), e)$. The *generated language* of G is defined as $L(G) = \{\sigma \in E^* \mid \delta^*(x_0, \sigma) \text{ is defined}\}$.

Assume that E is partitioned into $E = E_o \dot{\cup} E_{uo}$, where E_o and E_{uo} denote, respectively, the sets of *observable* and *unobservable events*. Based on the partition, the *natural projection* function $P : E^* \to E_o^*$ is defined as [2]:

$$P(\varepsilon) = \varepsilon \text{ and } P(\sigma e) = \begin{cases} P(\sigma)e & \text{if} \quad e \in E_o, \\ P(\sigma) & \text{if} \quad e \in E_{uo}. \end{cases} \tag{1}$$

The *inverse projection* $P^{-1} : E_o^* \to 2^{E^*}$ is defined as $P^{-1}(s) = \{\sigma \in E^* : P(\sigma) = s\}$.

The evolution of the plant G is observed by an operator. Assume that a plant G produces a word $\sigma \in E^*$. Due to the natural projection, the operator observes an observation $s = P(\sigma) \in E_o^*$. When no attack occurs, s is called an *uncorrupted observation*. We also define $\mathcal{S}(s) = P^{-1}(s) \cap L(G)$ the set of words consistent with observation s, i.e., the set of words in the language of G that produce the observation s.

The *unobservable reach* of state x is defined by a set of states $x' \in X$ reached from state $x \in X$ by executing an unobservable word $\sigma \in E_{uo}^*$, namely, $UR(x) = \{x' \in X \mid (\exists \sigma \in E_{uo}^*)\, \delta^*(x, \sigma) = x'\}$.

Given a plant $G = (X, E, \delta, x_0)$ with the set of observable events E_o, the observer of G is $Obs(G) = (B, E_o, \delta_{obs}, b_0)$, where $B \subseteq 2^X$ is the set of states, E_o is the set of observable events of G, $\delta_{obs} : B \times E_o \to B$ is the transition function defined as $\delta_{obs}(b, e_o) := \bigcup_{x \in b} UR(\{x' \mid \delta(x, e_o) = x'\})$, and the initial state is $b_0 := UR(x_0)$ (see [2] for details).

Given two automata $G_1 = (X_1, E_1, \delta_1, x_{01})$ and $G_2 = (X_2, E_2, \delta_2, x_{02})$, their *parallel composition* is denoted as $G = G_1 \parallel G_2 = (X_1 \times X_2, E_1 \cup E_2, \delta, (x_{01} \times x_{02}))$, where the transition function δ is defined as follows:

$$\begin{cases} \delta[(x_1, x_2), e] = (x_1', x_2') \text{ if } \delta_1(x_1, e) = x_1' \wedge \delta_2(x_2, e) = x_2', \\ \delta[(x_1, x_2), e] = (x_1', x_2) \text{ if } \delta_1(x_1, e) = x_1' \wedge e \notin E_2, \\ \delta[(x_1, x_2), e] = (x_1, x_2') \text{ if } \delta_2(x_2, e) = x_2' \wedge e \notin E_1, \\ \text{undefined} \qquad\qquad \text{otherwise.} \end{cases} \tag{2}$$

We notice that in the parallel composition, if there exist unreachable states from the initial state, then such states should be removed and only reachable states should be considered.

Let $E_f \subseteq E_{uo}$ denote the set of fault events. For the sake of simplicity, we do not distinguish among different fault types. In the remainder of this paper, we will consider a single fault class. The basic fault diagnosis problem is to determine at run-time, based on the observation $s \in E_o^*$, if a fault has occurred or not in the past. Solving a diagnosis problem requires constructing a diagnosis function.

Definition 1: Given a plant G with respect to set of fault events $E_f \subseteq E_{uo}$, a *diagnosis function*

$$\gamma : E_o^* \to \{N, F, U\}$$

associates any observation $s \in E_o^*$ to a diagnosis state $\gamma(s) \in \{N, F, U\}$ as follows:

1) $\gamma(s) = N$, if for all $\sigma \in \mathcal{S}(s)$, it holds $\|\sigma\| \cap E_f = \emptyset$;

2) $\gamma(s) = F$, if for all $\sigma \in \mathcal{S}(s)$, it holds $\|\sigma\| \cap E_f \neq \emptyset$;

3) $\gamma(s) = U$, if there exist $\sigma, \sigma' \in \mathcal{S}(s)$ such that $\|\sigma\| \cap E_f = \emptyset$ and $\|\sigma'\| \cap E_f \neq \emptyset$. $\diamond$

A more efficient way of computing a diagnosis function is by means of diagnosers [3]. Diagnosers are deterministic automata whose alphabet is the set of observable events of G, and their states have labels F and N attached to the states of G. Formally, the diagnoser $Diag(G) = (X_d, E_o, \delta_d, x_{d,0})$ is defined as

$$Diag(G) = Obs(Rec(G)) = Obs(G \parallel A_\ell) \tag{3}$$

where $Rec(G)$ is the *fault recognizer* obtained by the parallel composition of G and A_ℓ, in which A_ℓ is the *fault monitor* on alphabet E_f shown in Fig. 1.

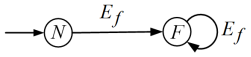
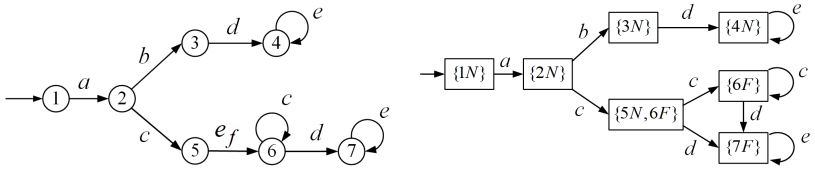


Figure 1: Fault monitor A_ℓ on alphabet E_f.

To each state x_d of $Diag(G)$ we associate a diagnosis value $\gamma(x_d)$ in the following manner: (i) $\gamma(x_d) = F$ (certain state), if $\ell = F$ for all $(x, \ell) \in x_d$, (ii) $\gamma(x_d) = N$ (normal state), if $\ell = N$ for all $(x, \ell) \in x_d$, and (iii) $\gamma(x_d) = U$ (uncertain state), if there exist $(x, \ell), (\tilde{x}, \tilde{\ell}) \in x_d, x$ not necessarily distinct from $\tilde{x}$, such that $\ell = F$ and $\tilde{\ell} = N$. Thus a diagnoser allows one to associate each observation $s \in E_o^*$ to a diagnosis state $\gamma(s) = \gamma(x_d)$, where $x_d = \delta_d^*(x_{d,0}, s)$ is the state reached in $Diag(G)$ by executing s. In this context, fault diagnosis can be performed online by examining the diagnoser states. Figs. 2(a) and 2(b) show, respectively, a plant G, for which $E = \{a, b, c, d, e, e_f\}$, $E_o = \{a, b, c, d, e\}$ and $E_f = \{e_f\}$, and the corresponding diagnoser $Diag(G)$.



(a) G (b) $Diag(G)$

Figure 2: (a) A plant G and (b) its diagnoser $Diag(G)$.

3 Attack Model

First, we define the fault diagnosis system model under attack, as depicted in Fig. 3. A plant G produces a word σ and the observation is $s = P(\sigma)$. The attacker intervenes in the communication channels between the system's sensor and the operator. In particular, the attacker may corrupt the observation s by inserting some events that do not occur or erasing some events that have occurred. Such a *corrupted observation* is denoted as s'. The operator is used

to recognize the corrupted observation s' and compute its diagnosis state $\gamma(s')$, where γ is the diagnosis function.

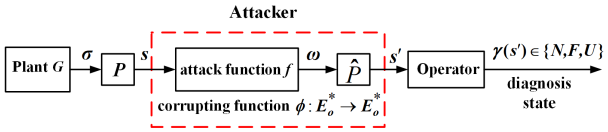


Figure 3: A fault diagnosis system model under attack.

Then, let us recall some preliminary notations from [14]. The subset of events that can be corrupted by the attacker is defined as the *compromised event set* and denoted by E_{com}. Without loss of generality, we assume that $E_{com} \subseteq E_o$. More specifically, the sets of events that can be inserted and erased are denoted as E_{ins} and E_{era}, respectively. Note that E_{ins} and E_{era} are not necessarily disjoint. Correspondingly, we define two new sets of events to describe the permitted actions of the attacker. The sets $E_+ = \{e_+ \mid e \in E_{ins}\}$ and $E_- = \{e_- \mid e \in E_{era}\}$ are defined as the sets of inserted and erased events, respectively. We also define $E_a = E_o \cup E_+ \cup E_-$ as the *attack alphabet*. Note that $E_o, E_+,$ and E_- are disjoint.

Definition 2: Given a plant G with the set of compromised events $E_{com} = E_{ins} \cup E_{era}$, an *attack function* is $f : P(L(G)) \to E_a^*$ satisfying the following constraints:
a) $f(\varepsilon) \in E_+^*$,
b) $\forall se \in P(L(G))$ with $s \in E_o^*$:

$$\begin{cases} f(se) \in f(s)\{e\}E_+^* & \textbf{if } e \in E_o \backslash E_{era}, \\ f(se) \in f(s)\,\{e_-, e\}\,E_+^* & \textbf{if } e \in E_{era}. \end{cases} \tag{4}$$

$\diamond$

Condition a) in Definition 2 indicates that the attacker may insert an arbitrary word $t \in E_+^*$ at the initial state before any generated word of G is observed. Condition b) means that the attacker cannot erase e when e does not belong to E_{era}. However, the attacker may insert an arbitrary word $t \in E_+^*$ after e. If an event $e \in E_{era}$ occurs, the attacker may either erase e or leave it intact, and then insert any arbitrary word $t \in E_+^*$.

The existence of an attack function induces a new language, called an *attack language*. The attack language is defined as $L(f, G) = f(P(L(G)))$. A word $\omega \in L(f, G)$ is called an *attack word*. Then, we define the *operator mask* $\widehat{P} : E_a^* \to E_o^*$ that treats the attack word ω as follows: (i) $\widehat{P}(\epsilon) = \epsilon$; (ii) $\widehat{P}(\omega e_+) = \widehat{P}(\omega)e$, if $e_+ \in E_+$; (iii) $\widehat{P}(\omega e_-) = \widehat{P}(\omega)$, if $e_- \in E_-$; (iv) $\widehat{P}(\omega e) = \widehat{P}(\omega)e$, if $e \in E$. Given an attack word ω, an observation s is said to be consistent with the attack word ω if $\omega = f(s)$ or $s = \widehat{P}(\omega)$ holds. In addition, the *corrupting*

function $\phi : E_o^* \to E_o^*$ is defined as $\phi(s) = \widehat{P}(f(s))$ taking an uncorrupted observation s as input and producing a corrupted observation s' as output, as in Fig. 3. Similarly, the *corrupted language* induced by the corrupting function is defined as $L(\phi, G) = \phi(P(L(G)))$. Finally, we point out that in this paper, the corrupting function ϕ is used to model the capabilities of the attacker to temper with sensor readings.

4 Problem Setting: Diagnosis Under Attack

In this section, we first describe the *stealthiness* and *harmfulness* of a corrupting function and then formalize the problem statement. Stealthiness implies that the attack remains undetected by the operator. This can be formalized as follows.

Definition 3: Let G be a plant with the set of observable events E_o. A corrupting function ϕ is said to be *stealthy* if $L(\phi, G) \subseteq P(L(G))$. $\diamond$

The stealthiness is guaranteed provided that any corrupted observation is contained in the set of uncorrupted observations that the plant may generate when no attack occurs. In this sense, the operator does not realize that the system is under attack. We additionally define the notion of *harmful attacks*.

Definition 4: Let G be a plant with the set of observable events E_o. Let $\gamma : E_o^* \to \{N, F, U\}$ be a diagnosis function. A corrupting function ϕ is *harmful* if there exists an observation $s \in P(L(G))$ generated by the plant, such that s can be corrupted into a word $s' = \phi(s) \in P(L(G))$, and $(\gamma(s), \gamma(s')) \in \{(F, N), (F, U)\}$. $\diamond$

According to the above definition, a corrupting function is harmful if an uncorrupted observation s which allows the operator to detect a fault (diagnosis state F) can be altered into a corrupted observation s' corresponding to the absence of fault or to the uncertain situation (diagnosis state N or U). In simple words, a harmful attack may prevent the operator from detecting the occurrence of the fault during the system evolution.

Given a plant G with the set of compromised events E_{com}, the main contribution of this paper is that of providing a diagnoser, called a stealthy joint diagnoser, which contains all possible stealthy attack actions that an attacker is able to execute. Then, it is shown how such a structure allows one to determine if there exist stealthy attacks which are harmful, thus allowing the operator to establish if the diagnoser is robust to attacks in the considered setting.

5 STEALTHY JOINT DIAGNOSER

5.1 Attacker Diagnoser and Operator Diagnoser

In this subsection, we introduce two special diagnosers, called *Attacker Di-*

agnoser and *Operator Diagnoser*, which serve to compute *Joint Diagnoser* used to solve the considered diagnosis problem under attack. In [14], given a plant G, the notions of attacker observer $Obs_{att}(G)$, operator observer $Obs_{opr}(G)$ and their construction algorithms are proposed to solve the state estimation problem in the presence of an attack. Given a plant G, due to the fact that the diagnoser of G is equivalent to the observer of $Rec(G)$, we define the attacker diagnoser and the operator diagnoser as follows.

Definition 5: Given a plant G with set of observable events E_o and set of fault events E_f, let Rec(G) be its fault recognizer. We define the *attacker diagnoser* as

$$Diag_{att}(G) = Obs_{att}(Rec(G)), \tag{5}$$

and the *operator diagnoser* as

$$Diag_{opr}(G) = Obs_{opr}(Rec(G)). \tag{6}$$

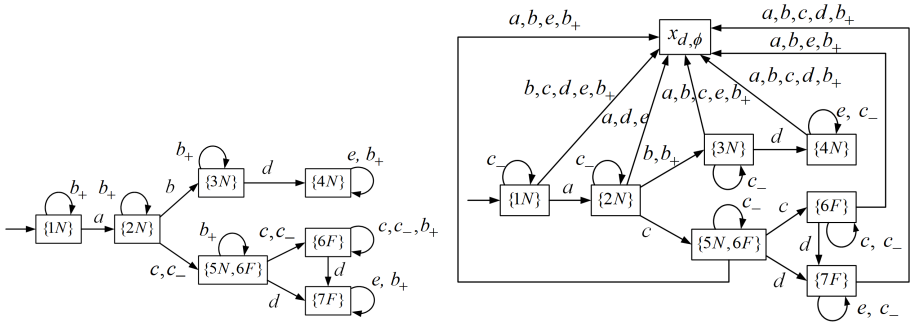
Given a plant G, one can compute $Diag_{att}(G)$ (resp. $Diag_{opr}(G)$) by applying Algorithm 1 (resp. Algorithm 2) in [14] to its fault recognizer $Rec(G)$. Note that a non-stealthy attack may transform an observation s into a corrupted observation s' that cannot be generated by the plant when no attack occurs. In this case, the operator diagnoser receives an attack word that is not consistent with any uncorrupted observation and yields a dummy state, denoted as $x_{d,\emptyset}$.

Based on an attack word ω, $Diag_{att}(G)$ and $Diag_{opr}(G)$ make their diagnostic decisions $\gamma_{att}(\omega)$ and $\gamma_{opr}(\omega)$, respectively, where $\gamma_{att} : E_a^* \to \{N, F, U\}$ and $\gamma_{opr} : E_a^* \to \{N, F, U\}$ are the attack diagnosis function and the operator diagnosis function, respectively. The computation of their diagnostic decisions will be discussed in Section 6.

Example 1: Consider the plant G in Fig. 2(a). Let $E_{ins} = \{b\}$ and $E_{era} = \{c\}$. Figs. 4(a) and 4(b) show $Diag_{att}(G)$ and $Diag_{opr}(G)$, respectively. We have self-loop transitions labeled with the inserted event b_+ at all the states of $Diag_{att}(G)$ since the attacker knows that the inserted event is fictitious. Analogously, we have self-loop transitions labeled with the erased event c_- at all the states of $Diag_{opr}(G)$ since the event has been erased by the attacker and the operator receives no information regarding such an event occurrence. There exists a transition labeled with the erased event c_- from state $\{2N\}$ to $\{5N, 6F\}$ in $Diag_{att}(G)$ since the attacker knows that the event c has occurred. There exists a transition labeled with the inserted event b_+ from state $\{2N\}$ to $\{3N\}$ in $Diag_{opr}(G)$ since the $Diag_{opr}(G)$ cannot distinguish between b_+ and the corresponding event b.

5.2 Joint Diagnoser and Pruning Process

Definition 6: A *joint diagnoser* $J\text{-}Diag(G) = (R, E_a, \delta_a, r_0)$ with respect to G and E_{com} is defined as $J\text{-}Diag(G) = Diag_{att}(G) \parallel Diag_{opr}(G)$. $\diamond$



(a) $Diag_{att}(G)$ (b) $Diag_{opr}(G)$

Figure 4: (a) Attacker diagnoser and (b) operator diagnoser.

A joint diagnoser $J\text{-}Diag(G)$ is obtained by performing the parallel composition of $Diag_{att}(G)$ and $Diag_{opr}(G)$. The joint diagnoser is an extension of the *joint estimator* proposed in [14] where we append to every state pair the fault diagnosis information. We notice that the joint diagnoser generates the same language as the joint estimator; thus all the results that put in relationship the language of the joint diagnoser and the language of the joint estimator are omitted here due to the limited space.

In general, $J\text{-}Diag(G)$ includes the non-stealthy attacks, i.e., attacks that reveal the attacker's presence to the operator. Therefore, $J\text{-}Diag(G)$ must be pruned, in the sense that some particular diagnosis states must be withdrawn. The states of $J\text{-}Diag(G)$ are defined by pairs, i.e., $r = (x_d, \bar{x}_d)$. We define the set of exposing states as $R_e := \{r \in R \mid r = (x_d, \bar{x}_d), \bar{x}_d = x_{d,\emptyset}\}$. Each time the joint diagnoser reaches an exposing state, the attack is no longer stealthy. Furthermore, there may exist some weakly exposing states defined as R_{we} from which an exposing state is necessarily reached. Then we define $\bar{R}_e = R_e \cup R_{we}$ as the weakly exposing region. At the weakly exposing state $r \in R_{we}$, neither the firing of a legitimate event e, the erasure of this event (if it is possible), nor the insertion of some $e'_+ \in E_{ins}$ can lead the joint diagnoser outside the weakly exposing region $\bar{R}_e$. One can recursively compute $\bar{R}_e$ in $J\text{-}Diag(G)$ following the algorithm in [15].

Definition 7: Given a joint diagnoser $J\text{-}Diag(G) = (R, E_a, \delta_a, r_0)$, the *stealthy joint diagnoser* of $J\text{-}Diag(G)$ is defined as $SJ\text{-}Diag(G) = (R_s, E_a, \delta_{sa}, r_0)$, where $R_s = R \setminus \bar{R}_e$, $\delta_{sa} = \delta_a|_{R_s \times E_a \to R_s}$. ◇

The notation $\delta_a|_{R_s \times E_a \to R_s}$ means that we are restricting δ_a to the smaller domain of the stealthy states R_s. The stealthy joint diagnoser $SJ\text{-}Diag(G)$ can be obtained from $J\text{-}Diag(G)$ by removing all states in $\bar{R}_e$ and their corresponding input and output arcs. As a result, $SJ\text{-}Diag(G)$ includes all the possible stealthy actions that an attacker may implement during the system evolution.

Example 2: Consider again the plant in Fig. 2(a). The joint diagnoser *J-Diag*(G) is shown in Fig. 5, where exposing states are highlighted in gray and weakly exposing states are highlighted in yellow. Consider the state $(\{2N\},\{3N\})$. Since there exists a transition labeled $b \in E_o \backslash E_{era}$ that yields from $(\{2N\},\{3N\})$ to the exposing state $(\{3N\},\{x_{d,\emptyset}\})$ in $\bar{R}_e$, and there does not exist a transition in E_+ yielding to a state not in $\bar{R}_e$, the state $(\{2N\},\{3N\})$ is added to $\bar{R}_e$. Note that the state $(\{5N,6F\},\{2N\})$ is not a weakly exposing state. The attacker can perform the insertion action b_+ before the execution of the event $d \in E_o \backslash E_{era}$, thus leading the joint diagnoser outside the set $\bar{R}_e$.

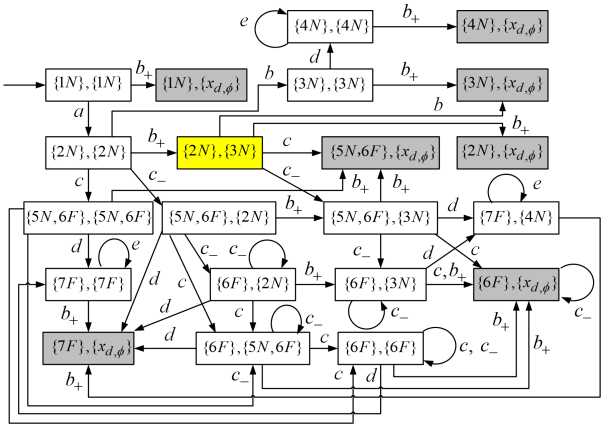


Figure 5: Joint diagnoser *J-Diag*(G) in Example 2.

Herein we provide the following result that characterizes the transition function of *SJ-Diag*(G).

Theorem 1: Consider a plant G with set of observable events E_o and set of fault events E_f, and its diagnoser $Diag(G) = (X_d, E_o, \delta_d, x_{d,0})$. Let $\phi : E_o^* \to E_o^*$ be a corrupting function. Given attack alphabet E_a, let *SJ-Diag*$(G) = (R_s, E_a, \delta_{sa}, r_0)$ be its stealthy joint diagnoser. For all uncorrupted observations $s \in P(L(G)))$ with the corresponding attack word $\omega = f(s)$ and corrupted observation $s' = \phi(s)$, it holds that:
$[\delta_{sa}^*(r_0,\omega) = (x_d,\bar{x}_d) \Longleftrightarrow \delta_d^*(x_{d,0},s) = x_d, \delta_d^*(x_{d,0},s') = \bar{x}_d]$.

The result follows from arguments similar to those in [14] (Theorem 1 therein). The state pair $(x_d,\bar{x}_d)$ reached in *SJ-Diag*(G) by an attack word ω describes the joint diagnosis state estimate. The first element x_d represents the *correct diagnosis state estimate* of the operator, denoted as $x_d = \delta_d^*(x_{d,0},s)$, as seen by the attacker for the uncorrupted observation s consistent with the attack word ω. The second element $\bar{x}_d$ represents the *corrupted diagnosis state estimate* of the operator, denoted as $\bar{x}_d = \delta_d^*(x_{d,0},s')$, which is the current state of its

realization based on the corrupted observation s' that it receives. As desired, a state pair in $SJ\text{-}Diag(G)$ embeds the necessary information for the operator to draw diagnostic conclusions based on the uncorrupted and the corrupted observations, respectively, which will be further discussed in the next section.

6 Fault Diagnosis Under Attack

Recall how the attacker compromises a system. Assume that the system generates an observation s. The attacker may corrupt the observation s, resulting in the corrupted observation $s' = \phi(s)$, where ϕ is the corrupting function. In more detail, this function ϕ proceeds in two steps: first, the observation s is transformed into an attack word $\omega = f(s)$ via the attack function f. Subsequently, the attack word ω is transformed into the corrupted observation $s' = \widehat{P}(\omega)$ using the operator mask $\widehat{P}$. The attacker aims to mislead the operator to make an incorrect diagnostic decision based on the corrupted observation that it receives.

Problem 1: Given a plant G with E_o, E_f and E_{com}, and given an attack word $\omega \in E_a^*$, the fault diagnosis problem under attack consists in determining if the operator takes consistent diagnostic decisions based on an uncorrupted observation s and the corresponding corrupted observation s', respectively.

In this section, we show that the stealthy joint diagnoser $SJ\text{-}Diag(G)$ can provide a solution to Problem 1. Let us first define the diagnosis pair function whose purpose is to classify the diagnosis state of $SJ\text{-}Diag(G)$.

Definition 8: Let $SJ\text{-}Diag(G) = (R_s, E_a, \delta_{sa}, r_0)$ be a stealthy joint diagnoser. Given a state r_s reached in $SJ\text{-}Diag(G)$ by the attack word $\omega \in E_a^*$, the *diagnosis pair function*

$$d : R_s \to \{N, F, U\} \times \{N, F, U\}$$

associates each state r_s to a diagnosis pair $d(r_s) = (\gamma_{att}(\omega), \gamma_{opr}(\omega))$, where $\gamma_{att} : E_a^* \to \{N, F, U\}$ and $\gamma_{opr} : E_a^* \to \{N, F, U\}$ are the attack diagnosis function and the operator diagnosis function, respectively. $\diamond$

According to Theorem 1, based on an attack word ω, the diagnostic decision $\gamma_{att}(\omega)$ of $Diag_{att}(G)$ is equivalent to that of the nominal diagnoser $Diag(G)$ based on the uncorrupted observation s consistent with the word ω, i.e., $\gamma_{att}(\omega) = \gamma(s)$. Likewise, $Diag_{opr}(G)$ makes its diagnostic decision $\gamma_{opr}(\omega) = \gamma(s')$, where s' is the corrupted observation consistent with the word ω. Therefore, $SJ\text{-}Diag(G)$ shows the joint diagnostic decision of the operator composed by the *correct diagnostic decision* based on s without attack and the *corrupted diagnostic decision* based on s' in the presence of an attack. Once the stealthy joint diagnoser has been constructed, fault diagnosis under attack can be performed by tracking the current stealthy joint diagnoser state in response to the attack word ω.

Then we define the set of all diagnosis pairs as $D = \{N, F, U\} \times \{N, F, U\}$ that can be classified as

$$D = D_{corr} \cup D_{wrng} \cup D_{harm},$$

where

- $D_{corr} = \{(N, N), (U, U), (F, F)\}$ is the set of *correct diagnosis pair*;
- $D_{wrng} = \{(N, U), (N, F), (U, N), (U, F)\}$ is the set of *wrong diagnosis pair*;
- $D_{harm} = \{(F, N), (F, U)\}$ is the set of *harmful diagnosis pair*.

Definition 9: Let $SJ\text{-}Diag(G) = (R_s, E_a, \delta_{sa}, r_0)$ be a stealthy joint diagnoser. A state $r_s \in R_s$ is *correct* if $d(r_s) \in D_{corr}$, *wrong* if $d(r_s) \in D_{wrng}$, and *harmful* if $d(r_s) \in D_{harm}$. $\diamond$

The state set R_s of $SJ\text{-}Diag(G)$ can be partitioned into correct state set $R_{s,corr}$, wrong state set $R_{s,wrng}$, and harmful state set $R_{s,harm}$ according to their diagnosis pairs. First, when the $SJ\text{-}Diag(G)$ is in a correct state, either the attack does not corrupt the system's observation, or the operator makes the correct diagnostic decisions even if an attack has occurred. In this case, we say that the attack is ineffective.

Then, when the $SJ\text{-}Diag(G)$ reaches a wrong state, the operator makes wrong diagnostic decisions based on the corrupted observation, which are inconsistent with the decisions based on the uncorrupted observation. In this case, due to the attack, the fault diagnosis result is degraded but does not pose a real danger.

Finally, we define as *harmful states* of the stealthy joint diagnoser those states that correspond to a harmful attack, namely those states that correspond to the detection of the fault in the case of uncorrupted observation, and no detection in the case of corrupted observation. The set of harmful states can be partitioned into two classes (the state with $d(r_s) = (F, N)$ and $d(r_s) = (F, U)$) in which the attacker has been able to hide the occurrence of a fault during the system evolution, thus thwarting the operator's effort. The stealthy joint diagnoser can be used to identify if an attack is harmful according to the presence of the harmful states.

Proposition 1: Given a plant $G = (X, E, \delta, x_0)$ with the set of compromised events E_{com}, let $D_{harm} = \{(F, N), (F, U)\}$ be the harmful diagnosis pair set, and $SJ\text{-}Diag(G) = (R_s, E_a, \delta_{sa}, r_0)$ be the stealthy joint diagnoser. There exists a stealthy corrupting function that is harmful iff $R_{s,harm} \neq \emptyset$, i.e., the $SJ\text{-}Diag(G)$ may reach a harmful state.

Proof: (If) Assume that there exists a harmful state $r_s = (x_d, \bar{x}_d)$ in $SJ\text{-}Diag(G)$ by executing the attack word ω such that $d(r_s) = (\gamma_{att}(\omega), \gamma_{opr}(\omega)) \in D_{harm}$. By Definition 2, there exists an uncorrupted observation $s \in P(L(G))$

such that $\omega = f(s)$. By Theorem 1, we have $r_s = \delta^*_{sa}(r_0, \omega) = (x_d, \bar{x}_d)$ such that $\delta^*_d(x_{d,0}, s) = x_d$ and $\delta^*_d(x_{d,0}, s') = \bar{x}_d$ with $s' = \phi(s)$, where ϕ is the corrupting function. Therefore, there exists a stealthy corrupting function ϕ transforming the observation s into s' such that $(\gamma(s), \gamma(s')) = (\gamma_{att}(\omega), \gamma_{opr}(\omega)) = d(r_s) \in D_{harm}$, i.e., $(\gamma(s), \gamma(s')) \in \{(F, N), (F, U)\}$. According to Definition 4, we conclude that the corrupting function ϕ is harmful.

(Only if) Assume that there exists a corrupting function ϕ that is harmful. By Definition 4, there exists an observation $s \in P(L(G))$, such that s can be corrupted into a word $s' = \phi(s) \in P(L(G))$, and $(\gamma(s), \gamma(s')) \in \{(F, N), (F, U)\}$. By $s, s' \in P(L(G))$, we have $\delta^*_d(x_{d,0}, s) = x_d$ and $\delta^*_d(x_{d,0}, s') = \bar{x}_d$. Since the stealthy joint diagnoser $SJ\text{-}Diag(G)$ includes all the possible stealthy attacks, according to Theorem 1, there necessarily exists a state $r_s = (x_d, \bar{x}_d)$ in $SJ\text{-}Diag(G)$ by executing word $\omega = f(s)$ such that $\delta^*_d(x_{d,0}, s) = x_d$ and $\delta^*_d(x_{d,0}, s') = \bar{x}_d$. By $d(r_s) = (\gamma_{att}(\omega), \gamma_{opr}(\omega)) = (\gamma(s), \gamma(s')) \in \{(F, N), (F, U)\}$, i.e., $d(r_s) \in D_{harm}$, and Definition 9, we conclude that the state r_s is harmful, i.e., $R_{s,harm} \neq \emptyset$.

Thus, the existence of a harmful state in the stealthy diagnoser is a necessary and sufficient condition for the existence of a stealthy harmful attack that can hide the occurrence of a fault.

Example 3: Recall the plant G in Fig. 2(a) and the joint diagnoser $J\text{-}Diag(G)$ in Fig. 5. The stealthy diagnoser $SJ\text{-}Diag(G)$ is shown in Fig. 6, where correct states are highlighted in brown, wrong states are highlighted in blue, and harmful states are highlighted in green.

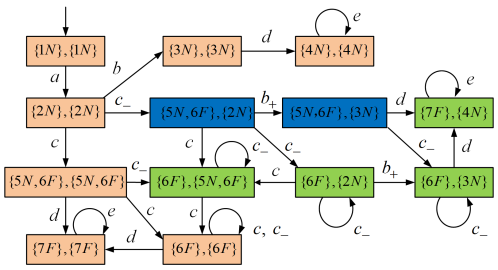


Figure 6: Stealthy diagnoser $SJ\text{-}Diag(G)$ in Example 3.

Table 1 presents the performance of fault diagnosis during the occurrence of the attack word $\omega_1 = ac\_ccc^n (n \in \mathbb{N})$. When the $SJ\text{-}Diag(G)$ reaches the harmful state $(\{6F\}, \{5N, 6F\})$ by executing ac_c as seen in the second row of Table 1, the system generates the observation acc and the operator may identify the corrupted observation ac. In the absence of an attack, the operator is able to detect the occurrence of a fault. However, due to the erasure of event c by the attack, the operator is no more sure if the fault has occurred or not. Indeed, as seen in the third row of Table 1, the fault occurrence cannot be detected by the

operator until the plant produces the third c, i.e., the uncorrupted observation is $accc$.

Table 2 presents the performance of fault diagnosis during the occurrence of the attack word $\omega_2 = ac\_b_+de^n (n \in \mathbb{N})$. When the $SJ\text{-}Diag(G)$ yields the harmful state $(\{7F\}, \{4N\})$ by executing ac_b_+d as seen in the second row of Table 2, the observation acd has been transformed into the corrupted observation abd by a harmful attack. As a result, the harmful attack induces the operator to draw the incorrect conclusion that the fault has not occurred based on the corrupted observation abd. Finally, the harmful attack can be implemented by first erasing the occurrence of event c, and then inserting b_+ when the plant generates the uncorrupted observation ac. We notice that the $SJ\text{-}Diag(G)$ will remain in a cycle formed with harmful states associated with the attack word $\omega_2 = ac\_b_+de^n (n \in \mathbb{N})$.

Table 1: Fault diagnosis during the occurrence of word $\omega_1 = ac\_ccc^n (n \in \mathbb{N})$

ω	r_s	s	s'	$d(r_s)$
$ac\_$	$(\{5N, 6F\}, \{2N\})$	ac	a	(U, N)
ac_c	$(\{6F\}, \{5N, 6F\})$	acc	ac	(F, U)
ac_cc	$(\{6F\}, \{6F\})$	$accc$	acc	(F, F)

Table 2: Fault diagnosis during the occurrence of word $\omega_2 = ac\_b_+de^n (n \in \mathbb{N})$

ω	r_s	s	s'	$d(r_s)$
ac_b_+	$(\{5N, 6F\}, \{3N\})$	ac	ab	(U, N)
ac_b_+d	$(\{7F\}, \{4N\})$	acd	abd	(F, N)
ac_b_+de	$(\{7F\}, \{4N\})$	$acde$	$abde$	(F, N)

7 Conclusions and future work

In the presence of an attack, we have considered the fault diagnosis of discrete event systems, where sensor readings may be corrupted by the attack in the form of insertions and erasures. In this context, we propose a stealthy joint diagnoser with the purpose to show how an attacker may affect the operator to perform the correct online fault diagnosis. Our future research in this framework will focus on diagnosability analysis under attack. We plan to first formalize the notion of system diagnosability under attack and and then provide its verification method. Second, we plan to use the stealthy joint diagnoser to synthesize suitable attacks that can avoid detection from the operator and result in a violation of system diagnosability.

References

[1] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

[2] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer, 2021.

[3] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.

[4] L. K. Carvalho, J. C. Basilio, and M. V. Moreira, "Robust diagnosis of discrete event systems against intermittent loss of observations," *Automatica*, vol. 48, no. 9, pp. 2068–2078, Sep. 2012.

[5] L. K. Carvalho, M. V. Moreira, J. C. Basilio, and S. Lafortune, "Robust diagnosis of discrete-event systems against permanent loss of observations," *Automatica*, vol. 49, no. 1, pp. 223–231, Jan. 2013.

[6] 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, Aug. 2021.

[7] R. Fritz and P. Zhang, "Modeling and detection of cyber attacks on discrete event systems," in *Proc. 14th IFAC Workshop Discrete Event Syst.*, Sorrento, Italy, May 2018, pp. 285–290.

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

[9] M. Wakaiki, P. Tabuada, and J. P. Hespanha, "Supervisory control of discrete-event systems under attacks," *Dynamic Games and Applications*, vol. 9, no. 4, pp. 965–983, Dec. 2019.

[10] Y. Wang and M. Pajic, "Supervisory control of discrete event systems in the presence of sensor and actuator attacks," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Nice, France, Dec. 2019, pp. 5350–5355.

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

[12] R. Su, "Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations," *Automatica*, vol. 94, pp. 35–44, Aug. 2018.

[13] R. Meira-Góes, 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, Nov. 2020.

[14] Q. Zhang, C. Seatzu, Z. Li, and A. Giua, "Joint state estimation under attack of discrete event systems," *IEEE Access*, vol. 9, pp. 168 068–168 079, 2021.

[15] Q. Zhang, Z. Li, C. Seatzu, and A. Giua, "Stealthy attacks for partially-observed discrete event systems," in *Proc. 23rd IEEE Int. Conf. Emerg. Technol. Factory Automat.*, Turin, Italy, Sep. 2018, pp. 1161–1164.