sensors

MDPI

Article

Sensor Network Attack Synthesis against Fault Diagnosis of Discrete Event Systems †

Tenglong Kang [1,2], Yifan Hou [1] and Ding Liu [1,*]

[1] School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China; tlkang@stu.xidian.edu.cn (T.K.); yfhou@xidian.edu.cn (Y.H.)
[2] Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy
* Correspondence: dliu@xidian.edu.cn
† This paper is partially presented at the 62nd IEEE Conference on Decision and Control (CDC), Singapore, 13–15 December 2023.

Abstract: This paper investigates the problem of synthesizing network attacks against fault diagnosis in the context of discrete event systems (DESs). It is assumed that the sensor observations sent to the operator that monitors a system are tampered with by an active attacker. We first formulate the process of online fault diagnosis under attack. Then, from the attack viewpoint, we define a sensor network attacker as successful if it can degrade the fault diagnosis in the case of maintaining itself as undiscovered by the operator. To verify such an attacker, an information structure called a joint diagnoser (JD) is proposed, which describes all possible attacks in a given attack scenario. Based on the refined JD, i.e., stealthy joint diagnoser (SJD), we present an algorithmic procedure for synthesizing a successful attacker if it exists.

Keywords: fault diagnosis; sensor network attack; discrete event systems

check for updates

Citation: Kang, T.; Hou, Y.; Liu, D. Sensor Network Attack Synthesis against Fault Diagnosis of Discrete Event Systems. *Sensors* **2024**, *24*, 4445. https://doi.org/10.3390/s24144445

Academic Editors: Gabriele Patrizi and Marco Carratù

Received: 25 June 2024
Revised: 7 July 2024
Accepted: 8 July 2024
Published: 9 July 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the advancement in computation and communication technologies, cyber–physical systems (CPSs) have emerged as the new generation of engineering systems and computation devices embedded within physical dynamics [1–5]. Examples of CPSs include sensor networks, networked computer systems, autonomous vehicles, etc. Usually, the undesired behavior of CPS may be caused by the occurrence of component faults. Hence, we wish that a fault can be diagnosed after its occurrence based on data collected by sensors. In a CPS, the transmission of data information relies on communication networks, which may introduce vulnerabilities to cyber-attacks.

In this paper, we study the behavior of a CPS within the framework of discrete event systems (DESs) [6]. One significant advantage of the DES approach to fault diagnosis is that it is a model-based approach. Therefore, DES models naturally capture faults as abrupt events in systems, which facilitates the analysis of faulty behaviors of the system based on limited observations. Another motivation is that, unlike complex models or large data-driven models, DES modeling offers an opportunity to make efficient decisions over an abstract discrete decision space. Specifically, a DES is an event-driven model-based formalism that can provide a time-abstract model for complex systems.

There are two main approaches and tools available for the fault diagnosis of a DES, i.e., automata theory [7–9] and Petri nets [10–13]. A comprehensive survey can be found in [14]. Within the context of automata theory, the underlying structure of a DES is that of a finite state automaton. This will be formally defined in Section 3. Online fault diagnosis in a DES aims to perform model-based inferencing at runtime based on the current observations, and determines if a fault event has occurred or not in the past. In order to achieve this, a process called a diagnoser is proposed in the seminal work [7], which associates to each observed word of events a diagnosis state, such as "negative", "positive", or "uncertain".

2. Related Work

The related literature in the context of robust fault diagnosis against disruptions in the communication channels includes, e.g., [15–17]. For a survey of the work in this domain, we refer to [17]. These works are concerned with the loss of observations, i.e., any event in a certain subset of the observable event set may become unobservable due to sensor failures or communication channel constraints. It is shown in [17] that, based on a general diagnosis framework that takes sensor failures into account, a procedure of online diagnosis can be presented. However, such disruptions are not necessarily caused by active attacks.

Motivated by this, our focus is on a specific class of cyber-attacks called sensor network attacks, which may potentially inject false network signals into the communication channels between sensors and the system operator, as shown in Figure 1. As will be clear in the context, in the worst-case scenario, due to the impact of an attacker, the operator cannot diagnose the fault occurrence as it should. Indeed, sensor network attacks have been investigated in the context of attack detection [18,19], state estimation [20,21], and supervisory control [22,23]. However, very few studies have been devoted to fault diagnosis under attack. Apart from our earlier work [24], two recent contributions in this context are [25,26]. In these works, from a defense viewpoint, the authors aim to capture the fact that fault diagnosis can be properly performed against malicious attacks.

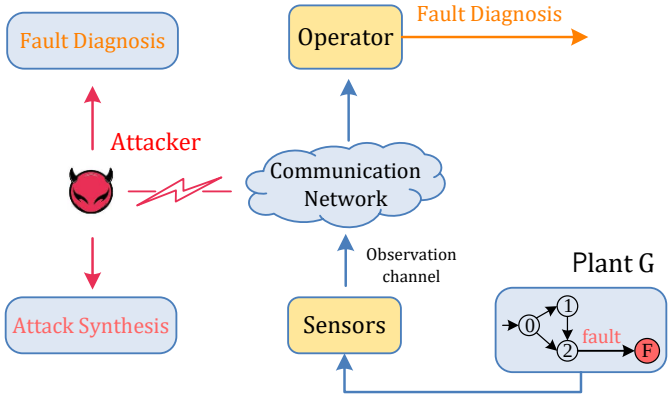


Figure 1. Fault diagnosis architecture under attack. To make the architecture more illustrative, different components are represented by different colors.

In this paper, we study from the attacker's viewpoint, which allows us to determine conditions under which the attacker is harmful and stealthy and synthesize a strategy to satisfy those conditions. Our objective is to verify a successful attacker that can achieve two goals: (i) induce the operator to no longer diagnose the fault occurrence, guaranteeing its harmfulness; (ii) at the same time, maintain itself as undetected by the operator, guaranteeing its stealthiness. To this end, we first propose a bipartite diagnoser called a joint diagnoser (JD) that embeds all possible attacks in a given attack scenario. To capture stealthy attacks only, a refined JD, i.e., stealthy joint diagnoser (SJD), is presented, which finally provides a necessary and sufficient condition for the existence of a successful attacker.

Another contribution of this work is the development of a successful attack synthesis algorithm. To ensure stealthiness, an attacker must consider the race between inserted events and internal system events. In essence, when synthesizing successful attacks, an attack event must be inserted at the desired time. In order to capture this race, we define a pre-empting state of the SJD as those states corresponding to the case where the internal system events are pre-empted by the inserted events. Based on this, a successful attack synthesis algorithm is finally proposed. By providing a general synthesis approach, our ultimate goal is to understand what kinds of attacks systems are not robust to and how to reduce their impact on diagnosis estimation performance. In Table 1, we compare the proposed approach with previous studies.

Table 1. Comparison of related works, where the "✓" symbol indicates the selection of an option.

Reference	Sensor Failures	Sensor Attacks	Attack Detection	Fault Diagnosis	Attack Synthesis
[15–17]	✓			✓	
[18,19]		✓	✓		
[20–23]		✓	✓		✓
[25,26]		✓		✓	
This work		✓	✓	✓	✓

Compared with our conference paper [24] that introduced the diagnosis setting without providing proofs of the results, we here provide formal proofs and detailed examples. Furthermore, we present an approach for online synthesizing a successful attacker.

3. Preliminaries

3.1. Finite-State Automata

We first review some common notations of DESs used throughout this paper; the reader is referred to [6] for more details. It is assumed that a DES to be diagnosed is modeled as a finite-state automaton (automaton for short) $G = (Q, \Sigma, \delta, q_0)$, where Q is the finite set of states, Σ is the finite set of events, $\delta : Q \times \Sigma \to Q$ is the partial transition function, and $q_0 \in Q$ is the initial state. Suppose that set Σ is divided into the observable event set Σ_o and the unobservable event set Σ_{uo}. The set of active events at state q is defined as $\Gamma(q) = \{\sigma \in \Sigma \,|\, \delta(q, \sigma)!\}$, where "!" means "defined". The language generated by G, denoted by $L(G)$ or simply L, is defined as $L = \{s \in \Sigma^* \mid \delta(q_0, s)!\}$, where Σ^* denotes the Kleene closure of Σ. Let s denote a word of L. The set of the prefixes of s is denoted as $\overline{s} = \{u \in \Sigma^* | (\exists v \in \Sigma^*)[uv = s]\}$.

The natural projection $P : \Sigma^* \to \Sigma_o^*$ is defined as (where ε is the empty word)

$$P(\varepsilon) = \varepsilon \text{ and } P(s\sigma) = \begin{cases} P(s)\sigma, & \text{if} \quad \sigma \in \Sigma_o; \\ P(s), & \text{if} \quad \sigma \in \Sigma_{uo}. \end{cases}$$

The inverse projection $P^{-1} : \Sigma_o^* \to 2^{\Sigma^*}$ is defined as $P^{-1}(w) = \{s \in L(G) \,|\, P(s) = w\}$, i.e., $P^{-1}(w)$ consists of all words s in $L(G)$ whose observations are w.

Let $G_1 = (Q_1, \Sigma_1, \delta_1, q_{01})$ and $G_2 = (Q_2, \Sigma_2, \delta_2, q_{02})$. The parallel composition between G_1 and G_2 is defined as $G_1 \parallel G_2 = Ac(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta, (q_{01}, q_{02}))$, where $\delta[(q_1, q_2), \sigma] = (q_1', q_2')$ if $\delta_1(q_1, \sigma) = q_1'$ and $\delta_2(q_2, \sigma) = q_2'$; $\delta[(q_1, q_2), \sigma] = (q_1', q_2)$ if $\delta_1(q_1, \sigma) = q_1'$ and $\sigma \notin \Sigma_2$; $\delta[(q_1, q_2), \sigma] = (q_1, q_2')$ if $\delta_2(q_2, \sigma) = q_2'$ and $\sigma \notin \Sigma_1$; otherwise, it is undefined. In the definition of $G_1 \parallel G_2$, $Ac(\cdot)$ denotes the accessible part of an automaton.

3.2. Fault Diagnosis

In the literature of fault diagnosis in DESs, the unobservable event set Σ_{uo} is further partitioned into the set of regular unobservable events Σ_{reg} and the set of fault events Σ_f. Let $\Psi(\Sigma_f) = \{sf \in L \mid s \in \Sigma^*, f \in \Sigma_f\}$ denote the set of all finite words in L that end with a fault event f. With some abuse of notation, $\Sigma_f \in s$ denotes that $\overline{s} \cap \Psi(\Sigma_f) \neq \emptyset$. A word s is said to be faulty (resp., normal) if $\Sigma_f \in s$ (resp., $\Sigma_f \notin s$). The fault diagnosis problem in a DES aims to decide, using the current observation $w \in \Sigma_o^*$, if a fault has already occurred or not. To solve this problem, one wishes to build a diagnosis function $\gamma : \Sigma_o^* \to \{N, F, U\}$ associating each observation to a diagnosis state such that

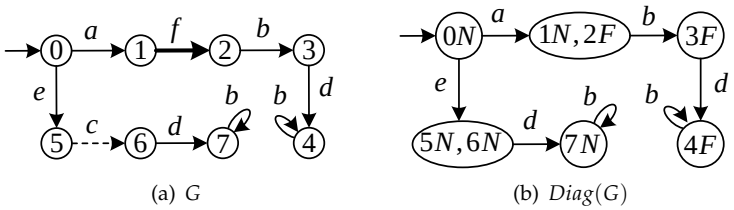
$$\gamma(w) = \begin{cases} N, & \text{if} \quad \forall s \in P^{-1}(w), \Sigma_f \notin s; \\ F, & \text{if} \quad \forall s \in P^{-1}(w), \Sigma_f \in s; \\ U, & \text{otherwise.} \end{cases}$$

In other words, an observation w is called normal if $\gamma(w) = N$ since, in this case, no word producing w contains a fault; faulty if $\gamma(w) = F$ since, in this case, all words producing w

contain a fault; ambiguous otherwise. A standard way to compute the diagnosis function is by using a diagnoser. The diagnoser of G is defined as

$$Diag(G) = (Q_d, \Sigma_o, \delta_d, q_{d,0}). \tag{1}$$

Its construction procedure is detailed in [6]. By construction, it holds $L(Diag(G)) = P(L(G))$. The diagnoser state space Q_d is a subset of $2^{Q\times\{N,F\}}$. The diagnoser allows one to associate every state to a diagnosis value $\gamma(q_d) = \gamma(w)$, where $q_d = \delta_d(q_{d,0}, w)$. The diagnoser state q_d is negative if $\gamma(q_d) = N$; positive if $\gamma(q_d) = F$; uncertain if $\gamma(q_d) = U$. Figure 2a shows a plant G, where $\Sigma_o = \{a, b, d, e\}$, $\Sigma_{uo} = \{c, f\}$, and $\Sigma_f = \{f\}$. The corresponding diagnoser is shown in Figure 2b.



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

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

4. Sensor Network Attacks

We start by defining a system model to be diagnosed under attack depicted in Figure 3, where the shadowed block denotes a sensor network attacker that intervenes in the communication channels between the sensors and the operator. It is assumed that the attacker can compromise a subset of the sensor network channels. Specifically, it may implement two types of sensor network attacks: (i) Sensor Erasure attack (SE-attack), which erases some readings generated by the plant; (ii) Sensor Insertion attack (SI-attack), which inserts some fake readings that have not occurred in the plant. If a plant generates a word $s \in \Sigma^*$, the attacker observes an original observation $w = P(s)$ and then produces a corrupted observation w'. The operator computes diagnosis state $\gamma(w')$ from the corrupted observation w'.

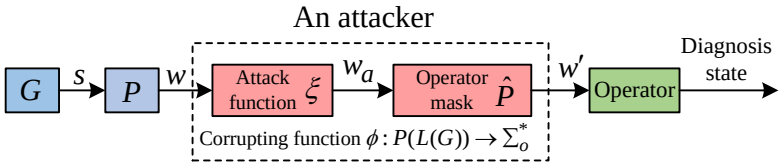


Figure 3. A fault diagnosis system under attack.

To formally define the attack function in Figure 3, we denote by $\Sigma_{era} \subseteq \Sigma_o$ (resp., $\Sigma_{ins} \subseteq \Sigma_o$) the set of events subject to SE-attacks (resp., SI-attacks). Correspondingly, we define $\Sigma_- = \{\sigma_- \mid \sigma \in \Sigma_{era}\}$ as the set of erased events and $\Sigma_+ = \{\sigma_+ \mid \sigma \in \Sigma_{ins}\}$ as the set of inserted events. We also define $\Sigma_a = \Sigma_o \cup \Sigma_+ \cup \Sigma_-$ as the attack alphabet.

Definition 1. *Given a plant G with sets Σ_{ins} and Σ_{era}, an attack function $\xi : P(L(G)) \to \Sigma_a^*$ satisfies the following constraints:*

(1) $\xi(\varepsilon) \in \Sigma_+^*$;

(2) $\forall w\sigma \in P(L(G))$, where $w \in \Sigma_o^*$ and $\sigma \in \Sigma_o$:

$$\xi(w\sigma) \in \begin{cases} \xi(w)\{\sigma\}\Sigma_+^* & \text{if } \sigma \in \Sigma_o \backslash \Sigma_{era}, \\ \xi(w)\{\sigma_-, \sigma\}\Sigma_+^* & \text{if } \sigma \in \Sigma_{era}. \end{cases} \tag{2}$$

Statement (1) in Definition 1 implies that the attacker may insert an arbitrary word $t \in \Sigma_+^*$ at the initial state before any generated word of G is observed. By Statement (2), the attacker cannot erase σ if σ does not belong to Σ_{era}. However, it may insert an arbitrary

word $t \in \Sigma_+^*$ after σ. If an event $\sigma \in \Sigma_{era}$ occurs, the attacker may either erase σ or leave it intact, and then insert any arbitrary word $t \in \Sigma_+^*$. Compared with the nondeterministic attack function in [22], the attack function ξ in Definition 1 is deterministic; therefore, in order to provide different classes of attack strategies, we define the set of all possible attack functions depending on relativity to sets Σ_{era} and Σ_{ins} as "$\Xi(\Sigma_{era}, \Sigma_{ins})$", abbreviated as Ξ.

An attack function ξ introduces an attack language, denoted as $L(\xi, G) = \{\xi(w) \mid w \in P(L(G))\}$ (abbreviated as $L(\xi)$). Let $w_a \in L(\xi)$ be an attack word. Given an attack word w_a, to describe the corrupted observation w' received by the operator, an operator mask $\widehat{P} : \Sigma_a^* \to \Sigma_o^*$ is defined as

$$\widehat{P}(\varepsilon) = \varepsilon, \widehat{P}(w_a\sigma') = \begin{cases} \widehat{P}(w_a)\sigma & \text{if } \sigma' = \sigma \in \Sigma_o \vee \sigma' = \sigma_+ \in \Sigma_+, \\ \widehat{P}(w_a) & \text{if } \sigma' = \sigma_- \in \Sigma_-. \end{cases} \tag{3}$$

We now define two diagnosis functions on Σ_a. The attacker diagnosis function $\gamma_{att} : \Sigma_a^* \to \{N, F, U\}$ is defined as $\gamma_{att}(w_a) = \gamma(w)$, where $w_a = \xi(w)$; the operator diagnosis function $\gamma_{opr} : \Sigma_a^* \to \{N, F, U\}$ is defined as $\gamma_{opr}(w_a) = \gamma(w')$, where $w' = \widehat{P}(w_a)$.

Definition 2. *A corrupting function $\phi : P(L(G)) \to \Sigma_o^*$ is defined as $\phi(w) = \widehat{P}(\xi(w))$, where $\xi : P(L(G)) \to \Sigma_a^*$ and $\widehat{P} : \Sigma_a^* \to \Sigma_o^*$ are the attack function and the operator mask, respectively.*

In this paper, a sensor network attacker is characterized by a corrupting function that takes an original observation w as input and produces a corrupted observation $w' = \phi(w) = \widehat{P}(\xi(w))$ as output, as shown in Figure 3. We define $L(\phi, G) = \widehat{P}(L(\xi, G))$, abbreviated as $L(\phi)$, as the corrupted language induced by function ϕ. The set of all possible corrupting functions on Σ_{era} and Σ_{ins} is defined as "$\Phi(\Sigma_{era}, \Sigma_{ins})$", abbreviated as Φ.

5. Problem Statement

In this section, we first state the fault diagnosis problem under attack as follows.

Problem 1. *(Online diagnosis under attack) Given an attack word $w_a \in L(\xi)$, determine if the diagnosis states based on the corresponding w and w' are consistent.*

In the worst-case scenario, a faulty observation w that allows for the detection of a fault (i.e., positive diagnosis state "F") can be corrupted into a normal or ambiguous observation corresponding to the absence of the fault (i.e., negative diagnosis state "N") or to an uncertain situation (i.e., uncertain diagnosis state "U"). This leads to the notion of a harmful attacker, defined as follows.

Definition 3. *An attacker ϕ for a language L is said to be harmful if there exists an observation $w \in P(L)$ with $\gamma(w) = \{F\}$, which may be corrupted into another observation $w' = \phi(w)$ and $\gamma(w') \in \{N, U\}$.*

Under such a harmful attacker (that receives w), the occurrence of a fault is hidden from the operator (that perceives w'); that is, a harmful attacker introduces a delay in the detection of the fault. Next, the stealthiness of an attacker is considered. To this end, we introduce an attack detection mechanism [20,22]. If an attacker can always keep its attacks undiscovered by the operator during system execution, we call it stealthy. We now give the condition that implies the stealthiness of an attacker.

Definition 4. *An attacker ϕ for a language L is stealthy if $L(\phi) \subseteq P(L(G))$.*

The stealthiness of an attacker is ensured when any corrupted observation received by the operator is contained in the observed language of G. Associated with Definition 4 are two sets of words $w \in \Sigma_a^*$ defined as follows. Given a plant G, the set of stealthy words on Σ_a is defined as $\mathcal{S}_s = \{w_a \in \Sigma_a^* \mid \widehat{P}(w_a) \in P(L(G))\}$, while the set of exposing

words on Σ_a is defined as $\mathcal{S}_e = \{w_a\sigma \in \Sigma_a^* \mid w_a \in \mathcal{S}_s, \sigma \in \Sigma_a, w_a\sigma \notin \mathcal{S}_s\}$. A stealthy word w_a produces a corrupted observation $w' = \widehat{P}(w_a) \in P(L(G))$, which does not reveal the attacker's presence. On the contrary, an exposing word results in the exposure of the attacker at the last observable step. Finally, the ideal attacker that can achieve both goals is said to be successful. Finally, from the attack viewpoint, we formalize the successful attacker existence and synthesis problem.

Problem 2. *(Existence and synthesis of a successful attacker) Given a language L, determine whether there exists a successful attacker ϕ for L. If it exists, synthesize the successful attacker ϕ.*

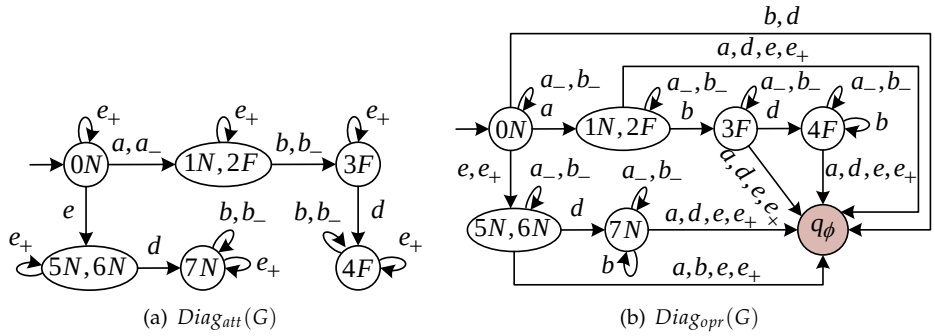
6. Stealthy Joint Diagnoser

6.1. Attacker Diagnoser and Operator Diagnoser

In our previous work [24], we detail the constructions of two special diagnosers, called Attacker Diagnoser and Operator Diagnoser. The former diagnoser describes all attack words w_a that can be generated under attack. The latter diagnoser describes how all words $w_a \in \Sigma_a^*$ are recognized by the operator. Starting from a diagnoser $Diag(G) = (Q_d, \Sigma_o, \delta_d, q_{d,0})$, we build the following:

- Attacker Diagnoser $Diag_{att}(G) = (Q_d, \Sigma_a, \delta_{att}, q_{d,0})$ through self-looping each state with all events in Σ_+ and then adding in parallel to each event $\sigma \in \Sigma_{era}$ the corresponding event $\sigma_- \in \Sigma_-$.
- Operator Diagnoser $Diag_{opr}(G) = (Q_d \cup q_\emptyset, \Sigma_a, \delta_{opr}, q_{d,0})$ through self-looping each state with all events in Σ_-, then adding in parallel to each event $\sigma \in \Sigma_{ins}$ the corresponding event $\sigma_+ \in \Sigma_+$, and finally adding a dump state $q_\emptyset$ that is reached by all undefined transitions.

Example 1. *Consider the plant G in Figure 2a. Let $\Sigma_{ins} = \{e\}$ and $\Sigma_{era} = \{a, b\}$. The corresponding $Diag_{att}(G)$ and $Diag_{opr}(G)$ are shown in Figure 4a and Figure 4b, respectively.*



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

Figure 4. (**a**) $Diag_{att}(G)$ and (**b**) $Diag_{opr}(G)$ for G in Figure 2a.

Proposition 1. *Given a nominal diagnoser $Diag(G) = (Q_d, \Sigma_o, \delta_d, q_{d,0})$, let $Diag_{att}(G) = (Q_d, \Sigma_a, \delta_{att}, q_{d,0})$ be the attacker diagnoser. It holds that*
(1) $w_a \in L(Diag_{att}(G)) \Leftrightarrow (\exists \xi \in \Xi)(\exists w \in L(Diag(G))\,[w_a = \xi(w)]$;
(2) $(\forall \xi \in \Xi)(\forall w \in L(Diag(G)))[\delta_{att}(q_{d,0}, \xi(w)) = \delta_d(q_{d,0}, w)]$.

Proof. Consider first Statement (1). ($\Leftarrow$) By the construction of $Diag_{att}(G)$, given a word $w \in L(Diag(G))$ and $\xi \in \Xi$, there exists a corresponding attack word $w_a \in L(Diag_{att}(G))$ such that $w_a = \xi(w)$. ($\Rightarrow$) Consider a word $w_a \in L(Diag_{att}(G))$ that reaches a state Q_d and only contains events in Σ, implying an original observation. At each state of $Diag_{att}(G)$, all events σ_+ are in a self-loop, which implies that w_a can be corrupted by inserting a word of fake events in Σ_{ins}. If the word w_a is generated by executing a transition $\delta_{att}(Q'_d, \sigma) = Q''_d$ with $\sigma \in \Sigma_{era}$, it may happen that the "parallel" transition $\delta_{att}(Q'_d, \sigma_-) = Q''_d$ is executed corresponding to an attack that erases σ. Therefore, there exists a function $\xi \in \Xi$ and an

observation $w \in L(Diag(G))$ such that $\xi(w) = w_a$. The proof of Statement (2) follows that of Statement (1). □

By Statement (1) in Proposition 1, the language of the attacker diagnoser consists of all possible attack words w. According to Statement (2), the diagnosis state estimation of $Diag_{att}(G)$ based on w is the same as that of $Diag(G)$ based on s, where $w_a = \xi(w)$.

Proposition 2. *Given a nominal diagnoser* $Diag(G) = (Q_d, \Sigma_o, \delta_d, q_{d,0})$, *let* $Diag_{opr}(G) = (Q_d \cup q_\emptyset, \Sigma_a, \delta_{opr}, q_{d,0})$ *be the operator diagnoser. It holds that*

(1) $L(Diag_{opr}(G)) = \mathcal{S}_s \cup \mathcal{S}_e$;

(2) $w_a \in \mathcal{S}_s \subseteq L(Diag_{opr}(G)) \Rightarrow (\exists w' \in L(Diag(G)))[w' = \widehat{P}(w_a)]$;

(3) $\forall w_a \in L(Diag_{opr}(G))$: *if* $w_a \in \mathcal{S}_s$, $\delta_{opr}(q_{d,0}, w_a) = \delta_d(q_{d,0}, \widehat{P}(w_a))$; *if* $w_a \in \mathcal{S}_e$, $\delta_{opr}(q_{d,0}, w_a) = q_\emptyset$.

Proof. Statement (1) is first considered. By construction the $Diag_{opr}(G)$, it includes all the words in $\mathcal{S}_s \cup \mathcal{S}_e$. The following is to prove that all the words $w_a \in L(Diag_{opr}(G))$ either belong to $\mathcal{S}_s$ or $\mathcal{S}_e$. Consider a word $w_a \in L(Diag_{opr}(G))$ that reaches a state $q_d \in Q_d$ and only contains events in Σ_o, implying an original observation received by the operator. At each state, all events σ_- are in a self-loop, which corresponds to the generation of w_a. By the definition of $\widehat{P}$, it holds that $\widehat{P}(w_a) \in P(L(G))$, i.e., $w_a \in \mathcal{S}_s$. If the word w_a is generated by executing a transition $\delta_{opr}(q'_d, \sigma_+) = q''_d$ with $\sigma \in \Sigma_{ins}$, it may happen that the "parallel" transition $\delta_{att}(q'_d, \sigma) = q''_d$ is executed and thus $\widehat{P}(w_a) \in P(L(G))$, i.e., $w \in \mathcal{S}_s$. Then, if the word w_a yields $d_\emptyset$, then $\widehat{P}(w_a) \notin P(L(G))$, i.e., $w_a \in \mathcal{S}_e$, which completes the proof of Statement (1). The proofs of Statement (2) and (3) follow that of Statement (1). □

By Statement (1) in Proposition 2, all words in $\mathcal{S}_s$ and $\mathcal{S}_e$ can be generated by $Diag_{opr}(G)$. By Statement (2), a word in $\mathcal{S}_s$ generated by $Diag_{opr}(G)$ can correspond to a corrupted observation $w' \in P(L(G))$ recognized by the operator. Statement (3) implies that (i) the diagnosis state estimation of $Diag_{opr}(G)$ based on a stealthy word $w_a \in \mathcal{S}_s$ is the same as that of $Diag(G)$ based on w', where $w' = \widehat{P}(w_a)$; (ii) all exposing words $w_a \in \mathcal{S}_e$ yield $q_\emptyset$.

6.2. Joint Diagnoser and Its Refining

Definition 5. *A joint diagnoser (JD for short) J-Diag*(G) *is defined as J-Diag*$(G) = (X, \Sigma_a, \delta_a, x_0) = Diag_{att}(G) \,\|\, Diag_{opr}(G)$.

As defined, every state of *J-Diag*(G) is a pair $x = (q_d, \overline{q}_d)$. The states of *J-Diag*$(G)$ can be partitioned into stealthy states and exposing states as follows.

Definition 6. *Given a joint diagnoser J-Diag*$(G) = (X, \Sigma_a, \delta_a, x_0)$, *the set of exposing states is defined as* $X_e = \{x = (q_d, \overline{q}_d) \in X \mid \overline{q}_d = q_\emptyset\}$; *the set of stealthy states is defined as* $X_s = X \backslash X_e$.

If an exposing word $w_a \in \mathcal{S}_e$ yields an exposing state, we conclude that the attacker that produces w_a is stealthy. However, if a stealthy word $w_a \in \mathcal{S}_s$ yields a stealthy state, it can only be inferred that the attacker producing w_a is currently undiscovered. There is a case where, following the future evolution of G, the attacker will be inevitably discovered. This leads to the notion of a weakly exposing region, denoted as $X_{we} \supseteq X_e$, which can be computed iteratively by a procedure in [21]. In the first iteration,

$$X_{we} := \{x \in X \mid (\exists \sigma \in \Sigma_o)[\delta_a(x, \sigma) \in X_e \Rightarrow \sigma \notin \Sigma_{era} \wedge (\forall \sigma' \in \Sigma_{ins})[\delta_a(x, \sigma') \in X_e]]\}. \quad (4)$$

The remaining iterations are executed similarly to Equation (4). We do not present the complete procedure here for the sake of brevity but illustrate it via Example 2. Dually, we define the strongly stealthy region as $X_{ss} = X \backslash X_{we} \subseteq X_s$.

Example 2. *The joint diagnoser for the preceding G is shown in Figure 5, where the exposing states in X_{we} are highlighted in brown while the stealthy states in X_{we} are highlighted in gray. Let us focus on a stealthy state $(\{3F\},\{1N,2F\})$. In the case that the occurrence of event d will take $(\{3F\},\{1N,2F\})$ to an exposing state $(\{4F\},\{q_\emptyset\})$, we find that i) d cannot be erased; ii) the inserted event $e \in \Sigma_{ins}$ will reach another exposing state $(\{3F\},\{q_\emptyset\})$. By Equation (4), $(\{3F\},\{1N,2F\}) \in X_{we}$ holds. In plain words, once such a stealthy state is reached, all attempts of an attacker to prevent it from reaching a subsequent exposing state will fail.*

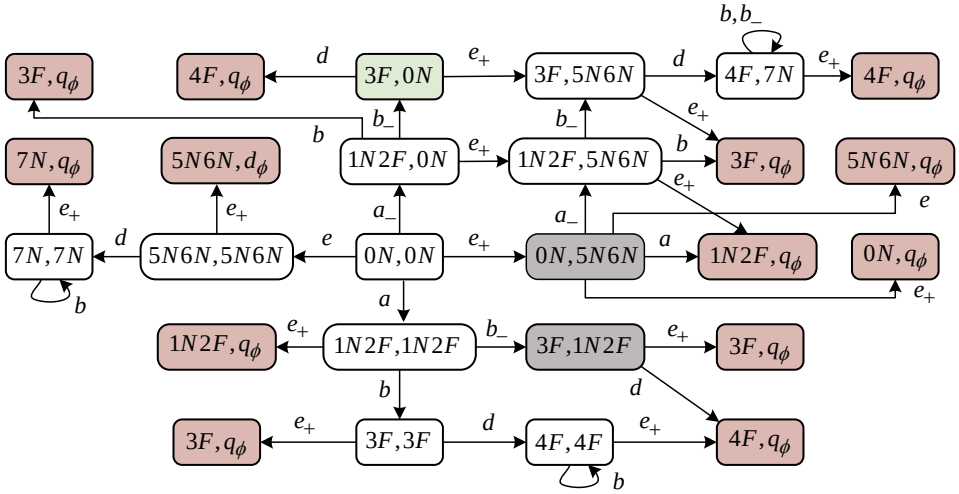


Figure 5. *J-Diag*(G) in Example 2.

Definition 7. *The stealthy joint diagnoser (SJD, for short) with respect to a joint diagnoser* $J\text{-}Diag(G) = (X,\Sigma_a,\delta_a,x_0)$ *is defined as* $SJ\text{-}Diag(G) = Ac(X_{ss},\Sigma_a,\delta_{sa},x_0)$, *where* $X_{ss} = X\backslash X_{we}$ *and* $\delta_{sa} = \delta_a|_{X_{ss}\times\Sigma_a\to X_{ss}}$.

The resulting SJD is obtained by removing all states in X_{we} from a JD and taking its accessible part. Then, we define a subset of the states of $SJ\text{-}Diag(G)$, called pre-empting states, which will be used in the procedure to extract an attack function from $SJ\text{-}Diag(G)$.

Definition 8. *Given a JD* $J\text{-}Diag(G) = (X,\Sigma_a,\delta_a,x_0)$ *with the corresponding SJD* $SJ\text{-}Diag(G) = (X_{ss},\Sigma_a,\delta_{sa},x_0)$, *the set of pre-empting states of* $SJ\text{-}Diag(G)$ *is defined as*

$$X_p = \{x_{ss} \in X_{ss} \mid (\exists\sigma \in \Sigma_o)\,[\delta_a(x_{ss},\sigma) \in X\backslash X_{ss} \wedge (\sigma \notin \Sigma_{era} \vee \delta_a(x_{ss},\sigma_-) \in X\backslash X_{ss})]\};$$

the set of non-pre-empting states of $SJ\text{-}Diag(G)$ *is defined as* $X_{np} = X_{ss}\backslash X_p$.

A state $x_{ss} \in X_{ss}$ is pre-empting if i) there exists an event $\sigma \in \Sigma_o$ whose occurrence (even if erased) causes $J\text{-}Diag(G)$ to lead outside X_{ss}; ii) however, σ can be definitely pre-empted by inserting an appropriate word of events in Σ_+ to reach a state in the strongly stealthy state. This state captures the races between inserted events and internal system events. Specifically, at a pre-empting state, the inserted word must be performed before the execution of event σ. In Example 2, state $(\{3F\},\{0N\})$ is a pre-empting state (highlighted in green), where event d must be pre-empted by inserting a fake event e_+ to reach a state $(\{3F\},\{5N,6N\}) \in X_{ss}$; otherwise, the occurrence of $d \in \Sigma_o\backslash\Sigma_{era}$ will take $(\{3F\},\{0N\})$ to an exposing state $(\{4F\},\{d_\emptyset\})$.

6.3. Synthesis of Attackers

In order to synthesize an attacker, the following definition is first required.

Definition 9. *Let* $x_{ss} \in X_{ss}$ *be a state and* X_{np} *be the set of non-pre-empting states in the SJD* $SJ\text{-}Diag(G) = (X_{ss},\Sigma_a,\delta_{sa},x_0)$.

(1) The set of events in Σ_a enabled at x_{ss} is defined as $\Gamma_{SJ}(x_{ss}) = \{\sigma \in \Sigma_a \mid \delta_{sa}(x_{ss}, \sigma)!\}$.
(2) The set of words consisting of events in Σ_+ that originate from x_{ss} and lead to a non-pre-empting state is defined as $L_+(SJ\text{-}Diag(G), x_{ss}) = \{w_{a,+} \in \Sigma_+^* \mid \delta_{sa}(x_{ss}, w_{a,+}) \in X_{np}\}$.

The following objective is to show how an attacker may determine an attack function from an SJD. In [21], an algorithmic procedure is presented for this purpose. The main idea is to select attack words that do not end in any state that has an inserted event as a successor transition, of course, including the pre-empting state. Compared with [21], it is now expected to synthesize the attack function producing an attack word that only does not end in pre-empting states. This means that an attacker cannot stay in the pre-empting state when determining its attack function. Now, we provide the following attacker synthesis algorithm, i.e., Algorithm 1.

Algorithm 1 Synthesis of the corrupting function ϕ

Require: $G = (Q, \Sigma, \delta, q_0)$ and $SJ\text{-}Diag(G) = (X_{ss}, \Sigma_a, \delta_{sa}, x_0)$
Ensure: corrupting function $\phi \in \Phi$

1: $\phi \leftarrow Synth(SJD)$
2: **procedure** SYNTH(SJD)
3: initialization: $w \leftarrow \varepsilon$
4: select a word $w_{a,+} \in L_+(SJ\text{-}Diag(G), x_0)$
5: $\xi(w) := w_{a,+}$; $\phi(w) := \widehat{P}(\xi(w))$
6: $x_{ss} := \delta_{sa}(x_0, w_{a,+})$
7: **while** a new observable event $\sigma \in \Sigma_o$ is produced by G **do**
8: $I := \emptyset$
9: **if** $\sigma \in \Gamma_{SJ}(x_{ss})$ **then** $I := I \cup \{\sigma\}$
10: **if** $\sigma_- \in \Gamma_{SJ}(x_{ss})$ **then** $I := I \cup \{\sigma_-\}$
11: select an event $\sigma' \in I$ and a word $w_{a,+} \in L_+(SJ\text{-}Diag(G), \delta_{sa}(x_{ss}, \sigma'))$
12: let $w_a := \sigma' w_{a,+}$ and $w'_a \in \{\sigma', w_a\}$
13: let $x_{ss} := \delta_{sa}(x_{ss}, \sigma')$
14: **if** $x_{ss} \in X_{np}$ **then** $\xi(w\sigma) \in \xi(w)w'_a$ and $x_{ss} := \delta_{sa}(x_{ss}, w'_a)$
15: **if** $x_{ss} \in X_p$ **then** $\xi(w\sigma) := \xi(w)w_a$ and $x_{ss} := \delta_{sa}(x_{ss}, w_a)$
16: $\phi(w\sigma) := \widehat{P}(\xi(w\sigma))$
17: $w \leftarrow w\sigma$
18: **end while**
19: **end procedure**

This algorithm may recursively associate each observation generated by G with a corresponding attack word. We now briefly explain how it works. When no event is generated by G, the attacker may insert a suitable word $w_{a,+}$ in the set $L_+(SJ\text{-}Diag(G), x_0)$ (Steps 3 and 4). In Step 5, $\xi(\varepsilon)$ is computed, and the new current state x_{ss} of $SJ\text{-}Diag(G)$ is updated. Then, we wait for G to generate a new observable event σ (Step 7). A new set I initialized at the empty set is defined. If σ is enabled at x_{ss}, σ is added to I. If σ_- is enabled at x_{ss}, σ_- is added to I (Steps 8 to 10). In Step 11, an event $\sigma' \in I$ and the insertion of a word $w_{a,+} \in L_+(SJ\text{-}Diag(G), \delta_{sa}(x_{ss}, \sigma'))$ are selected. If the current state x_{ss} with the occurrence of σ' is a non-pre-empting state, an attack word w'_a is produced, which is either σ' or the concatenation of σ' and $w_{a,+} \in L_+(SJ\text{-}Diag(G), \delta_{sa}(x_{ss}, \sigma'))$ (Step 14). Otherwise, another attack word w_a is produced as the concatenation of σ' and $w_{a,+}$ (Step 15). The corresponding corrupting function is computed in Step 16. Step 17 updates the observation w to $w\sigma$. This procedure goes to Step 7 when a new observable event is generated by G.

Proposition 3. *Given a plant G, function ϕ is stealthy if and only if, for all $w \in P(L(G))$, the corrupted observation $\phi(w)$ can be computed by Algorithm 1.*

According to the property of the SJD, the above results trivially hold, which means that the corrupting function synthesized by Algorithm 1 must be stealthy. Using the notion of pre-empting states, we provide a characterization of the SJD as follows.

Theorem 1. *Given a plant G, let $Diag(G) = (Q_d, \Sigma_o, \delta_d, q_{d,0})$ be the diagnoser and $SJ\text{-}Diag(G) = (X_{ss}, \Sigma_a, \delta_{sa}, x_0)$ be the SJD. The following implication holds:*

$$\begin{aligned}&(\exists\, \xi(w) \in \Sigma_a^*)\ [\delta_{sa}(x_0, \xi(w)) = x_{ss} = (q_d, \overline{q}_d) \wedge x_{ss} \in X_{np}]\\ \Longleftrightarrow\ &(\exists w \in P(L(G))), (\exists\, \xi \text{ synthesized by Algorithm 1})\\ &\qquad [q_d = \delta_d(q_{d,0}, w)\ \wedge\ \overline{q}_d = \delta_d(q_{d,0}, \widehat{P}(\xi(w))) = \delta_d(q_{d,0}, \phi(w))].\end{aligned}$$

Proof. (If) Assume that there exists an observation $w \in P(L(G))$ and an attack function ξ such that $q_d = \delta_d(q_{d,0}, w)$ and $\overline{q}_d = \delta_d(q_{d,0}, \widehat{P}(\xi(w)))$, where the attack function ξ is synthesized by Algorithm 1. By Proposition 3, the attack word $\xi(w)$ is a stealthy word, i.e., $\xi(w) \in \mathcal{S}_s$. By Propositions 1 and 2, we have $\delta_{att}(q_{d,0}, w_a) = \delta_d(q_{d,0}, w)$ and $\delta_{opr}(q_{d,0}, w_a) = \delta_d(q_{d,0}, \widehat{P}(w_a))$, i.e., $q_d = \delta_{att}(q_{d,0}, w_a)$ and $\overline{q}_d = \delta_{opr}(q_{d,0}, w_a)$. By $J\text{-}Diag(G) = Diag_{att}(G) \,\|\, Diag_{opr}(G)$, and by the definition of $SJ\text{-}Diag(G)$, we have $\delta_{sa}(x_0, w_a) = x_{ss} = (q_d, \overline{q}_d)$. Since $\xi(w)$ is computed by selecting w_a, which ends in x_{ss}, by Algorithm 1, $x_{ss} \in X_{np}$.

(Only if) Assume that there exists an attack word $\xi(w) \in \Sigma_a^*$ in $SJ\text{-}Diag(G)$ such that $\delta_{sa}(x_0, \xi(w)) = x_{ss} = (q_d, \overline{q}_d)$ and $x_{ss} \in X_{np}$. By $J\text{-}Diag(G) = Diag_{att}(G) \,\|\, Diag_{opr}(G)$, and by the definition of $SJ\text{-}Diag(G)$, it holds that $\delta_{att}(q_{d,0}, \xi(w)) = q_d$ and $\delta_{opr}(q_{d,0}, \xi(w)) = \overline{q}_d$. By Propositions 1 and 2, we have $\delta_{att}(q_{d,0}, \xi(w)) = \delta_d(q_{d,0}, w)$ and $\delta_{opr}(q_{d,0}, \xi(w)) = \delta_d(q_{d,0}, \widehat{P}(\xi(w)))$, i.e., $q_d = \delta_d(q_{d,0}, w)$ and $\overline{q}_d = \delta_d(q_{d,0}, \widehat{P}(\xi(w)))$. □

According to the above result, a state pair $x_{ss} = (q_d, \overline{q}_d)$ in the joint diagnoser reached by an attack word w_a represents the joint diagnosis state estimation, where q_d describes the original diagnosis state estimation of the attacker based on the original observation w, where $w_a = \xi(w)$, and $\overline{q}_d$ describes the corrupted diagnosis state estimation of the operator based on the corrupted observation $w' = \widehat{P}(w_a)$.

Finally, we conclude this section by the complexity analysis of the proposed approach. Let $Diag(G)$ be a nominal diagnoser with $|Q_d|$ states. By construction, the attacker diagnoser $Diag_{att}(G)$ has the same set of states of $Diag(G)$, and so does the operator diagnoser $Diag_{opr}(G)$ except for the dump state $d_\emptyset$. Since the JD $J\text{-}Diag(G)$ is the parallel composition of $Diag_{att}(G)$ and $Diag_{att}(G)$, its maximum number of states is $|Q_d| \times |Q_d + 1|$. Hence, the complexity of building an SJD is $O(|Q_d|^2)$, which is polynomial in the number of states of the nominal diagnoser. However, it is well known that the construction of the diagnoser is worst-case exponential in the number of states in the system. Hence, the overall computational complexity of an SJD is exponential in the number of states of G.

7. Fault Diagnoisis under Attack

We first address Problem 1, i.e., online fault diagnosis under attack. Similar to the nominal setting, we begin with the definition of a diagnosis function pair.

Definition 10. *Given an attack word w_a, a diagnosis pair function $r : \Sigma_a^* \to \{N, F, U\} \times \{N, F, U\}$ associating to $w_a \in \Sigma_a^*$ a diagnosis state pair is defined as $r(w_a) = (\gamma_{att}(w_a), \gamma_{opr}(w_a))$, where γ_{att} and γ_{opr} are the attacker and operator diagnosis functions, respectively.*

From the definitions of γ_{att} and γ_{opr}, it holds that $r(w_a) = (\gamma_{att}(w_a), \gamma_{opr}(w_a)) = (\gamma(w), \gamma(w'))$, where $w_a = \xi(w)$ and $w' = \widehat{P}(w_a)$. A more systematic way to compute the diagnosis pair function is by using an SJD. Let $x_{ss} = (q_d, \overline{q}_d) = \delta_{sa}(x_0, w_a)$. By Theorem 1, the SJD allows one to associate every state to a diagnosis state pair $r(x_{ss}) = r(w_a)$, i.e., $\gamma(q_d) = \gamma(w)$ and $\gamma(\overline{q}_d) = \gamma(w')$ are the diagnosis state of the attacker and operator, respectively. Therefore, Problem 1 can be solved by tracking the current state in SJD reached by the attack word w_a. The set of all diagnosis state pairs is defined as

$R = \{N, F, U\} \times \{N, F, U\}$, which can be partitioned into $R = R_c \cup R_w \cup R_h$, where $R_c = \{(N, N), (U, U), (F, F)\}$, $R_w = \{(N, U), (N, F), (U, N), (U, F)\}$, and $R_h = \{(F, N), (F, U)\}$.

Definition 11. *Let SJ-Diag(G) be an SJD. A state x_{ss} is correct if $r(x_{ss}) \in R_c$; wrong non-harmful if $r(x_{ss}) \in R_w$; harmful if $r(x_{ss}) \in R_h$. Denote the set of correct states, the set of wrong non-harmful states, and the set of harmful states by X_{sc}, X_{sw}, and X_{sh}, respectively.*

When the SJD is in a correct state, the operator correctly computes the diagnosis state regardless of the fact that an attack has occurred. When the SJD reaches a wrong non-harmful state, the operator computes a wrong diagnosis state from the corrupted observation due to an attack, which is inconsistent with the diagnosis state based on the original observation. Note that, in such a case, the fault diagnosis is manipulated due to the attack but does not pose a harmful danger. For example, there exists a special case where an attacker induces the operator to think that a fault has occurred, while the system is actually under nominal behavior. The false positive may make the operator decide to take some unnecessary actions, e.g., shut down the system and start it again, which may increase the operational expenses of operating a system.

Finally, harmful states of the SJD correspond to the detection of a fault based on the original observation, while no detection is based on corrupted observation. Its physical interpretation is that the attacker itself has already confirmed that the fault has occurred, but it induces the operator to be unable to claim the fault occurrence.

7.1. Verification of Successful Attackers

As discussed above, it is intuitive that the presence of a harmful state in the SJD is related to Problem 2, the existence of a successful attacker.

Theorem 2. *Given a plant G, there exists a successful attacker if and only if the SJD SJ-Diag(G) contains a harmful state, i.e., $X_{sh} \neq \emptyset$.*

Proof. (If) Suppose that *SJ-Diag*(G) contains a harmful state x_{ss} such that $r(x_{ss}) \in R_h$, where $x_{ss} = \delta_{sa}(x_0, w_a)$. By Theorem 1, associated with x_{ss} is an attack word w_a such that $r(w_a) = r(x_{ss}) \in R_h$. Hence, there exists an attacker ϕ that alters the observation w into w' such that $(\gamma(w), \gamma(w')) = (\gamma_{att}(w_a), \gamma_{opr}(w_a)) = r(w_a)$, i.e., $(\gamma(w), \gamma(w')) \in \{(F, N), (F, U)\}$. By Definition 3, the attacker ϕ is harmful. By the construction of *SJ-Diag*(G), ϕ must satisfy stealthiness. Hence, the attacker ϕ is successful.

(Only if) Suppose that there exists a successful attacker ϕ. From Definition 3, there exists an observation w such that w is corrupted into w' satisfying $(\gamma(w), \gamma(w')) \in \{(F, N), (F, U)\}$. By Theorem 1, there exists a reachable state x_{ss} in *SJ-Diag*(G) with the occurrence of w_a such that $r(x_{ss}) = (\gamma(q_d), \gamma(\overline{q}_d)) = (\gamma(w), \gamma(w')) \in \{(F, N), (F, U)\}$. □

Example 3. *Let us continue with Example 2. After refining, the SJD is shown in Figure 6. Let $w_{a,1} = a_e_+$ be an attack word that yields the wrong non-harmful state $(\{1N, 2F\}, \{5N, 6N\})$. This implies that the diagnosis state of the attacker based on $w = a$ is "U" while the diagnosis state of the operator based on $w' = e$ is "N". At this point, the attacker has doubted if the fault has occurred or not; however, the operator is certain that the fault has not occurred.*

Let the evolution continue. Another word $w_{a,2} = a_e_+b_d$ yields a harmful state $(\{4F\}, \{7N\})$ in SJ-Diag(G). By Theorem 2, there exists a successful attacker for G that produces the attack word $w_{a,2} = a_e_+b_d$. At this point, the attacker is certain that the fault has occurred based on $w = abd$; however, based on the corrupted observation $w' = ed$, the operator persists in its opinion that the fault has not occurred.

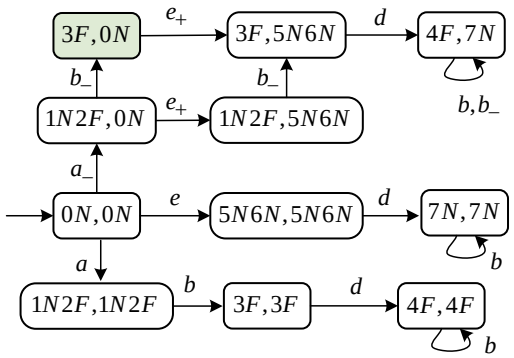


Figure 6. *SJ-Diag*(G) in Example 3.

7.2. Synthesis of Successful Attackers

We first provide the following proposition to ensure that a successful attacker can be synthesized from the SJD.

Proposition 4. *Let G be a plant and SJ-Diag$(G) = (X_{ss}, \Sigma_a, \delta_{sa}, x_0)$ be the SJD. A successful corrupting function ϕ can be computed if and only if $X_{sh} \cap X_{np} \neq \emptyset$.*

Proof. (If) Suppose that a state $x_{ss} \in X_{sh} \cap X_{np}$ in *SJ-Diag*(G) satisfies $x_{ss} = \delta_{sa}(x_0, w_a)$ and $w_a = \xi(w)$, where $w \in P(L(G))$. By $x_{ss} \in X_{np}$, the attack word w_a does not end in a pre-empting state. According to Algorithm 1 and Proposition 3, we see that ϕ is stealthy. Since $x_{ss} = (q_d, \overline{q}_d) \in X_{sh}$, by Theorem 1, we have $x_{ss} = \delta_{sa}(x_0, w_a) = (q_d, \overline{q}_d)$ such that $\delta_d(q_{d,0}, w) = q_d$ and $\delta_d(q_{d,0}, w') = \overline{q}_d$ with $w' = \phi(w)$, where the attacker ϕ alters the observation w into w' such that $(\gamma(w), \gamma(w')) = (\gamma_{att}(w_a), \gamma_{opr}(w_a)) = r(w_a) = r(x_{ss}) \in R_h$, i.e., $(\gamma(w), \gamma(w')) \in \{(F, N), (F, U)\}$. According to Definition 3, the corrupting function ϕ is also harmful.

(Only if) Suppose that ϕ is a successful corrupting function that can be computed. Since function ϕ is stealthy, by Algorithm 1 and Proposition 3, the attack word $w_a = \xi(w)$ does not end in a pre-empting state of *SJ-Diag*(G), i.e., $x_{ss} = \delta_{sa}(x_0, \xi(w)) \in X_{np}$. On the other hand, as function ϕ is harmful, according to Definition 3, there exists an observation $w \in P(L(G))$ that can be changed into an observation $w' = \phi(w) \in P(L(G))$, and $(\gamma(w), \gamma(w')) \in \{(F, N), (F, U)\}$. By $w, w' \in P(L(G))$, it holds that $\delta_d(q_{d,0}, w) = q_d$ and $\delta_d(q_{d,0}, w') = \overline{q}_d$. Since *SJ-Diag*$(G)$ contains all possible stealthy attacks, by Theorem 1, there must exist a state $x_{ss} = (q_d, \overline{q}_d)$ in *SJ-Diag*(G) with the occurrence of $w_a = \xi(w)$ such that $\delta_d(q_{d,0}, w) = q_d$ and $\delta_d(q_{d,0}, w') = \overline{q}_d$. By $r(x_{ss}) = r(w_a) = (\gamma_{att}(w_a), \gamma_{opr}(w_a)) = (\gamma(w), \gamma(w')) \in \{(F, N), (F, U)\}$, i.e., $r(x_{ss}) \in R_h$, we see that the state r_{ss} is harmful. It holds that $x_{ss} \in X_{sh} \cap X_{np}$, i.e., $X_{sh} \cap X_{np} \neq \emptyset$. □

This proposition implies that, from the synthesis viewpoint, only a successful attacker that determines an attack word yielding a non-pre-empting harmful state can make sense. We now give Algorithm 2 to synthesize a successful attacker from an SJD. In Algorithm 2, the condition $x_{ss} \in X_{sh} \cap X_{np}$ (Step 3) for selecting a state r guarantees the harmfulness of the stealthy corrupting function synthesized by Algorithm 1.

Example 4. *Following Algorithm 2, choose a harmful state $(4F, 7N) \in X_{sh} \cap X_{np}$. Associated with the state is an attack word $w_a = a_b_e_+d \in L(SJ\text{-}Diag(G))$. It corresponds to an observation $w = abd$. For all $w^* \in \overline{w} \cap P(L(G))$, the attack function ξ is computed by Algorithm 1—$\xi(\varepsilon) = \varepsilon$, $\xi(a) = a_$, $\xi(ab) = a_b_e_+$, and $\xi(abd) = a_b_e_+d$—and the corrupting function is synthesized accordingly: $\phi(\varepsilon) = \varepsilon$, $\phi(a) = \varepsilon$, $\phi(ab) = e$, and $\phi(abd) = ed$. The key point is that the attack word $a_b_$ is not selected in the synthesis procedure, i.e., it holds that $\xi(ab) = a_b_e_+$ rather than $\xi(ab) = a_b_$. This means that the attacker implements its attacks by first erasing the occurrence of event a, then erasing event b, and finally inserting e_+ while observing abd.*

Algorithm 2 Synthesis of the successful corrupting function ϕ

Require: $G = (Q, \Sigma, \delta, q_0)$ and $SJ\text{-}Diag(G) = (X_{ss}, \Sigma_a, \delta_{sa}, x_0)$
Ensure: successful corrupting function ϕ

1: $\phi \leftarrow Synthsuc(SJD)$
2: **procedure** SYNTHSUC(SJD)
3: **if** $X_{sh} \cap X_{np}$ **then** choose a harmful state $x_{ss} \in X_{sh} \cap X_{np}$ with $x_{ss} = \delta_{sa}(x_0, w_a)$
4: compute the observation w such that $w_a = \xi(w)$
5: **while** there is an observation $w^* \in \overline{w} \cap P(L(G))$ **do**
6: compute the corrupted observation $\phi(w^*)$ by Algorithm 1
7: **end while**
8: **end procedure**

8. Conclusions and Future Work

This paper investigated the problem of fault diagnosis under sensor network attacks. The notion of a successful attacker is proposed; it can degrade fault diagnosis without being discovered by the operator. We construct a joint diagnoser (JD) to describe all the behavior of the system under all possible sensor attacks. Then, the SD is refined to stealthy joint diagnoser (SJD), which includes stealthy attacks only. Based on it, a necessary and sufficient condition for the existence of a successful attacker is presented. Finally, we propose an algorithmic procedure to synthesize successful attacks from the SJD. In the future, we plan to extend the proposed diagnoser-based approach to diagnosability verification under attack. On the other hand, the decentralized setting should be further considered.

Author Contributions: T.K.: conceptualization, formal analysis, writing—original draft. Y.H.: formal analysis, methodology, writing and editing. D.L.: project administration, supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Complex Systems International Joint Research Center of Shaanxi Province and Xi'an Theory and Applications of Discrete Event Dynamic Systems International Science and Technology Cooperation Center.

Data Availability Statement: Enquiries about data availability should be directed to the authors.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DES	Discrete event system
CPS	Cyber–physical system
JD	Joint diagnoser
SJD	Stealthy joint diagnoser
G	Plant automaton
$Diag(G)$	Diagnoser
$Diag_{att}(G)$	Attacker diagnoser
$Diag_{opr}(G)$	Operator diagnoser
γ	Diagnosis function
γ_{att}	Attacker diagnosis function
γ_{opr}	Operator diagnosis function
r	Diagnosis pair function
P	Natural projection
ξ	Attack function
$\widehat{P}$	Operator mask

References

1. Chen, J.; Liu, B.; Li, T.; Hu, Y. Multiplicative Attacks with Essential Stealthiness in Sensor and Actuator Loops against Cyber-Physical Systems. *Sensors* **2023**, *23*, 1957. [CrossRef] [PubMed]
2. Suprabhath Koduru, S.; Machina, V.S.P.; Madichetty, S. Cyber attacks in cyber-physical microgrid systems: A comprehensive review. *Energies* **2023**, *16*, 4573. [CrossRef]

3. He, Z.; Tang, B.; Luan, F. An improved African vulture optimization algorithm for dual-resource constrained multi-objective flexible job shop scheduling problems. *Sensors* **2022**, *23*, 90. [CrossRef] [PubMed]
4. Yu, Z.; Gao, H.; Cong, X.; Wu, N.; Song, H.H. A Survey on Cyber-Physical Systems Security. *IEEE Internet Things J.* **2023**, *10*, 21670–21686. [CrossRef]
5. Goetz, C.; Humm, B. Decentralized real-time anomaly detection in cyber-physical production systems under industry constraints. *Sensors* **2023**, *23*, 4207. [CrossRef]
6. Cassandras, C.G.; Lafortune, S. *Introduction to Discrete Event Systems*; Springer: New York, NY, USA, 2019.
7. Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; Teneketzis, D. Diagnosability of discrete-event systems. *IEEE Trans. Autom. Control* **1995**, *40*, 1555–1575. [CrossRef]
8. Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; Teneketzis, D. Failure diagnosis using discrete event models. *IEEE Trans. Control Syst. Technol.* **1996**, *44*, 105–124. [CrossRef]
9. Lafortune, S.; Lin, F.; Hadjicostis, C.N. On the history of diagnosability and opacity in discrete event systems. *Annu. Rev. Control.* **2018**, *45*, 257–266. [CrossRef]
10. Basile, F.; Chiacchio, P.; De Tommasi, G. An efficient approach for online diagnosis of discrete event systems. *IEEE Trans. Autom. Control* **2009**, *54*, 748–759. [CrossRef]
11. Basile, F.; Cabasino, M.P.; Seatzu, C. Diagnosability analysis of labeled time Petri net systems. *IEEE Trans. Autom. Control* **2016**, *62*, 1384–1396. [CrossRef]
12. Cong, X.; Fanti, M.P.; Mangini, A.M.; Li, Z. Decentralized diagnosis by Petri nets and integer linear programming. *IEEE Trans. Syst. Man Cybern.* **2017**, *48*, 1689–1700. [CrossRef]
13. Yu, Z.; Qi, Y.; Cong, X. Decentralized Marking Fault Diagnosis of Labeled Petri Nets. *IEEE Access* **2023**, *11*, 99168–99177. [CrossRef]
14. Zaytoon, J.; Lafortune, S. Overview of fault diagnosis methods for discrete event systems. *Annu. Rev. Control.* **2013**, *37*, 308–320. [CrossRef]
15. Carvalho, L.K.; Basilio, J.C.; Moreira, M.V. Robust diagnosis of discrete event systems against intermittent loss of observations. *Automatica* **2012**, *48*, 2068–2078. [CrossRef]
16. Carvalho, L.K.; Moreira, M.V.; Basilio, J.C.; Lafortune, S. Robust diagnosis of discrete-event systems against permanent loss of observations. *Automatica* **2013**, *49*, 223–231. [CrossRef]
17. Takai, S. A general framework for diagnosis of discrete event systems subject to sensor failures. *Automatica* **2021**, *129*, 109669. [CrossRef]
18. Carvalho, L.K.; Wu, Y.C.; Kwong, R.; Lafortune, S. Detection and mitigation of classes of attacks in supervisory control systems. *Automatica* **2018**, *97*, 121–133. [CrossRef]
19. Yu, Z.; Duan, X.; Cong, X.; Li, X.; Zheng, L. Detection of actuator enablement attacks by Petri nets in supervisory control systems. *Mathematics* **2023**, *11*, 943. [CrossRef]
20. Zhang, Q.; Seatzu, C.; Li, Z.; Giua, A. Joint state estimation under attack of discrete event systems. *IEEE Access* **2021**, *9*, 168068–168079. [CrossRef]
21. Zhang, Q.; Seatzu, C.; Li, Z.; Giua, A. Selection of a successful attack function in discrete event systems. *Sci. Rep.* **2022**, *12*, 16302. [CrossRef]
22. Meira-Góes, R.; Kang, E.; Kwong, R.H.; Lafortune, S. Synthesis of sensor deception attacks at the supervisory layer of Cyber-Physical Systems. *Automatica* **2020**, *121*, 109172. [CrossRef]
23. Meira-Góes, R.; Marchand, H.; Lafortune, S. Synthesis of supervisors robust against sensor deception attacks. *IEEE Trans. Autom. Control* **2021**, *66*, 4990–4997. [CrossRef]
24. Kang T.; Seatzu, C.; Li, Z.; Giua, A. Fault Diagnosis of Discrete Event Systems Under Attack. In Proceedings of the 2023 62nd IEEE Conference on Decision and Control (CDC), Singapore, 13–15 December 2023; pp. 7923–7929.
25. Li, Y.; Hadjicostis, C.N.; Wu, N. Tamper-tolerant diagnosability under bounded or unbounded attacks. *IFAC-Paper* **2022**, *55*, 52–57. [CrossRef]
26. Hadjicostis, C.N.; Lafortune, S.; Lin, F.; Su, R. Cybersecurity and Supervisory Control: A Tutorial on Robust State Estimation, Attack Synthesis, and Resilient Control. In Proceedings of the 2022 IEEE 61st Conference on Decision and Control (CDC), Cancún, Mexico, 6–9 December 2022; pp. 3020–3040.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.