

Optimal Attack Strategy Compromising Diagnosability of Automated Manufacturing Systems in Labeled Petri Nets

Ruotian Liu, Agostino Marcello Mangini, Maria Pia Fanti

Abstract—This paper addresses the diagnosability analysis problem under malicious attacks of a discrete event system modeled by labeled Petri net. We focus on a stealthy replacement attack to alter or corrupt the observation of the system. The aim of this work is, from an attacker viewpoint, to design a stealthy replacement attack for violating the diagnosability of system. To this end, we first build a new structure, called attack verifier that is used to enumerate all the attack paths. Then an optimal attack synthesis problem in terms of minimum energy cost is formulated by determining whether a bad path is generated via solving a set of integer linear programming problems. Finally, an automated manufacturing system is provided to illustrate the proposed attack strategy.

I. INTRODUCTION

Fault diagnosis is critical for highly automated manufacturing systems (AMSs). The attack scenario of AMSs have attracted much attention in the community of discrete event systems (DESSs): mainly concerning the problems of attack detection and attack synthesis [1], [2]. This work focuses on the attack synthesis problem with optimization objectives for violating the diagnosability in DESSs. In this work, we consider the permanent attack [3] (i.e., once an attack is performed and each transition is either associated with its original or a replaced label, then all the transition labels remain unchanged) rather than intermittent setting [4] (i.e., the attacked label can be recovered in finite time).

Diagnosability is a system property determining whether the occurrence of a fault can be detected within finite steps, and has been studied in the framework of automata [5] and Petri nets [6]. Although recent works focus on diagnosability and fault diagnosis in the presence of attacks, they primarily deal with detecting or tolerating attacks rather than synthesizing attacks to violate diagnosability, which is the focus of our study. For example, the violation of safety caused by malicious interception and alteration of sensor readings is considered in [7]. Moreover, the work [8] investigates how an attacker with asymmetric observations can strategically corrupt outputs to maximize the operator's diagnostic uncertainty. In addition, the violation of opacity is studied in [9]. The studies [3], [4] works on attack-synthesis problems in DESSs that aim to hide the occurrence of faults.

This work is a part of the IN2CCAM project. This project has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No. 101076791. This content reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information this publication contains.

Ruotian Liu, Agostino Marcello Mangini and Maria Pia Fanti are with the Department of Electrical and Information Engineering, Polytechnic University of Bari, Italy (e-mails: ruotian.liu@poliba.it, agostinomangini@poliba.it, mariapia.fanti@poliba.it).

To tackle the attack synthesis problem, there are mainly several approaches in the literature. The first type of approaches [2] employs a discrete structure to model the game-like interaction between the supervisor and the attacker, which incorporates all possible attacks; the second one transforms the attack synthesis problem into the supervisor synthesis problem [1]. Different from the existing approaches, we introduce a modified labeling function to model an attack, for making a system non-diagnosable by leveraging techniques of unfolded verifier structure in labeled Petri nets.

The main contribution of this work is listed as follows. First, we formulate an optimal stealthy replacement attack problem in terms of minimum energy cost, which is determined by solving a set of integer programming problems. Since the classic verifier [10] can only update states when consecutive transition pairs have the same observation, it is unable to generate bad paths. To solve these issues, we then develop a new structure called attack verifier by integrating the attack capability. Finally, an illustrative example on automated manufacturing system is provided to show the proposed attack strategy.

II. PRELIMINARIES

A. Basic definitions of Petri net

Let \mathbb{N} be the set of non-negative integers. A Petri net is defined as a four-tuple $N = (P, T, Pre, Post)$, where $P = \{p_1, \dots, p_m\}$ is a set of $m \in \mathbb{N}$ places, $T = \{t_1, \dots, t_n\}$ is a set of $n \in \mathbb{N}$ transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ are the *pre*- and *post-incidence* matrices, respectively, denoting the weight of the arcs from places to transitions and transitions to places. The incidence matrix of a net is defined by $C = Post - Pre$. A Petri net is said to be *acyclic* if there is no directed cycle.

A marking is a mapping $M : P \rightarrow \mathbb{N}$ that assigns to a place of a Petri net a non-negative integer of tokens. $M(p_i)$ is the number of tokens in place p_i at a marking M . A net system $\langle N, M_0 \rangle$ is a net N with an initial marking M_0 . A transition $t \in T$ is enabled at a marking M if $M \geq Pre(\cdot, t)$ and may fire yielding a marking $M' = M + C(\cdot, t)$. We write $M[\sigma]$ to denote that a transition sequence $\sigma = t_1 t_2 \dots t_i \in T^*$ is enabled at M , and $M[\sigma]M'$ to denote that the firing of σ yields M' . The Parikh vector of σ is denoted by $\pi(\sigma) : T \rightarrow \mathbb{N}^n$ and maps a transition $t \in T$ to the number of occurrences of t in σ .

A marking M is reachable in $\langle N, M_0 \rangle$ if there exists a firing sequence $\sigma \in T^*$ such that $M_0[\sigma]M$. The set of all markings reachable from M_0 , denoted by $R(N, M_0)$, defines

the reachability set of $\langle N, M_0 \rangle$, i.e., $R(N, M_0) = \{M \in \mathbb{N}^m \mid M_0[\sigma]M\}$. The set of transition sequences enabled at the initial marking M_0 is defined as $L(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma]\}$. Given a set $H \subseteq L(N, M_0)$, we denote H/σ the post transition sequence of H after σ , i.e., $H/\sigma = \{\sigma' \in T^* \mid \sigma\sigma' \in H\}$. A net system $\langle N, M_0 \rangle$ is said to be: *bounded* if it exists an integer $k \in \mathbb{N}$ such that for all $M \in R(N, M_0)$ and for all $p_i \in P$, $M(p_i) \leq k$ holds; *deadlock-free* if for all $M \in R(N, M_0)$, there exists $t \in T$, $M[t]$.

B. Labeled Petri net

Given a Petri net $N = (P, T, Pre, Post)$ and an event set Σ , a labeling function $l : T \rightarrow \Sigma \cup \{\varepsilon\} = \Sigma_\varepsilon$ assigns to a transition either a symbol from the event set Σ or the empty string symbol ε . A labeled Petri net (LPN) system $S = \langle N, \Sigma, l, M_0 \rangle$ is a Petri net system $\langle N, M_0 \rangle$ with a labeling function l and an event set Σ . A transition t is said to be unobservable if it is associated with the empty string ε , i.e., $l(t) = \varepsilon$. The set of unobservable transitions is denoted by $T_u = \{t \in T \mid l(t) = \varepsilon\}$. The other transitions labeled with events from Σ are called observable transitions, denoted as $T_o = \{t \in T \mid l(t) \in \Sigma\}$. Furthermore, the set T_u can be divided into two disjoint sets T_f and T_{reg} with $T_u = T_f \cup T_{reg}$, where T_f and T_{reg} denote the sets of fault transitions and regular unobservable transitions, respectively.

The labeling function can be extended to a transition sequence $\sigma = t_1 t_2 \dots t_i$ such that $\omega = l(\sigma) = l(t_1)l(t_2) \dots l(t_i)$, which is called an *observation* corresponding to the sequence σ . Given a net system $\langle N, \Sigma, l, M_0 \rangle$, we define $l^{-1}(\omega)$ as the set of all transition sequences consistent with $\omega \in \Sigma_\varepsilon^*$, i.e., $l^{-1}(\omega) = \{\sigma \in L(N, M_0) \mid l(\sigma) = \omega\}$. The language generated by an LPN system S is defined as $\mathcal{L}(N, M_0) = \{\omega \in \Sigma_\varepsilon^* \mid \exists \sigma \in L(N, M_0) : \omega = l(\sigma)\}$.

C. Extended basis reachability graphs

In this part, we recall necessary notions of the extended basis markings in [6], [11], where the observable transition set is assumed to be $T_o^\alpha = T_o \cup T_f$. Given a transition sequence $\sigma \in T^*$, we denote by $P_{reg}(\sigma)$ (resp., $P_o^\alpha(\sigma)$) the projection of σ over T_{reg} (resp., T_o^α). Moreover, the restriction of incidence matrix C of an LPN system to T_{reg} is denoted by C_{reg} .

Definition 2.1 ([6]): Given a marking $M \in R(N, M_0)$ and a transition $t \in T_o^\alpha$ of an LPN system $S = \langle N, \Sigma, l, M_0 \rangle$, the set of explanations of t at M is defined by $\Sigma(M, t) = \{\sigma \in T_{reg}^* \mid M[\sigma]M', M'[t]\}$, and the set of their e -vector is denoted as $Y(M, t) = \{\pi(\sigma) \mid \sigma \in \Sigma(M, t)\}$. In addition, the set of minimal explanations of t at M is denoted by $\Sigma_{\min}(M, t) = \{\sigma \in \Sigma(M, t) \mid \nexists \sigma' \in \Sigma(M, t) : \pi(\sigma') \leq \pi(\sigma)\}$ and the minimal e -vector is defined as $Y_{\min}(M, t) = \{\pi(\sigma) \mid \sigma \in \Sigma_{\min}(M, t)\}$.

The set of *extended basis markings*, denoted as X_e , is recursively computed as follows: $M_0 \in X_e$; If $M \in X_e$, then for each $t \in T_o \cup T_f$, $y = \pi(\sigma) \in Y_{\min}(M, t)$, $(M' = M + C_{reg} \cdot y + C(\cdot, t)) \Rightarrow (M' \in X_e)$. The *extended basis reachability graph (EBRG)* of S is a nondeterministic finite state automaton $G_e = (X_e, E, \delta, M_0)$, where X_e is

the set of states; $E \subseteq (T_o \times \Sigma) \cup (T_f \times \{\varepsilon\})$ is the set of event labels; $\delta \subseteq X_e \times E \times X_e$ is the transition relation; and M_0 is the initial state. A *nonfailure EBRG*, denoted by $G_{e,n} = (X_{e,n}, E_n, \delta_n, M_{0,n})$, is the EBRG derived from $\langle N', \Sigma, l', M_0 \rangle$ following the assumption that the set of observable transitions is equal to T_o .

III. DIAGNOSABILITY ANALYSIS PROBLEM WITH ATTACKS

A. Replacement and stealthy attack

To characterize the capability of an intruder that masks the transition labels, an attack structure is presented as follows.

Definition 3.1 (Attack structure): Let $S = \langle N, \Sigma, l, M_0 \rangle$ be an LPN system. An attack structure is defined as the set $\mathcal{A} \subseteq 2^{(T_o \times \Sigma_\varepsilon) \times (T_o \times \Sigma_\varepsilon)}$, i.e., \mathcal{A} is a set of transition pairs, each associated to its label: the first transition is associated to the original label while the second one is associated to the replaced label.

To make the exposition clear, we denote by $T_a = \{t \in T_o \mid \exists e' \in \Sigma_\varepsilon, ((t, e), (t, e')) \in \mathcal{A}\}$ the set of *attacked transitions* that can be targeted with respect to \mathcal{A} , and denote by $\Sigma_{\mathcal{A}}(t) = \{e' \in \Sigma_\varepsilon \mid ((t, e), (t, e')) \in \mathcal{A}, t \in T_a, e = l(t)\}$ the set of replaced labels associated with transition t while taking into account the attack structure \mathcal{A} . We denote by $l_{\mathcal{A}}(t) = \{e \mid l(t) = e\} \cup \{e' \mid e' \in \Sigma_{\mathcal{A}}(t)\}$.

Definition 3.2 (Replacement attack): Let $S = \langle N, \Sigma, l, M_0 \rangle$ be an LPN system and \mathcal{A} be an attack structure. A replacement attack A (attack for short) is a modified labeling function that is the mapping $l_a : T^* \rightarrow \Sigma_\varepsilon^*$ where

- $l_a(\varepsilon) = \varepsilon$,
- $l_a(t) = \begin{cases} l(t) & \text{if } t \in T \setminus T_a, \\ l(t) \text{ or } e' \in \Sigma_{\mathcal{A}}(t) & \text{if } t \in T_a, \end{cases}$
- $l_a(\sigma t) = l_a(\sigma)l_a(t), \sigma \in T^*, t \in T$.

Under the attack A , each transition $t \in T_a$ is either associated with the original label $l(t)$ or a replaced label $e' \in \Sigma_{\mathcal{A}}(t)$. Subsequently, the given attack structure \mathcal{A} may cause multiple attack options. The considered replacement attack includes a particular removal case, such as when an observable transition is associated with an empty string.

Definition 3.3 (Stealthy attack): Given an LPN system $S = \langle N, \Sigma, l, M_0 \rangle$ under an attack A_i , the attack A_i is said to be *stealthy* if for any transition sequence σ , its corrupted observations are contained in the language of LPN system, i.e., $\forall \sigma \in L(N, M_0), l_{a_i}(\sigma) \in \mathcal{L}(N, M_0)$.

Precisely, stealthiness requires that the set of corrupted observations is contained in the set of observations without attacks.

B. Diagnosability

The fault transition set T_f can be partitioned into r classes T_f^i , where $i = 1, \dots, r$. For the sake of simplicity, this work considers an LPN with a single fault class, i.e., $T_f = T_f^1$. Nevertheless, the proposed approach could be extended to the nets with multiple fault classes with a slight modification of the method proposed in [5] for this purpose.

Let T' be a subset of T . We define $\psi(T') = \{\sigma t \in L(N, M_0) \mid \sigma \in T^*, t \in T'\}$ as the set of firing sequences in $L(N, M_0)$ that end with a transition $t \in T'$.

Definition 3.4 ([6]): An LPN system $S = \langle N, \Sigma, l, M_0 \rangle$ having no deadlock after the occurrence of a fault $t_f \in T_f$, is diagnosable with respect to the fault transition set T_f if

$$\forall \sigma' \in \psi(T_f), \exists K \in \mathbb{N}, \forall \sigma'' \in L(N, M_0)/\sigma',$$

$$|\sigma''| \geq K \implies \forall \sigma \in l^{-1}(l(\sigma'\sigma'')), \exists t_f \in T_f : t_f \in \sigma.$$

C. Problem statement

Before formulating the addressed problem, we first present a notion of optimality criterion, i.e., minimum cost. Specifically, each transition is associated with a replacement attack cost, i.e., a non-negative real value, which describes the difficulty of attack for replacing its transition label. The higher the value of the cost of the attack is, the more difficult the attack on the transition is, otherwise it is easier. In the rest of this work, we refer the optimal attack to the minimum cost one, and focus on the following problem:

Problem 1: Given an LPN system $S = \langle N, \Sigma, l, M_0 \rangle$ that is vulnerable to an attack structure \mathcal{A} , the objective is to design an optimal stealthy replacement attack A_i for violating the diagnosability in the LPN system.

The following assumptions hold for the diagnosability analysis under the attacks in the LPN systems.

A1) The LPN system is bounded and deadlock-free after the occurrence of a fault.

A2) The T_u -induced subnet is acyclic.

A3) There exists a nonempty set of predetermined stealthy attacks $\mathcal{A}_s = \{A_{s_1}, \dots, A_{s_\theta}\}$, with $1 \leq \theta \leq \Pi_{t \in T_a} [(|\Sigma_A| + 1)] - 1$ based on the attack structure \mathcal{A} .

IV. OPTIMAL ATTACK AGAINST DIAGNOSABILITY

In this section, we first present an attack verifier that lists all the attack paths to be transformed into bad paths leading to the violation of diagnosability. Then, an optimal attack can be obtained by solving a set of linear programming problems.

A. Attack Verifier

An attack verifier $U_a = (X_a^U, E_a^U, \delta_a^U, M_0^U)$ is a finite state automaton, that shows all the possible attacked paths, presented in Algorithm 1.

Step 1 initializes the set of states, transitions, events, and the initial state in the attack verifier. The main part in lines 2–15, at each untagged state, iteratively generates all the other states. In this process, for each transition pair (t_i, t_j) , where $t_i \in T_o \cup T_f$ and $t_j \in T_o$, the verifier adds a new state $(M'_1, \alpha'; M'_2)$ if the transitions $(M_1, t_i, M'_1) \in \delta$, $(M_2, t_j, M'_2) \in \delta_n$ and non-empty set $l_{\mathcal{A}}(t_1) \cap l_{\mathcal{A}}(t_2)$ hold from the current state $(M_1, \alpha; M_2)$. By accounting for transitions with the same observation under attack, the proposed verifier captures attack paths that may exploit differences in observation, which classical verifiers would ignore. Note that α can be either N to represent the normal behavior of system without the occurrence of fault from the initial state to this one, or F to denote the occurrence of faulty behavior of system.

Algorithm 1: Construction of an attack verifier

Input: EBRGs $G_e = (X_e, E, \delta, M_0)$, $G_{e,n} = (X_{e,n}, E_n, \delta_n, M_{0,n})$, and attack structure \mathcal{A}

Output: An attack verifier $U_a = (X_a^U, E_a^U, \delta_a^U, M_0^U)$

- 1 Let $X_a^U = \{(M_0, N; M_0)\}$, $E_a^U = \emptyset$, $\delta_a^U = \emptyset$, $M_0^U = (M_0, N; M_0)$ be the initial state ;
- 2 **while** states with no tag exist **do**
- 3 select a state $(M_1, \alpha; M_2)$ with no tag ;
- 4 **if** $(M_1, \alpha; M_2)$ is same as a state in the path from M_0^U to it **then**
- 5 tag it “duplicate” and go to Step 2;
- 6 **for all** $t_1 \in T_o \cup T_f$ and $t_2 \in T_o$, **do**
- 7 **if** $(M_1, t_1, M'_1) \in \delta$, $(M_2, t_2, M'_2) \in \delta_n$, $l_{\mathcal{A}}(t_1) \cap l_{\mathcal{A}}(t_2) \neq \emptyset$ **then**
- 8 add a state $(M'_1, \alpha; M'_2)$ and a transition (t_1, t_2) from $(M_1, \alpha; M_2)$ to $(M'_1, \alpha; M'_2)$;
- 9 **if** $t_1 \in T_f$, $(M_1, t_1, M'_1) \in \delta$ **then**
- 10 add a state $(M'_1, F; M_2)$ and a transition (t_1, λ) from $(M_1, \alpha; M_2)$ to $(M'_1, F; M_2)$;
- 11 **if** $t_1 \in T_o$, $(M_1, t_1, M'_1) \in \delta$, $\varepsilon \in l_{\mathcal{A}}(t_1)$ **then**
- 12 add a state $(M'_1, \alpha; M_2)$ and a transition (t_1, λ) from $(M_1, \alpha; M_2)$ to $(M'_1, \alpha; M_2)$;
- 13 **if** $t_2 \in T_o$, $(M_2, t_2, M'_2) \in \delta_n$, $\varepsilon \in l_{\mathcal{A}}(t_2)$ **then**
- 14 add a state $(M_1, \alpha; M'_2)$ and a transition (λ, t_2) from $(M_1, \alpha; M_2)$ to $(M_1, \alpha; M'_2)$;
- 15 tag the state $(M_1, \alpha; M_2)$ “old”.

The following definition presents the notion of bad path that leads to the violation of diagnosability. Precisely, the existence of this type of path shows that, two arbitrarily long transition sequences of LPN system have the same observation under attack and one of them contains the fault transition, such that the occurrence of the fault cannot be detected in a finite number of steps. Given an automaton G , we write $M \xrightarrow[\sigma]{\sigma'} M'$ to denote that M' is reached in G from M with a sequence σ .

Definition 4.1: Let $S = \langle N, \Sigma, l, M_0 \rangle$ be an LPN system, X_e be the set of extended basis markings, and U_a be an attack verifier constructed by its two EBRGs G_e and $G_{e,n}$. A path $\tilde{\sigma} = (\gamma_{i_1}, \gamma_{j_1})(\gamma_{i_2}, \gamma_{j_2}) \dots (\gamma_{i_k}, \gamma_{j_k})$ in U_a is called a bad path if, by letting $\sigma_\alpha = \gamma_{i_1} \dots \gamma_{i_q}$, $\sigma_\beta = \gamma_{i_{q+1}} \dots \gamma_{i_k}$, $\sigma'_\alpha = \gamma_{j_1} \dots \gamma_{j_q}$, and $\sigma'_\beta = \gamma_{j_{q+1}} \dots \gamma_{j_k}$, there exist $M, M' \in X_e$ and an attack A_i corresponding to the modified labeling function l_{a_i} satisfying:

- (1) $M_0 \xrightarrow[\sigma_\alpha]{\sigma_\alpha} M \xrightarrow[\sigma_\beta]{\sigma_\beta} M$;
- (2) $M_0 \xrightarrow[\sigma_\alpha]{\sigma'_\alpha} M' \xrightarrow[\sigma_\beta]{\sigma'_\beta} M'$;
- (3) $l_{a_i}(\sigma_\alpha) = l_{a_i}(\sigma'_\alpha)$ and $l_{a_i}(\sigma_\beta) = l_{a_i}(\sigma'_\beta)$;
- (4) $t_f \in \sigma_\alpha \sigma_\beta$;

(5) no prefix of $\tilde{\sigma}$ satisfies items (1)–(4).

Proposition 4.2: An LPN system $\langle N, \Sigma, l, M_0 \rangle$ satisfying (A1)–(A2) is diagnosable if and only if its attack verifier U_a has no bad paths.

Proof: Following Definition 4.1 and a result proved in [10], [12] that the LPN system is diagnosable if and only if its classic unfolded verifier has no such a path, it could easily extend into our result. ■

Using Algorithm 1, we enumerate all paths but focus only on those that can be attacked into a bad path that violates the diagnosability of the system. The following definition presents such an attack path.

Definition 4.3: A path $\tilde{\sigma} = (\gamma_{i_1}, \gamma_{j_1})(\gamma_{i_2}, \gamma_{j_2}) \cdots (\gamma_{i_k}, \gamma_{j_k})$ in U_a is called attack path if $\sigma_\alpha = \gamma_{i_1} \cdots \gamma_{i_q}$, $\sigma_\beta = \gamma_{j_{q+1}} \cdots \gamma_{i_k}$, $\sigma'_\alpha = \gamma_{j_1} \cdots \gamma_{j_q}$, and $\sigma'_\beta = \gamma_{j_{q+1}} \cdots \gamma_{j_k}$, there exist $M, M' \in X_e$, such that (i) the conditions (1)(2)(4)(5) in Definition 4.1 hold and (ii) an attacked transition $t \in T_a$ exists such that $t \in \sigma_\alpha \sigma_\beta$ or $t \in \sigma'_\alpha \sigma'_\beta$.

B. Optimal attack determination

Before formally describing the attack strategy, we present some necessary notions of minimum cost attack for violating the diagnosability. We denote by the row vector $\mathbf{c} = [c_1, \dots, c_j, \dots, c_{|T|}]$ the attack cost coefficient vector, that associates a non-negative real value c_j to each possible attacked transition $t_j \in T_o$, and assigns $c_j = 0$ to each transition $t_j \in T_u$, where $c_j = 0$ means that transition t_j cannot be attacked. Now, given an attack path $\tilde{\sigma}_k = (\sigma_{k,1}, \sigma_{k,2})$, in order to select the possible attacks for violating the diagnosability, we define a vector $\mathbf{v}_k = [v_{t_1,k}, \dots, v_{t_j,k}, \dots, v_{t_{|T|},k}]^T$. In particular, $v_{t_j,k} \in \{0, 1\}$ for $j = 1, \dots, |T|$ is a binary decision variable and $v_{t_j,k} = 1$ (resp., $v_{t_j,k} = 0$) means that the transition label of t_j has (resp., has not) been replaced under the replacement attack A . Moreover, it holds $v_{t_j,k} = 0$ for the set of transitions $T \setminus T_a$ that cannot be attacked.

For the sake of clarity, we denote by e_j and e'_j the original label of transition and one replaced label of transition t_j , i.e., $e_j = l(t_j)$ and $e'_j \in \Sigma_A(t_j)$, respectively. Based on the aforementioned notions, given a transition t_j and its corresponding transition label pair (e_j, e'_j) , the corrupted observation of transition t_j can be expressed as $v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k}) \cdot e_j$. Precisely, in the case that $v_{t_j,k} = 1$ has the corrupted observation $1 \cdot e'_j = e'_j$, while $v_{t_j,k} = 0$ holds its original observation with $(1 - 0)e_j = e_j$. When the sequence σ is under an attack A_i , we can derive its compromised observation using the following expression

$$l_{a_i}(\sigma) = \prod_{j=1}^{|\sigma|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j].$$

Definition 4.4 (Corrupted option): Given an LPN system S vulnerable to an attack structure \mathcal{A} , a corrupted option \mathcal{C} is defined as: each transition $t_j \in T_a$ is associated with a corrupted label $e'_j \in \Sigma_A(t_j)$, in which the amount of corrupted options is equal to $\prod_{t \in T_a} |\Sigma_A(t)|$. And the set of corrupted options is denoted by $\mathcal{C}(\mathcal{A}) = \{\mathcal{C}_i \mid i = 1, \dots, \prod_{t \in T_a} |\Sigma_A(t)|\}$.

Given an attack path and a corrupted option, the following lemma characterizes the possible attacks A_i for violating the diagnosability.

Lemma 4.5: Let us consider the k -th attack path $\tilde{\sigma}_k = (\sigma_{k,1}, \sigma_{k,2})$ with $\sigma_{k,1} = \sigma_{\alpha,k} \sigma_{\beta,k}$ and $\sigma_{k,2} = \sigma'_{\alpha,k} \sigma'_{\beta,k}$. Given an attack structure \mathcal{A} and a corrupted option \mathcal{C} , the attacks A_i to generate a bad path for violating the diagnosability are determined by the vectors $\mathbf{v}_k = [v_{t_1,k}, \dots, v_{t_j,k}, \dots, v_{t_{|T|},k}]^T$ that satisfy the following constraints:

$$\begin{cases} \text{a) } v_{t_j,k} \in \{0, 1\} & \forall t_j \in T_a \\ \text{b) } v_{t_j,k} = 0 & \forall t_j \in T \setminus T_a \\ \text{c) } \prod_{j'=j_1}^{j|\sigma_{\alpha,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \\ \quad = \prod_{j'=j'_1}^{j|\sigma'_{\alpha,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \\ \text{d) } \prod_{j'=j_1}^{j|\sigma_{\beta,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \\ \quad = \prod_{j'=j'_1}^{j|\sigma'_{\beta,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \end{cases} \quad (1)$$

Proof: Constraints (a) show that each transition $t_j \in T_a$ is associated with a binary decision variable $v_{t_j,k} \in \{0, 1\}$. Constraints (b) impose $v_{t_j,k} = 0$ for the set of transitions $T \setminus T_a$. By performing an attack A_i that is the modified labeling function l_{a_i} , constraints (c)(d) guarantee that two sequences $\sigma_{k,1} = \sigma_{\alpha,k} \sigma_{\beta,k}$ and $\sigma_{k,2} = \sigma'_{\alpha,k} \sigma'_{\beta,k}$ generate the same observation, i.e., $l_{a_i}(\sigma_{\alpha,k}) = l_{a_i}(\sigma'_{\alpha,k})$ and $l_{a_i}(\sigma_{\beta,k}) = l_{a_i}(\sigma'_{\beta,k})$. ■

In the following, constraints (1.c) and (1.d) are linearized by replacing them with the constraints (2.c) and (2.d):

$$\begin{cases} \text{a) } v_{t_j,k} \in \{0, 1\} & \forall t_j \in T_a \\ \text{b) } v_{t_j,k} = 0 & \forall t_j \in T \setminus T_a \\ \text{c) } \sum_{j'=j_1}^{j|\sigma_{\alpha,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \\ \quad = \sum_{j'=j'_1}^{j|\sigma'_{\alpha,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \\ \text{d) } \sum_{j'=j_1}^{j|\sigma_{\beta,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \\ \quad = \sum_{j'=j'_1}^{j|\sigma'_{\beta,k}|} [v_{t_j,k} \cdot e'_j + (1 - v_{t_j,k})e_j] \end{cases} \quad (2)$$

The following lemma establishes the relationship between vectors \mathbf{v}_k satisfying constraints (1) and vectors \mathbf{v}_k satisfying constraints (2).

Lemma 4.6: If vector \mathbf{v}_k satisfies constraints (1), then it also satisfies constraints (2).

Proof: Constraints (2.c) (2.d) ensure that two sequences $\sigma_{k,1}$ and $\sigma_{k,2}$ generate the same amount of transition labels (without requirement on the label order), while constraints (1.c) (1.d) guarantee that two sequences generate the same observation (with requirement on the label order). Hence, it can be deduced that vector \mathbf{v}_k satisfies constraints (1), then it also satisfies constraints (2). ■

By Lemma 4.6, we can deduce that the set of feasible solutions for constraints (1) is a subset of the set of feasible solutions for constraints (2). To formalize the new problem we introduce a set of vectors $\bar{\mathbf{v}}_k$, where each element $\bar{\mathbf{v}}_k \in \bar{\mathbf{v}}_k$ satisfies constraints (2) but violates constraints (1). To ensure these infeasible solutions $\bar{\mathbf{v}}_k$ are excluded from

the solution of following ILP Problem 2, additional linear constraints are defined using a vector \mathbf{y}_k for each $\bar{\mathbf{v}}_k \in \bar{V}_k$ with $y_{t_j,k} \in \{0, 1\}$ for $j = 1, \dots, |T|$.

ILP Problem 1: Consider an attack structure \mathcal{A} , an attack cost coefficient vector \mathbf{c} and a set of vectors \bar{V}_k that satisfies constraints (2) but does not satisfy constraints (1). The minimum attack cost z_k in terms of k -th attack path and a corrupted option \mathcal{C} , corresponding to attack A_k for violating the diagnosability can be obtained by solving the following ILP problem:

$$\begin{cases} z_k = \min \mathbf{c} \cdot \mathbf{v}_k, \\ \text{s.t. the set of constraints (2),} \\ \text{e.1) } y_{t_j,k} \in \{0, 1\} & \forall t_j \in T, \\ \text{e.2) } \sum_{j=1}^{|T|} y_{t_j,k} \geq 1 \\ \text{e.3) } y_{t_j,k} \leq v_{t_j,k} + \bar{v}_{t_j,k} \leq 2 - y_{t_j,k} & \forall t_j \in T, \forall \bar{\mathbf{v}}_k \in \bar{V}_k. \end{cases}$$

Constraints (e.2) and (e.3) linearize $\mathbf{v}_k \neq \bar{\mathbf{v}}_k$ by enforcing that there exists at least one variable $y_{t_j,k} \geq 1$. Indeed, if $\mathbf{v}_k = \bar{\mathbf{v}}_k$, then each $y_{t_j,k} = 0$ and constraint (e.2) is not satisfied.

C. Algorithm for optimal stealthy replacement attack

In the following, Algorithm 2 outlines the steps to compute this optimal stealthy replacement attack.

Algorithm 2: Computation of an optimal stealthy replacement attack

Input: An attack verifier $U_a = (X_a^U, E_a^U, \delta_a^U, M_0^U)$, an attack structure \mathcal{A} , a set of stealthy attacks \mathcal{A}_s and an attack cost coefficient vector \mathbf{c} .

- 1 List all the attack paths $\Pi = \{\tilde{\sigma}_1, \dots, \tilde{\sigma}_k, \dots, \tilde{\sigma}_h\}$ from U_a by Definition 4.3 and obtain all the corrupted options $\mathcal{C}(\mathcal{A})$;
- 2 Initialize $k = 1, Z = \emptyset, V_k = \emptyset$;
- 3 **for** all $\tilde{\sigma}_k \in \Pi$ **do**
- 4 **for** each corrupted option $\mathcal{C} \in \mathcal{C}(\mathcal{A})$ **do**
- 5 Reset $\bar{V}_k = \emptyset$;
- 6 **if** the ILP Problem 1 is feasible **then**
- 7 **if** the solution \mathbf{v}_k satisfies constraints (1) and the corresponding $A_k \in \mathcal{A}_s$ **then**
- 8 z_k is the objective value of solution \mathbf{v}_k ;
- 9 **if** $z_k = \hat{c}$ **then**
- 10 $\hat{z}_k = z_k, \hat{A}_k = A_k$, go to Step 17;
- 11 **else**
- 12 $Z = \{z_k\} \cup Z, V_k = V_k \cup \{\mathbf{v}_k\}$;
- 13 **else**
- 14 $\bar{\mathbf{v}}_k = \mathbf{v}_k$;
- 15 $\bar{V}_k = \bar{V}_k \cup \{\bar{\mathbf{v}}_k\}$ and go to Step 6;
- 16 $\hat{z} = \min_{z_k} Z$;
- 17 Return \mathbf{v}_k and \hat{A}_k ;

Step 1 obtains all the attack paths in the attack verifier U_a by the conditions in Definition 4.3 and all the corrupted

options $\mathcal{C}(\mathcal{A})$. Step 2 initializes the index k , the set Z that contains the attack costs corresponding to the computed attacks, and the set V_k containing the feasible solutions of ILP problem 2 that satisfy constraints (1). The main part (steps 3–15) analyzes each attack path until the attack cost is optimal (i.e., minimum cost) or all the attack paths have been analyzed.

Theorem 4.7: Given an LPN system S satisfying Assumptions (A1)–(A3) vulnerable to an attack structure \mathcal{A} , the attack A for violating the diagnosability of the system S that is computed by Algorithm 2 is optimal.

Proof: For each path $\tilde{\sigma}_k$, the algorithm obtains all the possible attacks A_k corresponding to the path $\tilde{\sigma}_k$ and all the corrupted options $\mathcal{C}(\mathcal{A})$. For each corrupted option \mathcal{C} , the set \bar{V}_k containing the feasible solutions of ILP problem 1 and not satisfying constraints (1) is firstly reset as empty set. By Lemma 4.7, the part (steps 6–15) analyzes the minimum cost attack corresponding the path $\tilde{\sigma}_k$ and this corrupted option by guaranteeing the feasibility of solution, satisfying the constraints (1) and ensuring the stealthiness of attack (step 7), i.e., the obtained attack should be contained in the set of given stealthy attack \mathcal{A}_s . Particularly, if the total attack cost z_k is equal to its minimum coefficient \hat{c} , i.e., $\hat{c} = \min_{\{j|t_j \in T_o\}} c_j$, it implies that a minimum cost attack \hat{z}_k is obtained, such that the algorithm returns the corresponding attack \hat{A}_k . Otherwise it stores the attack cost z_k in the cost set Z . After analyzing all paths, we obtain the minimum cost \hat{z}_k from Z and its corresponding attack \hat{A}_k . ■

V. A CASE STUDY ON AUTOMATED MANUFACTURING SYSTEM

In this section, to demonstrate the proposed attack approach, we consider an automated manufacturing system (AMS) taken from [13], [14], as shown in Fig. 1. This system consists of two entries (I1 and I2), two exits (O1 and O2), five machines (M1–M5), two buffers with capacity 4 (B1 and B2), four robots (R1–R4), and two AGVs (AGV1 and AGV2). It incorporates two separate production lines that manufacture different products, as summarized below. During operation, Robots R1 and R2 serve both lines: Robot R1 supports Machines M1, M2, and M4, while Robot R2 handles parts from M3 and M5.

Line 1: R1 loads stock from I1 into M1/M2, R3 routes the identical intermediates through B1 to M3, and R2 puts the finished pieces on AGV1, which delivers them to O1 and brings fresh material. Line 2 mirrors this process where R1 feeds M4 from I2, R4 moves intermediates via B2 to M5, and R2 places the output on AGV2 for O2 before picking up new stock.

The Petri net model of the considered AMS is shown in Fig. 2. The meaning of each place and each transition is described in detail in [14]. In Fig. 2, the set of unobservable transitions $T_u = \{t_7, t_9, t_{12}, t_{17}, t_{21}, t_{22}\}$ with two fault transitions $t_{21}(f_1)$ and $t_{22}(f_2)$. In detail, f_1 represents a situation that a raw material from entry I1 is directly put into buffer B1 without being processed by M1 or M2 and

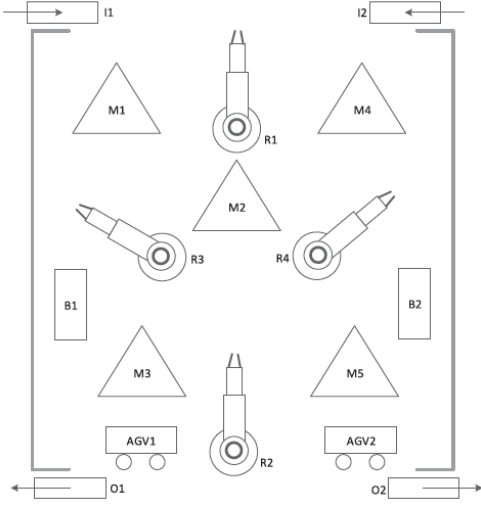


Fig. 1. An automated manufacturing system.

f_2 denotes another situation that an intermediate part after processing by M4 is then machined by M5, without being put into the buffer B2.

The set of observable transitions $T_o = T \setminus T_u$ with $l(t_1) = a, l(t_2) = l(t_3) = l(t_{13}) = b, l(t_4) = l(t_5) = c, l(t_6) = d, l(t_8) = l(t_{15}) = e, l(t_{10}) = l(t_{19}) = f, l(t_{11}) = g, l(t_{14}) = l(t_{16}) = h, l(t_{18}) = l(t_{20}) = k$. We now explore the effectiveness of the attack strategy proposed in this paper. Suppose that an attacker can hijack the sensors at Robots 1–4, and has the attack capability by removing the label of transition t_2 and replacing the label of transition t_4 (resp., t_{13}) from c to d (resp., from b to c), i.e., the attack structure $\mathcal{A} = \{(t_2(b), t_2(\varepsilon)), (t_4(c), t_4(d)), (t_{13}(b), t_{13}(c))\}$.

By implementing Algorithm 2, it returns the following results with a computational time 2.35 mins that refers to the CPU seconds of a laptop with Intel CPU Core 2.3 GHz, 8GB memory and a Matlab tool. Specifically, an attack path $\tilde{\sigma}_1 = (t_1, t_1)(t_{21}, t_2)(t_{13}, t_{13})(t_{14}, t_{14})(t_{15}, t_{15})(t_{16}, t_{16})(t_{17}, t_{17})(t_{18}, t_{18})(t_{19}, t_{19})(t_{20}, t_{20})$ and the corrupted option $\mathcal{C}_1 = \{e'_2 = \varepsilon, e'_4 = d, e'_{13} = c\}$, we get the solution $v_1 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ and its minimum attack cost $z_1 = 1$, which generates a bad path with $l_{a_1}(\sigma_{1,1}) = l_{a_1}(\sigma_{1,2}) = a(bhekf k)$, such that the system becomes non-diagnosable.

VI. CONCLUSION

Algorithms addressing stealthy replacement attacks that undermine diagnosability in discrete event systems have been developed. We constructed an attack verifier to enumerate all attack paths leading to the violation of diagnosability. In addition, we formulated the optimal attack synthesis as a set of ILP problems, and solved them to obtain the optimal attack. Future work will first extend the proposed attack strategy by incorporating more advanced scenarios.

REFERENCES

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

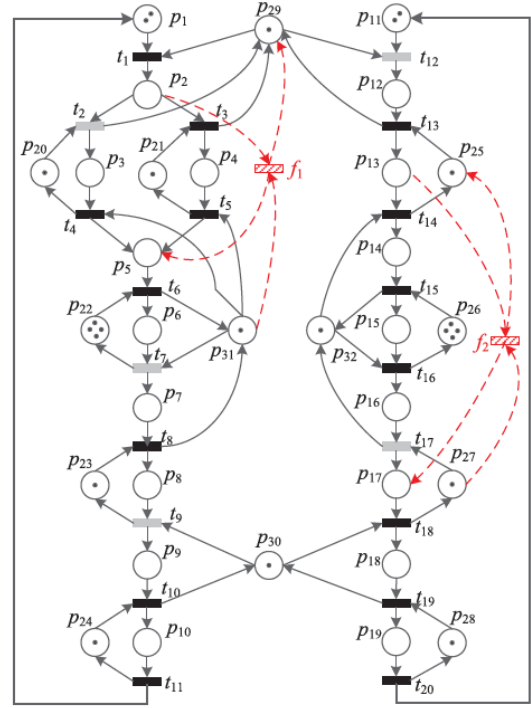


Fig. 2. An automated manufacturing system under attack modeled by LPN.

- [2] 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, 2020.
- [3] R. Liu, A. M. Mangini, and M. P. Fanti, "Synthesis of optimal stealthy attacks against diagnosability in labeled Petri nets," *IEEE/CAA Journal of Automatica Sinica*, 2025.
- [4] R. Liu, Y. Hu, A. M. Mangini, and M. P. Fanti, "K-corruption intermittent attacks for violating the codiagnosability," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 1, pp. 159–172, 2025.
- [5] 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, 1995.
- [6] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete event systems using labeled Petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 144–153, 2014.
- [7] T. Li, H. Ren, R. Liu, M. P. Fanti, and Z. Li, "Fault diagnosis of labeled Petri nets under attacks using integer linear programming," *IEEE Transactions on Automation Science and Engineering*, 2025.
- [8] R. Liu, W. Duan, A. M. Mangini, and M. P. Fanti, "Attack synthesis in discrete event systems under asymmetric observation setting," *IFAC-PapersOnLine*, vol. 58, no. 1, pp. 186–191, 2024.
- [9] J. Yao, S. Li, and X. Yin, "Sensor deception attacks against security in supervisory control systems," *Automatica*, vol. 159, p. 111330, 2024.
- [10] N. Ran, A. Giua, and C. Seatzu, "Enforcement of diagnosability in labeled Petri nets via optimal sensor selection," *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 2997–3004, 2018.
- [11] Y. Hu, R. Liu, M. P. Fanti, and Z. Li, "Robust fault diagnosis of networked discrete event systems using labeled Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2025.
- [12] S. Hu, Z. Li, and R. Wisniewski, "Optimal sensor selection for diagnosability enforcement in labeled Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 5, pp. 2965–2977, 2024.
- [13] G. Zhu, Z. Li, N. Wu, and A. Al-Ahmari, "Fault identification of discrete event systems modeled by Petri nets with unobservable transitions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 333–345, 2017.
- [14] M. Zhou and F. DiCesare, *Petri net synthesis for discrete event control of manufacturing systems*. Springer Science & Business Media, 2012, vol. 204.