Robust Stealthy Attacks Based on Uncertain Costs and Labeled Finite Automata With Inputs

Rabah Ammour, Said Amari, Leonardo Brenner, Isabel Demongodin, Dimitri Lefebvre

Robust stealthy attacks based on uncertain costs and labeled finite automata with inputs

Rabah Ammour[1] *Member, IEEE,* Said Amari[2], Leonardo Brenner[1], Isabel Demongodin[1] and, Dimitri Lefebvre[3], *Senior Member, IEEE*

***Abstract*—This paper deals with the vulnerability analysis of cyber-physical systems subject to malicious actions. For this purpose, the considered system is assumed to be abstracted as a discrete event system. Labeled finite automata with inputs are used to model the system's behavior along with the information that circulates in both the input and output channels. In particular, we study here stealthy, i.e., undetectable, cyber-attacks that aim to drive the system from a given normal state to a set of forbidden states. We assume that the attacker has limited resources, i.e., a credit, to insert and delete control and sensors events. The proposed analysis evaluates the costs of such attacks on the controlled system depending on its structure, the cost of the malicious actions and possible uncertainties that may affect those costs. It provides systematic methods that aim to compute attacks of minimal cost and robust attacks that are weakly impacted by uncertainties. A case study representing a manufacturing plant is considered to illustrate the results.**

***Index Terms*—Discrete Event Systems, Automata, Cyber-Physical Systems, Robust Attacks.**

I. Introduction

Cyber-Physical Systems (CPSs) have been widely used in numerous applications such as networked control systems, smart power grids, healthcare systems, advanced communication processes and autonomous transportation networks. CPSs integrate computational and communication capabilities to control and monitor physical processes. The communication and data exchange networks between controllers /supervisors and the operative part of the process increase the vulnerability of CPSs to various types of attacks that may lead to critical and dangerous situations. Examples of cyber attacks include the StuxNet strike on industrial control systems [1], and the spoofing of global positioning systems to capture unmanned aircrafts [2]. As a consequence, vulnerability analysis of CPSs is an increasingly important problem and several works addressed this issue in recent years [3]. In the context of Discrete Event Systems (DESs), automata formalism is suitable to model CPSs to analyze their vulnerability.

Authors of [4] use this formalism and investigate the problem of synthesizing an attack strategy for a given controlled DES by adopting an attacker's viewpoint. They assume that the attacker is able to manipulate the sensor measurements in order to mislead the controller and drive the system to unsafe or undesirable states without being detected. A similar problem has been addressed by authors of [5] considering actuator attacks where the intruder partially observes the execution of the closed-loop system and can modify the control generated by the supervisor. In the work of [6], attacks on both sensors and actuators layers in networked supervisory control systems are addressed. Thus, the intruder can hide, insert or replace events with the objective to drive the system to reach unsafe states.

Attacks on both sensors and actuators are also considered in [7] but in the discrete-time distributed multi-agent systems framework. The authors show how an attack on a compromised agent can propagate and affect other agents that are reachable from it. An adaptive attack compensator is designed to limit the attack effect and its propagation. Attacks and defense graphs [8], [9] and kill chains [10] are widely used tools to deal with the problem of security assessment due to their great ability to detail network attacks. Basically, the nodes of such graph or chains represent vulnerabilities or devices, and the edges represent the possible evolution within nodes, e.g., the attacker gains more and more privilege in the network by exploiting successive vulnerabilities. Different analysis methods (such as path search, Bayesian netwoks) are then applied on the graph to assess the vulnerabilities of the network. In the meantime, some approaches have been extended to the stochastic context [11], [12], [13] allowing to quantify, in a probabilistic sense, the attack strategies.

An other concept has been proposed in [14] where a fixed cost has been assigned to each possible attack action. This cost captures the expense (in terms of time, complexity or data packet size) of the attacker when trying to alter the exchanged data in the CPS. This concept makes it possible to model the attacker's cost constraints and to characterize feasible attacks scenario with respect to intruder's available resources.

In this paper, we assume that the network security barriers have failed and we consider malicious stealthy actions on both sensors and control events allowing the attacker to drive the system from its current state to a set of target (forbidden or dangerous) states. We refer to such attacks as *stealthy moving attacks* in the rest of the paper. The context is similar to

Manuscript received: September 22, 2022; Revised: December 15, 2022; Accepted: February 6, 2023.

This paper was recommended for publication by Editor Jingang Yi upon evaluation of the Associate Editor and Reviewers' comments.

This work has been partially supported by the CPSecurity project (CNRS-INS2I grant) and by the French National Research Agency under grant agreement ANR-22-CE10-0002.

[1]Rabah Ammour, Leonardo Brenner, and Isabel Demongodin are with Aix-Marseille University, CNRS, LIS, Marseille, France rabah.ammour, leonardo.brenner, isabel.demongodin@lis-lab.fr

[2]Said Amari is with LURPA, ENS Paris-Saclay, France. samari@ens-paris-saclay.fr

[3]Dimitri Lefebvre is with Université Le Havre Normandie, GREAH, 76600, Le Havre, France. dimitri.lefebvre@univ-lehavre.fr

Digital Object Identifier (DOI): 10.1109/LRA.2023.3250007.

[14], in the sense that the attacker is assumed to have a certain credit to manipulate the control *symbols* sent to the actuators and the output *labels* returned by the sensors. It is supposed that each action, which can be an insertion or deletion of symbols/labels, has a certain cost. Consequently, stealthy cyber-attacks of limited cost could be considered. The stealthiness characterizes the ability of the attacker to hide its traces and to remain undetectable while it moves the system's current state. Assigning constant values to the attack costs has been developed in [15] with a vulnerability analysis based on Dijkstra algorithm. In the present work, we introduce uncertain costs represented by intervals that may be used when costs are varying or not perfectly known. This new setting is consistent with many practical situations for which the cost values cannot be exactly estimated. By enlarging the cost intervals, the approach can be implicitly extended to situations that include some unknown costs. This setting needs the use of a new approach based on *min-max regret* [16].

With this new concept, the main objective of this paper is to evaluate and discuss the vulnerability of the CPS by defining and computing attacks of maximal robustness. Such attacks are characterized by the lowest impact of the uncertainties on their global cost. Note that exploring robust attacks that aim to minimize the maximal regret is interesting not only for the attacker but also for the defender in the sense that this new notion helps to design and refine a defence strategy.

The rest of the paper is organized as follows. Section II is about the motivations and backgrounds. Section III introduces uncertain cost graphs based on labeled finite automata with inputs. Section IV is devoted to the vulnerability analysis of CPS affected by stealthy moving attacks. Section V is a case study and Section VI concludes the paper.

II. Backgrounds

We consider attacks that aim to drive the system from a given (normal) state to a target (forbidden or dangerous) state. Such a *moving attack* is able to change the information that circulates in both the input and output channels of the system as represented in Figure 1. Consequently, it can replace the true control sequence i by a wrong control sequence i_a. In the same time, the attacker is able to erase the traces generated by its malicious actions or to insert wrong traces o_a that are similar to the expected traces to be observed by the user which makes such attacks stealthy. The following assumptions are considered:

- the attacker knows the model of the system,
- the attacker knows (or is able to estimate) the current state of the system to perform the attack,
- the attacker can manipulate (insert or delete) the input symbols and output labels, each action corresponds to a given cost and the attacker has a limited credit to perform its attack.

For the sake of brevity, we restrict the proposed analysis to the insertion of input events and deletion of output labels and consider the cases where the attacker can manipulate all events and labels without any restriction, as far as it has enough credit. It is worth noting that the previous assumptions can be relaxed, in particular, by increasing some cost values.

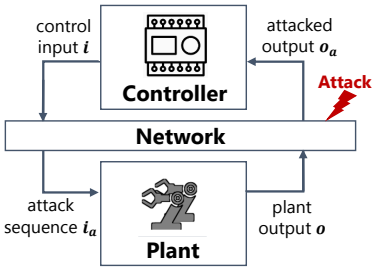

Fig. 1: Cyber-physical system under attack

In our previous work [17], a particular class of Petri nets, called *output synchronized Petri nets*, has been defined to model the controlled plant and the information that circulates through the CPS. To represent and analyze the behavior of such a model, *labeled finite state automaton with inputs* is derived from this formalism. It represents the states space of the system as well as the input and output information.

Definition 1: A *labeled finite automaton with inputs* (LFAI) is a 6-tuple $G = (X, E_\lambda, \delta, x_0, Q, Obs)$, where

- X is a finite set of *states*,
- E is a finite set of *symbols (i.e., external input events)* and $E_\lambda = E \cup \{\lambda\}$ where λ is an *internal* and "always occurring" event,
- $\delta : X \times E_\lambda \rightarrow X$ is a (possibly partially defined) *transition function*,
- $x_0 \in X$ is an *initial state*,
- Q is a finite set of *labels (i.e., output events)* and $Q_\varepsilon = Q \cup \{\varepsilon\}$, where ε denotes the absence of label,
- $Obs : X \times E_\lambda \rightarrow 2^Q \cup \{\varepsilon\}$ is a *labeling function*. ▲

We consider that the LFAI is *deterministic* with respect to the symbols i.e., $\forall x \in X, \forall e \in E_\lambda, |\delta(x,e)| \in \{0,1\}$ where $|.|$ stands for the cardinality of a set. If $|\delta(x,e)| = 1$ with $e \in E$, then e is said to be *active* at state x. $\delta(x,\lambda) = x'$ means that the system will move from x to x' according to the "always occurring" event λ, i.e., without waiting for any external input symbol. In this case, x is said to be a λ-state. λ-transitions and λ-states are used to represent explicitly internal switches, for example high priority switches, that do not require any action from the controller and are directly generated by the system. Moreover, when a λ-transition (i.e.,

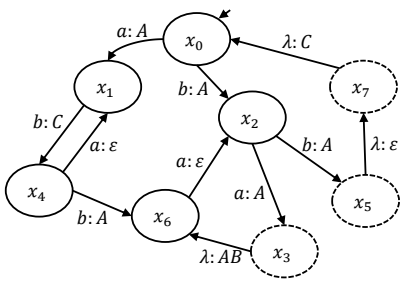

Fig. 2: Example of labeled finite state automaton with inputs

a transition associated with event λ) exists from a given state x then it will be the single transition outgoing from x, i.e., $\sum_{e\in E_\lambda} |\delta(x,e)| = 1$. Note that multiple labels (a subset of Q) could be provided by one transition and ε is used when no label is generated.

Example 1: Consider the LFAI of Figure 2. It is defined by a set of 8 states $X = \{x_0, \cdots, x_7\}$ (with three λ-states x_3, x_5, x_7 represented by dashed circles) where x_0 is the initial state. The set of symbols is $E = \{a, b\}$ and the set of labels is $Q = \{A, B, C\}$. The notation "$a : A$" means that the system switches from state x_0 to state x_1 when it receives symbol a and that this change delivers label A. Thus, it holds $\delta(x_0, a) = x_1$ and $Obs(x_0, a) = A$. □

A control sequence of length n sent by a controller is denoted by $i = e_1 \dots e_n$ with $e_h \in E, h = 1 \dots n$. It is completed by the λ events that are generated spontaneously by the system leading to the corresponding *executed sequence* $i' = e'_1 \dots e'_m$, $e'_h \in E_\lambda$, $h = 1 \dots m$ with $m \geq n$. Due to the determinism of the considered formalism, a single i' is associated to a given i.

We introduce δ^* and Obs^* as the trivial extensions of δ and Obs functions defined recursively, for an executed sequence i', by $\delta^*(x, ei') = \delta^*(\delta(x, e), i')$ and $Obs^*(x, ei') = Obs(x, e)Obs^*(\delta(x, e), i')$. A trajectory, denoted $\sigma(x, i')$, of $m + 1$ successive states could be obtained from x as:

$$x_{j_0} \xrightarrow{e'_1 : Obs(x_{j_0}, e'_1)} x_{j_1} \dots x_{j_{m-1}} \xrightarrow{e'_m : Obs(x_{j_{m-1}}, e'_m)} x_{j_m} \quad (1)$$

where $x_{j_0} = x$ and $x_{j_m} = \delta^*(x, i')$. The sequence of sets of labels generated by i' is denoted as $o = Obs^*(x, i') = Obs(x_{j_0}, e'_1) \dots Obs(x_{j_{m-1}}, e'_m)$. We use $(x_{j_{h-1}}, e'_h) \in \sigma(x, i')$ to refer to a transition from the state $x_{j_{h-1}}$ driven by the symbol e'_h in the trajectory $\sigma(x, i')$. Finally, when an attacker inserts orders or manipulates the control sequence i sent by the controller, the resulted sequence is called an *attack sequence* which is denoted as i_a. Its corresponding executed sequence i'_a is called an *executed attack sequence* and the generated sequence of sets of labels to be erased is denoted as o_a.

III. Uncertain cost graph

In this section we consider that the attacker knows the model of the system, and can manipulate the symbol and label events. In particular, we consider moving attacks that are composed by sequences of symbols generated exclusively by the attacker, i.e., $i_a = i$, and sequences of sets of outputs erased by the attacker. An insertion cost c_I is defined for each symbol and a deletion cost c_E is defined for each label. Note that $c_I(\lambda) = 0$ and $c_E(\varepsilon) = 0$. For $e \in E$ and $q \in Q$, insertion and deletion costs may be either defined as single values or as intervals when some uncertainties exist about such costs. Uncertainties may exist for various reasons, including the risk (from the attacker's viewpoint) that the controller performs some actions during an attack.

In our previous work [15], fixed values of costs have been considered and an *Adding Control Graph* (ACG) has been developed. The objective was to characterize the attack sequences of minimal cost. The nodes of an ACG are those of the LFAI while each edge corresponds to a transition of the LFAI with a weight given as follows. Let $x, x' \in X$ be two states in the LFAI and $e \in E_\lambda$ such that $\delta(x, e) = x'$. The weight $c_{ACG}(x, e)$ of the arc corresponding to transition $\delta(x, e) = x'$, is given by:

$$c_{ACG}(x, e) = c_I(e) + \sum_{q \in Obs(x,e)} c_E(q) \quad (2)$$

Now, let us introduce the *Uncertain Adding Control Graph* (UACG) that is similar to the ACG except that the weight of each edge is a positive interval. For this purpose let us first define for each symbol $e \in E$ the *insertion cost interval* as:

$$C_I(e) = [c_I^-(e), c_I^+(e)], \quad (3)$$

and for each label $q \in Q$, the *deletion cost interval* as:

$$C_E(q) = [c_E^-(q), c_E^+(q)]. \quad (4)$$

Note that $C_I(\lambda) = C_E(\varepsilon) = [0, 0]$. Let $x, x' \in X$ be two states in the LFAI and $e \in E_\lambda$, such that $\delta(x, e) = x'$. The weight $c_{UACG}(x, e)$ of the edge corresponding to transition $\delta(x, e) = x'$ is associated to the interval[1]:

$$C_{UACG}(x, e) = [c^-_{UACG}(x, e), c^+_{UACG}(x, e)], \quad (5)$$

with

$$\begin{array}{l} c^-_{UACG}(x, e) = c_I^-(e) + \sum_{q \in Obs(x,e)} c_E^-(q), \\ c^+_{UACG}(x, e) = c_I^+(e) + \sum_{q \in Obs(x,e)} c_E^+(q). \end{array} \quad (6)$$

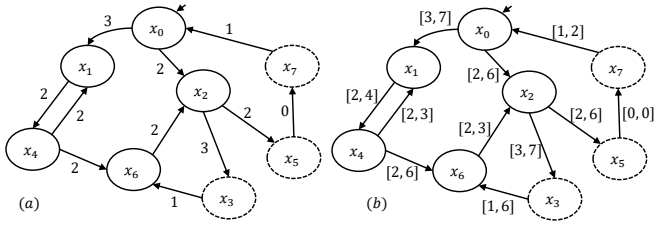

Fig. 3: (a) Adding control graph, (b) Uncertain adding control graph

Example 2: Let us consider that the costs to insert or delete each symbol and label for the LFAI of Figure 2 are given by $c_I(a) = 2, c_I(b) = 1, c_E(A) = 1, c_E(B) = 0, c_E(C) = 1$. From these values, the cost of each transition of the corresponding ACG could be computed for the "certain case" as reported in Figure 3 (a). For instance, the cost to drive the system from state x_0 to state x_1 is equal to 3 which corresponds to the sum of the cost to insert the symbol a and of the cost to erase the label A that results from the state switch, i.e., $c_{ACG}(x_0, a) = c_I(a) + c_E(A) = 2 + 1 = 3$.

Concerning the "uncertain case" and considering the cost intervals, $C_I(a) = [2, 3], C_I(b) = [1, 2], C_E(A) =$

[1]Intervals are manipulated according to the IEEE 1788 standard for interval arithmetic [18]. In particular, $[a, b] + [c, d] = [a + c, b + d]$.

$[1,4], C_E(B) = [0,2], C_E(C) = [1,2]$, one can compute the cost of each transition of the UACG as reported in Figure 3 (b). For instance, the cost interval to drive the system from state x_0 to state x_1 is equal to $C_{UACG}(x_0, a)$=$[c_I^-(a)+c_E^-(A), c_I^+(a)+c_E^+(A)]$ = $[3,7]$. □

IV. Analysis of stealthy attacks

In this section, we consider stealthy moving attacks that aim to drive the system from the current state to a state from a subset $\mathcal{F} \subseteq X$ of forbidden states. This set could include deadlocks or other dangerous states. To perform a stealthy moving attack, the attacker inserts an attack sequence (i.e., a wrong sequence of symbols) and, in the same time, it erases the observable traces (i.e., a sequence of sets of labels) that the attack sequence has generated. Uncertain costs, associated to the attacker's actions, are used to take into account the situations where the insertion and disabling costs may vary depending on the system state and other reasons, e.g., the risk that the controller sends symbols during the attack. Such variations are defined by a given UACG. Observe that compared to our previous works [15] the proposed analysis aims to evaluate the robustness of the attack with respect to the system.

Definition 2: Let $i_a = e_1 \dots e_n$ be a stealthy moving sequence that drives the system from a state x to a forbidden state in $\mathcal{F}$. The cost of i_a is included in interval $c_{MA}(i_a, x)$ defined by

$$c_{MA}(i_a, x) = \sum_{h=1,\dots,n} C_{UACG}(x_{jh}, e_h).$$

▲

Since many possible attacks with variable costs exist to reach a forbidden state, two particular attacks are considered: the attack of minimal cost and the attack of maximal robustness. The search of such attacks is motivated by the assumption that the attacker has a limited credit to perform its attack.

Definition 3: An attack sequence i_a^* *of minimal cost* c^* is an attack that drives the system from a state x to a forbidden state in $\mathcal{F}$ such that other attacks from x to $\mathcal{F}$ have a cost at least equal to c^*. ▲

Observe that the attack i_a^* corresponds to the worst case from the controller perspective since the attacker could reach $\mathcal{F}$ from x with the lowest cost.

Definition 4: An attack sequence i_a^r *of maximal robustness (or robust attack)* is an attack that drives the system from a state x to a forbidden state in $\mathcal{F}$ such that the possible cost variation, with respect to the minimal cost attack to $\mathcal{F}$, is minimal. ▲

A. Moving attack of minimal cost

In order to compute the cost of a stealthy moving attack in an UACG, the notion of *scenario* is first introduced [16].

Definition 5: A *scenario* S is an assignment of a single value $c_{UACG}(x, e, S) \in C_{UACG}(x, e)$ for each state $x \in X$ and event $e \in E$ active in x. ▲

We refer to $ACG(S)$ as to the adding control graph obtained from the UACG associated to scenario S. The cost of the stealthy moving attack from x to $x_f \in \mathcal{F}$ in scenario S that inserts i_a at x and erases the corresponding generated sequence of sets of labels o_a is obtained according to its corresponding executed attack sequence i'_a and the resulting trajectory $\sigma(x, i'_a)$ of the form (1) that ends in x_f. It is named as the *cost of attack sequence* i_a *at state* x in scenario S and computed as:

$$c_{MA}(x, i_a, S) = \sum_{(x',e)\in\sigma(x,i'_a)} c_{UACG}(x', e, S).$$

We denote by $\mathcal{S}$ the set of possible scenarios S. Observe that it remains non tractable to enumerate all possible scenarios.

***Proposition** 1:* The attack sequence i_a^* of *minimal cost* from state $x \in X$ to a forbidden state in $\mathcal{F}$ is defined by

$$i_a^* = \underset{i_a \in I_a(x,\mathcal{F})}{\arg\min} \{c_{MA}(x, i_a, S_{min})\}. \quad (7)$$

where $I_a(x, \mathcal{F})$ is the set of attacks that move the state from x to any state in $\mathcal{F}$ and S_{min} is the *scenario of minimal costs* in which $c_{UACG}(x', e, S) = c^-_{UACG}(x', e)$ for each state $x' \in X$ and event $e \in E$ active in x'.

Proof: Let us first consider a given scenario $S \in \mathcal{S}$. The stealthy moving attack of minimal cost from state x to a given state $x_f \in \mathcal{F}$ in scenario S can be obtained by using the well-known Dijkstra algorithm in $ACG(S)$ and the cost $c^*_{MA}(x, x_f, S)$ of such attack satisfies:

$$c^*_{MA}(x, x_f, S) = \min_{i_a \in I_a(x,x_f)} \{c_{MA}(x, i_a, S)\}.$$

where $I_a(x, x_f)$ is the set of attacks that move the state from x to x_f. Now by repeating the use of Dijkstra algorithm to each state $x_f \in \mathcal{F}$, one can compute the minimal cost $c^*_{MA}(x, \mathcal{F}, S)$ from current state x to any state $x_f \in \mathcal{F}$ in scenario S:

$$\begin{aligned} c^*_{MA}(x, \mathcal{F}, S) &= \min_{x_f \in \mathcal{F}} \{c^*_{MA}(x, x_f, S)\} \\ &= \min_{i_a \in I_a(x,\mathcal{F})} \{c_{MA}(x, i_a, S)\}. \end{aligned}$$

Finally, the minimal cost over all scenarios $S \in \mathcal{S}$ is given by:

$$c^*_{MA}(x, \mathcal{F}) = \min_{S\in\mathcal{S}} \{c^*_{MA}(x, \mathcal{F}, S)\}.$$

Observe that for $S \in \mathcal{S}$ and $x_f \in \mathcal{F}$, we have $c^*_{MA}(x, x_f, S) \geq c^*_{MA}(x, x_f, S_{min})$ and $c^*_{MA}(x, \mathcal{F}, S) \geq c^*_{MA}(x, \mathcal{F}, S_{min})$. Consequently, $c^*_{MA}(x, \mathcal{F}) = c^*_{MA}(x, \mathcal{F}, S_{min})$ and Proposition 1 holds. □

B. Robust moving attack

In this section we are interested in determining robust attacks with respect to a set of known cost intervals that model the uncertainties. The criterion used here to classify an attack as robust or not is the *maximal regret*. This criterion initially proposed in the context of game theory was adapted to robust optimization in [19], [20], [21] and is extended here for defining robust attacks. Let us consider an UACG and first introduce the notion of *regret*.

Definition 6: The *regret* $r_{MA}(x, x_f, i_a, S)$ of a moving attack i_a from a state x to a state $x_f \in \mathcal{F}$ in a given scenario S is the difference between the costs of the attack sequence i_a and the one of minimal cost to reach x_f in scenario S:

$$r_{MA}(x, x_f, i_a, S) = c_{MA}(x, i_a, S) - c^*_{MA}(x, x_f, S). \quad (8)$$

▲

The regret reflects the additional cost that the attacker may spend to reach x_f with respect to the attack of minimal cost. The notion of regret can be extended to a set of forbidden states $\mathcal{F}$ for i_a in scenario S by:

$$r_{MA}(x, \mathcal{F}, i_a, S) = c_{MA}(x, i_a, S) - c^*_{MA}(x, \mathcal{F}, S). \quad (9)$$

***Proposition** 2:* The attack sequence i^r_a of *maximal robustness* from state $x \in X$ to a forbidden state in $\mathcal{F}$ is given by

$$i^r_a = \underset{i_a \in I_a(x,\mathcal{F})}{\arg\min} \{r_{MA}(x, \mathcal{F}, i_a, S^{max}_{min}(i_a))\} \quad (10)$$

with $S^{max}_{min}(i_a)$ the scenario where the cost of each transition $(x', e) \in \sigma(x, i'_a)$ (i'_a is the executed attack sequence that corresponds to i_a) is assumed to take its maximal value $c^+_{UACG}(x', e)$, whereas the costs of all other transitions in the UACG take their minimal values.

Proof: The regret $r_{MA}(x, x_f, i_a, S)$ of an attack sequence i_a from x to a given state $x_f \in \mathcal{F}$ reaches its maximal value for the particular scenario $S^{max}_{min}(i_a)$ [16]:

$$r_{MA}(x, x_f, i_a, S^{max}_{min}(i_a)) = \max_{S \in \mathcal{S}} \{r_{MA}(x, x_f, i_a, S)\}.$$

It can be extended to a set of forbidden states $\mathcal{F}$:

$$r_{MA}(x, \mathcal{F}, i_a, S^{max}_{min}(i_a)) = \max_{S \in \mathcal{S}} \{r_{MA}(x, \mathcal{F}, i_a, S)\}.$$

The stealthy moving attack of maximal robustness from x to $\mathcal{F}$ corresponding to attack sequence i^r_a is the attack with the minimal value of maximal regret. i^r_a is obtained in three steps:

- compute the set $I_a(x, \mathcal{F})$ of attacks that correspond to non cycling trajectories from x to $\mathcal{F}$ in the LFAI (it is not necessary to consider the trajectories with one or more cycles because such trajectories include an additional cost and regret),
- for each attack i_a from x to $\mathcal{F}$, compute the maximal regret $r_{MA}(x, \mathcal{F}, i_a, S^{max}_{min}(i_a))$,
- compute the minimal value $r^*_{MA}(x, \mathcal{F})$ of the maximal regret of the attacks from x to $\mathcal{F}$:

$$r^*_{MA}(x, \mathcal{F}) = \min_{i_a \in I_a(x,\mathcal{F})} \{r_{MA}(x, \mathcal{F}, i_a, S^{max}_{min}(i_a))\}.$$

Consequently, the stealthy moving attack from state x to $\mathcal{F}$ of maximal robustness is obtained according to the attack sequence of Equation (10) and Proposition 2 holds. Note that, in general, i^r_a does not coincide with i^*_a. □

Remark 1: the time complexity to determine the robust attack that drives the system from state x to a state in $\mathcal{F}$ is $\mathcal{O}(|I_a(x, \mathcal{F})|.|V|.\log|X|)$. It depends on the number of non-cycling trajectories $|I_a(x, \mathcal{F})|$ which can be bounded by $|\mathcal{F}| \times \sum_{k=0}^{n} \frac{n!}{(n-k)!}$ with $n = |X| - (|\mathcal{F}| + 1)$ and on the complexity of Dijkstra's algorithm given by $\mathcal{O}(|V|.\log|X|)$ where $|V|$ is the number of UACG edges.

Example 3: Consider again the LFAI of Figure 2 and an attack that aims to change the current state and mask the traces generated by the attack. Specifically, the attacker aims to drive the plant from state x_0 to state x_6. There exist several attack sequences to reach x_6, in particular the sequences of symbols, $ab(ab)^*b$ and ba. According to Dijkstra algorithm applied on ACG in Figure 3(a), the trajectory of minimal cost is $x_0 \xrightarrow{b:A} x_2 \xrightarrow{a:A} x_3 \xrightarrow{\lambda:AB} x_6$ of cost $c^*_{MA}(x_0, x_6) = 6$. Thus, the attacker needs to insert symbol b, next symbol a and, to erase labels A three times and next B once. The attack sequence of minimal cost is then given by $i^*_a = ba$ which corresponds to the executed attack sequence $(i'_a)^* = ba\lambda$.

Consider now the UACG of Figure 3(b) and attacks that aim to drive the system from x_0 to x_6. Let us first remark that $ACG(S_{min})$ associated to this UACG is the one represented in Figure 3(a). Thus $c^*_{MA}(x_0, x_6, S_{min}) = 6$ and $i^*_a = ba$. For the robust attack that drives the plant from state x_0 to state x_6, observe that only two attack sequences that correspond to non cycling trajectories exist to move the system from x_0 to x_6 given by $I_a(x_0, x_6) = \{abb, ba\}$. The maximal regret of the two attacks are respectively $r_{MA}(x_0, x_6, abb, S^{max}_{min}(abb)) = 7 + 4 + 6 - 2 - 3 - 1 = 11$ whereas $r_{MA}(x_0, x_6, ba, S^{max}_{min}(ba)) = 6 + 7 + 6 - 3 - 2 - 2 = 12$. To conclude, $r^*_{MA}(x_0, x_6) = 11$ and thus the robust attack sequence is given by $i^r_a = abb$. Note that the attack of minimal cost is different from the robust one. □

Remark 2: The proposed approach, dedicated to CPS, could be enlarged to any system provided that its associated model explicitly defines the states space and the input/output information.

V. CASE STUDY

We consider for this case study a part of an industrial plant (see Figure 4) composed by two production lines and shared robots. Each line comprises several mobile agents to transport the products that will undergo three operations by robots of different types (R1, R2, and R3). Each robot executes a particular operation on the product.

A. Description

The considered system is represented by the output synchronized Petri net (OutSynPN) given in Figure 5 (see [17] for more details on this formalism). Places p_1 and p_5 represent,

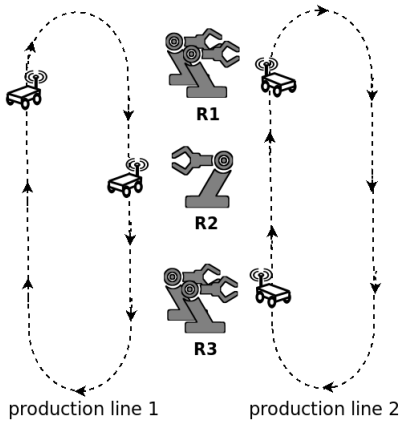

Fig. 4: Considered manufacturing plant

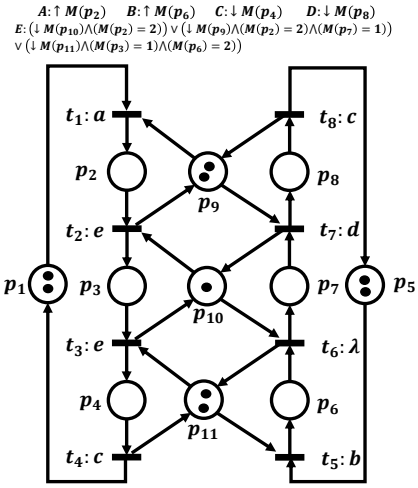

Fig. 5: Output synchronized Petri net of the case study

respectively, the availability of mobile agents of production lines 1 and 2 (in other terms, the agents are located at the electric charging stations). The availability of shared robots of types R1, R2, and R3 are represented by places p_9, p_{10}, and p_{11}, respectively. Transported by mobile agents, the products of line 1 have the following sequence of operations, represented by $t_1, p_2, t_2, p_3, t_3, p_4, t_4$ in the OutSynPN: operation 1 on robot R1 (place p_2), next operation 2 on robot R2 (place p_3) and finally operation 3 on robot R3 (place p_4). The one of products of line 2, represented by $t_5, p_6, t_6, p_7, t_7, p_8, t_8$, is as follow: operation 1 on robot R3 (place p_6), next operation 2 on robot R2 (place p_7) and finally operation 3 on robot R1 (place p_8).

The controller, via a wireless network, gives the orders to guide the mobile agents through the areas of robots. These control inputs, associated with the transitions of the Petri net model, are described in Table I. Note that the moving of a mobile agent of line 2 from area R3 to area R2 is done autonomously, i.e., without any control input sent by the controller, as it is represented by symbol λ associated with transition t_6.

Several sensors are placed in the physical system in order to detect the presence of mobile agents in the different areas or the availability of robots. According to these sensors, the physical system provides some feedback to the controller as resumed in Table II, where $\uparrow m_i$ and $\downarrow m_i$ represent, respectively, an increasing or decreasing of the marking of place p_i ($m_i = M(p_i)$) and MA stands for mobile agent.

TABLE I: Control inputs

Symbol	Application	Control input
a	mobile agent of line 1	enter in line 1 & move to area R1
b	mobile agent of line 2	enter in line 2 & move to area R3
c	mobile agent of both lines	evacuate the line (i.e., the product) and move to the station
d	mobile agent of line 2	move to area R1
e	mobile agent of line 1	move to area R2 and/or area R3

TABLE II: Plant output

Label	Definition	Plant output
A	$\uparrow m_2$	a MA enters in line 1
B	$\uparrow m_6$	a MA enters in line 2
C	$\downarrow m_4$	a MA leaves line 1
D	$\downarrow m_8$	a MA leaves line 2
E	$(\downarrow m_{10} \wedge (m_2 = 2)) \vee$ $(\downarrow m_9 \wedge (m_2 = 2) \wedge (m_7 = 1)) \vee$ $(\downarrow m_{11} \wedge (m_6 = 2) \wedge (m_3 = 1))$	a blocked situation

Note that if we consider two robots R1, one robot R2 and two robots R3, noted as configuration $2R1/1R2/2R3$, with two mobile agents per production line, the physical system is blocked in two situations: robot R2 is used by line 1 and both robots R3 are used by line 2 ; both robots R1 are used by line 1 and robot R2 is used by line 2. These both situations are detected by the feedback E provided by the physical plant to the controller.

B. Results

Let us now consider the costs associated with an insertion of symbol or a deletion of label by the attacker given by Table III. At the initial state, in configuration $2R1/1R2/2R3$ with two mobile agents per production line, all robots are available and all mobile agents are located at the electric charging station. From the Petri net model, the LFAI could be determined and the uncertain adding control graph is obtained. This UACG, shown in Figure 6, is composed of 41 states where x_0 is the initial state and, x_{18} and x_{19} are deadlocks corresponding to the two blocked situations previously described.

We consider the set of forbidden states given by $\mathcal{F} = \{x_{13}, x_{14}, x_{18}, x_{19}\}$ that correspond to the two deadlocks and two additional dangerous states from which the system cannot come back to a normal state.

The attacker wants to determine an attack, weakly impacted by costs uncertainties, that drives the system from the initial state x_0 to one state of $\mathcal{F}$. According to the approach proposed in this paper, the robust attack i_a^r is computed by evaluating the maximal regret from x_0 to states x_{13}, x_{14}, x_{18} and x_{19} (see Table IV). Thus, according to Proposition 2, the robust attack $i_a^r = aba$ is obtained which corresponds to the smallest

TABLE III: Insertion and deletion costs

Symbol	C_I	Label	C_E
a	$[2, 3]$	A	$[1, 4]$
b	$[1, 2]$	B	$[0, 2]$
c	$[1, 2]$	C	$[1, 2]$
d	$[2, 3]$	D	$[1, 3]$
e	$[1, 5]$	E	$[3, 4]$

TABLE IV: Minimal costs and min-max regret

State	Minimal cost	min-max regret
x_{13}	10	11
x_{14}	9	14
x_{18}	12	21
x_{19}	11	15

regret $r^*_{MA}(x_0, \mathcal{F}) = 11$. The detail of the trajectory is as follow: $x_0 \xrightarrow{a:A} x_1 \xrightarrow{b:B} x_8 \xrightarrow{a:AE} x_{13}$. Due to uncertainties in the insertion and deletion costs (see Table III), the actual cost of the attack aba may vary in range $[11, 21]$. In the worst case, the cost may increase up to 21.

In order to point out the importance of computing the robust attack for both attacker and defender, consider also the attack $i^*_a = aebb$ of minimal cost $c^*_{MA}(x_0, \mathcal{F}) = 9$ that differs from $i^r_a = aba$. It corresponds to the trajectory $x_0 \xrightarrow{a:A} x_1 \xrightarrow{e} x_3 \xrightarrow{b:B} x_7 \xrightarrow{b:BE} x_{14}$. When uncertainties exist, the cost of $aebb$ may vary within $[9, 23]$ and one can observe that, in the worst case the cost of $aebb$ exceeds 21 and that input sequence aba becomes more performent than input sequence $aebb$.

C. Computation time evaluation

To evaluate our approach, a set of simulation has been run using Matlab® on an Intel® Core™ i7-8650U 2.11GHz CPU with 16Go RAM memory. First of all, the time needed to compute the previously robust attack, $i^r_a = aba$, is 286.55 seconds. A simulation campaign has been conducted with different robots configurations, number of mobile agents and, cost intervals.

Table V shows the impact of the number of mobile agents and robots on the size of the UACG and on the computation time. Note that for the first four configurations, the computation time is bounded when the number of mobile agents per line is equal to the total number of robots. In fact, the size of the UACG, and consequently the number of attack trajectories, remain constant. This result can be generalised to all configurations as it is a structural property of the system's OutSynPN model. The number of attack trajectories, $|I_a(x, \mathcal{F})|$, increases exponentially as the size of the UACG (represented by the number of nodes $|X|$ and the number of edges $|V|$) grows. As a consequence, the computational time becomes large (out of time (o.t.) is used when the computational time exceeds 3 hours).

The results of Table VI are obtained for a fixed robots configuration $2R1/1R2/2R2$ and two mobile agents in each production line. Firstly, it is shown how the number of attack trajectories (and thus the computation time) evolves with respect to the number of forbidden states. Secondly, we consider a variation of the cost intervals. The values, based on Table III, are given by $C_I = [c^-_I, c^+_I + \Delta]; C_E = [c^-_E, c^+_E + \Delta]$ where Δ is the parameter. It is important to point out that the variation of the cost intervals does not impact the computation time (insignificant time variation). This result is not surprising as the proposed approach does not consider an exhaustive investigation of all scenarios but considers only extremum scenarios that use minimal and maximal cost bounds.

TABLE V: Simulation results

Robots config.	Agents per line	$\lvert X\rvert$	$\lvert V\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert I_a(x,\mathcal{F})\rvert$	Comp. time (sec.)
Config.1:	1	8	10	2	3	0.016
$1R1$	2	15	24	2	5	0.018
$1R2$	3	17	28	2	7	0.035
$1R3$	4	17	28	2	7	0.035
Config.2:	1	11	17	1	6	0.026
$2R1$	2	28	55	2	332	1.546
$1R2$	3	36	74	2	1282	8.674
$1R3$	4	38	78	2	2389	21.762
	5	38	78	2	2389	21.762
Config.3:	1	11	18	0	–	–
$1R1$	2	25	47	2	37	0.182
$2R2$	3	36	67	3	145	1.071
$1R3$	4	38	71	3	201	1.564
	5	38	71	3	201	1.564
Config.4:	1	10	15	1	4	0.017
$1R1$	2	25	48	2	77	0.359
$1R2$	3	33	66	2	175	1.046
$2R3$	4	35	70	2	231	1.481
	5	35	70	2	231	1.481
Config.5:	1	11	19	0	–	–
$1R1$	2	33	73	1	1671	13.392
$2R2$	3	51	116	2	31525	338.360
$2R3$	4	69	153	3	566839	6947.07
	5	71	157	3	892942	*o.t.*
Config.6:	1	12	21	0	–	–
$2R1$	2	41	91	2	22264	179.670
$1R2$	3	66	155	2	2641486	*o.t.*
$2R3$	4	75	178	2	13299270	*o.t.*
Config.7:	1	22	21	0	–	–
$2R1$	2	89	91	1	16334	161.39
$2R2$	3	146	155	3	4490807	*o.t.*
$1R3$						

TABLE VI: Computation times with different sets $\mathcal{F}$ and variable cost intervals

		Computation time (seconds)				
$\mathcal{F}$	$\lvert I_a(x,\mathcal{F})\rvert$	$\Delta = 0$	$\Delta = 2$	$\Delta = 5$	$\Delta = 10$	$\Delta = 100$
$\{x_{13}\}$	6969	61.80	61.21	62.52	62.75	61.12
$\{x_{14}\}$	2124	19.03	20.49	18.91	18.82	18.95
$\{x_{18}\}$	5064	45.08	45.16	45.64	44.94	44.09
$\{x_{19}\}$	17200	160.64	157.41	157.43	162.94	154.86
$\{x_{14}, x_{18}\}$	7188	64.11	65.65	64.55	63.76	63.04
$\{x_{18}, x_{19}\}$	22264	179.67	203.57	203.07	207.88	198.95
$\{x_{13}, x_{19}\}$	24169	222.44	218.62	219.95	225.69	215.98
$\{x_{13}, x_{14}$ $x_{18}, x_{19}\}$	31357	286.55	284.27	284.50	289.45	279.02

VI. Conclusions

This paper concerns vulnerability analysis of cyber-physical systems that include input and output events. By using a labeled finite automaton with inputs (LFAI) and an uncertain weighted graph that encodes the cost of the malicious actions, we propose an approach to evaluate the robustness of the attack and the vulnerability of the controlled system. This analysis can be used to isolate the weaknesses of the system (in terms of cyber-security) such that improvements can be considered to compensate the vulnerabilities. This approach is illustrated in the case of moving attacks that aim to drive the system's current state into a set of forbidden states. Limitations of the proposed approach lie in the difficulty to estimate the values of the cost intervals for real system, but also in its complexity related to the size of the graph that may grow exponentially.

The solutions obtained by robust optimization techniques such as the min-max regret could be too conservative [22]. Re-

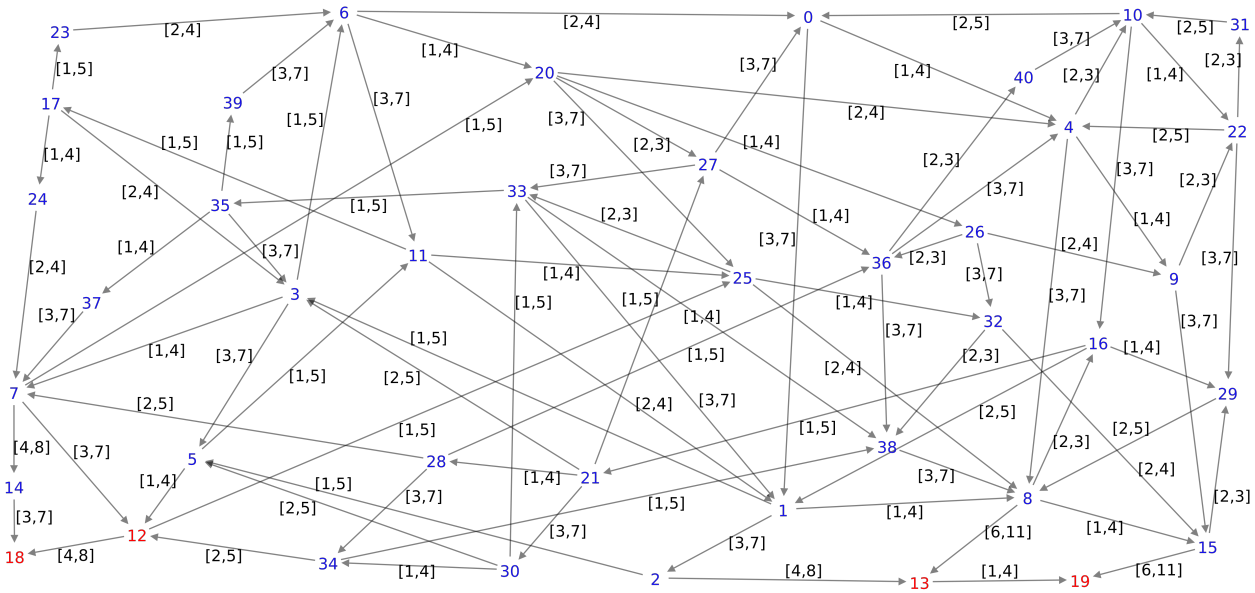

Fig. 6: UACG of the case study

placing the intervals by probabilistic distributions and connecting the probabilistic aspects to the costs leads to several open questions, such as the use of less conservative approaches, that will be considered in our future works. In addition, we will consider more general attack scenarios, including partial controllability and observability properties from the perspective of the attacker. We will also consider situations where the controller and attacker interact like a two-player game. Finally, we will explore situations where the attacker wants to freeze the state of the system in some particular situations.

REFERENCES

[1] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, 2011.

[2] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via gps spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[3] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022.

[4] 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, pp. 109–172, 2020.

[5] L. Lin, Y. Zhu, and R. Su, "Synthesis of covert actuator attackers for free," *Discrete Event Dynamic Systems*, vol. 30, pp. 561–577, 2020.

[6] P. M. Lima, M. V. S. Alves, L. K. Carvalho, and M. V. Moreira, "Security against communication network attacks of cyber-physical systems," *Journal of Control, Automation and Electrical Systems*, vol. 30, no. 1, pp. 125–135, 2019.

[7] A. Mustafa and H. Modares, "Attack analysis and resilient control design for discrete-time distributed multi-agent systems," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 369–376, 2019.

[8] S. Wang, Z. Zhang, and Y. Kadobayashi, "Exploring attack graph for cost-benefit security hardening: A probabilistic approach," *Computers & security*, vol. 32, pp. 158–169, 2013.

[9] J. Zeng, S. Wu, Y. Chen, R. Zeng, and C. Wu, "Survey of attack graph analysis methods from the perspective of data and knowledge processing," *Security and Communication Networks*, vol. 2019, 2019.

[10] A. Hahn, R. K. Thomas, I. Lozano, and A. Cardenas, "A multi-layered and kill-chain based security analysis framework for cyber-physical systems," *International Journal of Critical Infrastructure Protection*, vol. 11, pp. 39–50, 2015.

[11] S. Abraham and S. Nair, "Cyber security analytics: a stochastic model for security quantification using absorbing Markov chains," *Journal of Communications*, vol. 9, no. 12, pp. 899–907, 2014.

[12] A. Sadu, M. Stevic, N. Wirtz, and A. Monti, "A stochastic assessment of attacks based on continuous-time Markov chains," in *6th IEEE Int. Energy Conference (ENERGYCon)*. IEEE, 2020, pp. 11–16.

[13] D. Lefebvre, C. Seatzu, C. N. Hadjicostis, and A. Giua, "Probabilistic state estimation for labeled continuous time Markov models with applications to attack detection," *Discrete Event Dynamic Systems*, pp. 1–24, 2021.

[14] Y. Li, C. N. Hadjicostis, and N. Wu, "Error- and tamper-tolerant decentralized diagnosability of discrete event systems under cost constraints," in *Europ. Control Conf. (ECC), Netherlands, June 29 - July 2*, 2021.

[15] R. Ammour, S. Amari, L. Brenner, I. Demongodin, and D. Lefebvre, "Costs analysis of stealthy attacks with bounded output synchronized Petri nets," in *17th IEEE Int. Conference in Automation Science and Engineering (CASE), France, August 23-27*, 2021, pp. 799–804.

[16] I. Carvalho, A. Coco, T. Noronha, and C. Duhamel, "A min-max regret approach for the steiner tree problem with interval costs," in *52nd Brazilian Operational Research Symposium, Brazil*, 2020, pp. 1–16.

[17] R. Ammour, S. Amari, L. Brenner, I. Demongodin, and D. Lefebvre, "Observer design for output synchronized Petri nets," in *European Control Conference (ECC), Netherlands, June 29 - July 2*, 2021.

[18] N. Revol, "Introduction to the ieee 1788-2015 standard for interval arithmetic," in *Numerical Software Verification: 10th International Workshop, NSV 2017, Heidelberg, Germany, July 22-23, 2017, Proceedings 10*. Springer, 2017, pp. 14–21.

[19] P. Kouvelis and G. Yu, *Robust discrete optimization and its applications*. Springer, Boston, MA, 1997.

[20] H. Aissi, C. Bazgan, and D. Vanderpooten, "Min-max and min-max regret versions of combinatorial optimization problems: A survey," *Europ. Journal of Oper. Research*, vol. 197, no. 2, pp. 427–438, 2009.

[21] O. Karasan, M. Pinar, and H. Yaman, "The robust shortest path problem with interval data," *Optimization Online*, pp. 1–16, 2001.

[22] D. Bertsimas and M. Sim, "The price of robustness," *Operations research*, vol. 52, no. 1, pp. 35–53, 2004.