Attack-Resilient Supervisory Control under Energy-Bounded Attacks [⋆]

Jingshi Yao [*] Shaoyuan Li [*] Xunyuan Yin [**] Xiang Yin [*]

[*] *Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (E-mail: yaojingshi, syli,yinxiang@sjtu.edu.cn).*
[**] *School of Chemistry, Chemical Engineering and Biotechnology, Nanyang Technological University, 62 Nanyang Drive, Singapore, 637459 (E-mail: xunyuan.yin@ntu.edu.sg).*

Abstract: In this paper, we investigate the problem of synthesizing safe supervisors for discrete-event systems under attacks. Specifically, we consider an important class of attacks called the *actuator enablement attacks* (AE-attacks), where an attacker can additionally enable events that are disabled by the supervisor originally. We assume that the attacker consumes certain units of energy each time to launch an attack, and its total energy budget is bounded. Our goal is to synthesize an *attack-resilient* supervisor such that the closed-loop system is still safe under any attack whose capability is constrained by the energy bound. We provide a necessary and sufficient condition for the existence of such an attack-resilient supervisor based on the notion of *safety threshold.* When the existence condition holds, an attack-resilient supervisor can be effectively synthesized by dynamically estimating the remaining energy of the attacker. We show the synthesized supervisor is maximally permissive in terms of the nominal behavior without attack.

Keywords: Energy-Bounded Attacks, DES, Supervisory Control, Safety

1. INTRODUCTION

The supervisory control theory in the context of discrete-event systems (DES) is an important formal methodology for the design of high-level logical controllers for dynamic systems Cassandras and Lafortune (2008). In this framework, uncertainties in the environment are abstracted as uncontrollable events and the design objective is to synthesize a supervisor that disables/enables events dynamically so that the closed-loop system under control satisfies some desired specifications Liu et al. (2022b); Yin and Lafortune (2016).

Due to the development of communication and network technologies, in many modern applications, supervisors are implemented over networks. Such a networked architecture, on the one hand, makes the implementation of the supervisory control systems more flexible, but on the other hand, makes the overall system more vulnerable to cyber-attacks. For example, an attacker may manipulate the sensor readings of the supervisor/actuator by intruding the communication channels so that incorrect control logic will be executed. In the past five years, modeling and control of discrete-event systems under attacks have drawn considerable attention in the literature; see, e.g., Gao et al. (2019); He et al. (2021); Khoumsi (2019); Lin et al. (2020); Liu et al. (2022a); Ma and Cai (2021); Matsui and Cai (2019); Tong et al. (2022); Xie et al. (2022); Yang and Yin (2022); Yao et al. (2020); Zhu et al. (2019) and the recent survey Rashidinejad et al. (2019).

Existing works on cyber-attacks in DES mostly focus on the following problems: (i) how to model different attacks; (ii) how to detect and defend against attacks from the supervisor's point of view; and (iii) how to synthesize attack strategies from the attacker's point of view. For example, in Carvalho et al. (2018), the authors investigate the modeling of three different types of attacks including actuator enablement attacks, sensor erasure attacks, and sensor insertion attacks; attack detection methods as well as defence strategies are also provided. Modeling and detection of cyber-attacks have also been studied in Fritz et al. (2019); Wang et al. (2021) by using Petri nets. In Lin et al. (2019); Meira-Góes et al. (2020, 2019); Su (2018), the problem of synthesizing sensor deception attacks and actuator attackers against a given supervisor are investigated. In Meira-Góes et al. (2020); Wakaiki et al. (2019); Wang and Pajic (2019), the authors investigate how to synthesize robust supervisors that are safe under all possible attacks.

Our work also investigates supervisory control under attacks from the supervisor's point of view. That is, we want to synthesize an attack-resilient supervisor such that the closed-loop system stays safe under possible attacks. In practice, synthesizing an absolutely safe system for any possible attacks is very restrictive or even impossible. For example, in denial-of-service (DoS) attacks, the attacker can always consume more bandwidth than that can be provided by the user to achieve its goal. Therefore, it is of practice to investigate the rationality of the attacker in

[⋆] This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61833012). Xunyuan Yin acknowledges the financial support from Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RS15/21).

terms of the trade-off between its energy consumption and its attack benefits.

To this end, in the work, we formulate an attack-resilient supervisory control problem under *energy-bounded attacks*. Specifically, we consider the class of *actuator enablement attacks* (AE-attacks), where an attacker can additionally enable events that are disabled by the supervisor originally. However, in contrast to existing works, here we investigate this problem more quantitatively by assuming that the attacker will consume certain units of energy each time to launch an attack, and its total energy budget is upper-bounded by a value $\Delta \geq 0$. Our design objective is to design a so-called Δ-safe supervisor, which is safe under any attackers whose total energy consumption along any string is no more than Δ. Our approach essentially follows the idea of "assume-guarantee synthesis" in the sense that we only guarantee the correctness of the solution when the environment's action satisfies the assumption on its total capability and its rationality. In our context, the energy-bound Δ can either be interpreted as the actual energy budget of the attacker, or be interpreted as the security-level the supervisor can provide in the sense that under how powerful attacks the system is still safe.

To solve the attack-resilient supervisor synthesis problem, we generalize the standard supervisor synthesis problem following the idea of "defend the boundary". Specifically, we introduce the notion of safety threshold, which is the smallest energy unit the attacker needs at each state to threaten the safety of the system. In fact, the safety threshold of a state corresponds to the security-level of the system when this state is the initial state. Then the nominal supervisor synthesized tries to prevent the system from reaching states whose safety thresholds are smaller than or equal to the energy bound of the attacker. When the attacker launches an enablement attack, the supervisor may realize the presence of attack by observing events that are not in the nominal behavior. Then the supervisor can estimate the remaining energy budget of the attacker and use this information to improve the permissiveness of the behavior under attack. We show that the synthesized strategy is Δ-safe and maximally permissive in terms of the nominal behavior of the system.

The remaining part of the paper is organized as follows. Some basic preliminaries on supervisory control of DES are introduced in Section 2. In Section 3, we formulate the attack-resilient supervisor control problem under energy-bounded attacks. The solvability of this problem is established in Section 4. In Section 5, the attack-resilient supervisor is proposed for our problem. Finally, we conclude the paper in Section 6.

2. PRELIMINARY

Let Σ be a set of events. A string over Σ is a finite sequence $s = \sigma_1 \cdots \sigma_n, \sigma_i \in \Sigma$. We denote by Σ^* the set of all strings over Σ including the empty string ϵ. A language $L \subseteq \Sigma^*$ is a set of strings. The prefix closure of language L is $\overline{L} = \{s \in \Sigma^* : \exists t \in \Sigma^*, st \in L\}$ and a language L is said to be prefix-closed if $\overline{L} = L$. We say string t is a prefix of s denoted by $t \leq s$ if $t \in \overline{\{s\}}$.

We consider a discrete-event system modeled by a deterministic finite-state automaton (DFA)

$$G = (X, \Sigma, \delta, x_0),$$

where X is a finite set of states, Σ is a finite set of events, $\delta : X \times \Sigma \to X$ is a (partial) transition function such that $\delta(x, \sigma) = x'$ means that there exists a transition from state x to x' labeled by event σ, and $x_0 \in X$ is the initial state. The transition function can be extended to $\delta : X \times \Sigma^* \to X$ recursively in the usual manner. For the sake of simplicity, we write $\delta(x_0, s)$ as $\delta(s)$. Then the language generated by G is defined by $\mathcal{L}(G) = \{s \in \Sigma^* : \delta(s)!\}$, where ! means "is defined". We define $E_G(s) = \{\sigma \in \Sigma : s\sigma \in \mathcal{L}(G)\}$ as the set of active events defined at state $\delta(s)$.

To enforce the system G to satisfy some desired specifications, one is interested in synthesizing a *supervisor*, which disables the occurrences of events dynamically on the system based on its observation. In the context of supervisory control of discrete event systems, we assume that the event set is partitioned as $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$, where Σ_c is the set of controllable events and Σ_{uc} is the set of uncontrollable events. Then a supervisor can be presented by a function:

$$S : \mathcal{L}(G) \to \Gamma,$$

where $\Gamma = \{\gamma \in 2^{\Sigma} : \Sigma_{uc} \subseteq \gamma\}$ is the set of *admissible control decisions* or *control patterns*, which means that uncontrollable events can never be disabled by the supervisor. We denote by $\mathbb{S}(G)$ the set of all possible supervisors for G. The closed-loop system under control is denoted by S/G and the *nominal language* generated by S/G, denoted by $\mathcal{L}(S/G)$, is defined recursively by:

- $\epsilon \in \mathcal{L}(S/G)$;
- for any $s \in \Sigma^*, \sigma \in \Sigma$, we have $s\sigma \in \mathcal{L}(S/G)$ iff $$[s \in \mathcal{L}(S/G)] \wedge [s\sigma \in \mathcal{L}(G)] \wedge [\sigma \in S(s)].$$

In this work, we consider *safety* as the main design objective of the supervisor. Formally, the safety specification is described by a prefix-closed sub-language $K = \overline{K} \subseteq \mathcal{L}(G)$. The supervisor S is said to be *safe* if $\mathcal{L}(S/G) \subseteq K$. In the nominal case, it is well-known that a maximally permissive safe supervisor exists and it can be synthesized by computing the supremal controllable sub-language of K; see, e.g., Cassandras and Lafortune (2008).

Suppose that the specification language K is recognized by a DFA $H = (X', \Sigma, \delta', x_0)$. We assume that H is a *strict sub-automaton* of G, i.e., (i) $\forall s \in \mathcal{L}(H) : \delta(s) = \delta'(s)$; and (ii) $\forall s \in \mathcal{L}(G) \setminus \mathcal{L}(H) : \delta(s) \notin X'$. Note that this assumption is without loss of generality because one can always refine the state-space of G, in polynomial time, such that the strict sub-automation relation holds for the refined system; see, e.g., the appendix of Yin and Lafortune (2017). Then under this assumption, we can define $X_{bad} = X \setminus X'$ as the set of unsafe/illegal states. Therefore, a string s is in K if and only if $\delta(s) \notin X_{bad}$.

3. SUPERVISORY CONTROL UNDER ENERGY-BOUNDED ATTACKS

In the standard supervisory control framework, upon the occurrence of string s, supervisor S will issue a new control decision $S(s)$ and send it to each actuator to prevent events in $\Sigma \setminus S(s)$ from happening. However, in networked cyber-physical systems, such control command may not be perfectly received by the actuators. The control decision value may be manipulated by a malicious attacker, either by intruding the communication channel or by physically attacking the actuators.

In this work, since our main concern for the closed-loop system is safety, there is always no harm if the attacker further restricts the behavior of the system by disabling events additionally. Therefore, we focus on a more meaningful type of attacks called the AE-attacks. Under AE-attacks, a supervisor, whose nominal behavior is safe, may become unsafe because some undesired behaviors that are originally disabled in the nominal language may still occur.

Formally, we denote by $\Sigma_a \subseteq \Sigma_c$ the set of controllable events vulnerable to AE-attacks, and the AE-attacker can be presented by a function:

$$A : \mathcal{L}(G) \to 2^{\Sigma_a}$$

that decides which events to enable additionally based on its current observation.

Then, given system G, supervisor S and AE-attacker A, the total control effect can be considered as the disjunction of their decisions, and the closed-loop language under attack, denoted by $\mathcal{L}(S \vee A/G)$, is defined recursively by

- $\epsilon \in \mathcal{L}(S \vee A/G)$;
- for any $s \in \Sigma^*, \sigma \in \Sigma$, we have $s\sigma \in \mathcal{L}(S \vee A/G)$ iff $$[s \in \mathcal{L}(S \vee A/G)] \wedge [s\sigma \in \mathcal{L}(G)] \wedge [\sigma \in S(s) \cup A(s)]$$

If the AE-attacker can enable evens in Σ_a arbitrarily without any constraint, then it suffices to also consider Σ_a as uncontrollable events from the designer's point of view for the purpose of safety. Then this problem can be solved by the standard SCT. In this work, we consider a more practical setting by assuming that

- the attacker consumes a certain units of energy to launch an enablement attack for events in Σ_a at each instant;
- the total energy of the attacker is bounded.

To this end, we consider a cost function for attacks:

$$w : \Sigma \to \mathbb{N},$$

that assigns each event an attack cost or energy consumption units. Furthermore, we assume that $\forall \sigma \in \Sigma_{uc} : w(\sigma) = 0$, i.e., the attacker is free to enable uncontrollable events because those events are always enabled by default, and $\forall \sigma \in \Sigma_c \setminus \Sigma_a : w(\sigma) = \infty$, i.e., the attacker cannot attack those events that are not vulnerable. We also extend the cost function to $w : \Sigma^* \to \mathbb{N}$ by, for any string $s = \sigma_1\sigma_2 \dots \sigma_n$, we have $w(s) = \sum_{i=1}^{n} w(\sigma_i)$ with $w(\epsilon) = 0$.

Note that, at each state, the attacker may choose to enable a set of events and we assume that the cost is additive. Besides, an event may be activated by the supervisor and attacked by the attacker at the same time where the attacker will not consume energy for this attack. Then for any string $s \in \mathcal{L}(S \vee A/G)$, we use $\mathsf{cost}_A(s)$ to denote the upper limit of the total energy consumption of the attacker as:

$$\mathsf{cost}_A(s) = \sum_{s' \in \overline{\{s\}}} \sum_{\sigma \in A(s')} w(\sigma)$$

In this sense, Let $\Delta \geq 0$ be a non-negative integer that represents an energy budget. Given system G, we say that attacker A is Δ-*bounded* w.r.t. a given supervisor S if

$$\forall s \in \mathcal{L}(S \vee A/G) : \mathsf{cost}_A(s) \leq \Delta.$$

We assume that the cost function w is known to the supervisor. Note that this does not mean the supervisor need to know the exact cost of the attacker. Instead, it is reasonable that the supervisor have some knowledge of the system and can analyze the vulnerability of the system through which the supervisor can have a robust estimate of the real cost function. In fact, the outcome of function w used by the supervisor is larger than the real cost.

Then for a given energy bound Δ, we denote by $\mathbb{A}_\Delta(S)$ the set of all Δ-bounded AE-attackers w.r.t. supervisor S. Our design objective is to synthesize a supervisor such that the closed-loop system is safe under any Δ-bounded AE-attackers, which is formulated as the following *Attack-Resilient Supervisor Control Problem under Energy-Bounded Attacks* (ARSC-EBA).

Problem 1. (ARSC-EBA) Given system $G = (X, \Sigma, \delta, x_0)$, safety specification $K = \overline{K} \subseteq \mathcal{L}(G)$, attack cost function $w : \Sigma \to \mathbb{N}$ and an energy bound $\Delta \geq 0$, synthesize a supervisor S that is safe *under any* Δ-*bounded AE-attacks*, i.e.,

$$\forall A \in \mathbb{A}_\Delta(S) : \mathcal{L}(S \vee A/G) \subseteq K, \tag{1}$$

or equivalently,

$$\forall A \in \mathbb{A}_\Delta(S), \forall s \in \mathcal{L}(S \vee A/G) : \delta(s) \notin X_{bad}. \tag{2}$$

Such a supervisor is also referred to as a Δ-*safe* supervisor.

Remark 1. Essentially, here we are considering an assume guarantee type of synthesis problem in the sense that we only need to guarantee safety under the environment's assumption that the attacker is Δ-bounded. Such an energy-bounded condition has different interpretations in different contexts:

- If the system is known to be working against an attacker that has at most Δ units of energy budget, then clearly such a solution provides us an *attack-resilient* supervisor that guarantees safety by taking the energy limit of the attacker into account.
- In most cases, however, the energy bound of the attacker is unknown. In this case, however, parameter Δ is still meaningful because it can be interpreted as the *security-level* the supervision system can guarantee in the sense that how powerful attacks the supervision system can tolerate robustly.

Here we give a simple example to illustrate how the bounded energy assumption of the attacker affects supervisor synthesis for safe property.

Example 1. Let us consider system G shown in Figure 1, where the set of controllable events is $\Sigma_c = \{b, c, d, e\}$ and the set of vulnerable events is $\Sigma_a = \{c, d, e\}$. The attack cost for each vulnerable event is given by: $w(c) = 1, w(d) = 3$, and $w(e) = 2$. The safety specification is $K = \{\epsilon, a, ad, ab, abc\}$, i.e., $X_{bad} = \{x_4\}$. Let Δ denote the total energy units of the attacker. If $\Delta < 1$, then the intruder does not have enough energy to launch any attack. Then the optimal supervisor S_1 can be obtained

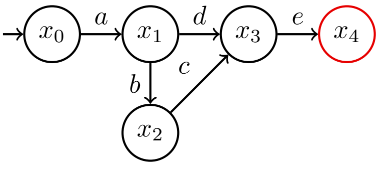


Fig. 1. System G with $X_{bad} = \{x_4\}$ $\Sigma_c = \{b, c, d, e\}$, $\Sigma_a = \{c, d, e\}$, $w(c) = 1, w(d) = 3$ and $w(e) = 2$

by the standard supervisory control algorithm, which just disables event e at state x_3. If $1 \leq \Delta < 2$, then although the attacker can enable event c, S_1 is still safe because event e still cannot be attacked. If $2 \leq \Delta < 3$, then the attacker can enable event e and S_1 is not safe. In order to achieve safety, one needs to use supervisor S_2 that disables event d at state x_1, disables event c at state x_2 and disables event e at state x_3. In this case, if the attack choose to enable c, then it will not have enough energy remaining to enable e. If $3 \leq \Delta < 5$, then S_2 is not safe again and we need to disable event b in addition to S_2. Finally, if $\Delta \geq 5$, then there is no safe supervisor because the attacker can always spend 5 energy units to enable events d and e. Based on the above discussion, we observe that, if the energy bound Δ is smaller than five, then there exists at least one attack-resilient supervisor for G. In other words, for the system G the highest security-level the supervisor can guarantee is 5.

4. EXISTENCE OF THE ATTACK-RESILIENT SUPERVISOR

In this section, we investigate the solvability of Problem 1, i.e., under what condition there exists a Δ-safe supervisor. In order to synthesize an Δ-safe supervisor or to decide whether or not such a supervisor exists, our approach is to think from the attacker's point of view. Specifically, we want to determine, for each state, the minimum energy units an AE-attacker needs in order to threaten the safety of the system. When the system enters a state and this value of the state is smaller than the current energy units of the attacker, then the attacker has the potential to achieve its objective against safety no matter what the supervisor does. This concept is formulated as the *safety threshold* as follows.

Definition 1. Given system $G = (X, \Sigma, \delta, x_0)$ and unsafe states X_{bad}, the safety threshold for state $x \in X$, denoted by $\theta_G(x)$, is the minimum energy units an AE-attack needs to have at state x so that it can make sure that the system is unsafe. Formally, we have

$$\theta_G(x) = \arg\min_{\Delta} \left\{ \Delta : \begin{array}{c} \forall S \in \mathbb{S}(G_x), \exists A \in \mathbb{A}_\Delta(S), \\ \exists s \in \mathcal{L}(S \vee A/G_x), \\ \text{s.t. } \delta(x, s) \in X_{bad} \end{array} \right\}, \quad (3)$$

where $G_x = (X, \Sigma, \delta, x)$ is the DFA by changing the initial state from x_0 to x.

The following result shows how to characterize the solvability of Problem 1 in terms of the safety threshold, which is straightforward based on the definition.

Theorem 1. Problem 1 has no solution if and only if $\theta_G(x_0) \leq \Delta$.

Proof 1. (if) When $\theta_G(x_0) \leq \Delta$, according to the Definition 1, for any supervisor $S \in \mathbb{S}(G)$, there exists an attacker $A \in \mathbb{A}_{\theta_G(x_0)}(S) \subseteq \mathbb{A}_\Delta(S)$ and a string $s \in \mathcal{L}(S \vee A/G)$ such that $\delta(s) \in X_{bad}$. This contradicts to Equation (2) and Problem 1 has no solution. (only if) When $\theta_G(x_0) > \Delta$, according to Definition 1, we know that there exists a supervisor $S \in \mathbb{S}(G)$ such that for any attacker $A \in \mathbb{A}_\Delta(S)$ and string $s \in \mathcal{L}(S \vee A/G)$, we have $\delta(s) \notin X_{bad}$, which means that Problem 1 has at least one solution.

However, the definition of safety threshold itself is mainly motivated by the existential quantifiers and the universal quantifiers in Problem 1. It does not provide a constructive way to compute $\theta_G(x)$ for each state $x \in X$. Recall that in the standard safe supervisor synthesis algorithm without attacks (or the well-known supremal controllable sub-language algorithm), whether or not a state is safe is computed by back-tracking from a unsafe state taking the issue of uncontrollable events into account. In other words, a state is not safe if there exists a sequence of uncontrollable events from this state to an unsafe state in X_{bad}. Here, we are essentially considering a quantitative version of this problem. In particular, we say a state is not Δ-safe w.r.t. attackers with energy budget Δ if there exists a sequence of events from this state to an unsafe state, and the total attack cost incurred along this string is no more than Δ.

Formally, for any state $x \in X$, we denote by $L_{bad}(x) = \{s \in \Sigma^* : \delta(x, s) \in X_{bad}\}$ the set of strings that lead to unsafe states from state x. Then the above discussed idea is formalized by the following result.

Proposition 1. Given system $G = (X, \Sigma, \delta, x_0)$ and unsafe states X_{bad}, for each state $x \in X$, we have

$$\theta_G(x) = \min_{s \in L_{bad}(x)} w(s). \quad (4)$$

Clearly, if $\theta_G(x) < \infty$, then it means that there exists at least one string $s \in (\Sigma_{uc} \cup \Sigma_a)^*$ such that $\delta(x, s) \in X_{bad}$. Then to compute $\theta_G(x)$, it suffices to compute the smallest cost from state x to region X_{bad} via strings where only uncontrollable events and vulnerable events are involved. This is essentially a shortest-path-like problem, which can be done by back-tracking the threshold value from states X_{bad}. Algorithm 1 provides one of the methods to compute the threshold by back-tracking.

Algorithm 1: Compute Safety Threshold

```
Input: controlled system $G = (X, \Sigma, \delta, x_0)$, cost
       function $w$, set of bad states $X_{bad}$
Output: safety threshold $\theta_G$
for $x \in X_{bad}$ do
    $\theta_G(x) \leftarrow 0$
end
for $x \in X \setminus X_{bad}$ do
    $\theta_G(x) \leftarrow \infty$
end
repeat
    for each transition $x = \delta(x', \sigma)$, $\sigma \in \Sigma_{uc} \cup \Sigma_a$ do
        $\theta_G(x') = \min\{\theta_G(x) + w(\sigma), \theta_G(x')\}$
    end
until $\theta_G$ converges;
```

5. SYNTHESIS OF ATTACK-RESILIENT SUPERVISORS

According to Theorem 1, starting from any state $x \in X$, there exists an attack-resilient supervisor against Δ-bounded AE-attacks only when $\theta_G(x) > \Delta$. Specifically, we follow the idea of "defend the boundary" by avoiding reaching states whose safety threshold is no more than Δ. Formally, we define

$$X_{bad}^{\Delta} = \{x \in X : \theta_G(x) \leq \Delta\}$$

as the set of *extended unsafe states* w.r.t. parameter Δ. Therefore, the basic role of the supervisor is to keep the nominal behavior within $X \setminus X_{bad}^{\Delta}$. However, the question is that what if the system under control goes beyond the nominal behavior due to attacks, i.e., when a state in X_{Bad}^{Δ} is reached. Here, we propose the following supervision strategy.

Specifically, if the supervisor observes the occurrence of an event σ that is originally disabled by the nominal supervisor, then it knows for sure that the attacker has spent at least $w(\sigma)$ units of energy. Therefore, the supervisor can deduce that the attack has at most $\Delta' = \Delta - w(\sigma)$ energy budget remained. Then by taking the information into account, the "boundary to defend" can be relaxed from X_{bad}^{Δ} to $X_{bad}^{\Delta'}$. Based on this idea, we propose the notion of *optimistic supervisor* $S_{optm} : \mathcal{L}(G) \rightarrow \Gamma$ that works online according to Algorithm 2. As line 6-8 shows, once the attack behaviour is detected by observing events inconsistent with the nominal behavior, the value of Δ, which is our estimate of the remaining energy budget of the attacker, will decrease. Since X_{bad}^{Δ} becomes smaller, the control decision will have more event choices.

The following theorem shows that the optimistic supervisor is safe against any Δ-bounded AE-attacks.

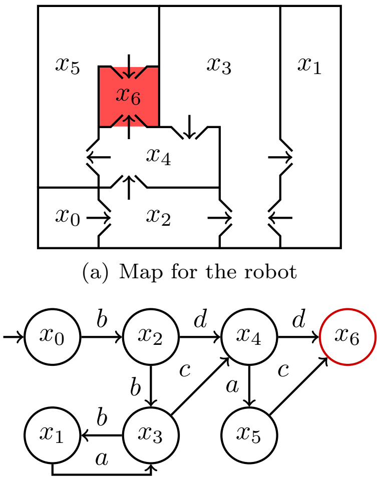
Theorem 2. If $\theta_G(x_0) > \Delta$, then S_{optm} is Δ-safe as defined in Problem 1, i.e., for any $A \in \mathbb{A}_{\Delta}(S_{optm})$, $\forall s \in \mathcal{L}(S_{optm} \vee A/G), \delta(s) \notin X_{bad}$.

Furthermore, we show that the optimistic supervisor is not only optimal (least restrictive) in terms of nominal behavior, but is also maximally-permissive under any attack.

Theorem 3. For any Δ-safe supervisor S', we have

(1) $\mathcal{L}(S'/G) \subseteq \mathcal{L}(S_{optm}/G)$; and
(2) for any $A \in \mathbb{A}_{\Delta}(S_{optm}) \cap \mathbb{A}_{\Delta}(S')$, we further have $\mathcal{L}(S' \vee A/G) \subseteq \mathcal{L}(S_{optm} \vee A/G)$.

Proof. The first result can be regarded as a special case of the second result by adopting the 0-bounded attacker $A : \mathcal{L}(G) \rightarrow \emptyset$. Hereafter, we prove (ii) by contradiction. Let Δ_0 record the initial value of Δ. Assume that there exists a Δ_0-safe supervisor S' such that $\mathcal{L}(S' \vee A/G) \not\subseteq \mathcal{L}(S_{optm} \vee A/G)$ for an attacker $A \in \mathbb{A}_{\Delta_0}(S_{optm}) \cap \mathbb{A}_{\Delta_0}(S')$. This means that there exists a string $s = t\sigma$ such that $\sigma \in S'(t) \setminus S_{optm}(t)$ and $\forall t' < t, S'(t') \subseteq S_{optm}(t')$. Then since $\sigma \notin S_{optm}(t)$, $\delta(t\sigma) \in X_{bad}^{\Delta}$, where $\Delta = \Delta_0 - \sum_{t\sigma \leq s \wedge \sigma \notin S_{optm}(t)} w(\sigma)$ according to Algorithm 2. For the worst case, for any $t\sigma \leq s$, $A(t) = \sigma$ when $\sigma \notin S_{optm}(t)$ and $A(t) = \emptyset$ otherwise. Then the remaining energy of the intruder is $\Delta_0 - \mathsf{cost}_A(s) = \Delta$. So the system under S' will reach state $\delta(s\sigma)$ such that $\theta_G(\delta(s\sigma)) \leq \Delta$ and the attacker has Δ units of energy. Therefore, S' is not Δ_0-safe, which is a contradiction.



(a) Map for the robot

(b) Automaton G in Example 2

Fig. 2. Workspace for the robot and the automaton model in Example 2, where $\Sigma_c = \{a, c, d\}$, $\Sigma_a = \{c, d\}$, $X_{bad} = \{x_6\}$, $w(c) = 1, w(d) = 3$, $w(a) = \infty$ and $w(b) = 0$

We provide an example to illustrate the concept of the safety threshold and the optimistic supervisor.

Algorithm 2: Supervisor Synthesis

Input: $G = (X, \Sigma, \delta, x_0)$, w, $\theta_G(\cdot)$ and Δ
Output: control decision $S_{optm}(s)$ at each instant

```
1  $s \leftarrow \epsilon$
2  $\gamma \leftarrow \{\sigma' \in \Sigma : \delta(s\sigma') \notin X_{bad}^{\Delta}\}$
3  apply control decision $S_{optm}(\epsilon) = \gamma$
4  while event $\sigma$ is observed do
5      $s \leftarrow s\sigma$
6      if $\sigma \notin \gamma$ then
7          $\Delta \leftarrow \Delta - w(\sigma)$
8      end
9      $\gamma \leftarrow \{\sigma' \in E_G(s) : \delta(s\sigma') \notin X_{bad}^{\Delta}\}$
10     apply control decision $S_{optm}(s) = \gamma$
11 end
```

Example 2. We consider a robot moving in a workspace shown in Fig. 2(a). The workspace consists of several rooms connected by one-way doors as depicted in the figure. Each time when the robot enters a room, the supervisor can determine which doors to close, where the rightward doors cannot be closed. Initially, the robot is in room x_0, and our objective is to keep the robot from reaching room x_6. Furthermore, we assume that the downward doors and the upward doors can be forced to open by an attacker with one unit and three units of energy, respectively, for each instant.

We use automaton G to model the mobility of the robot as shown in Fig. 2(b), where events a, b, c and d represent that the robot passes a leftward door, a rightward door, a downward door and a upward door, respectively. Then the set of controllable events set is $\Sigma_c = \{a, c, d\}$, the set of vulnerable events set is $\Sigma_a = \{c, d\}$ and the safety specification is captured by unsafe states $X_{bad} = \{x_6\}$.

The cost function for attacker is $w(c) = 1, w(d) = 3$, $w(a) = \infty$ and $w(b) = 0$. Using Algorithm 1, we can calculate the safety threshold for each state as $\theta_G(x_1) = \infty, \theta_G(x_0) = \theta_G(x_2) = \theta_G(x_3) = 4, \theta_G(x_4) = 3$ and $\theta_G(x_4) = 1$. Now we set the security level for the system as $\Delta = 3$ and synthesize the most permissive supervisor S_{optm} that is resilient to all attackers with energy bound Δ by Algorithm 2. Nominally, S_{optm} will disable event d at state x_2 and disable event c at state x_3. If the supervisor observes event c from state x_3, it will updated Δ to $3 - w(c) = 2$. For this case, we have $X^2_{bad} = \{x_5, x_6\}$ and S_{optm} will disable event a at state x_4. However, when the supervisor observes event d from state x_2, it will update Δ to $3 - w(d) = 0$. For this case, we have $X^0_{bad} = \{x_6\}$ and S_{optm} will enable event a at state x_4.

6. CONCLUSION

In this paper, we formulated and solved an attack-resilient supervisory control problem under the assumption that the total energy of the attacker is bounded. A simple necessary and sufficient condition of the problem was obtained for the existence of Δ-safe supervisor by introducing the concept of safety threshold. Furthermore, we showed how to synthesize a maximally-permissive Δ-safe supervisor so that the closed-loop system is safe under any possible AE-attacks whose energy is upper-bounded by Δ and has minimum constraints. Note that, in this work, we only considered the actuator enablement attacks because the main design objective considered is safety. In future work, we plan to investigated non-blocking supervisor synthesis under both actuator enablement and actuator disablement attacks. Also, we plan to investigate how to synthesize Δ-safe supervisors under the partial observation setting.

REFERENCES

Carvalho, L., Wu, Y., Kwong, R., and Lafortune, S. (2018). Detection and mitigation of classes of attacks in supervisory control systems. *Automatica*, 97, 121–133.

Cassandras, C. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer.

Fritz, R., Schwarz, P., and Zhang, P. (2019). Modeling of cyber attacks and a time guard detection for ICS based on discrete event systems. In *European Control Conference (ECC)*, 4368–4373.

Gao, C., Seatzu, C., Li, Z., and Giua, A. (2019). Multiple attacks detection on discrete event systems. In *IEEE Conf. Systems, Man and Cybernetics*, 2352–2357.

He, Z., Ma, Z., and Tang, W. (2021). Performance safety enforcement in strongly connected timed event graphs. *Automatica*, 128, 109605.

Khoumsi, A. (2019). Sensor and actuator attacks of cyber-physical systems: A study based on supervisory control of discrete event systems. In *8th International Conference on Systems and Control (ICSC)*, 176–182.

Lin, L., Thuijsman, S., Zhu, Y., Ware, S., Su, R., and Reniers, M. (2019). Synthesis of supremal successful normal actuator attackers on normal supervisors. In *American Control Conference (ACC)*, 5614–5619.

Lin, L., Zhu, Y., and Su, R. (2020). Synthesis of covert actuator attackers for free. *Discrete Event Dynamic Systems*, 30, 561–577.

Liu, S., Trivedi, A., Yin, X., and Zamani, M. (2022a). Secure-by-construction synthesis of cyber-physical systems. *Annual Reviews in Control*.

Liu, Z., Yin, X., Shu, S., Lin, F., and Li, S. (2022b). Online supervisory control of networked discrete event systems with control delays. *IEEE Transactions on Automatic Control*, 67(5), 2314–2329.

Ma, Z. and Cai, K. (2021). Optimal secret protections in discrete-event systems. *IEEE Transactions on Automatic Control*.

Matsui, S. and Cai, K. (2019). Secret securing with multiple protections and minimum costs. In *58th IEEE Conference on Decision and Control (CDC)*, 7635–7640.

Meira-Góes, R., Kang, E., Kwong, R., and Lafortune, S. (2020). Synthesis of sensor deception attacks at the supervisory layer of cyber–physical systems. *Automatica*, 121, 109172.

Meira-Góes, R., Kwong, R., and Lafortune, S. (2019). Synthesis of sensor deception attacks for systems modeled as probabilistic automata. In *American Control Conference (ACC)*, 5620–5626.

Rashidinejad, A., Wetzels, B., Reniers, M., Lin, L., Zhu, Y., and Su, R. (2019). Supervisory control of discrete-event systems under attacks: an overview and outlook. In *European Control Conference*, 1732–1739.

Su, R. (2018). Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations. *Automatica*, 94, 35–44.

Tong, Y., Wang, Y., and Giua, A. (2022). A polynomial approach to verifying the existence of a threatening sensor attacker. *IEEE Control Systems Letters*, 6, 2930–2935.

Wakaiki, M., Tabuada, P., and Hespanha, J. (2019). Supervisory control of discrete-event systems under attacks. *Dynamic Games and Applications*, 9(4), 965–983.

Wang, Y., Li, Y., Yu, Z., Wu, N., and Li, Z. (2021). Supervisory control of discrete-event systems under external attacks. *Information Sciences*.

Wang, Y. and Pajic, M. (2019). Attack-resilient supervisory control of discrete-event systems. In *IEEE Conference on Decision and Control (CDC)*, 2015–2020.

Xie, Y., Yin, X., and Li, S. (2022). Opacity enforcing supervisory control using nondeterministic supervisors. *IEEE Tran. Automatic Control*, 67(12), 6567–6582.

Yang, S. and Yin, X. (2022). Secure your intention: On notions of pre-opacity in discrete-event systems. *IEEE Transactions on Automatic Control*.

Yao, J., Yin, X., and Li, S. (2020). On attack mitigation in supervisory control systems: A tolerant control approach. In *59th IEEE Conference on Decision and Control (CDC)*, 4504–4510. IEEE.

Yin, X. and Lafortune, S. (2017). Synthesis of maximally-permissive supervisors for the range control problem. *IEEE Trans. Automatic Control*, 62(8), 3914–3929.

Yin, X. and Lafortune, S. (2016). A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 61(8), 2140–2154.

Zhu, Y., Lin, L., and Su, R. (2019). Supervisor obfuscation against actuator enablement attack. In *European Control Conference*, 1760–1765.