Moving Target Defense based on Switched Supervisory Control: A New Technique for Mitigating Sensor Deception Attacks [1]

Rômulo Meira-Góes * **Stéphane Lafortune** *

* *Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA, (e-mail: {romulo,stephane}@umich.edu)*

Abstract: In this paper, we introduce a new notion of switched supervisory control in the context of discrete event systems. We assume a single event-driven system (plant) controlled by a set of supervisors, where each pair plant/supervisor generates a different event-driven system. A switching mechanism coordinates which plant/supervisor is active at a given time. We investigate the problem of synthesizing a set of supervisors and a switching mechanism such that safety, liveness and maximal permissiveness specifications are satisfied. Sufficient and necessary conditions to solve this problem are presented. Second, a direct application of this new switched supervisory control theory is provided in the context of cyber-security. Namely, this new framework is used as basis for a Moving Target Defense paradigm. Again, sufficient and necessary conditions to solve this problem are presented.

Keywords: Supervisory control theory; Modeling tools: Petri nets, automata; Security.

1. INTRODUCTION

The notion of switched systems has been studied in the control community, e.g., Liberzon (2013); Reveliotis and Fei (2017). In discrete event systems (DES), the idea of switched DES (sDES) was previously investigated in different contexts, e.g., Garcia and Ray (1996); Darabi et al. (2003); Faraut et al. (2009); Macktoobian and Wonham (2017); Reveliotis and Fei (2017), where sDES are composed of event-driven (operational) systems with discrete switching. The problem investigated in Reveliotis and Fei (2017) considers only uncontrollable switches between modes. In Garcia and Ray (1996); Faraut et al. (2009); Macktoobian and Wonham (2017), the problem of reconfiguration of DES was investigated. In reconfiguration of DES, the decision to switch between modes of operation is given by an external higher level decision maker (coordinator). Finally, the switching supervisory control problem when sensor failures occur is investigated in Darabi et al. (2003).

Our work investigates the problem of sDES with controllable switches. More specifically, we assume a single event-driven system (plant) with multiple supervisors, where each pair plant/supervisor generates a different event-driven system. This new framework is denoted as switched supervisory control theory (swSCT). In this framework, we investigate the problem of synthesizing a set of supervisors and a switching mechanism such that safety, liveness and maximal permissiveness specifications are satisfied. This problem is hereafter referred to as the switched supervisory control (swSC) problem. Sufficient and necessary conditions to solve this problem are presented.

Our work differs from previous works in several respects. First, the switching mechanism decides when to switch modes which implies that the switching is controllable. Second, the switching mechanism is part of the underlying controlled system which differs from the external coordinator approach in reconfiguration of DES. Lastly, our switching mechanism continually switches between supervisors and it is not only triggered by sensor failures.

The framework of swSCT does not offer any advantage compared to the standard framework of SCT with respect to language generation. Namely, it is known that there exists a unique maximally permissive supervisor that generates the supremal controllable sublanguage of safety and liveness specifications. Nonetheless, an interesting property arises in the swSC framework. Namely, the dynamic switching between supervisors creates a controlled nondeterministic behavior. This dynamical change of modes (here, supervisors) has advantages over the static single supervisor framework. These advantages are shown in the context of cyber-security.

Cyber-security has been recently investigated in SCT, where the supervisory control framework is assumed to be under attack (Rashidinejad et al., 2019). In this work, we focus on a specific type of attack class denoted as sensor deception attacks. Prior work on sensor deception attacks in the field of DES so far can be divided into three classes of problems: (i) designing attack strategies for *fixed supervisors* (Meira-Góes et al., 2017; Su, 2018; Meira-Góes et al., 2019a,b); (ii) designing intrusion detection modules for *fixed supervisors* (Carvalho et al., 2018; Lima et al., 2018); and (iii) designing robust supervisors against sensor deception attacks (Wakaiki et al., 2018; Su, 2018; Meira-Góes et al., 2019c,d).

In this paper, we propose the first application, to the best of our knowledge, of the defense mechanism of Moving Target Defense to DES. Moving Target Defense (MTD) is a proactive defense paradigm used to mitigate attacker's effectiveness in systems (Jajodia et al., 2011). The framework of swSCT provides the necessary results to study the problem of MTD in the supervisory control framework. Namely, the switching mechanism provides the proactive defense mechanism for the controlled system. The problem investigated, denoted as the MTD problem, is to synthesize an swSC that mitigates the attacker's actions.

[1] The work of R.M.G. and S.L. was supported in part by US NSF grants CNS-1738103 and CNS-1801342.

The contributions of our paper are two fold. First, we introduce a new framework of swSC in DES and provide necessary and sufficient conditions for the existence of an swSC. Second, a direct application of this new swSCT is provided. Namely, this new framework is used in the context of the MTD paradigm. We provide a sufficient condition for the non-existence of supervisors such that the actions of a sensor deception attacker are mitigated. Moreover, we provide an exhaustive algorithm that provides necessary and sufficient conditions for the existence of supervisors that solve the MTD problem.

The paper is organized as follows. Section 2 introduces the supervisory control theory background. The framework of switched supervisory control theory (swSCT) and its synthesis problem are presented in Section 3. In Section 4, we investigate the swSC framework under sensor deception attacks. Finally, the moving target defense problem and discussion about its solution are presented in Section 5. We conclude the paper in Section 6. Several proofs are omitted due space limitations.

2. SUPERVISORY CONTROL THEORY

We consider systems that are modeled by deterministic or nondeterministic automata.

Definition 1. A nondeterministic finite-state automaton (NFA) is a tuple $G = (X_G, \Sigma, \delta_G, X_{0,G})$, where X_G is the finite set of states, Σ is the finite set of events, $\delta_G \subseteq X_G \times \Sigma \times X_G$ is the transition relation, and $X_{0,G} \subseteq X_G$ is the set of initial states. G is deterministic (DFA) if $|X_{0,G}| = 1$ and $(x, e, y_1), (x, e, y_2) \in \delta_M$ implies $y_1 = y_2$ for all $x \in X_G$ and $e \in \Sigma$. For a DFA, we use $\delta_G : X_G \times \Sigma \to X_G$ as a partial transition function and $x_{0,G}$ as the initial state.

The set of all finite strings of events in Σ is denoted by Σ^*. For any string $s = s[1]s[2]\dots s[|s|] \in \Sigma^*$, $s[i] \in \Sigma$ denotes the ith event of s and s^i denotes the ith prefix of s, i.e., $s^i = s[1]\dots s[i]$ and $s^0 = \epsilon$. Finally, $\mathbb{N}$, $[n]$ and $[n]^+$ are the set of natural numbers, the set of natural numbers bounded by n, and the set of positive natural numbers bounded by n, respectively.

The transition relation δ_G is extended in the usual manner to $X_G \times \Sigma^* \times X_G$. By an abuse of notation, we define $\delta_G(X, s) = \{y \in X_G \mid \exists x \in X \text{ s.t. } (x, s, y) \in \delta_G\}$ for $s \in \Sigma^*$. The same notions apply to a DFA. The language generated by G is defined by $\mathcal{L}(G) = \{s \in \Sigma^* | \delta_G(X_{0,G}, s)!\}$, where ! means that the function is defined for these arguments.

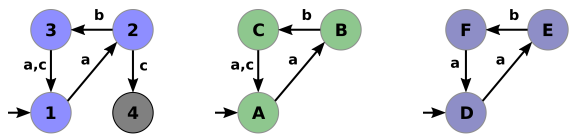
A *run* in G is defined as a sequence $x_0 e_1 x_1 e_2 \dots e_n x_n \in (X_G \Sigma)^* X_G$ such that $(x_i, e_{i+1}, x_{i+1}) \in \delta_G$ for $i \in [n]$; a run is *initial* if $x_0 \in X_{0,G}$. We denote by $Runs(G, X)$ the set of all runs in G starting from X and $Runs(G) = Runs(G, X_{0,G})$. Each $s \in \mathcal{L}(G)$ generates at least one run in $Runs(G)$; and if G is a DFA, then the run generated by s is unique.

A DFA G, in supervisory control theory, is considered as the *plant* (uncontrolled system) that needs to be controlled to meet a desired specification. The event set Σ is partitioned as $\Sigma = \Sigma_c \cup \Sigma_{uc}$, where Σ_c (Σ_{uc}) is the set of controllable (uncontrollable) events. A supervisor dynamically enables/disables controllable events of the plant so that a desired specification is achieved. Formally, a supervisor is defined as $S : \Sigma^* \to \Gamma$, where $\Gamma = \{\gamma \subseteq \Sigma | \Sigma_{uc} \subseteq \gamma\}$ is the set of admissible control decisions. The closed-loop behavior of the controlled system is denoted by S/G and defined by the language $\mathcal{L}(S/G)$; see, e.g., Cassandras and Lafortune (2008). Without loss of generality, S is realized by a DFA $R = (X_R, \Sigma, \delta_R, x_{0,R})$. For convenience, let $R(x) = \{e \in \Sigma \mid \delta_R(x, e)!\}$ be the control decision issued by R in state $x \in X_R$.

In this work, we are interested in safety and liveness specifications. Without loss of generality, we assume that there exists a unique $x_{crit} \in X_G$ such that x_{crit} is the *only deadlock state in* G. The safety specification is given by $K = Ac(G, x_{crit})$, where the accessible operator Ac deletes state x_{crit} from G. We define $\mathcal{C}(\mathcal{L}(K)) = \{L \subseteq \mathcal{L}(K) \mid L = pre(L),\ L \text{ is live, and } L\Sigma_{uc} \cap \mathcal{L}(G) \subseteq L\}$ to be the set of all controllable sublanguages of $\mathcal{L}(K)$, where $pre(L)$ is the set of all prefixes of L and L is live if and only if its automaton representation has no deadlocks. It is known that the supremal element of this set exists and is achieved by a DFA (Cassandras and Lafortune, 2008; Yin and Lafortune, 2016). We define the maximally permissive supervisor R_{max} to be a fixed supervisor realization such that $\mathcal{L}(R_{max}/G)$ is the supremal element of $\mathcal{C}(\mathcal{L}(K))$.

Finally, we recall the parallel composition operator $||$ as defined in Cassandras and Lafortune (2008). Given two DFAs G, R, $G||R = (X_{G||R}, \Sigma, \delta_{G||R}, x_{0,G||R})$, where $x = (x_1, x_2) \in X_{G||R}$ with $x_1 \in X_G$ and $x_2 \in X_R$. The same concept applies when the operator $||$ is applied to multiple DFAs.

Example 1. Consider the plant G depicted in Fig. 1(a), where $\Sigma_c = \{b, c\}$ and the critical state $x_{crit} = 4$. The supervisors R_1 and R_2, shown in Fig. 1(c) and Fig. 1(b), guarantee that the controlled system is safe and live. Supervisor R_1 is the maximally permissive supervisor.



(a) Plant G (b) Supervisor R_1 (c) Supervisor R_2

Fig. 1. Running example

3. SWITCHED SUPERVISORY CONTROL

3.1 Problem description

In this section, we describe the framework of switched supervisory control. The idea of switched supervisory control is inspired by the notion of switched hybrid systems (Liberzon, 2013).

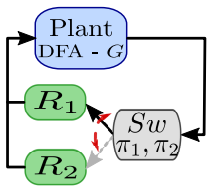


Fig. 2. The switching supervisory control framework

The framework of switched supervisory control is depicted in Fig. 2. We have a plant G that is controlled by two supervisors R_1, R_2. For simplicity, we assume that there are only two supervisors but this can be generalized to a finite set of supervisors. At any given point only one supervisor is controlling the plant and a switching mechanism, depicted as Sw in Fig. 2, controls which supervisor is controlling the plant. Moreover, the switching mechanism "re-initializes" the supervisors when a switch happens using functions π_1, π_2. In other words, Sw

controls the switching between supervisors R_1, R_2 and π_1, π_2 control the updates of the supervisor controlling G.

Similar problems as the ones in switched hybrid systems, e.g., stability, safety, existence of a switching mechanism, etc., must be addressed in this switched supervisory control framework. In this work, we want to investigate the problem of constructing a switching mechanism for a set of supervisors such that safety and liveness specifications on G are satisfied.

3.2 Problem Formulation

Informally, the switching mechanism works as follows. Let R_1 be the supervisor controlling the plant G when G executes event e. The switching mechanism receives the information that e was executed and decides which supervisor controls the plant next. If Sw decides that the next supervisor is R_1 then π_1 updates R_1. On the other hand, if Sw decides that the next supervisor is R_2 then π_2 must restart R_2 based on the previous information received. Therefore, the switching mechanism is defined based on functions Sw, π_1, and π_2.

For simplicity and without loss of generality, we formalize these definitions using a state-based formalization instead of a language-based one. Namely, the switching function is defined as the nondeterministic function $Sw : X_G \to 2^{[2]^+}$ and the update functions as the nondeterministic $\pi_i : X_G \times (X_{R_i} \cup \{sw\}) \times \Sigma \to 2^{X_{R_i}}$. Inspired by Liu et al. (2013), we define a valid behavior of the switched controlled system.

Definition 2. A *valid run* $\phi : \mathbb{N} \to \Sigma \times X_G \times [2]^+ \times \cup_{i=0}^{2} X_{R_i}$ of the switched controlled system defined by G, R_1, R_2, Sw, π_1, and π_2 is a tuple $\phi(i) = (e_i, x_i, \sigma_i, y_i)$ that satisfies for $i \in \mathbb{N}$:

(1) $\phi(0) \in \{(\epsilon, x_{0,G}, 1, x_{0,R_1}), (\epsilon, x_{0,G}, 2, x_{0,R_2})\}$
(2) $e_{i+1} \in R_{\sigma_i}(y_i)$
(3) $x_{i+1} = \delta_G(x_i, e_{i+1})$
(4) $\sigma_{i+1} \in Sw(x_{i+1})$
(5) $y_{i+1} \in \begin{cases} \pi_{\sigma_{i+1}}(x_{i+1}, y_i, e_{i+1}) & \text{if } \sigma_{i+1} = \sigma_i \\ \pi_{\sigma_{i+1}}(x_{i+1}, sw, e_{i+1}) & \text{if } \sigma_{i+1} \neq \sigma_i \end{cases}$

The set of all valid runs in the switched controlled system is denoted as Φ.

Remark 1: Although the valid runs of the switched control system are defined for live systems (no deadlocks), finite runs can be defined by augmenting Σ with a "deadlock" event.

Remark 2: As was mentioned, the swSC framework *does not* provide any advantage compared to the standard SCT nor with the nondeterministic SC framework (Fabian and Lennartson, 1996) with respect to language generation. However in the case of the standard SCT, the supervisors are deterministic and generate deterministic controlled behavior while the swSC generates a nondeterministic controlled behavior. In the case of the nondeterministic SC, nondeterministic supervisors are allowed. However, the structure of the switching framework better captures the relationship among supervisors which allow us to enforce properties involving the switching mechanism. These would be challenging to express in the nondeterministic SC framework. These points will be made clear in Sections 4 and 5.

For a valid run ϕ with $\phi(i) = (e_i, x_i, \sigma_i, y_i)$, let $\Psi(\phi(i)) = e_i$ be the event projection of $\phi(i)$. By an abuse of notation, we define $\Psi(\phi[i,j]) = e_i \dots e_j$ for $j \geq i$ and $\Psi(\phi) = \cup_{i \geq 0} \Psi(\phi[0,i])$.

Definition 3. The language generated by the switched controlled system defined by G, R_1, R_2, Sw, π_1 and π_2, denoted as $\mathcal{L}(\{R_i, \pi_i\}_{i=0}^{2} /_{Sw} G)$, is defined as:

$$\mathcal{L}(\{R_i, \pi_i\}_{i=0}^{2} /_{Sw} G) := \bigcup_{\phi \in \Phi} \Psi(\phi) \tag{1}$$

A string $s \in \mathcal{L}(\{R_i, \pi_i\}_{i=0}^{2} /_{Sw} G)$ is generated by a valid run in the switched controlled system, i.e., $\exists \phi \in \Phi$ s.t. $s = \Psi(\phi[0, |s|])$. Such runs are not unique and it is possible that $\exists \phi_1, \phi_2 \in \Phi$ s.t. $s = \Psi(\phi_1[0, |s|]) = \Psi(\phi_2[0, |s|])$.

Now we are ready to formally state the switching synthesis problem.

Problem 1. Given a system G with a deadlock critical state x_{crit}, a specification $K = Ac(G, x_{crit})$, synthesize, if they exist, supervisors R_1, R_2, update functions π_1, π_2 and a switching mechanism Sw such that:

(1) $\mathcal{L}(R_1/G) \neq \mathcal{L}(R_2/G)$
(2) $\mathcal{L}(R_1/G)$, $\mathcal{L}(R_2/G) \in \mathcal{C}(\mathcal{L}(K))$
(3) $\mathcal{L}(\{R_i, \pi_i\}_{i=0}^{2} /_{Sw} G)$ is the supremal element of $\mathcal{C}(\mathcal{L}(K))$; and,
(4) $\forall s \in \mathcal{L}(\{R_i, \pi_i\}_{i=0}^{2} /_{Sw} G)$, $\exists t \in \Sigma^* \setminus \{\epsilon\}$ s.t. $st \in \mathcal{L}(\{R_i, \pi_i\}_{i=0}^{2} /_{Sw} G)$ and $Sw(\delta_G(x_{0,G}, st)) = \{1, 2\}$.

Based on conditions (1), (2) and (3) of Problem 1, we look for a switched supervisory control system that satisfies a desired specification and generates the supremal controllable sublanguage of this specification. Condition (4) in Problem 1 states that the switched system cannot be "stuck" in only one supervisor. Namely, it is always possible to switch supervisors at some future point.

3.3 Solution

Our solution methodology leverages the results of synthesizing supervisors in the standard supervisory control framework. In this manner, we partition the problem of synthesizing supervisors R_1, R_2 and the switching mechanism into two problems. First, we select supervisors R_1, R_2 based on the standard supervisory control framework. Based on these supervisors, we define a switching mechanism to obtain a solution to Problem 1.

In our presentation, we invert the solution sequence. Given supervisors R_1, R_2, we present the switching mechanism for them. Next, we present some properties of this mechanism. Following that, we provide a solution for Problem 1 by choosing appropriate supervisors R_1, R_2. We start by defining the function Sw for two given safe and live supervisors.

Definition 4. Let R_1, R_2 be two supervisors s.t. $\mathcal{L}(R_1/G)$, $\mathcal{L}(R_2/G) \in \mathcal{C}(\mathcal{L}(K))$, the switching function Sw is defined for any $x \in X_G$ as:

$$Sw(x) = \{i \in [2]^+ \mid \exists y \in X_{R_i} \text{ s.t. } (x, y) \in X_{G||R_i}\} \tag{2}$$

Next, we define the supervisor update functions π_1 and π_2.

Definition 5. Let R_1, R_2 be two supervisors s.t. $\mathcal{L}(R_1/G)$, $\mathcal{L}(R_2/G) \in \mathcal{C}(\mathcal{L}(K)$, the supervisor update function π_i, for $i \in [2]^+$, is defined for any $x \in X_G$, $y \in X_{R_i} \cup \{sw\}$, and $e \in \Sigma$ as:

$$\pi_i(x, y, e) = \begin{cases} \{\delta_{R_i}(y, e)\} & \text{if } y \in X_{R_i} \\ \{z \in X_{R_i} \mid (x, z) \in X_{G||R_i}\} & \text{if } y = sw \end{cases} \tag{3}$$

To help us solve Problem 1, we define an NFA based on supervisors R_1, R_2, the switching function Sw and the supervisor update functions π_1, π_2.

Definition 6. Let R_1, R_2 be two supervisors s.t. $\mathcal{L}(R_1/G)$, $\mathcal{L}(R_2/G) \in \mathcal{C}(\mathcal{L}(K))$, the switching mechanism Sw defined by Eq. (2) and the update functions π_1, π_2 defined by Eq. (3), we define the NFA $M(R_1, R_2) = (X_M, \Sigma, \delta_M, X_{0,M})$ as follows:

1: $X_M \subseteq X_G \times \bigcup_{i=1}^{2} X_{R_i} \times [2]^+$

2: Σ

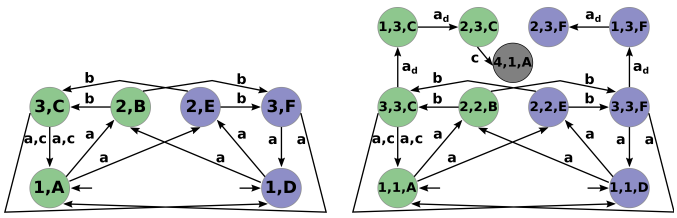
3: $\big((x_1, x_2, i), e, (y_1, y_2, i)\big) \in \delta_M$ if $\begin{cases} e \in R_i(x_2) \\ y_1 = \delta_G(x_1, e) \\ i \in Sw(y_1) \\ y_2 \in \pi_i(y_1, x_2, e) \end{cases}$

$\big((x_1, x_2, i), e, (y_1, y_2, j)\big) \in \delta_M$ if $\begin{cases} j \neq i \\ e \in R_i(x_2) \\ y_1 = \delta_G(x_1, e) \\ j \in Sw(y_1) \\ y_2 \in \pi_j(y_1, sw, e) \end{cases}$

4: $X_{0,M} = \{(x_{0,G}, x_{0,R_1}, 1), (x_{0,G}, x_{0,R_2}, 2)\}$

The NFA $M(R_1, R_2)$ captures all valid behaviors of the switched controlled system defined based on R_1, R_2, the switching mechanism, and the update functions inferred from the given R_1, R_2, based on Equations (2) and (3). Moreover, $\mathcal{L}(M(R_1, R_2)) = \mathcal{L}(\{R_i, \pi_i\}_{i=1}^{2}/_{Sw}G)$ by construction of M (to see this, it suffices to compare the conditions in Definitions 2 and 6). When there is no confusion, we use $M = M(R_1, R_2)$. The following example clarifies the construction of M.

Example 2. We return to our running example. We build the switched controlled system based on supervisors R_1, R_2 depicted in Fig. 1. Figure 3(a) shows the NFA M that captures the behavior of the switched controlled system. We omit the indices of the active supervisors since their state names are non-conflicting. Supervisor R_1 is active on green states while supervisor R_2 is active on blue states. It is important to note that the switched controlled system M does not reach the critical state in this example. Next, we provide general guarantees for this result.



(a) Switched controlled system M

(b) Attacked switched controlled system M_a

Fig. 3. Switched controlled systems

The switching criterion and the update functions are important constraints on the switched controlled system. Without correctly constraining the switching and the updates according to Eqs. (2) and (3), the switched controlled system could be unsafe, i.e., it reaches the critical state x_{crit}. Next, we show that $\mathcal{L}(M)$ is indeed safe and live.

First, we state the following lemma related to specification K.

Lemma 1. Let G be a plant, $K = Ac(G, x_{crit})$ be a safety specification on G, and R be a supervisor such that $\mathcal{L}(R/G) \in \mathcal{C}(\mathcal{L}(K))$. For any $(x, y) \in X_{G||R}$ s.t. $\exists e \in \Sigma$ and $\delta_G(x, e) = x_{crit}$, then $e \notin R(y)$.

Since the specification K is a state-based specification, knowing the pair $(x, y) \in X_{G||R}$ is sufficient to allow us to take the correct control decisions such that the specification is not violated. Lemma 1 states this condition based on a one-step transition. If the pair $(x, y) \in X_{G||R}$ is one event away from reaching the critical state, then supervisor R always disables the event that reaches the critical state. This property is crucial for the safety of M since the definitions of Sw, π_1 and π_2 depend on $G||R_1$ and $G||R_2$.

Theorem 2. $(x_1, x_2, i) \in X_M$ if and only if $(x_1, x_2) \in X_{G||R_i}$.

As a direct consequence of Theorem 2, we have the following corollary.

Corollary 1. $\mathcal{L}(M(R_1, R_2)) \in \mathcal{C}(\mathcal{L}(K))$.

Corollary 1 is an important result since it guarantees that the switching mechanism defined by Definition 4 and Definition 5 is safe and live as long as the selected supervisors are safe and live. Therefore, a safe and live switched controlled system can be constructed by leveraging the standard results in supervisory control theory.

The following property of the switched controlled system follows from the construction of $M(R_1, R_2)$.

Proposition 3. Given $M(R_1, R_2)$ then $\mathcal{L}(R_i||G) \subseteq \mathcal{L}(M(R_1, R_2))$ for $i \in [2]^+$.

It is also important to characterize the switched controlled system when one of the supervisors generates the supremal element of $\mathcal{C}(\mathcal{L}(K))$.

Lemma 4. Let $R_1 = R_{max}$ be a maximally permissive supervisor based on $\mathcal{C}(\mathcal{L}(K))$. For any supervisor $R_2 \neq R_{max}$ such that $\mathcal{L}(R_2/G) \in \mathcal{C}(\mathcal{L}(K))$, we have that $\mathcal{L}(M(R_1, R_2)) = \mathcal{L}(R_1||G)$.

Theorem 5. Let $R_1 = R_{max}$, R_2 be any supervisor s.t. $\mathcal{L}(R_1/G) \neq \mathcal{L}(R_2/G)$ and $\mathcal{L}(R_2/G) \in \mathcal{C}(\mathcal{L}(K))$, Sw defined based on Definition 4 and π_1, π_2 defined based on Definition 5, then $R_1, R_2, Sw, \pi_1, \pi_2$ are a solution of Problem 1 if for any $(x_1, x_2) \in X_{G||R_1}$ there exist $t \in \Sigma^* \setminus \{\epsilon\}$ and $z \in X_{R_2}$ s.t. $(y_1, y_2) = \delta_{G||R_1}((x_1, x_2), t)$ and $(y_1, z) \in X_{G||R_2}$.

Proof. Condition (1) of Problem 1 is satisfied by the assumption that $\mathcal{L}(R_1/G) \neq \mathcal{L}(R_2/G)$. Conditions (2) and (3) are satisfied via Lemma 4. We only need to prove condition (4). It is known that since $K \sqsubset G$ the supervisor $R_1 \sqsubset G$, where $\sqsubset$ means a strict subautomaton (see Cassandras and Lafortune (2008); Yin and Lafortune (2016)). Intuitively, $R_{max} = Ac(G, X_{del})$ for some set $X_{del} \subseteq X_G$ and $x_{crit} \in X_{del}$. In this manner, the set $X_{max} = X_G \setminus X_{del}$ contains all possible states that can be reached safely by a supervisor and do not lead to a deadlock. The states of G reached when controlled by any other supervisor R_2 is a subset of X_{max}. For this reason, R_2 can always switch to R_1. If R_1 satisfies that for any $(x_1, x_2) \in X_{G||R_1}$ there exist $t \in \Sigma^* \setminus \{\epsilon\}$ and $z \in X_{R_2}$ s.t. $(y_1, y_2) = \delta_G((x_1, x_2), t)$ and $(y_1, z) \in X_{G||R_2}$, then R_1 can always switch back to R_2. □

Theorem 5 provides a sufficient condition for Problem 1. Namely, a way to obtain a solution for Problem 1 is to select the maximally permissive supervisor and any other supervisor that satisfies Theorem 5. However, it is possible that there does not exist any other supervisor that satisfies Theorem 5.

A naive algorithm to check if Problem 1 has a solution is to iterate over a finite number of possible supervisors. Note that, this is possible since K is a state based specification. Even though there exists an infinite number of supervisors, it

is possible to iterate within a finite number of them in order to provide a complete solution of this problem. In this paper, we only describe this algorithm intuitively and we leave for future work to improve the efficiency of this naive algorithm. This algorithm is based on results presented in Yin and Lafortune (2016).

The algorithm is to construct the All-Enforcement-Supervisor (AES) structure described in Yin and Lafortune (2016) based on G and K. Based on the AES, we select two supervisors and verify if they satisfy all the conditions to be a solution for Problem 1. If the selected supervisors are a solution for Problem 1, then we know that Problem 1 has a solution. Otherwise, we select two other supervisors and repeat the test. If no supervisor pair satisfy the conditions of Problem 1, then Problem 1 has no solution.

4. ATTACKED SWITCHED SYSTEM DESCRIPTION

4.1 Notation

We define $\Sigma_a \subseteq \Sigma$ as the set of compromised events, i.e., the attacker can insert/delete events in this set on the communication channel between the sensors and the supervisor. We use subscripts to identify attacker modifications; the sets $\Sigma_i = \{e_i \mid e \in \Sigma_a\}$ and $\Sigma_d = \{e_d \mid e \in \Sigma_a\}$ are the sets of inserted and deleted events, respectively. Events without subscripts are legitimate events generated by the plant G, whereas events with subscripts are events altered by the attacker. Finally, let $\Sigma_m = \Sigma \cup \Sigma_d \cup \Sigma_i$ be the complete event set.

The mask $\mathcal{M} : \Sigma_m \to \Sigma$ removes the subscripts from events in $\Sigma_d \cup \Sigma_i$, i.e., $\mathcal{M}(e_d) = \mathcal{M}(e_i) = e$. Let P^G (P^R) be a projection operator that projects events in Σ_m to events in Σ generated by the plant (observed by the supervisor). Namely, P^G outputs the event that is executed in G, i.e., $P^G(e_i) = \epsilon$ and $P^G(e_d) = P^G(e) = e$. On the other hand, P^R outputs the event observed by the supervisor, i.e, $P^R(e_d) = \epsilon$ and $P^R(e_i) = P^R(e) = e$. Lastly, we denote $e^G = P^G(e)$ and $e^R = P^R(e)$ as the plant projection and the supervisor observation of event $e \in \Sigma_m$.

4.2 Attacked switched controlled systems

We assume that the attacker has the power of hijacking the events in Σ_a in the communication channel between the plant and the supervisor. It means that the attacker can insert fictitious information about these events in the channel or it can delete the information about these events from the channel, see e.g., Meira-Góes et al. (2017); Su (2018); Meira-Góes et al. (2019a). We also assume that the attacker has full knowledge of the system, i.e., the attacker knows the model of the plant G, the model of the supervisors R_1, R_2, and the switching mechanism Sw, π_1 and π_2. In other words, it knows the NFA M. Moreover, we assume that the attacker observes the same events as the switching mechanism. These assumptions are common in security analysis of systems. In our case, it means that we are studying the worst-case scenario.

On the other hand, the attacker does not know which supervisor is active at a given point, it can only infer which supervisor is active based on observations. Only the switching mechanism knows which supervisor is active.

Intuitively, the attacker observes the events executed by G and based on Σ_a, it applies some *strategy* to attain its goal. Since we do not have any information about the attack strategy used by the attacker, we assume that the attacker could use any strategy constrained only by Σ_a. This procedure is called the *"all-out"* attack strategy and it was introduced in Carvalho et al. (2018); Lima et al. (2018), where the attacker attacks whenever it is possible. Although this model is simple, it is well-suited for the investigated problem since we want to analyze the switched controlled system under an arbitrary sensor deception attack.

Based on the "all-out" attack strategy, we analyze the behavior of the switched controlled system under attack. Namely, we construct an NFA M_a that captures how an attacker affects a given switched controlled system. First, we slightly modify supervisors R_1 and R_2 such that they embed an intrusion detection module.

Based on supervisors R_1, R_2, we construct supervisors $\tilde{R}_1, \tilde{R}_2$ that are equivalent to R_1, R_2, and distinguish $\mathcal{L}(R/G)$ from the language in $\mathcal{L}(G) \setminus \mathcal{L}(R/G)$. Intuitively, $\tilde{R}_i = (X_{\tilde{R}_i} = X_{R_i} \cup \{dead\}, \Sigma, \delta_{\tilde{R}_i}, x_{0,\tilde{R}_i})$ is a copy of R_i augmented with a deadlock state called $dead$ that is only reached via strings in $\mathcal{L}(G) \setminus \mathcal{L}(R_i/G)$, where $i \in [2]^+$. The state $dead$ is used to capture an intrusion detection mechanism, where an attacker is detected when the supervisor reaches this state.

Definition 7. Given a switched controlled system defined by the NFA $M(R_1, R_2)$ and the set of compromised events, we define the attacked switched controlled system as the NFA $M_a(\tilde{R}_1, \tilde{R}_2) = (X_{M_a}, \Sigma_m, \delta_{M_a}, X_{0,M_a})$ as follows:

1: $X_{M_a} \subseteq X_G \times X_G \times \bigcup_{i=1}^{2} X_{R_i} \times [2]^+$

2: Σ_m

3: $\big((x_1, x_2, x_3, i), e, (y_1, y_2, y_3, i)\big) \in \delta_{M_a}$ if

$$\begin{cases} \mathcal{M}(e) \in \tilde{R}_i(x_3) \\ y_1 = \delta_G(x_1, e^G) \\ y_2 = \delta_G(x_2, e^S) \\ i \in Sw(y_2) \\ y_3 \in \pi_i(y_2, x_2, e^S) \end{cases}$$

$\big((x_1, x_2, x_3, i), e, (y_1, y_2, y_3, j)\big) \in \delta_{M_a}$ if

$$\begin{cases} j \neq i \\ \mathcal{M}(e) \in \tilde{R}_i(x_3) \\ y_1 = \delta_G(x_1, e^G) \\ y_2 = \delta_G(x_2, e^S) \\ j \in Sw(y_2) \\ y_3 \in \pi_j(y_2, sw, e^S) \end{cases}$$

4: $X_{0,M_a} = \{(x_{0,G}, x_{0,G}, x_{0,R_1}, 1), (x_{0,G}, x_{0,G}, x_{0,R_2}, 2)\}$

Similarly as M, the DFA M_a embeds all the information about the attacked system. Let us clarify the definition of M_a by comparing with the definition of M. States in M_a have two elements in X_G, one more than the states in M. These two elements in X_G capture the exact state of G (based on the executed events) and the fictitious state of G (based on the modifications made by the attacker). The other two elements of the states of M_a are exactly the same as in M.

The transition function δ_{M_a} is quite similar to δ_M. First, we update both the exact and the fictitious states of G based on $e^G = P^G(e)$ and $e^S = P^S(e)$, respectively. Next, the updates of the supervisor state and the active supervisor index are performed based on the fictitious state of G since the switching mechanism receives the events modified by the attacker.

In M_a, we denote by $L_{un}(M_a)$ to be the set of all unsafe strings in M_a. Namely, any string $s \in \mathcal{L}(M_a)$ such

that $x_0 s[1] \dots s[|s|] x_{|s|} \in Runs(M_a)$ and $x_{|s|} = (x_{crit}, x^2_{|s|}, x^3_{|s|}, j)$.

We show an example to illustrate the DFA M_a.

Example 3. Back to our running example, we construct the attacked switched controlled system M_a based on M shown in Fig. 3(a) and $\Sigma_a = \Sigma_{uc} = \{a\}$. Part of M_a is shown in Fig. 3(b), where unsuccessful attack strategies were removed and the *dead* states of each supervisor are omitted. Note that, an attacker can reach the critical state $(4, 1, A)$, in gray, if R_1 is the active supervisor when the state pair of G (correct state and fictitious state) is $(3, 3)$. On the other hand, the same attack strategy fails to reach the critical state if R_2 is the active supervisor when the state pair of G is $(3, 3)$. It is exactly this result that we exploit in the next section.

5. MOVING TARGET DEFENSE PROBLEM

5.1 Intuition on the problem formulation

Let us analyze Example 1, where a maximally permissive supervisor enforces the safety and liveness of the plant. This supervisor is fixed since the safety and liveness specifications for the controlled system are static. This static nature of the supervisory control framework becomes a liability for this controlled system when a sensor deception attacker is included in this framework. In fact, a simple attack strategy reaches the critical state without being detected by an intrusion detection module. Analyzing only the green states in Fig. 3(b) shows that deleting event a twice when the plant is in state 3 is sufficient to reach state x_{crit}.

Even adding a defense mechanism does not help in this scenario since this defense mechanism is designed based on the static nature of the supervisory control framework. As described previously, the traditional defense techniques are based on fixed parameters or static techniques, e.g., fixed plant, fixed supervisor and fixed intrusion detection modules. These vulnerabilities can be exploited by a persistent attacker to develop its strategy.

Moving Target Defense (MTD) is a new proactive defense paradigm used to mitigate attacker's effectiveness in systems. MTD protocols countermeasure attacker actions by dynamically and unpredictability changing the parameters of the system. This countermeasure creates uncertainty to the attacker, which makes it more difficult for the attacker to succeed.

Back to our example, let us analyze the switched supervisory control framework presented in Example 2. Although the behavior allowed by this switched controlled system is the same as in Example 1, a sensor deception attacker might not succeed if it attacks the switched controlled system. The attacker might fail since it does not know which supervisor is currently active. Back to Fig. 3(b) but now analyzing the entire figure shows that when the state pair of the plant is $(3, 3)$, either R_1 or R_2 can be active (states $(3, 3, C)$ and $(3, 3, F)$). The attacker is successful if R_1 is the active supervisor. On the other hand, the attacker fails to reach the the critical state if R_2 is active since the controlled system reaches a deadlock.

5.2 Problem formulation

Based on the previous discussion we state the following problem, named the Moving Target Defense (MTD) supervisory control problem.

Problem 2. Given a system G and Σ_a find supervisors R_1, R_2, switching mechanism Sw, π_1 and π_2 such that:

(1) The supervisors R_1, R_2 and the switching mechanism Sw, π_1, π_2 defined based on Definitions 4 and 5 are a solution for Problem 1; and,

(2) $\forall s \in L_{un}(M_a)$, $\exists k < |s|$ s.t. s^k generates a run $(x^1_0, x^2_0, x^3_0, i_0) s[1] \dots s[k] (x^1_k, x^2_k, x^3_k, i_k) \in Runs(M_a)$ with:
   (a) $s[k+1] \notin \tilde{R}_{i_k}(x^3_k)$; and
   (b) $\forall (y^1_0, y^2_0, y^3_0, j_0) t[1] \dots t[|t|] (y^1_{|t|}, y^2_{|t|}, y^3_{|t|}, j_{|t|}) \in Runs(M_a, (x^1_k, x^2_k, x^3_k, i_k))$, then $y^1_l \neq x_{crit}$ for $l \in [|t|]$.

The first condition of Problem 2 simply states that we need to find supervisors R_1, R_2 that are a solution for Problem 1 disregarding the attacker. The second condition states that actively switching between these supervisors reduces the effectiveness of sensor deception attacks. Any attack that may (nondeterministically) be successful, may also (again, nondeterministically) be prevented by the switching to a different supervisor, which happens unbeknown to the attacker. By "prevented", we mean that at some point during the attack, some desired action (insertion or deletion) of the attacker will not be possible. This is the meaning of the index k in condition (2) of Problem 2. Moreover, from that point on, the attacker will never have a way to steer the system to the critical state.

Condition (2) of Problem 2, as stated, is a strong requirement. One could weaken it to condition (2a) alone, which would mean that the attacker is *temporarily* unable to reach the critical state.

5.3 Existence of supervisors

First, we use the results of the work in Meira-Góes et al. (2019c,d) to derive a sufficient condition for the non-existence of a solution for Problem 2. In Meira-Góes et al. (2019c,d), the problem of synthesizing robust supervisors against sensor deception attacks is investigated. Next, we use the AES structure mentioned in Section 3 to provide necessary and sufficient conditions on the existence of these supervisors.

We informally introduce the concept of a robust supervisor against sensor deception attacks, for a formal definition see Meira-Góes et al. (2019c,d). Following the definition of robust supervisors, we restate an important result about synthesis of a robust supervisor.

Definition 8. (Meira-Góes et al., 2019d) A supervisor R is robust against sensor deception attacks over the set $\Sigma_a \subseteq \Sigma$, if $\nexists (x_{crit}, x_2, x_3, 1) \in X_{M_a(R,\emptyset)}$, where $M_a(R, \emptyset)$ is the attacked system constructed based only on supervisor R.

Sufficient and necessary conditions for the existence of a robust supervisors w.r.t. Σ_a are presented in Meira-Góes et al. (2019c,d). An efficient algorithm to compute a supervisor realization R_{rob} that realizes a supremal controllable and robust sublanguage of the specification $\mathcal{L}(K)$ is provided in Meira-Góes et al. (2019d). Based on supervisor R_{rob}, we state a sufficient condition for Problem 2.

Theorem 6. Let $R_1 = R_{max}$, then if $R_{rob} = \emptyset$, then $\nexists R_2$ such that R_1, R_2 form a solution of Problem 2.

Theorem 6 states a sufficient condition for the non-existence of a solution for Problem 2. If we select $R_1 = R_{max}$ and the robust supervisor w.r.t. Σ_a does not exist (empty supervisor), then there is no supervisor R_2 such that R_1, R_2 form a solution for Problem 2. However, a non-empty R_{rob} does not guarantee a solution for this problem. Similarly to the solution of Problem 1, we present an exhaustive algorithm that provides sufficient and necessary conditions for Problem 2.

Again, we use the AES structure from Yin and Lafortune (2016) to present necessary and sufficient conditions to solve Problem 2. There are finitely many supervisors that we can select in the AES to check if a solution for Problem 2 exists. In other words, we select two supervisors from the AES, then we verify if these two supervisors are a solution for our problem.

Example 4. We go back to our running example. Supervisor R_2, shown in Fig. 1(b), is the robust supervisor of G w.r.t. to $\Sigma_a = \{a\}$. Therefore, this system has a robust supervisor, which implies that it is possible that a solution for Problem 2 exists. In fact, R_1 and R_2 depicted in Fig. 1 are a solution for Problem 2. As we have analyzed the attacked switched controlled system M_a at the beginning of this section, the attacker is successful if R_1 is the active supervisor when the pair of states of the plant is $(3, 3)$. However, it is unsuccessful if R_2 is the active supervisor in this same pair of states. This exactly satisfies the conditions of Problem 2.

6. CONCLUSION

We presented a new framework for switched supervisory control theory (swSCT). In this new framework, a single plant is controlled by two supervisors, where only one supervisor is active at a certain point and a switching mechanism coordinates which supervisor is active. We provided necessary and sufficient conditions for the existence of supervisors and a nondeterministic switching mechanism such that the switched controlled system enforces safe, live and maximally permissive specifications. An exhaustive algorithm on how to obtain the supervisors and the switching mechanism is provided. It is left for the future work to investigate an efficient algorithm for this problem.

The nondeterminism created by the active switching between these supervisors is exploited in the context of cyber-security in SCT. Namely, the swSCT framework provides the necessary results to study the moving target defense paradigm in SCT. A second problem was posed in this context, where we investigated the problem of synthesizing an swSC such that it mitigates the actions of a sensor deception attacker. We provided a sufficient condition for the non-existence of such switched controlled systems based on previous results on the synthesis of robust supervisors against this class of attacks. Also, necessary and sufficient conditions are provided for this problem based on an exhaustive algorithm. It is left for future work to investigate an efficient algorithm for this problem.

REFERENCES

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

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition.

Darabi, H., Jafari, M.A., and Buczak, A.L. (2003). A control switching theory for supervisory control of discrete event systems. *IEEE Transactions on Robotics and Automation*, 19(1), 131–137.

Fabian, M. and Lennartson, B. (1996). On non-deterministic supervisory control. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 2, 2213–2218 vol.2.

Faraut, G., Pietrac, L., and Niel, E. (2009). Formal approach to multimodal control design: Application to mode switching. *IEEE Transactions on Industrial Informatics*, 5(4), 443–453.

Garcia, H.E. and Ray, A. (1996). State-space supervisory control of reconfigurable discrete event systems. *International Journal of Control*, 63(4), 767–797.

Jajodia, S., Ghosh, A.K., Swarup, V., Wang, C., and Wang, X.S. (2011). *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*. Springer Publishing Company, Incorporated, 1st edition.

Liberzon, D. (2013). Switched systems : Stability analysis and control synthesis.

Lima, P.M., Carvalho, L.K., and Moreira, M.V. (2018). Detectable and undetectable network attack security of cyber-physical systems. In *14th IFAC Workshop on Discrete Event Systems WODES 2018*, volume 51, 179 – 185.

Liu, J., Ozay, N., Topcu, U., and Murray, R.M. (2013). Synthesis of reactive switching protocols from temporal logic specifications. *IEEE Transactions on Automatic Control*, 58(7), 1771–1785.

Macktoobian, M. and Wonham, W.M. (2017). Automatic reconfiguration of untimed discrete-event systems. In *2017 14th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 1–6.

Meira-Góes, R., Kang, E., Kwong, R., and Lafortune, S. (2017). Stealthy deception attacks for cyber-physical systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 4224–4230.

Meira-Góes, R., Kang, E., Kwong, R., and Lafortune, S. (2019a). Synthesis of sensor deception attacks at the supervisory layer of cyber-physical systems. *under review*.

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

Meira-Góes, R., Lafortune, S., and Marchand, H. (2019c). Synthesis of supervisors robust against sensor deception attacks. *under review*.

Meira-Góes, R., Marchand, H., and Lafortune, S. (2019d). Towards resilient supervisors against sensor deception attacks. In *2019 IEEE 58th Annual Conference on Decision and Control (CDC)*.

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 *2019 18th European Control Conference (ECC)*, 1732–1739.

Reveliotis, S. and Fei, Z. (2017). Invariant-based supervisory control of switched discrete event systems. *IEEE Transactions on Automatic Control*, 62(2), 921–927.

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

Wakaiki, M., Tabuada, P., and Hespanha, J.P. (2018). Supervisory control of discrete-event systems under attacks. *Dynamic Games and Applications*.

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.