Efficient Synthesis of Sensor Deception Attacks Using Observation Equivalence-Based Abstraction

Downloaded from: https://research.chalmers.se, 2021-06-23 02:25 UTC

Citation for the original published paper (version of record):
Mohajerani, S., Meira-Góes, R., Lafortune, S. (2020)
Efficient Synthesis of Sensor Deception Attacks Using Observation Equivalence-Based Abstraction
IFAC-PapersOnLine, 53(4): 28-34
http://dx.doi.org/10.1016/j.ifacol.2021.04.069

N.B. When citing this work, cite the original published paper.
Efficient Synthesis of Sensor Deception Attacks Using Observation-Based Abstraction

Sahar Mohajerani∗ Rômulo Meira-Góes**
Stéphane Lafortune**

∗ Department of Electrical Engineering,
Chalmers University of Technology, Göteborg, Sweden,
(e-mail: mohajeran@chalmers.se)
** Department of Electrical Engineering and Computer Science,
University of Michigan, Ann Arbor, MI, USA, (e-mail:
{romulo,stephane}@umich.edu)

Abstract: This paper investigates the synthesis of successful sensor deception attack functions in supervisory control using abstraction methods to reduce computational complexity. In sensor deception attacks, an attacker hijacks a subset of the sensors of the plant and feeds incorrect information to the supervisor with the intent on causing damage to the supervised system. The attacker is successful if its attack causes damage to the system and it is not identified by an intrusion detection module. The existence test and the synthesis method of successful sensor deception attack functions are computationally expensive, specifically in partially observed systems. For this reason, we leverage results on abstraction methods to reduce the computational effort in solving these problems. Namely, we introduce an equivalence relation called restricted observation equivalence, that is used to abstract the original system before calculating attack functions. Based on this equivalence relation we prove that the existence of successful attack functions in the abstracted supervised system guarantees the existence of successful attack functions in the unabstracted supervised system and vice versa. Moreover, successful attack functions synthesized from the abstracted system can be exactly mapped to successful attack functions on the unabstrcted system, thereby providing a complete solution to the attack synthesis problem.

Keywords: Automaton, Deception Attacks, Abstraction, Supervisory Control Theory.

1. INTRODUCTION

Over the past decades, human dependability on cyber-
physical systems has rapidly increased. Usually, these
systems are highly complex, and appear in settings that
are safety critical, where small failures may result in
huge financial and/or human losses. These failures may
be caused by external attacks that manipulate the sensor
measurements received by the controller or the supervisor.
This class of attacks is called sensor deception attacks.

Prior work on sensor deception attacks in the field of Dis-
crete Event Systems (DES) focuses on characterizing suc-
cessful attack strategies for fixed supervisors (Meira-Góes
et al., 2017; Su, 2018; Meira-Góes et al., 2019a; Meira-Góes
et al., 2019c), on designing intrusion detection modules for
fixed supervisors (Thorsley and Teneketzis, 2006; Carvalho
et al., 2018; Lima et al., 2019), or on designing robust
supervisors against sensor deception attacks (Wakaiki et
al., 2018; Su, 2018; Meira-Góes et al., 2019d; Meira-Góes
et al., 2019b); see (Rashidinejad et al., 2019) for a review in
this area. These works mostly focus on the effect of the at-
tacker on the supervisory control framework. Formally, the
attacker is modelled by an attack function that edits the
behavior generated by the system and feeds this edited be-
havior to the supervisor. However, these recent results do
not provide computationally efficient methods, especially
in the case of partially observed systems. For more infor-
mation on the supervisory control framework under sensor
decception attacks see (Meira-Góes et al., 2017; Meira-Góes
et al., 2019a).

This work focuses on synthesizing successful sensor deception attack strategies for fixed supervisors in an efficient manner. The work of (Meira-Góes et al., 2019a) provides a methodology to solve this problem based on a game-graph structure called the All Insertion-Deletion Attack (AIDA) structure. The AIDA structure captures the interaction between the supervisor and the environment (which includes the system and the attacker). Based on the AIDA, a successful attack strategy, if one exists, can be obtained.

The construction of the AIDA is based on the dis-
crete model of the system and the supervisor that con-
tains admissible control decisions and differentiates nor-
mal/abnormal behavior. Usually the discrete model of a
cyber-physical system is complex and as a result the
computation of the AIDA could be potentially costly. To
mitigate this issue, in this paper an abstraction method
is investigated to reduce the size of the system before
calculating the AIDA. The proposed abstraction method

1 The work of the first author was supported by the Swedish
Research Council (VR 2016-00529). The work of the second and third
authors was supported in part by US NSF grants CNS-1738103 and
CNS-1801342.
is based on observation equivalence, which is a well-known abstraction method (Milner, 1989). Observation equivalence considers states as equivalent if they have the same future behavior. Observation equivalence however, can not be used to abstract a system before calculating the AIDA and some adjustments are necessary. Abstraction-based observation equivalence was used in (Zhang and Zaman, 2017; Mohajerani and LaFortune, 2019; Mohajerani et al., 2019) in opacity setting.

This paper introduces a restricted version of observation equivalence, which is used to abstract the system before calculating the AIDA. Since the abstraction method reduces the size of the system by merging some states the computational complexity of the AIDA can be reduced. Moreover, the abstraction method can be calculated in polynomial time and it guarantees that successful attack functions synthesized from the abstracted system can be exactly mapped to successful attack functions on the original system.

The presentation of our results is organized as follows. Sect. 2 gives a brief background on modeling and the All Insertion-Deletion Attack structure. Next, Sect. 3 explains the abstraction method that is used in this paper and shows how it is leveraged for synthesizing successful attack functions. Finally, some concluding remarks are given in Sect. 4.

2. SYSTEM AND ATTACK MODEL
2.1 Supervisory Control System Model

Discrete system behaviors can be modeled by deterministic or nondeterministic automata.

Definition 1. A (nondeterministic) finite-state automaton is a tuple \( G = (\Sigma, Q, \delta, q_0) \), where \( \Sigma \) is a finite set of events, \( Q \) is a finite set of states, \( \delta : Q \times \Sigma \rightarrow Q \) is the state transition relation, and \( q_0 \in Q \) is the initial state. \( G \) is deterministic, if \( x \xrightarrow{\sigma} y \) and \( x \xrightarrow{\sigma} y \) always implies \( y_1 = y_2 \).

\( \Sigma^* \) is the set of all finite traces of events from \( \Sigma \), including the empty trace \( \varepsilon \).

The transition relation is written in infix notation \( x \xrightarrow{\sigma} y \), and is extended to strings in \( \Sigma^* \) by letting \( x \xrightarrow{\sigma} y \) for all \( x \in Q \), and \( x \xrightarrow{\sigma} z \) if \( x \xrightarrow{\sigma} y \) and \( y \xrightarrow{\sigma} z \) for some \( y \in Q \). Furthermore, \( x \xrightarrow{\sigma} y \) means that \( x \xrightarrow{\sigma} y \) for some \( y \in Q \), and \( x \xrightarrow{\sigma} y \) if \( x \xrightarrow{\sigma} y \) for some \( y \in Q \), and \( x \xrightarrow{\sigma} y \) for \( t \in \Sigma^* \). These notations also apply to state sets, \( X \xrightarrow{\sigma} Y \) for \( X, Y \subseteq Q \) means that \( x \xrightarrow{\sigma} y \) for some \( x \in X \) and \( y \in Y \), and to automata, \( G \xrightarrow{\sigma} \) means that \( x \xrightarrow{\sigma} Y \), etc.

The language of an automaton \( G \) is \( \mathcal{L}(G) = \{ s \in \Sigma^* \mid G \xrightarrow{\sigma} \} \). In addition, for \( q \in Q \) let \( \Gamma_G(q) = \{ \sigma \in \Sigma \mid q \xrightarrow{\sigma} \} \) be the set of active events at state \( q \). By a slight abuse of notation, we write \( \Gamma_G(S) = \cup q \in G \Gamma_G(q) \) for \( S \subseteq Q \).

One common automaton operation is the quotient modulo an equivalence relation on the state set.

Definition 2. Let \( Z \) be a set. A relation \( \sim \subseteq Z \times Z \) is called an equivalence relation on \( Z \) if it is reflexive, symmetric, and transitive. Given an equivalence relation \( \sim \) on \( Z \), the equivalence class of \( z \in Z \) is \( \{ z' \in Z \mid z \sim z' \} \), and \( Z = \{ \{ z \} \mid z \in Z \} \) is the set of all equivalence classes modulo \( \sim \).

Definition 3. Let \( G = (\Sigma, Q, \delta, q_0) \) be an automaton and let \( \sim \subseteq Q \times Q \) be an equivalence relation. The quotient automaton of \( G \) modulo \( \sim \) is

\[ G = (\Sigma, \tilde{Q}, \tilde{\delta}, \tilde{q}_0) \],

where \( \tilde{\delta} : \tilde{Q} \times \Sigma \rightarrow \tilde{Q} \) and \( \tilde{q}_0 \in \tilde{Q} \). In supervisory control theory, an automaton \( G \), called the plant, models the uncontrolled behavior of a system. This automaton is controlled by a supervisor \( S_P \) that dynamically enables and disables the controllable events such that it enforces some safety property on \( G \). The limited actuation capabilities of \( G \) are modeled by a partition of the event set \( \Sigma = \Sigma_u \cup \Sigma_{uc} \), where \( \Sigma_{uc} \) is the set of uncontrollable events and \( \Sigma_u \) is the set of controllable events. In the notation of the theory of supervisory control of DES initiated in (Ramadge and Wonham, 1987), the resulting controlled behavior is a new DES denoted by \( S_P/G \) and resulting in the closed-loop language \( \mathcal{L}(S_P/G) \), which is defined in the usual manner (Cassandras and LaFortune, 2008). The set of admissible control decisions is defined as \( \Gamma = \{ \gamma \subseteq \Sigma \mid \Sigma_{uc} \subseteq \gamma \} \), where admissibility guarantees that a control decision never disables uncontrollable events.

In addition, due to the limited sensing capabilities of \( G \), the event set is also partitioned into \( \Sigma = \Sigma_o \cup \Sigma_{uo} \), where \( \Sigma_o \) is the set of observable events and \( \Sigma_{uo} \) is the set of unobservable events. Based on this second partition, the projection function \( P : \Sigma^* \rightarrow \Sigma_o^* \) is defined as \( P(\varepsilon) = \varepsilon \), and for \( s \in \Sigma_o^* \), \( e \in \Sigma \) then \( P(\varepsilon) = P(s) = P(s) \) if \( e \in \Sigma_o \) or \( P(\varepsilon) = P(s) \) if \( e \in \Sigma_{uo} \). The inverse projection \( P^{-1} : \Sigma_o^* \rightarrow 2^{\Sigma^*} \) is defined as \( P^{-1}(t) = \{ s \in \Sigma^* \mid P(t) = s \} \).

For brevity, \( p \xrightarrow{\sigma} q \), with \( s \in \Sigma_o^* \), denotes the existence of a string \( t \in \Sigma^* \) such that \( P(t) = s \) and \( p \xrightarrow{\sigma} \).

Similarly, \( p \xrightarrow{\sigma} q \) means there exists \( t \in \Sigma^* \) such that \( p \xrightarrow{\sigma} \).

Formally, a partial observation supervisor is a function \( S_P : \mathcal{L}(G) \rightarrow \Gamma \). Without loss of generality, we assume that \( S_P \) is realized (i.e., encoded) as a deterministic automaton \( R = (Q_R, \Sigma_o \xrightarrow{\rightarrow} R, q_0) \), see (Cassandras and LaFortune, 2008). Based on supervisor \( R \), we construct a supervisor \( R_{dead} \) that is equivalent to \( R \) but captures \( P(\mathcal{L}(R/G)) \) and also detects the language \( P(\mathcal{L}(G)) \) \( P(\mathcal{L}(R/G)) \). Formally, \( R_{dead} = (Q_{R_{dead}} = Q_R \cup \{ dead \}, \Sigma_o \xrightarrow{\rightarrow} R, q_0) \) is a copy of \( R \) augmented with a deadlock state called dead that is only reached via strings in \( P(\mathcal{L}(G)) \) \( P(\mathcal{L}(R/G)) \).

The supervisor \( R_{dead} \) allows us to merge an intrusion detection mechanism with supervisor \( R \), where strings that reach the dead state are detectable.

For convenience, we define two operators that are used in this paper together with some useful notation. The unobservable reach of the subset of states \( S \subseteq Q \) under the subset of events \( \gamma \subseteq \Sigma \) is given by:

\[ U_R(\gamma)(S) := \{ x \in Q \mid \exists u \in S \{ \exists s \in (\Sigma_{uo} \cap \gamma)^* \text{ s.t. } u \xrightarrow{\sigma} x \} \} \]

The observable reach (or next states) of the subset of states \( S \subseteq Q \) given the execution of the observable event \( e \in \Sigma_o \) is defined as:

\[ NX_e(S) := \{ x \in Q \mid \exists u \in S \text{ s.t. } u \xrightarrow{\sigma} x \} \]
2.2 The Insertion-Deletion Attack Structure

The problem introduced in (Meira-Góes et al., 2017; Meira-Góes et al., 2019a) is to synthesize an attack function such that a critical state is reachable and the state dead is unreachable in the attacked controlled behavior.

In (Meira-Góes et al., 2019a), the framework of supervisory control under sensor deception attacks is introduced, where an attacker is modelled by an attack function. We assume that $Q$ contains a set of critical states defined as $Q_{crit} \subseteq Q$, which are never reached when $SP$ controls $G$ and no attacker is present. The problem of synthesizing “successful” attack functions is posed, where means successful an attacker that causes the system to reach $Q_{crit}$ without being detected. A two-player structure called the All-Insertion-Deletion Attack structure (AIDA) is introduced in (Meira-Góes et al., 2019a) as a general solution methodology to synthesize stealthy deception attacks against a fixed supervisor $R_{dead}$. The AIDA captures the interaction between the supervisor and the environment that is defined by the plant executions and attacker actions.

Namely, the AIDA enumerates all deception attack actions and all possible plant executions based on the control decisions of the supervisor. Thus, we define the pair $IS \in 2^{Q} \times Q_{dead}$ to be the information state, and $I = 2^{Q} \times Q_{dead}$ the set of all information states. As defined, an $IS$ embeds the necessary information for either the supervisor or the environment to make a decision.

In order to construct the AIDA the set of compromised events $\Sigma_a \subseteq \Sigma_n$ needs to be introduced. This event set specifies the events that the attacker can edit. To identify the attacker actions, let $\Sigma' = \{e_i | e \in \Sigma_n\}$ and $\Sigma^d = \{e_d | e \in \Sigma_n\}$ be the sets of inserted and deleted events, respectively. These events sets clearly identify the attacker actions at the communication channel between the plant $G$ and the supervisor $R_{dead}$. Note that the subscripts are introduced for convenience and the supervisor receives the actual event execution in $G$. Let $\Sigma_m = \Sigma \cup \Sigma' \cup \Sigma^d$ be the complete event set.

We define $M(e_i) = M(e_d) = M(e) = e$ for $e \in \Sigma$, $P^G(e) = M(e)$ for $e \in \Sigma \cup \Sigma^d$ and $P^d(e) = e$ for $e \in \Sigma'$, and $P^S(e) = M(e)$ for $e \in \Sigma \cup \Sigma'$ and $P^S(e) = e$ for $e \in \Sigma^d$. The mask $M$ removes substitutions, when present, from events in $\Sigma_m$, the projection $P^G$ projects an event in $\Sigma_m$ to its actual event execution in $G$, and the projection $P^S$ projects an event in $\Sigma_m$ to its event observation by $R_{dead}$.

For simplicity, in Def. 4 $\mu_d$ is considered as the transition function of $R_{dead}$, where $\mu_d(q, e) = p$ is the same as $q \xrightarrow{e_d} p$.

Definition 4. (Meira-Góes et al., 2019a) An All-Insertion-Deletion Attack structure (AIDA) $A$ w.r.t. $G$, $\Sigma_n$, and $R_{dead}$, is defined as

$$A = (Q_S, Q_E, h_{SE}, h_{ES}, \Sigma_m, y_0)$$

where:

- $Q_S \subseteq I$ is the set of $S$-states, where $S$ stands for Supervisor and where each $S$-state is of the form $y = (I_G(y), I_S(y))$, where $I_G(y)$ and $I_S(y)$ denote the state estimate of the plant and the state of the supervisor, respectively;
- $Q_E \subseteq I$ is the set of $E$-states, where $E$ stands for Environment; each $E$-state is defined in the same way as in the $S$-states case;
- $h_{SE} : Q_S \times \Gamma \rightarrow Q_E$ is the partial transition function from $S$-states to $E$-states, defined for $y \in Q_S$ and $\gamma = \Gamma_{dead}(I_S(y))$ as:

$$h_{SE}(y, \gamma) := \{(UR_\gamma(I_G(y)), I_S(y))\} \quad (5)$$

- $h_{ES} : Q_E \times (\Sigma_o \cup \Sigma' \cup \Sigma^d) \rightarrow Q_S$ is the partial transition function from $E$-states to $S$-states. It is defined as follows: for any $y \in Q_S$, $(z_g, z_h) \in Q_E$ and $e \in \Sigma_o \cup \Sigma' \cup \Sigma^d$, $h_{ES}(z_g, z_h, e) = \gamma$ where:

$$y = \begin{cases} (NX_c(z_g), \mu_d(z_h, P^S(e))) & \text{if } e \in Q_{dead} \cap \Gamma G(z_g) \\ (z_g, \mu_d(z_h, P^S(e))) & \text{if } e \in \Sigma' \text{ and } \mu_d(z_h, P^S(e)) \\ (NX_{\mu_d}(z_g), z_h) & \text{if } e \in \Sigma^d \text{ and } M(e) = \mu_d(z_h, P^S(e)) \in \Gamma G(z_h) \\ (\mu_d(z_h, P^S(e)), z_h) & \text{if } e \in \Sigma_o \text{ and } M(e) = \mu_d(z_h, P^S(e)) \in \Gamma G(z_h) \\ (z_h, \mu_d(z_h, P^S(e))) & \text{if } e \in \Sigma^d \text{ and } M(e) = \mu_d(z_h, P^S(e)) \in \Gamma G(z_h) \end{cases} \quad (6)$$

- $y_0 := \{(x^o), q_0\} \in Q_S$ is the initial state.

An $S$-state is an IS where the supervisor issues its control decision and an $E$-state is an IS at which the environment (system or attacker) selects one among the observable events to occur. A transition from an $S$-state to an $E$-state represents the updated unobservable reach in $G$’s state estimate together with the current supervisor state. On the other hand, a transition from an $E$-state to an $S$-state represents the “observable reach” immediately following the execution of the observable event by the environment. In this case, both the system’s state estimate and the supervisor’s state are updated. However, these updates depend on the type of event generated by the environment: (i) true system event unaltered by the attacker; (ii) (fictitious) event insertion by the attacker; or (iii) deletion by the attacker of an event just executed by the system. Thus, the transition rules are split into three cases as defined by Equation (6).

Definition 4 also defines a construction algorithm of the AIDA, where starting from the initial state $y_0$ a breadth first search is performed to compute all states and transitions in the AIDA. Since the goal of the attacker is to reach any state in $Q_{crit}$, no transitions are defined in $E$-states $z \in Q_E$ that satisfy $I_G(z) \subseteq Q_{crit}$.

After calculating the AIDA of a system a pruning process that removes non-stealthy attacks from the AIDA is applied. This pruning process can be mapped to a supervisory control problem, as explained in (Meira-Góes et al., 2019a). Specifically, any event $e \in \Sigma_o \cup \Sigma' \cup \Sigma^d$ is considered controllable and any event $e \in \Sigma_n \setminus \Sigma_o$ is uncontrollable. The plant is AIDA $A$ and the specification $A_{trim}$ is obtained by removing all states of the AIDA where the supervisor $R_{dead}$ reaches a dead state, $y = (S, dead)$. The pruning process in (Meira-Góes et al., 2019a) is a fixed-point algorithm that removes states from $A_{trim}$ iteratively until convergence. The pruned AIDA, where only
stealthy attack functions are present, is denoted as the Interruptible Stealthy Deceptive Attack (ISDA) structure.

3. ABSTRACTED INSERTION-DELETION ATTACK STRUCTURE

The objective of this paper is to calculate an abstracted ISDA with less states and the same interruptible and stealthy attack functions as present, is denoted as the Interruptible Stealthy Deceptive Attack (ISDA) structure.

Definition 5. (Milner, 1989) Let $G = \langle \Sigma, Q, \rightarrow, x^0 \rangle$ be a non-deterministic automaton. An equivalence relation $\approx \subseteq Q \times Q$ is called an observation equivalence on $G$, if the following holds for all $x_1, x_2 \in Q$ such that $x_1 \approx x_2$: if $x_1 \xrightarrow{\sigma} y_1$ for some $\sigma \in \Sigma^*$, then there exists $y_2 \in Q$ such that $x_2 \xrightarrow{\sigma} y_2$, and $y_1 \approx y_2$.

In order to be able to use observation equivalence in the attack framework of this paper, we need to consider the critical states.

Definition 6. Let $G = \langle \Sigma, Q, \rightarrow, x^0 \rangle$ be a non-deterministic automaton. An equivalence relation $\sim \subseteq Q \times Q$ is called restricted version of observation equivalence on $G$, if the following holds for all $x_1, x_2 \in Q$ such that $x_1 \sim x_2$:

- if $x_1 \xrightarrow{\sigma} y_1$ and $\sigma \in \Sigma_o$ then $x_2 \xrightarrow{\sigma} y_2$ and $y_1 \sim y_2$
- if $x_1 \xrightarrow{\sigma} y_1$ and $\sigma \in \Sigma_{uo}$ then $x_2 \xrightarrow{\sigma} y_2$ and $y_1 \sim y_2$
- $x_1 \in Q_{crit}$ if and only if $x_2 \in Q_{crit}$

In the restricted version of observation equivalence, similar to observation equivalence, two states are equivalent if they have the same future behavior. Moreover, the restricted version of observation equivalence requires the equivalent states to have the same critical status, i.e., either all are critical or none of them are critical states.

Example 1. Consider the system $G$ with the set of critical states $Q_{crit} = \{q_6\}$, $\Sigma_{uc} = \{\beta\}$, $\Sigma_{uo} = \{\nu\}$ and the compromised event set $\Sigma_a = \{\alpha\}$. Automaton $G$ and $R_{dead}$ are shown in Fig. 1. The figure also shows the original AIDA of the system, $A$. In the figure the initial states are marked by an arrow pointing into them, the $S$-states of $A$ are shaded grey and the critical states of $G$ and the $dead$-state of the supervisor $R_{dead}$ are crossed out. The initial state of $A$ is $(q_0, 0)$. The active event of $R_{dead}$ at state 0 is $\Gamma_{R_{dead}}(0) = \{\nu, \beta, \alpha\}$ and $UR_{\{\nu, \beta, \alpha\}} = \{(q_0, q_1, q_2, q_3)\}$. Thus, event $\{\nu, \beta, \alpha\}$ is executable at the initial state of $A$ leading the AIDA to the $E$-state $(q_9q_1q_2q_3, 0)$. Now consider event $\beta$, which is an active event at state $q_9$ of $G$. As $\beta \in (\Gamma_{R_{dead}}(z_8) \cap \Gamma_G(z_9))$ it holds that $(NX_e((q_9q_1q_2q_3), \mu_{d}(0, \beta)) = (q_1, 1)$ is an S-state of AIDA $A$ and it is reached from state $(q_9q_1q_2q_3, 0)$ by executing $\beta$. The whole structure is interpreted in a similar way. To obtain the ISDA, we must eliminate the non-stealthy strategies. Applying the pruning process previously described in Sect. 2, we obtain the ISDA $A_t$.
depicted in Fig. 2. State $q_k$ is reachable in $A_1$, which implies that there exists a successful stealthy attack function.

In the framework considered in this paper, the system automaton $G$ is abstracted by applying the restricted version of observation equivalence before the construction of the AIDA. States $q_0$, $q_1$ and $q_2$ are equivalent as $q_i \Rightarrow q_3$ for $i = 0, 1, 2$. Moreover, states $q_4$ and $q_5$ are also equivalent as $q_4 \Rightarrow q_6$ and $q_5 \Rightarrow q_6$. Merging equivalent states results in automaton $\tilde{G}$ shown in Fig. 1. In the figure the critical states of $\tilde{G}$ are crossed out.

### 3.2 AIDA with Abstracted Model

In the previous section, we have introduced the restricted version of observation equivalence as a mean to abstract a system before calculating the AIDA of the system. In this section, we prove that the abstracted AIDA and the original AIDA contain the same attack strategies if the restricted version of observation equivalence is used to abstract the system.

In the following, Proposition 3 establishes that the abstracted AIDA and the original AIDA contain the same attack strategies. Using the results from Proposition 3, Theorem 4 proves that the abstracted ISDA embeds all possible interruptible stealthy insertion-deletion attack strategies. First Lemmas 1 and 2 provide auxiliary results to prove Proposition 3.

#### Lemma 1

Let $G = (\Sigma, Q, \rightarrow, x^c)$ and let $\sim$ be the restricted version of observation equivalence. Let $\tilde{G}$ be the quotient automaton of $G$ by applying $\sim$. Consider $S \subseteq \tilde{Q}$ and $\tilde{S} \subseteq \tilde{Q}$ such that $x \in S$ if and only if $[x] \in \tilde{S}$, where $x \in [x]$. Then $y \in UR_x(S)$ if and only if $[y] \in UR_x(\tilde{S})$, where $y \in [y]$.

#### Proof

($\Rightarrow$) Assume that $y \in UR_x(S)$. Based on (2), it holds that there exists $t \in (\Sigma_x \cup \gamma)^*$ such that $x \stackrel{t}{\rightarrow} y$. Based on Def. 6 it holds that there exists $t' \in (\Sigma_x \cup \gamma)^*$ such that $[x] \stackrel{t'}{\rightarrow} [y]$ and $y \in [y]$.

($\Leftarrow$) A similar argument as above holds.

#### Lemma 2

Let $G = (\Sigma, Q, \rightarrow, x^c)$ and let $\sim$ be the restricted version of observation equivalence on $G$. Consider $S \subseteq \tilde{Q}$ and $\tilde{S} \subseteq \tilde{Q}$ such that $x \in S$ if and only if $[x] \in \tilde{S}$, where $x \in [x]$. Then $y \in NX_x(S)$ if and only if $[y] \in NX_x(\tilde{S})$, where $y \in [y]$.

#### Proof

($\Rightarrow$) Assume that $y \in NX_x(S)$. Then based on (3) it holds that there exists $x \rightarrow y$. Based on Def. 6 it holds that $[x] \stackrel{t}{\rightarrow} [y]$ and $y \in [y]$.

($\Leftarrow$) A similar argument as above holds.

Similarly to Lemma 1, Lemma 2 proves that the observable reach of the two subsets that contain restricted observation equivalent states are also restricted observation equivalent.

#### Proposition 3

Let $G = (\Sigma, Q, \rightarrow, x^c)$ and let $\sim$ be the restricted version of observation equivalence on $G$. Let $A$ be the AIDA with respect to $G$, $\Sigma_a$ and $R_{dead}$ and $\tilde{A}$ be the AIDA with respect to $\tilde{G}$, $\Sigma_a$ and $R_{dead}$. Then $A \sim p$ if and only if $\tilde{A} \sim \tilde{p}$ such that:

1. $p$ is an $S$-state, $E$-state, if and only if $\tilde{p}$ is an $S$-state, $E$-State,
2. $I_G(p) \subseteq Q_{crit}$ if and only if $I_{\tilde{G}}(\tilde{p}) \subseteq \tilde{Q}_{crit}$,
3. $I_S(p) = \text{dead}$ if and only $I_{\tilde{S}}(\tilde{p}) = \text{dead}$.

#### Proof

We show that a transition is defined in $G$ if and only if the same transition is defined in $A$. It is shown by induction on $n \geq 0$ that $y_0 \stackrel{\gamma_0}{\rightarrow} p_n$ in $A$ if and only if $\tilde{y}_0 \sim \tilde{p}_n$ in $\tilde{A}$. Note that $p_n$ or $\tilde{p}_n$ consist of $S$-states and $E$-states. Since there is no reduction on $R_{dead}$ it holds that $I_S(p_n) = I_{\tilde{S}}(\tilde{p}_n)$.

**Base case**: ($\Rightarrow$) Let $x^c$ be the initial state of $\tilde{G}$ and $q^c$ be the initial state of $R_{dead}$. Then the initial state of $A$ is $y_0 = (x^c, q^c)$ and the initial state of $\tilde{A}$ is $\tilde{y}_0 = (\tilde{x}^c, \tilde{q}^c)$ and they both are $S$-states. First assume that $y_0 \not\sim \tilde{y}_0$ in $A$, which is an $h_{SE}$ transition. We need to show that there exists $\tilde{y}_0 \not\sim \tilde{y}_0$ in $\tilde{A}$. Based on Def. 4 it holds that $I_G(z_0) = UR_{\gamma_0}((x^c))$, $I_{\tilde{S}}(\tilde{y}_0) = I_{\tilde{S}}(\tilde{y}_0) = \tilde{q}^c$ and $\gamma_0 = \Gamma_{R_{dead}}(I_S(y_0)) = \Gamma_{R_{dead}}(\tilde{q}^c)$. Since $x^c \in \tilde{x}^c$ it means that $v \in I_G(y_0)$ if and only if there exists $[v] \in I_{\tilde{G}}(\tilde{y}_0)$. Based on Lemma 1 it holds that $u \in UR_{\gamma_0}((x^c))$. This means that there exists $I_{\tilde{S}}(\tilde{y}_0) = UR_{\gamma_0}((\tilde{x}^c))$, where for every $u \in I_G(y_0)$ there exists a $[u] \in I_{\tilde{G}}(\tilde{y}_0)$.

**Inductive step**: Assume that the claim holds for some $n \geq 0$, i.e., $y_0 \not\sim \tilde{p}_n$ in $A$ and if and only if $\tilde{y}_0 \not\sim \tilde{p}_n$ in $\tilde{A}$, $p_n$ is an $S$-state, $E$-state, if and only if $\tilde{p}_n$ is an $S$-state, $E$-state and $I_G(p_n) \subseteq Q_{crit}$ if and only if $I_{\tilde{G}}(\tilde{p}_n) \subseteq \tilde{Q}_{crit}$. Thus, $y_0 \not\sim \tilde{y}_0$ in $A$, where $I_G(z_0) = UR_{\gamma_0}((x^c))$, $I_{\tilde{S}}(\tilde{z}_0) = I_{\tilde{S}}(\tilde{y}_0) = \tilde{q}^c$ and $\gamma_0 = \Gamma_{R_{dead}}(I_S(y_0)) = \Gamma_{R_{dead}}(\tilde{q}^c)$.

($\Leftarrow$) Now assume that $\tilde{y}_0 \not\sim \tilde{y}_0$ in $\tilde{A}$.

**Construction step**: Assume that the claim holds for some $n \geq 0$, i.e., $y_0 \not\sim \tilde{p}_n$ in $A$ and if and only if $\tilde{y}_0 \not\sim \tilde{p}_n$ in $\tilde{A}$, $p_n$ is an $S$-state, $E$-state, if and only if $\tilde{p}_n$ is an $S$-state, $E$-state, and $I_G(p_n) \subseteq Q_{crit}$ if and only if $I_{\tilde{G}}(\tilde{p}_n) \subseteq \tilde{Q}_{crit}$. Now it must be shown that if $p_n \not\sim \tilde{p}_{n+1}$ in $A$ then $\tilde{p}_n \not\sim \tilde{p}_{n+1}$ in $\tilde{A}$ such that $I_G(p_{n+1}) \subseteq Q_{crit}$ if and only if $I_{\tilde{G}}(\tilde{p}_{n+1}) \subseteq \tilde{Q}_{crit}$ and $I_S(p_n) = \text{dead}$ if and only $I_{\tilde{S}}(\tilde{p}_n) = \text{dead}$ and $v \in I_G(p_n)$ if and only if there exists $[v] \in I_{\tilde{G}}(\tilde{p}_n)$.

Consider the following two possibilities:

- $p_n \in Q_S$, which implies that $p_n \not\sim p_{n+1}$ is an $h_{SE}$ transition. Then based on Def. 4 it holds that $I_G(p_{n+1}) = UR_{\gamma_0}(I_G(p_n))$. $I_S(p_n) = I_{\tilde{S}}(\tilde{p}_n)$ and $\gamma_n = \Gamma_{R_{dead}}(I_S(p_n))$. Since based on inductive assumption it holds that $v \in I_G(p_n)$ if and only if there exists $[v] \in I_{\tilde{G}}(\tilde{p}_n)$ it follows from Lemma 1 that $u \in UR_{\gamma_0}(I_G(p_n))$ if and only if there exists $[u] \in UR_{\gamma_0}(I_{\tilde{G}}(\tilde{p}_n))$. This means there exists $I_G(p_{n+1}) = UR_{\gamma_0}(I_{\tilde{G}}(\tilde{p}_n))$, where for every $u \in$
After constructing the abstracted AIDA $\tilde{A}$, the non-stealthy interruptible attack strategies must be removed via the pruning process. The following theorem is the main contribution of the paper and it proves that successful attack functions synthesized from the abstracted system can be exactly mapped to successful attack functions on the original system. To establish this result, the theorem proves that the original AIDA $A$ and the abstracted AIDA $\tilde{A}$ are bisimilar and the pruning process removes from $A$ and $\tilde{A}$ bisimilar states. Consequently, the abstracted ISDA $A_t$ and the original ISDA $\tilde{A}_t$ are bisimilar and have the same structure.

![Diagram of Examples 2 and 3](image)

**Theorem 4.** Let $G = (\Sigma, Q, \delta, \tau, x^0)$ and let $\sim$ be the restricted version of observation equivalence on $G$. Let $\tilde{A}_t$ be the ISDA with respect to $\tilde{G}$, $\Sigma_o$ and $R_{dead}$. Then $\tilde{A}_t$ embeds all possible interruptible stealthy insertion-deletion attack strategies.

**Proof.** To prove the theorem we show the following:

(i) the AIDA of the original system and the abstracted system are bisimilar;

(ii) the specification of the original system and the abstracted system are bisimilar;

(iii) the pruning process removes from A and $\tilde{A}$ bisimilar states.

(i): Let $A$ be the AIDA based on $\Sigma_o$ and $R_{dead}$ and $G$ and $\tilde{A}$ be the AIDA based on $\Sigma_o$, $R_{dead}$ and $\tilde{G}$. It was shown in Proposition 3 that $A \xrightarrow{\gamma} \tilde{A}$ if and only if $\tilde{A} \xrightarrow{\gamma} \tilde{A}$, which implies that $A$ and $\tilde{A}$ are bisimilar.

(ii): Let $A_{trim}$ and $\tilde{A}_{trim}$ be the specifications obtained by removing states of $A$ and $\tilde{A}$, where $I_S(p) = \text{dead}$ and $I_S(\tilde{p}) = \text{dead}$. Based on Proposition 3 it holds that the $A \xrightarrow{\gamma} p$ if and only if $\tilde{A} \xrightarrow{\gamma} \tilde{p}$ and $I_S(p) = \text{dead}$ if and only $I_S(\tilde{p}) = \text{dead}$, which implies that the specifications $A_{trim}$ and $\tilde{A}_{trim}$ are bisimilar.

(iii): It has been proven in (Mohajerani et al., 2014) that the synthesis procedure that removes uncontrollable states from two deterministic bisimilar automata produces bisimilar supervisors. This proves that the uncontrollable states removed from $A$ and $\tilde{A}$ are bisimilar. The pruning process in (Meira-Góes et al., 2019a) is a fixed-point algorithm that also removes all the E-states $q \in Q_A$, such that $e \in \Gamma_A(q)$ but both $e$ and $e_d$ are absent from $\Gamma_{A_t}(q)$, where $A_t$ is an intermediate result in the fixed-point pruning process. Now we need to prove that bisimilar $E$-states are removed from $A$ and $\tilde{A}$. This is shown by induction on the iteration steps.

**Base case:** $i = 0$.

($\Rightarrow$) Assume that $q \in Q_A$. Based on Proposition 3 it holds that if $A \rightarrow q$ then $\tilde{A} \rightarrow \tilde{q}$, which implies that $\tilde{q} \in \tilde{Q}_{A_t}$.

($\Leftarrow$) Assume that $\tilde{q} \in \tilde{Q}_{A_t}$. A similar argument as above holds.
Inductive steps: Assume that the claim holds at iteration \( n \), i.e., \( q \in Q_{A_n} \) if and only if \( \tilde{q} \in Q_{\tilde{A}_n} \). Now we need to show that the claim holds for iteration \( n + 1 \). First, let \( q \in Q_{A^{n+1}} \), which implies that \( e \in \Gamma_A(q) \) and \( e \in \Gamma_{A^*}(q) \) or \( e_d \in \Gamma_{A^*}(q) \). From Proposition 3 it holds that if \( A_n+1 \rightarrow q \xrightarrow{\tilde{\omega}} \tilde{q} \xrightarrow{\tilde{\omega}} \tilde{q} \), which implies that \( e \in \Gamma_{\tilde{A}(\tilde{q})} \). A similar argument as above holds for \( A_{trim} \) and \( \tilde{A}_{trim} \), which means that \( e \in \Gamma_{\tilde{A}_{trim}}(\tilde{q}) \). This proves that \( q \in Q_{A^{n+1}} \) if \( q \in Q_{A^{n+1}} \). Now assume \( \tilde{q} \in Q_{\tilde{A}_{n+1}} \). A similar argument as above holds.

This means that \( A_i \xrightarrow{\tilde{\omega}} p \) if and only if \( \tilde{A}_i \xrightarrow{\tilde{\omega}} \tilde{p} \), which implies that \( A_t \) embeds all possible interruptible stealthy insertion-deletion attack strategies.

Example 3. Continuing Example 2, the corresponding abstracted AIDA \( \tilde{A} \) is shown in Fig. 3. To calculate the abstracted ISDA, the pruning process explained in Sect. 2 needs be applied on \( \tilde{A} \). First, state \((\{q_4\}, \text{dead})\) is removed producing specification \( \tilde{A}_{trim} \) which makes \( E \)-state \((\{q_4\}, 1) \) and then \( S \)-state \((\{q_0\}, 1) \) uncontrollable. This results in abstracted ISDA \( A_t \), shown in Fig. 3, with 2 less states compared to the original \( A_t \), shown in Fig. 2.

4. CONCLUSION

In this paper, we investigate abstraction-based attack function synthesis when the attacker feeds incorrect information to the supervisor in order to damage the supervised system. To synthesize successful stealthy sensor deception attacks, the All-Insertion-Deletion Attack structure (AIDA) is constructed. Calculating the AIDA can be computationally expensive, specifically in partially observed systems. To mitigate the computational complexity of constructing the AIDA an abstraction method called restricted version of observation equivalence is introduced that can abstract the system by merging some states before constructing the AIDA. We prove that the abstracted AIDA provides a complete solution to the attack synthesis problem. It would be of interest to investigate the abstraction methods not only for the system also for the supervisor. Moreover, developing abstraction-based AIDA-like structures for modular systems consisting of interacting subsystems is also of interest.

REFERENCES

Carvalho, Lilian K., Yi-Chin Wu, Raymond Kwong and Stéphane Laforetune (2018). Detection and mitigation of classes of attacks in supervisory control systems. *Automatica* **97**, 121–133.

Cassandras, Christos G. and Stéphane Laforetune (2008). *Introduction to Discrete Event Systems*. 2nd ed., Springer Science & Business Media. New York, NY, USA.

Lima, Pêblio Macedo, Marcos Vinicius Silva Alves, Lilian Kawakami Carvalho and Marcos Vicente Moreira (2019). Security against communication network attacks of cyber-physical systems. *Journal of Control, Automation and Electrical Systems* **30**(1), 125–135.

Meira-Góes, Rômulo, Eunsuk Kang, Raymond Kwong and Stéphane Laforetune (2019a). Synthesis of sensor deception attacks at the supervisory layer of cyber-physical systems. Submitted.