Attack Synchronizing Sequences for Output Synchronized Petri Nets with Multiple Deadlocks

Khalid Hamada, Rabah Ammour, Leonardo Brenner, Isabel Demongodin

Attack Synchronizing Sequences for Output Synchronized Petri Nets with Multiple Deadlocks

Khalid Hamada [1] Rabah Ammour [1] Leonardo Brenner [2] Isabel Demongodin [1]

[1]Aix-Marseille University, LIS, Marseille, France [2]Université de Reims Champagne-Ardenne, LICIIS - LRC DIGIT, Reims, France

Introduction

Context: Cyber attacks on a Cyber Physical System (CPS) characterized by a physical part, a cyber part (controller) and a network facilitating the transmission of commands from the controller and the observation of outputs from the sensors.

Attack assumptions:
i) the attacker has a complete knowledge of the CPS model;
ii) the attacker is able to insert control inputs and to read/delete sensors outputs;
iii) the attacker does not know (or is unable to estimate) the current state of the system.

Objective: Determine an **attack synchronizing sequence**, $\bar{\omega}_j$, that is an input control sequence driving the system to one of its deadlocks whatever its current state.

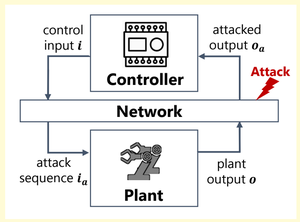

Figure 1. CPS under attack

Modeling with Output Synchronized Petri nets

The modeling formalism is a particular class of synchronized Petri nets, called Output Synchronized Petri nets (OutSynPN). This formalism allows the system to be controlled by input events associated with transitions and to be observed thanks to output events linked to marking change events and/or marking values of states.

An *output synchronized Petri net* (OutSynPN) is a structure $N_{os} = \langle N, E, f, \Sigma, \Gamma, Q, g\rangle$. A marked OutSynPN is $\langle N_{os}, M_0\rangle$ such that:

- $N = \langle P, T, Pre, Post\rangle$ is a Petri net, where P is a set of m places, T is a set of n transitions, $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the $pre-$ and $post-$ incidence matrices that specify the weights of directed arcs from places to transitions and vice versa;
- A marking is a vector $M : P \to \mathbb{N}^m$ that assigns to each place a non-negative integer;
- E is an alphabet of external input events;
- $f : T \to E_\lambda = E \cup \{\lambda\}$ is a labeling function that associates with each transition t either an external input event $f(t) \in E$ or λ, where λ is the "always occurring" event;
- $\Sigma \subseteq \{\uparrow m_i, \downarrow m_i \mid p_i \in P\}$ is a non empty set of events associated with a marking change of places, where $\downarrow$ and $\uparrow$ represent a decreasing and an increasing of a place marking, respectively;
- $\Gamma \subseteq \{m_i \sim h, \mid p_i \in P, h \in N, \sim \in \{=, \neq, \geq, \leq, >, <\}\}$ is a set of conditions on the place marking;
- Q is an alphabet of output events;
- $g : Q \to \{0, 1\}$ is an output function such that $q_i \in Q$, $g(q_i) = \Upsilon(F_\Gamma(q_i)) \wedge \Theta(F_\Sigma(q_i))$ where $F_\Gamma(q_i) \in F_{|\Gamma|}, F_\Sigma(q_i) \in F_{|\Sigma|}$ and:
  - $\Upsilon : F_{|\Gamma|} \to \{0, 1\}$ is a boolean function depicting the conditions on the marking value of places to generate output q_i and $\Upsilon() = 1$ when no condition on the marking values is involved for output q_i;
  - Θ: $F_{|\Sigma|} \to \{0, 1\}$ is a boolean function depicting the conditions on the marking change events to generate output q_i and $\Theta() = 0$ when no event on the marking change is involved for output q_i. ■

From $M_0 = (3\ 0\ 0\ 0\ 0\ 0)^T$,
input control sequence $i = e_2e_2$ leads to the following evolution:

- $3p_1 \xrightarrow{e_2|t_3:A} p_4 \xrightarrow{e_2|t_4:B} 3p_1$
- It generates the output sequence $o = AB$.

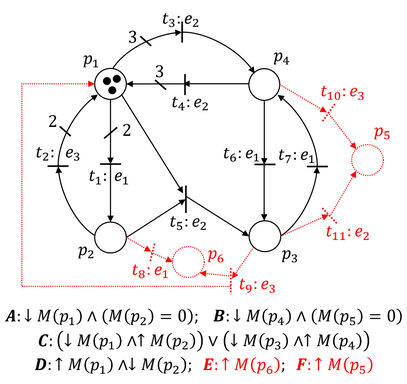

Figure 2. OutSynPN

State space with Labeled Finite state Automaton with Inputs (LFAI)

The reachability graph of an OutSynPN is represented by a particular *labeled finite state automaton with inputs*.

The labeling and output functions could be completed to obtain a completely specified LFAI.

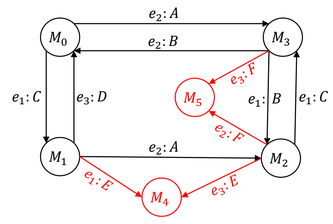

Figure 3. LFAI G

Synchronizing sequence

Consider a completely specified LFAI, $\tilde{G} = \langle X, E, \delta, M_0, Q, Obs\rangle$, and a single target state $\bar{M} \in X$. The input sequence $\bar{\omega} = e_1e_2 \dots e_k \in E^*$ is called synchronizing for state $\bar{M}$ if it drives the LFAI $\tilde{G}$ to the target state $\bar{M}$, regardless of the initial state, i.e., $\forall M \in X$ it holds $\delta^*(M, \bar{\omega}) = \bar{M}$.

Outputs sequences set

The outputs sequences set of synchronizing sequence $\bar{\omega}$, denoted $\vartheta_{\bar{\omega}}$, represents all the outputs sequences generated from all states of $\tilde{G}$, i.e., $\vartheta_{\bar{\omega}} = \{\theta^i_{\bar{\omega}} | \theta^i_{\bar{\omega}} = Obs^*(M_i, \bar{\omega}), M_i \in X\}$.

Restricted LFAI construction algorithm

In a *Restricted LFAI*, all deadlocks of the LFAI are merged into a single one.

Input: A LFAI $G = \langle X, E, \delta, M_0, Q, Obs\rangle$ with a set $\{M^d_1, M^d_2, ..., M^d_k\}$ of deadlocks.
Output: A restricted LFAI $G^r = \langle X_r, E, \delta_r, M_0, Q, Obs\rangle$.

1. Let $\mathcal{M}^d$ be a new (deadlock) state
2. $X_r \leftarrow \{X \backslash \{M^d_1, ..., M^d_k\}\} \cup \mathcal{M}^d$
3. For each node $M^d_i \in \{M^d_1, ..., M^d_k\}$:
   - For each arc $\delta(M, e) = M^d_i$ such that $e \in E$ and $M \in X \backslash \{M^d_1, ..., M^d_k\}$: $\delta_r(M, e) = \mathcal{M}^d$
4. For each node $M \in \{X_r \backslash \mathcal{M}^d\}$:
   - For each $e \in E$, if $\delta_r(M, e)$ is not defined and $\delta(M, e)$ exists: $\delta_r(M, e) = \delta(M, e)$

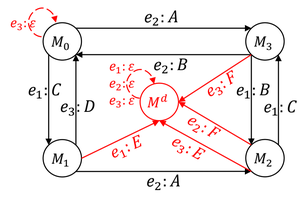

Figure 4. Completely specified and Restricted LFAI (RLFAI) $\tilde{G}^r$

Proposed method

Given an OutSynPN $\langle N_{os}, M_0\rangle$,

1. Compute G, the LFAI.
2. Construct the Restricted LFAI G^r.
3. Deduce the completely specified RLFAI $\tilde{G}^r$.
4. Construct the auxiliary graph with outputs $\mathcal{A}(\tilde{G}^r)$.
5. Determine for target marking $\bar{M}$, a synchronizing sequence $\bar{\omega}$ and its associated outputs sequences set $\vartheta_{\bar{\omega}}$.

Auxiliary graph with outputs

Let $\tilde{G} = \langle X, E, \delta, M_0, Q, Obs\rangle$ be a completely specified LFAI.
The *auxiliary graph with outputs* $\mathcal{A}(\tilde{G})$, consists of $(|X|.(|X| + 1)/2)$ nodes, representing every unordered pair (M', M'') of states within $\tilde{G}$, including pairs of identical states (M, M).
One edge is present from node (M', M'') to $(\hat{M}', \hat{M}'')$, labeled with $e : (o', o'')$, iff $\delta(M', e) = \hat{M}'$, $\delta(M'', e) = \hat{M}''$, $Obs(M', e) = o'$ and, $Obs(M'', e) = o''$, for $e \in E$ and $o', o'' \in Q$.

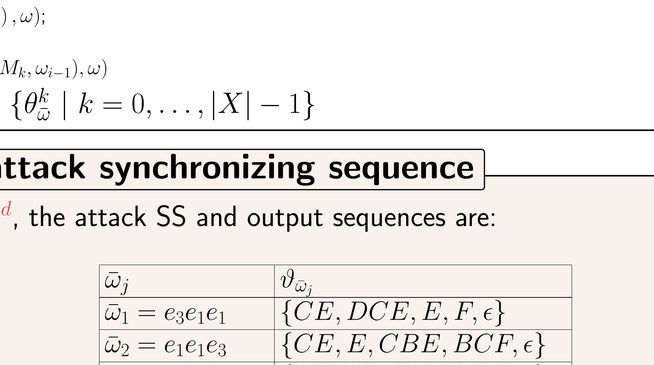

Figure 5. Auxiliary graph with outputs $\mathcal{A}(\tilde{G}^r)$

Greedy algorithm with outputs

Input: Completely specified LFAI $\tilde{G}$, auxiliary graph with outputs $\mathcal{A}(\tilde{G})$ and, target state $\bar{M} \in X$.
Output: A SS $\bar{\omega}$ **for** state $\bar{M}$, outputs sequences set $\vartheta_{\bar{\omega}}$

1. Let $i \leftarrow 0$; $\omega_0 \leftarrow \varepsilon$; $\phi(\omega_0) \leftarrow X$; $\theta^k_{\omega_0} \leftarrow \varepsilon$ for $k = 0, \dots, |X| - 1$
2. **While** $\phi(\omega_i) \neq \{\bar{M}\}$:
   - $i \leftarrow i + 1$
   - pick two states $M', M'' \in \phi(\omega_{i-1})$ such that $M' \neq M''$;
   - **If** there does not exist any path in $\mathcal{A}(\tilde{G})$ from node (M', M'') to node $(\bar{M}, \bar{M})$: stop the computation, there exists no SS for $\bar{M}$.
   - **Else** find the shortest path from node $\left(M', M''\right)$ to $(\bar{M}, \bar{M})$ and let ω be the input sequence along this path, do:
     - $\omega_i \leftarrow \omega_{i-1}\omega$;
     - $\phi(\omega_i) \leftarrow \delta^*(\phi(\omega_{i-1}), \omega)$;
     - **For** all $M_k \in X$:
     - $\theta^k_{\omega_i} \leftarrow \theta^k_{\omega_{i-1}} Obs^*(\delta^*(M_k, \omega_{i-1}), \omega)$
3. $\bar{\omega} \leftarrow \omega_i$ and $\vartheta_{\bar{\omega}} \leftarrow \{\theta^k_{\bar{\omega}} \mid k = 0, \dots, |X| - 1\}$

Example: best attack synchronizing sequence

For the target state $\mathcal{M}^d$, the attack SS and output sequences are:

$\bar{\omega}_j$	$\vartheta_{\bar{\omega}_j}$
$\bar{\omega}_1 = e_3e_1e_1$	$\{CE, DCE, E, F, \epsilon\}$
$\bar{\omega}_2 = e_1e_1e_3$	$\{CE, E, CBE, BCF, \epsilon\}$
$\bar{\omega}_3 = e_1e_3e_1e_1$	$\{CDCE, E, CF, BE, \epsilon\}$
$\bar{\omega}_4 = e_2e_3e_1e_1$	$\{AF, AE, F, BCE, \epsilon\}$
$\bar{\omega}_5 = e_1e_2e_3e_1e_1$	$\{CAE, E, CBCE, BF, \epsilon\}$

$$\Omega = \{\bar{\omega}_1, \bar{\omega}_2, \bar{\omega}_3, \bar{\omega}_4, \bar{\omega}_5\}$$

The attack synchronizing sequences, $\bar{\omega}_j \in \Omega$, with the minimal number of inputs events and outputs labels to manipulate in the worst case are given by $\bar{\omega}_j \in \arg\min_{\bar{\omega} \in \Omega}\{|\bar{\omega}| + \max_{\theta^i_{\bar{\omega}} \in \vartheta_{\bar{\omega}}} |\theta^i_{\bar{\omega}}|\}$.

$\bar{\omega}_j$	$\lvert\bar{\omega}_j\rvert$	$\max_{\theta^i_{\bar{\omega}_j} \in \vartheta_{\bar{\omega}_j}} \lvert\theta^i_{\bar{\omega}_j}\rvert$	$\lvert\bar{\omega}_j\rvert + \max_{\theta^i_{\bar{\omega}_j} \in \vartheta_{\bar{\omega}_j}} \lvert\theta^i_{\bar{\omega}_j}\rvert$
$\bar{\omega}_1 = e_3e_1e_1$	3	3 (DCE)	6
$\bar{\omega}_2 = e_1e_1e_3$	3	3 (CBE/BCF)	6
$\bar{\omega}_3 = e_1e_3e_1e_1$	4	4 $(CDCE)$	8
$\bar{\omega}_4 = e_2e_3e_1e_1$	4	3 (BCE)	7
$\bar{\omega}_5 = e_1e_2e_3e_1e_1$	5	4 $(CBCE)$	9

The minimal number of inputs events and outputs labels length to manipulate in the worst case is equal to 6, given by $|\bar{\omega}_1| + |\theta^1_{\bar{\omega}_1}| = 6$ and $|\bar{\omega}_2| + |\theta^2_{\bar{\omega}_2}| = |\bar{\omega}_2| + |\theta^3_{\bar{\omega}_2}| = 6$.

Thus, to remain stealthy and save efforts, the attacker has to choose between $\bar{\omega}_1$ or $\bar{\omega}_2$.

Conclusions

- The proposed method, based on extensions: the auxiliary graph with outputs and greedy algorithm computation of SS and output sequences, involves representing all deadlocks with a single state, that acts like a unique deadlock.
- Several attack synchronizing sequences are obtained for this unique deadlock and a criteria to choose the best attack SS is proposed.

K. Hamada, R. Ammour, L. Brenner, I. Demongodin. Attack Synchronizing Sequence Computation for Output Synchronized Petri Nets with Multiple Deadlocks. 20th IEEE International Conference on Networking, Sensing and Control (ICNSC), 2023.

14ème colloque sur la Modélisation des Systèmes Réactifs (MSR'23), LAAS-CNRS, Toulouse (France) - 22 au 24 novembre 2023