Supplementary material for "Multi-step Sensor Attackability in Cyber-Physical Systems"

Wenli Duo, *Graduate Student Member*, *IEEE*, Shouguang Wang, *Senior Member*, *IEEE*, and MengChu Zhou, *Fellow*, *IEEE*

I. NOMENCLATURE

$\mathbb{N}$	Set of natural numbers
$\mathbb{N}^+$	Set of positive integers
G	$(X, \Sigma, \delta, x_0, X_m)$, physical plant
S	$(X_S, \Sigma, \delta_S, x_{0,S}, X_{m,S})$, supervisor
R	$(X_R, \Sigma_R, \delta_R, x_{0,R}, X_{m,R})$, 1A automaton
$L(G)$	language generated by G
$L_m(G)$	language marked by G
V/G	Closed-loop system
$\Sigma_{\bar{o}}$	Set of unobservable events
Σ_o	Set of observable events
Σ_{RD}	Set of replacement and deletion operations
Σ_I	Set of insertion operations
Σ_A	Set of all events and possible actions under attacks
G_A	$(X_A, \Sigma_A, \delta_A, x_{0,A}, X_{m,A})$, general attack structure
G_P	$(X_P, \Sigma_A, \delta_P, x_{0,P}, X_{m,P})$, refined and stealthy attack structure
X_u	Set of unsafe states
X_{dead}	Set of dead states
$Ac(\cdot)$	Accessible part of an automaton
$Trim(\cdot)$	Accessible and coaccessible part of an automaton
$\Gamma(x)$	Set of active events at state x in G
$P_G(s)$	P_G: $\Sigma_A^* \to \Sigma^*$, actual generation of G with respect to s
$P_S(s)$	P_S: $\Sigma_A^* \to \Sigma_o^*$, observation of S with respect to s
$P_A(s)$	P_A: $\Sigma_A^* \to (\Sigma_{RD} \cup \Sigma_I)^*$, attack operations required by s
$\lvert s\rvert$	Length of a string s
$L_k(G_P)$	$\{s \in L_m(G_P) \mid \lvert P_A(s)\rvert \le k\}$, set of strings in $L_m(G_P)$ that requires within k attack operations
$G_{AR}(G)$	$(X_{AR}, \Sigma_{AR}, \delta_{AR}, x_{0,AR}, X_{m,AR})$, attack recognizer for G and G_P
$G_{AR}^k(G)$	$(X_{AR}^k, \Sigma_{AR}^k, \delta_{AR}^k, x_{0,AR}^k, X_{m,AR}^k)$, k-step attack recognizer for G and G_P
G_K	$(X_K, \Sigma_K, \delta_K, x_{0,K}, X_{m,K})$, automaton that recognizes $L_k(G_P)$
$\tilde{N}(x)$	$\{x' \in X_P \mid (\exists s \in \Sigma^*)\delta_P(x, s)=x'\}$, set of normal reach of x

II. ALGORITHMS

In this section, explanations of Algorithms 1 and 2 as well as their computational analysis are provided.

Algorithm 1: It is designed for construction of a refined and stealthy RDI attack structure, which extends the work in [1] and [2]. The algorithm first constructs a general attack structure based on Definition 2. After obtaining G_A, Algorithm 1 computes the 1A automaton R and the closed-loop system V/G. Then, it calls *PruningGa* to prune G_A. In *PruningGa*, let

---

Algorithm 1 Compotation of a refined and stealthy RDI attack structure G_P

Input: a plant $G = (X, \Sigma, \delta, x_0, X_m)$, its supervisor $S = (X_S, \Sigma, \delta_S, x_{0,S}, X_{m,S})$, and a set of unsafe states $X_u \subseteq X$.

Output: A refined and stealthy RDI attack structure $G_P = (X_P, \Sigma_A, \delta_P, x_{0,P}, X_{m,P})$.

1. construct G_A based on Definition 2;
2. construct 1A automaton R;
3. $V/G \leftarrow G\|S$;
4. $G_P \leftarrow PruningGa(V/G, R, G_A)$;

Output: G_P.

Function $G_P = PruningGa(V/G, R, G_A)$

Input: A closed-loop system $V/G = (X_{V/G}, \Sigma, \delta_{V/G}, x_{0,V/G}, X_{m,V/G})$, RDI attack structure $G_A = (X_A, \Sigma_A, \delta_A, x_{0,A}, X_{m,A})$, and 1A rule $R = (X_R, \Sigma_R, \delta_R, x_{0,R}, X_{m,R})$.

Output: A pruned automaton $G_P = (X_P, \Sigma_A, \delta_P, x_{0,P}, X_{m,P})$.

1. Let $X_{dead} \leftarrow \emptyset$, $B \leftarrow \emptyset$, $i \leftarrow 0$;
2. $Y_0 \leftarrow Ac(G_A\|R)$;
3. **for** each $x \in X_{Yi} \setminus (X_{m,Yi} \cup (X_{m,V/G} \times X_R))$ **do**
4.   **if** $\Gamma_{Yi}(x) = \emptyset$ **then**
5. $X_{dead} \leftarrow X_{dead} \cup \{x\}$;
6.   **end if**
7. **end for**
8. **for** each $x \in X_{Yi}$ **do**
9.   **if** $\exists e \in \Sigma_{\bar{c}}$, s. t. $\delta_{Yi}(x, e) \in X_{dead}$ **then**
10. $X_{dead} \leftarrow X_{dead} \cup \{x\}$;
11.   **else if** $P_G(\Gamma_{Yi}(x)) \neq P_G(\Gamma_{Y0}(x)) \wedge \Gamma_{Yi}(x) \cap \Sigma_I = \emptyset$ **then**
12. $X_{dead} \leftarrow X_{dead} \cup \{x\}$;
13.   **end if**
14. **end for**
15. $Y_{i+1} \leftarrow Ac(Y_i, X_{dead})$;
16. **if** $Y_{i+1} \neq Y_i$ **then**
17. $i \leftarrow i + 1$;
18.   go to step 8;
19. **else**
20. $Y' \leftarrow Trim(Y_{i+1})$;
21. **end if**
22. **for** each $x \in X_Y$ **do**
23.   **if** $\exists e \in \Sigma$, s. t. $\delta_Y(x, e) \in X_{m,Y}$ **then**
24. $B \leftarrow \bigcup_{\sigma \in \Gamma_Y(x) \cap \{\sigma \in \Sigma_{RD} \mid P_G(\sigma) = e\}} \delta_Y(x, \sigma)$;
25.   **end if**
26. **end for**
27. $G_P \leftarrow Trim(Y, B)$;
28. **Output:** G_P.

X_{dead} be a set of dead states and B be a global variable. First, *PruningGa* initializes X_{dead} and B. Let $i = 0$ and we obtain an automaton by $Y_i = Ac(G_A \| R) = (X_{Yi}, \Sigma_A, \delta_{Yi}, x_{0,Yi}, X_{m,Yi})$. This step removes strings that violate the 1A rule from G_A. Steps 3-7 compute X_{dead} in Y_i. Steps 8-21 adopt a pruning process from [2] to eliminate strings that can lead the system to dead states, which can be considered as a standard supervisory control problem. It iteratively removes dead states from Y_i as well as those that can reach dead ones via uncontrollable events. More details can be found in [2].

In practice, there is no need to launch any attack once an unsafe state is reached. For $x \in X_{Y'}$ and $e \in \Sigma$, $\delta_Y(x, e) \in X_{m,Y'}$ indicates that an unsafe state is reached when e occurs. Modifications on e are considered redundant. $\forall \sigma \in \Gamma_{Y'}(x) \cap \{\sigma \in \Sigma_{RD} | P_G(\sigma) = e\}$, $\delta_{Y'}(x, \sigma)$ is a redundant state and added to set B. Steps 20-27 of *PruningGa* remove all the states in B from Y', and derive an accessible and co-accessible automaton G_P, which is the refined and stealthy attack structure.

The burdensome parts of Algorithm 1 are construction and pruning of G_A. G_A has at most $|X| \times |X_S|$ states. For each state in X_A, we have to enumerate each event in Σ_A at it, where $|\Sigma_A| = |\Sigma| + |\Sigma_{RD}| + |\Sigma_I| = |\Sigma| + |\Sigma_o| \times |\Sigma_o| + |\Sigma_o|$. The complexity to construct G_A is $O(|X| \times |X_S| \times (|\Sigma| + |\Sigma_o| \times |\Sigma_o| + |\Sigma_o|) \approx O(|X|^2 \times |\Sigma|^2)$. In *PruningGa*, steps 2-7 compute $G_A \| R$ and find dead states, which takes $O(|X_A| \times |X_R|)$. There are at most $|X_A| \times |X_R|$ states in Y_i. For each state in X_{Yi}, steps 9-13 check active events at it, which takes $O(|\Sigma_A|)$. The complexity of steps 8-14 is $O(|X_A| \times |X_R| \times |\Sigma_A|)$ and that of 15 is $O(|X_A| \times |X_R|)$. After removing dead states from Y_i, we check if $Y_{i+1} \neq Y_i$. If so, a "go to" procedure is called, which runs $|X_A| \times |X_R|$ times in the worst case. Then, steps 22-26 take $O(|X_A| \times |X_R|)$ to remove redundant states. The overall complexity of *PruningGa* is $O(|X_A| \times |X_R|) + O(|X_A|^2 \times |X_R|^2 \times |\Sigma_A|) + O(|X_A|^2 \times |X_R|^2) + O(|X_A| \times |X_R|) \approx O(|X_A|^2 \times |X_R|^2 \times |\Sigma_A|) \approx O(|X|^4 \times |\Sigma|^2)$. Thus, the overall complexity of Algorithm 1 is $O(|X|^2 \times |\Sigma|^2) + O(|X|^4 \times |\Sigma|^2) \approx O(|X|^4 \times |\Sigma|^2)$.

Algorithm 2: It is provided to compute $L_k(G_P)$ for a weakly k-step attackable system. Algorithm 2 starts with constructing the k-step attack recognizer $G^k_{AR}(G)$. It then initializes a set φ, and searches for states in $G^k_{AR}(G)$ at which the required number of attacks exceeds k. For each $(x_{AR}, \omega) \in X^k_{AR}$, (x_{AR}, ω) is added into φ if $\omega = -1$. Then, $Trim(G^k_{AR}(G), \varphi)$ is called to remove states in φ from $G^k_{AR}(G)$. It returns an automaton $H = (X_H, \Sigma_H, \delta_H, x_{0,H}, X_{m,H})$, which is a subautomaton of $G^k_{AR}(G)$, and $\forall (x_{AR}, \omega) \in X_H$, $\omega \geq 0$, if $H \neq \varnothing$. Namely, $\forall s \in L_m(H)$, $|s| \leq k$. Then, we perform $Trim(G_P \| H)$ to remove strings in G_P that require more than k attack operations. The resultant automaton is denoted as $G_K = (X_K, \Sigma_K, \delta_K, x_{0,K}, X_{m,K})$, from which $L_k(G_P)$ can be derived.

In Algorithm 2, we first construct $G^k_{AR}(G)$, which can be done in $O((k + 2) \times 2^{2|X|^2+1} \times |\Sigma_o|)$. The computation of φ and the trim operation $Trim(G^k_{AR}(G), \varphi)$ take the same complexity of $O((k + 2) \times 2^{2|X|^2})$, since there are at most $(k + 2) \times 2^{2|X|^2}$ states in $G^k_{AR}(G)$. H is a subautomaton of $G^k_{AR}(G)$, which also contains at most $(k + 2) \times 2^{2|X|^2}$ states. The composition of G_P and H takes $|X_P| \times |X_H| = 2|X|^2 \times (k + 2) \times 2^{|X|^2}$ operations. Thus, the total complexity of Algorithm 2 is $O((k + 2) \times 2^{2|X|^2+1} \times |\Sigma_o| + (k + 2) \times 2^{2|X|^2} + (k + 2) \times 2^{2|X|^2} + 2|X|^2 \times (k + 2) \times 2^{|X|^2})$, which is simplified to $O((k + 2) \times 2^{|X|^2+1} \times |\Sigma_o|)$.

Algorithm 2 Computing $L_k(G_P)$ for a weakly k-step attackable system

Input: A plant G, an attack structure G_P, and $k \in \mathbb{N}^+$.
Output: An automaton G_K that recognizes $L_k(G_P)$.

```
construct G^k_AR(G);
let φ ← ∅;
for each (x_AR, ω) ∈ X^k_AR do
  if ω = -1 then
    φ ← φ ∪ {(x_AR, ω)};
  end if
end for
H ← Trim(G^k_AR(G), φ);
G_K ← Trim(G_P || H);
Output: G_K.
```

III. Proofs

***Proposition* 1**: Given a plant G, its supervisor S and a set of unsafe states X_u as inputs of Algorithm 1, the output G_P provides stealthy attacks that adhere to the 1A rule.

Proof: *PruningGa* shows that $Y_0 = Ac(G_A \| R)$ and $\Sigma_A = \Sigma_R$. We have $L(Y_0) \subseteq L(G_A) \cap L(R) \Rightarrow L(Y_i) \subseteq L(G_A) \cap L(R)$, for $i \in \mathbb{N}$. When $Y_{i+1} = Y_i$, we have $Y' = Trim(Y_{i+1})$ and $G_P = Trim(Y', B)$. It holds that $L(G_P) \subseteq L(G_A) \cap L(R)$. Hence, G_P follows the 1A rule. As for stealthiness, it trivially holds due to the construction of G_A and pruning process in [2]. ■

***Proposition* 2**: If G is k-step attackable w.r.t. S and G_P, then G is k'-step attackable w.r.t. S and G_P for any $k' > k$, where $k, k' \in \mathbb{N}^+$.

Proof: Suppose that G is not k'-step attackable. By Definition 4, it implies that G is not k-step attackable for any $k < k'$. By contrapositive, the proposition is true. ■

***Theorem* 1**: Given a plant G, its supervisor S, an attack structure G_P, there exists an integer $k \in \mathbb{N}^+$, such that G is k-step attackable w.r.t. S and G_P iff its attack recognizer $G_{AR}(G)$ is loop-free.

Proof: ($\Rightarrow$) By contrapositive, assume that there is a loop l_1: $x_{1,AR} \xrightarrow{\sigma_1} x_{2,AR} \xrightarrow{\sigma_2} \ldots\ x_{n,AR} \xrightarrow{\sigma_n} x_{1,AR}$ in $G_{AR}(G)$, where $n \in \mathbb{N}^+$, $\sigma_i \in (\Sigma_{RD} \cup \Sigma_I)$ for $i \in \{1, 2, \ldots, n\}$. Based on Definition 5, there exists a sequence l_2 in G_P:

$$x_{1,P} \xrightarrow{\sigma_1} x_{2,P} \xrightarrow{t_1} x'_{2,P} \xrightarrow{\sigma_2} x_{3,P} \xrightarrow{t_2} \ldots \xrightarrow{t_{n-1}} x'_{n,P} \xrightarrow{\sigma_n} x_{n+1,P},$$

where $x_{1,P}$ is a component of $x_{1,AR}$, $t_m \in \Sigma^*$, $x_{j,AR} = \widetilde{N}(x_{j,P})$, for $m \in \{1, 2, \ldots, n - 1\}$ and $j \in \{2, \ldots, n\}$. We have $\delta_P(x_{n,P}, t_{n-1}\sigma_n) =$

$x_{n+1,P}$, and $x_{1,AR} = \widetilde{N}(x_{n+1,P})$. Since $x_{1,P}$ is a component of $x_{1,AR}$, $x_{1,P} \in \widetilde{N}(x_{n+1,P})$, $\exists w \in \Sigma^*$, such that $x_{1,P} = \delta_P(x_{P,n+1}, w)$, leading to l_3:

$$x_{1,P} \xrightarrow{\sigma_1} x_{2,P} \xrightarrow{t_1} x'_{2,P} \xrightarrow{\sigma_2} x_{3,P} \xrightarrow{t_2} \dots \xrightarrow{t_{n-1}} x'_{n,P} \xrightarrow{\sigma_n} x_{n+1,P} \xrightarrow{w} x_{1,P},$$

which is a loop that contains attack operations in G_P. It implies that the system may stay in the loop when an attack strategy involving l_3 is performed, which results in an infinite number of attacks. It contradicts that G is k-step attackable w.r.t. G_P.

($\Leftarrow$) By contrapositive, assume that there does not exist an integer $k \in \mathbb{N}^+$, such that G is k-step attackable w.r.t. G_P. It implies that $\exists s \in L_m(G_P)$ and s contains an infinite number of attack operations. We have $P_A(s) = w = \sigma_1\sigma_2\sigma_3\dots$, and $w \in L(G_{AR}(G))$, where $\sigma_i \in \Sigma_{RD} \cup \Sigma_I$ and $i \in \mathbb{N}^+$. Since $G_{AR}(G)$ is a finite state automaton, an infinite string w should be regular with the form $w = (\sigma_1\sigma_2\sigma_3\dots\sigma_i)^*$, which is a loop in $G_{AR}(G)$. It contradicts that $G_{AR}(G)$ is loop-free. ∎

***Lemma* 1:** For any string $s \in (\Sigma_{RD}\cup\Sigma_I)^*$ with $|s| \geq k + 1$, we have $\delta^k_{AR}((x_{0,AR}, k), s) = (\delta_{AR}(x_{0,AR}, s), -1)$ if $\delta_{AR}(x_{0,AR}, s)!$.

Proof: Let $s = \sigma_1\sigma_2\dots\sigma_{k+1} \in L(G^k_{AR}(G))$, whe re $\sigma_i \in \Sigma_{RD} \cup \Sigma_I$ and $i \in \{1, 2, \dots, k+1\}$. $\delta^k_{AR}((x_{0,AR}, k), s) = \delta^k_{AR}((\delta_{AR}(x_{0,AR}, \sigma_1), k\text{-}1), \sigma_2\dots\sigma_{k+1}) = \dots = \delta^k_{AR}((\delta_{AR}(x_{0,AR}, \sigma_1\sigma_2\dots\sigma_k), k\text{-}k), \sigma_{k+1}) = (\delta_{AR}(x_{0,AR}, \sigma_1\sigma_2\dots\sigma_k\sigma_{k+1}), -1)$. It is intuitive that similar results can be obtained for any string $s' \in (\Sigma_{RD}\cup\Sigma_I)^*$ with $|s'| \geq |s|$. ∎

***Theorem* 2**: Given a plant G, its supervisor S, an attack structure G_P, and $k \in \mathbb{N}^+$, $G^k_{AR}(G)$ is the k-step attack recognizer. G is k-step attackable w.r.t. S and G_P iff $\forall(x_{AR}, \omega) \in X^k_{AR}$, $\omega \geq 0$.

Proof: ($\Rightarrow$) By contrapositive, suppose that $\exists(x_{AR}, \omega) \in X^k_{AR}$, $\omega = -1$. It implies that $\exists s \in L(G^k_{AR}(G))$, such that $\delta^k_{AR}((x_{0,AR}, k), s) = (x_{AR}, -1)$. It leads to a sequence $(x_{0,AR}, k) \xrightarrow{\sigma_1} (x_{1,AR}, k\text{-}1) \xrightarrow{\sigma_2} \dots\ \sigma_k(x_{k,AR}, 0) \xrightarrow{\sigma_{k+1}} (x_{k+1,AR}, -1) \xrightarrow{\sigma_{k+2}} \dots \xrightarrow{\sigma_{|s|}} (x_{AR}, -1)$, where $\sigma_i \in \Sigma_{RD} \cup \Sigma_I$ and $i \in \{1, 2, \dots, |s|\}$. It shows that $|s| = |\sigma_1\sigma_2\dots\sigma_{k+1}\dots\sigma_{|s|}| \geq k+1$. Since $L(G^k_{AR}(G)) = P_A(L(G_P))$, $\exists s' \in L(G_P)$, such that $P_A(s') = s$. Since G_P is a trim, $\exists v \in \Sigma^*_A$, such that $s'v \in L_m(G_P)$, and $|P_A(s'v)| \geq k+1$, which contradicts that G is k-step attackable.

($\Leftarrow$) By contrapositive, assume that G is not k-step attackable w.r.t. S and G_P, which means $\exists s \in L_m(G_P)$ such that $|P_A(s)| \geq k+1$. Let $s' = P_A(s)$. We have $s' \in L(G^k_{AR}(G))$. By Lemma 1, $\delta^k_{AR}((x_{0,AR}, k), s') = (\delta_{AR}(x_{0,AR}, s'), -1) \in X^k_{AR}$, which contradicts to $\forall(x_{AR}, \omega) \in X^k_{AR}$, $\omega \geq 0$. ∎

***Theorem* 3**: Given a plant G and a supervisor S, G_P is the attack structure w.r.t. S. For any $k' > k = 2^{2|X|^2}\text{-}1$, G is k'-step attackable w.r.t. S and G_P, iff G is k-step attackable w.r.t. S and G_P.

Proof: ($\Rightarrow$) For any $k' > k = 2^{2|X|^2}\text{-}1$, we prove that G is k-step attackable if G is k'-step attackable. Assuming that G is not k-step attackable, it implies that $\exists s \in L_m(G_P)$, such that $|P_A(s)| > k$. We have $\exists s' \in L_m(G_{AR}(G))$, such that $s' = P_A(s)$ and $|s'| > k$. Without loss of generality, let $|s'| = k + 1$, since the proof can be generalized inductively to any $k' > k$. The string s' leads to a sequence l in $G_{AR}(G)$: $x_{0,AR} \xrightarrow{\sigma_1} x_{1,AR} \xrightarrow{\sigma_2} \dots\ x_{k,AR} \xrightarrow{\sigma_{k+1}} x_{k+1,AR}$,where $s' = \sigma_1\sigma_2\dots\sigma_{k+1}$, and $|s'| = |\sigma_1\sigma_2\dots\sigma_{k+1}| = k + 1$. It is obvious that l contains $k + 1$ ($= 2^{2|X|^2}$) attack operations and $k + 2$ ($= 2^{2|X|^2}+1$) states. Since $G_{AR}(G)$ contains at most $2^{2|X|^2}$ states, at least two states $x_{i,AR}$ and $x_{j,AR}$ exist in l, such that $x_{i,AR} = x_{j,AR}$, where $i, j \in \{0, 1, \dots, k\}$. It indicates that there is a loop in $G_{AR}(G)$. By Theorem 1 and Proposition 2, we conclude that G is not k'-step attackable.

($\Leftarrow$) It is obvious that G is k'-step attackable if G is k-step attackable by Proposition 2. ∎

***Theorem* 4**: Given a plant G, its supervisor S, an attack structure G_P, and $k \in \mathbb{N}^+$, $G^k_{AR}(G) = (X^k_{AR}, \Sigma_{RD}\cup\Sigma_I, \delta^k_{AR}, x^k_{0,AR}, X^k_{m,AR})$ is the k-step attack recognizer. G is weakly k-step attackable w.r.t. S and G_P iff $\exists(x_{AR}, \omega) \in X^k_{m,AR}$, $\omega \geq 0$.

Proof: ($\Rightarrow$) If G is weakly k-step attackable, $\exists s \in L_m(G_P)$, such that $|P_A(s)| \leq k$, i.e., $\exists s' \in L_m(G^k_{AR}(G))$, such that $P_A(s) = s'$ and $|s'| \leq k$. It leads to sequence in $G^k_{AR}(G)$: $(x_{0,AR}, k) \xrightarrow{\sigma_1} (x_{1,AR}, k\text{-}1) \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_{|s'|}} (x_{AT}, k\text{-}|s'|)$. Since $|s'| \leq k$, we have $k\text{-}|s'| \geq 0$.

($\Leftarrow$) If $\exists(x_{AR}, \omega) \in X^k_{m,AR}$, $\omega \geq 0$, there exists a string $s \in L_m(G^k_{AR}(G))$, such that $\delta^k_{AR}((x_{0,AR}, k), s) = (x_{AR}, s)$ and $|s| \leq k$. Since $L_m(G^k_{AR}(G)) = P_A(L_m(G_P))$, we have $\exists s' \in L_m(G_P)$, such that $P_A(s') = s$ and $|P_A(s')| \leq k$. Hence, G is weakly k-step attackable w.r.t. S and G_P. ∎

***Proposition* 3**: Given a plant G, an attack structure G_P, and $k \in \mathbb{N}^+$, let G_K be the output of Algorithm 2 and $L_k(G_P) = \{s \in L_m(G_P) |\ |P_A(s)| \leq k\}$. It holds that $L_m(G_K) = L_k(G_P)$.

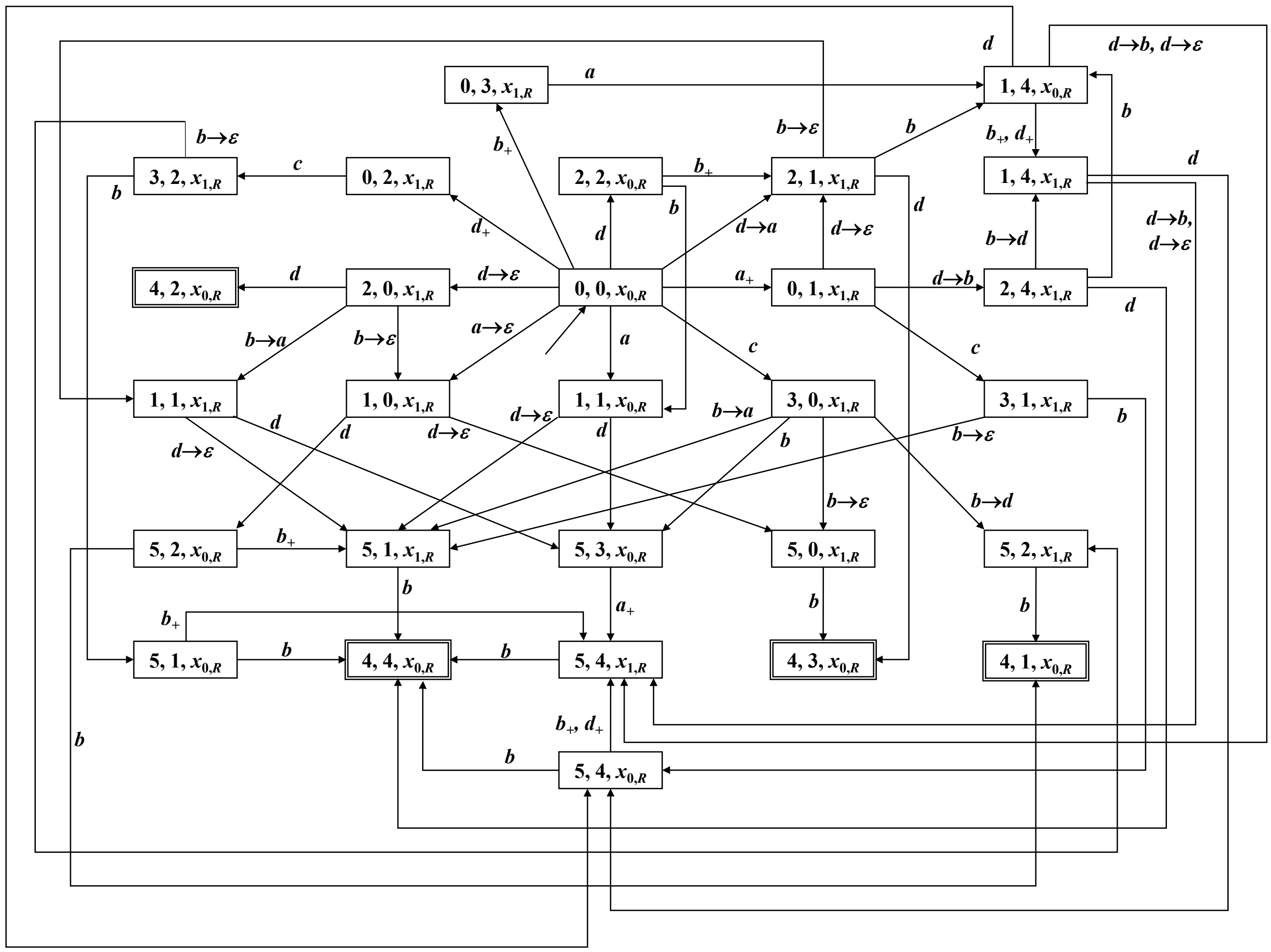
Proof: Consider the construction of G_K. Since $\Sigma_H = \Sigma_{RD} \cup \Sigma_I$ and $\Sigma_P = \Sigma_A = \Sigma_{RD} \cup \Sigma_I \cup \Sigma$, we have $L_m(G_K) = L_m(G_P \parallel H) = P^{-1}_{(\Sigma_P\cup\Sigma_H)^*\to\Sigma^*_P} L_m(G_P) \cap P^{-1}_{(\Sigma_P\cup\Sigma_H)^*\to\Sigma^*_H} L_m(H) = L_m(G_P) \cap P^{-1}_A L_m(H)$. We show that $L_m(G_K) \subseteq L_k(G_P)$. Since $\forall(x_{AR}, \omega) \in X_H$, $\omega \geq 0$, we have $\forall w \in L_m(H)$, $|w| \leq k$. For any $s \in L_m(G_K)$, we have $s \in L_m(G_P) \cap P^{-1}_A L_m(H)$. Since $P_A(L_m(G_P) \cap P^{-1}_A L_m(H)) \subseteq P_A(L_m(G_P)) \cap L_m(H)$, we have $P_A(s) \in L_m(H) \Rightarrow |P_A(s)| \leq k$. It is obvious that $\forall s \in L_m(G_K)$, $s \in L_k(G_P)$.

Then, we show that $L_k(G_P) \subseteq L_m(G_K)$. Let $s \in L_k(G_P)$. By $s \in L_m(G_P)$ and $|P_A(s)| \leq k$, we know that $P_A(s) \in L_m(G^k_{AR}(G))$ and $\delta^k_{AR}((x_{0,AR}, k), P_A(s)) = (x_{AR}, \omega) \in X^k_{m,AR}$, where $\omega \geq 0$. $P_A(s)$ is retained in H since (x_{AR}, ω) is not removed. Thus, we have $P_A(s) \in L_m(H) \Rightarrow s \in P^{-1}_A L_m(H) \Rightarrow s \in L_m(G_K)$.

In summary, $L_k(G_P) = L_m(G_K)$. ∎

IV. Supplementary Contents on Examples

A. G_P in Example 1



B. Details of $G_{AR}(G)$ in Example 2

TABLE I
DETAILS OF STATES IN $G_{AR}(G)$

State	Components
$x_{0,AR}$	$\{(0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 0, x_{1,R}), (5, 3, x_{0,R})\}$
$x_{1,AR}$	$\{(2, 1, x_{1,R}), (0, 3, x_{1,R}), (1, 4, x_{0,R}), (4, 3, x_{0,R}), (5, 4, x_{0,R}), (4, 4, x_{0,R})\}$
$x_{2,AR}$	$\{(0, 2, x_{1,R}), (3, 2, x_{1,R}), (5, 1, x_{0,R}), (4, 4, x_{0,R})\}$
$x_{3,AR}$	$\{(0, 1, x_{1,R}), (3, 1, x_{1,R}), (5, 4, x_{0,R}), (5, 4, x_{1,R}), (4, 4, x_{0,R})\}$
$x_{4,AR}$	$\{(2, 1, x_{1,R}), (4, 3, x_{1,R}), (1, 4, x_{0,R}), (5, 4, x_{0,R}), (4, 4, x_{0,R})\}$
$x_{5,AR}$	$\{(2, 0, x_{1,R}), (4, 2, x_{0,R}), (5, 1, x_{1,R}), (4, 4, x_{0,R})\}$
$x_{6,AR}$	$\{(1, 0, x_{0,R}), (5, 2, x_{0,R}), (4, 1, x_{0,R})\}$
$x_{7,AR}$	$\{(5, 2, x_{1,R}), (4, 1, x_{0,R})\}$
$x_{8,AR}$	$\{(2, 4, x_{1,R}), (1, 4, x_{0,R}), (5, 4, x_{0,R}), (4, 4, x_{0,R})\}$
$x_{9,AR}$	$\{(5, 4, x_{1,R}), (4, 4, x_{0,R})\}$
$x_{10,AR}$	$\{(1, 1, x_{0,R}), (5, 3, x_{0,R})\}$
$x_{11,AR}$	$\{(1, 4, x_{1,R}), (5, 4, x_{0,R}), (4, 4, x_{0,R})\}$
$x_{12,AR}$	$\{(5, 0, x_{1,R}), (4, 3, x_{0,R})\}$
$x_{13,AR}$	$\{(5, 1, x_{1,R}), (4, 4, x_{0,R})\}$
$x_{14,AR}$	$\{(5, 1, x_{0,R}), (4, 4, x_{0,R})\}$

C. Details of $G^1_{AR}(G)$ in Example 5

TABLE I
DETAILS OF STATES $x^1_{0,AR}$ - $x^1_{10,AR}$ IN $G^1_{AR}(G)$

State	x_{AR}	ω
$x^1_{0,AR}$	$\{(0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (4, 4, x_{0,R})\}$	1
$x^1_{1,AR}$	$\{(0, 1, x_{1,R}), (2, 3, x_{1,R}), (5, 4, x_{0,R})\}$	0
$x^1_{2,AR}$	$\{(2, 1, x_{1,R}), (5, 4, x_{0,R}), (0, 4, x_{1,R})\ (4, 0, x_{1,R}), (0, 4, x_{0,R}), (4, 0, x_{0,R})\}$	0
$x^1_{3,AR}$	$\{(2, 4, x_{1,R}), (5, 0, x_{0,R}), (0, 0, x_{1,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (4, 4, x_{0,R})\}$	0
$x^1_{4,AR}$	$\{(1, 4, x_{1,R}), (2, 0, x_{0,R}), (3, 1, x_{0,R}), (4, 4, x_{0,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (3, 4, x_{1,R}), (0, 4, x_{0,R}), (4, 0, x_{0,R}), (4, 0, x_{1,R})\}$	0
$x^1_{5,AR}$	$\{(1, 4, x_{1,R}), (2, 0, x_{0,R}), (3, 1, x_{0,R}), (4, 4, x_{0,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (3, 4, x_{1,R}), (0, 4, x_{0,R}), (4, 0, x_{0,R})\}$	0
$x^1_{6,AR}$	$\{(2, 4, x_{1,R}), (5, 0, x_{0,R}), (0, 0, x_{1,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (4, 4, x_{0,R})\}$	-1
$x^1_{7,AR}$	$\{(2, 4, x_{1,R}), (5, 0, x_{0,R}), (0, 4, x_{1,R}), (0, 4, x_{0,R}), (4, 0, x_{0,R}), (4, 0, x_{1,R}), (2, 1, x_{1,R}), (5, 4, x_{0,R}), (4, 4, x_{1,R}), (0, 0, x_{1,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (4, 4, x_{0,R})\}$	-1
$x^1_{8,AR}$	$\{(2, 1, x_{1,R}), (5, 4, x_{0,R}), (0, 1, x_{1,R}), (2, 3, x_{1,R}), (4, 1, x_{1,R}), (0, 2, x_{0,R}), (1, 3, x_{0,R}), (3, 3, x_{1,R}), (4, 4, x_{0,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R})\}$	-1
$x^1_{9,AR}$	$\{(2, 0, x_{1,R}), (3, 1, x_{0,R}), (4, 4, x_{0,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (0, 0, x_{1,R}), (2, 4, x_{1,R}), (5, 0, x_{0,R}), (4, 0, x_{1,R}), (0, 4, x_{0,R}), (4, 0, x_{0,R})\}$	-1
$x^1_{10,AR}$	$\{(2, 4, x_{1,R}), (5, 0, x_{0,R}), (0, 4, x_{1,R}), (0, 4, x_{0,R}), (4, 0, x_{0,R}), (4, 0, x_{1,R}), (4, 4, x_{1,R}), (0, 0, x_{1,R}), (0, 0, x_{0,R}), (1, 1, x_{0,R}), (2, 2, x_{0,R}), (3, 1, x_{1,R}), (3, 3, x_{0,R}), (4, 4, x_{0,R})\}$	-1

REFERENCES

[1] R. Su, "Supervisor synthesis to thwart cyber attack with bounded sensor reading alterations," *Automatica,* vol. 94, pp. 35-44, 2018.

[2] R. Meira-Góes, E. Kang, R. H. Kwong, and S. Lafortune, "Synthesis of sensor deception attacks at the supervisory layer of cyber–physical systems," *Automatica,* vol. 121, p. 109172, 2020.