Supplementary file for “Identification of Periodic Sensor-Reading Modification Attacks 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
$L(G)$	Language generated by G
$\Sigma_{\bar{o}}$	Set of unobservable events
Σ_o	Set of observable events
$\bar{R}(x)$	$\{x' \in X \| (s \in \Sigma_{\bar{o}}^*)\ \delta(x, s) = x'\}$, set of unobservable reach of x
$I(B, \sigma)$	$\{x \in B \| \delta(x, \sigma)!\}$, states in B at which σ is defined
P	P: $\Sigma^* \to \Sigma_o^*$, natural projection
$Obs(G)$	(Z, Σ_o, η, z_0), observer of G
A	(T, d, f), periodic sensor-reading modification (PSM) attack
T	Attack interval of PSM attacks
d	Attack duration of PSM attacks
f	f: $\Sigma_o \to 2^{\Sigma_o} \backslash \varnothing$, attack function of PSM attacks
$\hat{f}$	$\hat{f}$: $\Sigma_o \to 2^{\Sigma_o} \backslash \varnothing$, largest attack function of PSM attack
G_P	$(Q, \Sigma, \delta_P, q_0)$, attack model
G_A	$(X_A, \Sigma, \delta_A, x_{0,A})$, attacked plant
$\tilde{L}$	Set of corrupted observations under PSM attacks
Δ	$\{F, C, U\}$, set of state labels
E	$(Z_E, \Sigma_o, \eta_E, z_{0,E})$, A-estimator
$\widetilde{E}$	Subautomaton of E that recognizes $\tilde{L}$

II. ALGORITHMS

In this section, we provide explanations of function *ConstructGa* and Algorithm 1 as well as their computational analysis.

Function *ConstructGa*: It is designed to construct the attacked plant. It embeds all modified strings defined by A and displays the state evolutions of G that are consistent with these strings. Given a live plant G and a model G_P that captures a PSM attack A, function *ConstructGa* operates as follows. It first initializes an automaton $G_A = (X_A, \Sigma, \delta_A, x_{0,A})$, where $x_{0,A} = (x_0, q_0)$ and $X_A = \{x_{0,A}\}$. Each $x_A \in X_A$ can be denoted as $x_A = (x, q)$, where $x \in X$ and $q \in Q$. If an event σ is defined at x, step 4 verifies whether it can be modified at (x, q). If $\delta_P(q, \sigma) = q_i^\sigma$ for $i \in \{1, 2, \ldots, d\}$, it indicates that σ is observed during an active attack phase and can be altered to any event in $f(\sigma)$. Steps 5-9 define each $\sigma' \in f(\sigma)$ at (x, q) as follows: σ' leads to a transition $(x, q) \xrightarrow{\sigma'} (\delta(x, \sigma), \delta_P(\delta_P(q, \sigma), \sigma'))$, where the transitional state q_i^σ is ignored. It represents that an observable event σ is generated at x in G, while being modified to σ' and sent to estimators. Then, X_A and δ_A are updated based on such a transition. If $\delta_P(q, \sigma) \neq q_i^\sigma$, there are two cases: $\sigma \in \Sigma_{\bar{o}}$ or the attack is in a sleeping phase, with no modification in either case. Once σ occurs, it leads to a state transition $(x, q) \xrightarrow{\sigma} (\delta(x, \sigma), \delta_P(q, \sigma))$. Similarly, X_A and δ_A are accordingly updated based. The function repeats the above steps for each state of G_A until there are no further updates. It finally returns an automaton that embeds all modified strings, which is an NFA.

Function G_A = *ConstructGa*(G, G_P)

Input: a plant $G = (X, \Sigma, \delta, x_0)$ and a PSM attack model $G_P = (Q, \Sigma, \delta_P, q_0)$.
Output: an attacked plant $G_A = (X_A, \Sigma, \delta_A, x_{0,A})$.

1) initialize $G_A = (X_A, \Sigma, \delta_A, x_{0,A})$ with $x_{0,A} \leftarrow (x_0, q_0)$, $X_A \leftarrow \{x_{0,A}\}$ and $\delta_A \leftarrow \varnothing$;
2) **for** each $(x, q) \in X_A$ **do**
3)   **for** each $\sigma \in \Gamma(x)$ **do**
4)   **if** $\delta_P(q, \sigma) \in \{q_1^\sigma, \ldots, q_d^\sigma\}$ **then**
5)   **for** each $\sigma' \in f(\sigma)$ **do**
6) $x_A' \leftarrow (\delta(x, \sigma), \delta_P(\delta_P(q, \sigma), \sigma'))$;
7) $X_A \leftarrow X_A \cup \{x_A'\}$;
8)   add transition $x_A \xrightarrow{\sigma'} x_A'$ to δ_A;
9)   **end for**
10)   **else**
11) $x_A' \leftarrow (\delta(x, \sigma), \delta_P(q, \sigma))$;
12) $X_A \leftarrow X_A \cup \{x_A'\}$;
13)   add transition $x_A \xrightarrow{\sigma} x_A'$ to δ_A;
14)   **end if**
15)   **end for**
16) **end for**
17) **output**: G_A.

G_A synchronously displays the state evolution of G and G_P while ignoring the transitional state q_i^σ, i.e., $X_A \subseteq X \times \{q_0, q_1^\#, \ldots, q_d^\#, q_{d+1}, \ldots, q_{T\text{-}1}\}$. Thus, there are at most $T \times |X|$ states in G_A. For each $x_A \in X_A$, steps 3-15 of *ConstructGa* check active events and modifications on it, which takes $|\Sigma| \times |\Sigma_o|$. The overall complexity of *ConstructGa* is $O(T \times |X| \times |\Sigma| \times |\Sigma_o|) \approx O(T \times |X| \times |\Sigma|^2)$.

Algorithm 1 Construct A-estimator E

Input: a plant G, the largest PSM attack model G_P, and a set of labels $\Delta = \{F, C, U\}$.
Output: an A-estimator $E = (Z_E, \Sigma_o, \eta_E, z_{0,E})$.
1) $G_A \leftarrow ConstructGa(G, G_P)$;
2) construct $Obs(G)$ and $Obs(G_A)$;
3) initialize $E = (Z_E, \Sigma_o, \eta_E, z_{0,E})$ with $z_{0,E} \leftarrow z_{0,A} \times \{U\}$, $Z_E \leftarrow \{z_{0,E}\}$ and $\eta_E \leftarrow \varnothing$;
4) **for each** $z_E \in Z_E$ **do**
5) **for each** $\sigma \in \Sigma_o$ do
6) find $z_A \in Z_A$ such that $z_E \subseteq z_A \times \Delta$ and $|z_A| = |z_E|$;
7) find $z_A' \in Z_A$ such that $\eta_A(z_A, \sigma) = z_A'$;
8) $z_E' \leftarrow \varnothing$;
9) **for each** $x_A l \in z_E$ **do**
10) **for each** $x_A' \in \overline{R}(\delta_A(x_A, \sigma))$ **do**
11) **if** (C1), (C2) or (C3) is true w.r.t. x_A, z_A, and z_A' **then**
12) $z_E' \leftarrow z_E' \cup \{x_A'F\}$;
13) **end if**
14) **end for**
15) **for each** $x_A' \in \overline{R}(\delta_A(x_A, \sigma))$ s.t. $\{x_A'\} \times \Delta \cap z_E' = \varnothing$ **do**
16) **if** C4 is true w.r.t. x_A, z_A, and z_A' **then**
17) $z_E' \leftarrow z_E' \cup \{x_A'C\}$;
18) **else**
19) $z_E' \leftarrow z_E' \cup \{x_A'U\}$;
20) **end if**
21) **end for**
22) **end for**
23) add transition $z_E \xrightarrow{\sigma} z_E'$ to η_E;
24) $Z_E \leftarrow Z_E \cup \{z_E'\}$;
25) **end for**
26) **end for**
27) **Output**: E.

Algorithm 1: It constructs an A-estimator. First, it computes an attack structure G_A embedding function $\hat{f}$ by calling $G_A = ConstructGa(G, G_P)$, and constructs both $Obs(G)$ and $Obs(G_A)$. Next, it initializes A-estimator E with $Z_E = \{z_{0,E}\}$ and $z_{0,E} = z_{0,A} \times \{U\}$, i.e., each component of $z_{0,A}$ is assigned a label U. For each $z_E \in Z_E$ and $\sigma \in \Sigma_o$, we find two states in $Obs(G_A)$: a) a state z_A obtained by erasing labels of z_E; b) a state z_A' that is reached from z_A by enabling σ, i.e., $\eta_A(z_A, \sigma) = z_A'$. We then compute state z_E' that is reached from z_E after firing σ. To be precise, z_E' is first initialized. For each $x_A l \in z_E$ and $x_A' \in \overline{R}(\delta_A(x_A, \sigma))$, where $l \in \Delta$, the algorithm determines the label l' of x_A' based on C1 - C4 and add $x_A'l'$ into z_E'. After verifying each $x_A l \in z_E$, we add a transition $z_E \xrightarrow{\sigma} z_E'$ to η_E and update Z_E to $Z_E = Z_E \cup \{z_E'\}$.

In Algorithm 1, *ConstructGa* is first called, with a computational cost of $O(T \times |X| \times |\Sigma|^2)$. The computation of $Obs(G)$ and $Obs(G_A)$ requires $O(2^{|X|})$ and $O(2^{T \times |X|})$ time. Next, an iteration process is adopted for at most $2^{3 \times T \times |X|}$ states in Z_E, where at most $|\Sigma|$ events can be verified at a state $z_E \in Z_E$. Steps 6 and 7 require finding two states in Z_A, which both take $O(2^{T \times |X|})$. For each $(x_A, l) \in z_E$ the algorithm evaluates conditions C1 - C4 at each $x_A' \in \overline{R}(\delta_A(x_A, \sigma))$, where both z_E and $\overline{R}(\delta_A(x_A, \sigma))$ contain at most $T \times |X|$ components. Specifically, C1 takes $O((T \times |X|)^2 \times |\Sigma|^2)$, due to checking all pairs of events and states, while C2 and C3 each require $O(T \times |X|)$ time. The complexity to determine if C4 holds is $O(T \times |X| \times 2^{|X|})$, since it has to map z_A to a state in $Obs(G)$. As a result, the complexity of steps 9 - 22 is $O(T \times |X| \times T \times |X| \times ((T \times |X|)^2 \times |\Sigma|^2 + T \times |X| + T \times |X| + T \times |X| \times 2^{|X|})$. Therefore, the overall complexity of Algorithm 1 is $O(T \times |X| \times |\Sigma|^2) + O(2^{|X|}) + O(2^{T \times |X|}) + O(2^{3 \times T \times |X|} \times |\Sigma| \times [2^{T \times |X|} + 2^{T \times |X|} + (T^2 \times |X|^2 \times ((T \times |X|)^2 \times |\Sigma|^2 + T \times |X| + T \times |X| + T \times |X| \times 2^{|X|})])$, which can be simplified to $O(2^{4 \times T \times |X|} \times |\Sigma|)$.

III. Proofs

Proposition 1: Given a plant G, a PSM attack model G_P, and $G_A = ConstructGa(G, G_P)$, we have $P(L(G_A)) = f(P(L(G)))$.

Proof: ($\subseteq$) Let $s = \sigma_1\sigma_2 \dots \sigma_{T+1} \in P(L(G_A))$ with $|s| = T + 1$ (proofs for $|s| < T + 1$ and $|s| > T + 1$ can be easily covered by this case), where $\sigma_i \in \Sigma_o$ for $i \in \{1, 2, \dots, T + 1\}$. $\exists w \in L(G_A)$, such that $P(w) = s$. Let $w = t_1\sigma_1t_2\sigma_2t_3 \dots t_{T+1}\sigma_{T+1}$, where $t_i \in \Sigma_{\bar{o}}^*$. According to *ConstructGa*, it leads to a trajectory in G_A:
$(x_0, q_0) \xrightarrow{t_1} (x_1, q_0) \xrightarrow{\sigma_1} (x_1', q_1^{\#}) \xrightarrow{t_2} \dots \xrightarrow{\sigma_d} (x_d', q_d^{\#}) \xrightarrow{t_{d+1}} (x_{d+1}, q_d^{\#}) \xrightarrow{\sigma_{d+1}} (x'_{d+1}, q_{d+1}) \xrightarrow{t_{d+2}} \dots \xrightarrow{\sigma_T} (x_T', q_0) \xrightarrow{t_{T+1}} (x_{T+1}, q_0) \xrightarrow{\sigma_{T+1}} (x'_{T+1}, q_1^{\#})$.
Based on steps 4-6 and 10-11, we have

1) $\exists \sigma_j' \in \Sigma_o$ for $j \in \{1, 2, \dots, d, T + 1\}$, such that $\sigma_j \in f(\sigma_j')$, $\delta_A((x_1, q_0), \sigma_1) = (\delta(x_1, \sigma_1'), \delta_P(\delta_P(q_0, \sigma_1'), \sigma_1) = (x_1', q_1^{\#})$ if $j = 1$, and $\delta_A((x_j, q^{\#}_{j-1}), \sigma_j) = (\delta(x_j, \sigma_j'), \delta_P(\delta_P(q^{\#}_{j-1}, \sigma_j'), \sigma_j) = (x_j', q_j^{\#})$ if $j \in \{2, \dots, d\}$. It means that $\delta(x_j, \sigma_j')!$ and $\delta(x_j, \sigma_j') = x_j'$. Note that in this context, σ_j denotes a modified event observed under attack, and σ_j' is the possible original event that may have been altered into σ_j.
2) $\forall \sigma_m \in \Sigma_o$ for $m \in \{d + 1, d + 2, \dots, T\}$, we have $\delta_A((x_m, q_{m-1}), \sigma_m) = (\delta(x_m, \sigma_m), \delta_P(q_{m-1}, \sigma_m)) = (x_m', q_m)$ if $m \in \{d + 2, \dots, T - 1\}$, $\delta_A((x_{d+1}, q_d^{\#}), \sigma_{d+1}) = (x'_{d+1}, q_{d+1})$, and $\delta_A((x_T, q_{T-1}), \sigma_T) = (x_T', q_0)$.

Thus, there exists a trajectory in G:
$x_0 \xrightarrow{t_1} x_1 \xrightarrow{\sigma_1'} x_1' \xrightarrow{t_2} \dots \xrightarrow{\sigma_d'} x_d' \xrightarrow{t_{d+1}} x_{d+1} \xrightarrow{\sigma_{d+1}} x'_{d+1} \xrightarrow{t_{d+2}} \dots \xrightarrow{\sigma_T} x_T' \xrightarrow{t_{T+1}} x_{T+1} \xrightarrow{\sigma'_{T+1}} x'_{T+1}$.
Let $w' = t_1\sigma_1't_2 \dots \sigma_d't_{d+1}\sigma_{d+1} \dots t_{T+1}\sigma'_{T+1}$. Intruder's observation on it is $P(w') = \sigma_1'\sigma_2' \dots \sigma_d'\sigma_{d+1}\sigma_{d+2} \dots \sigma_T\sigma'_{T+1}$. Possible modification on $P(w')$ under a PSM attack is $f(P(w')) = f(\sigma_1')f(\sigma_2') \dots f(\sigma_d')\sigma_{d+1}\sigma_{d+2} \dots \sigma_T f(\sigma'_{T+1})$. According to case 1), we know that $\sigma_j \in f(\sigma_j')$ for $j \in \{1, 2, \dots, d, T + 1\}$. Hence, $w = \sigma_1\sigma_2 \dots \sigma_{T+1} \in f(P(w'))$. Since $f(P(w')) \subseteq f(P(L(G)))$, we have $w \in f(P(L(G)))$.
($\supseteq$) Let $s \in f(P(L(G)))$ and $s = \sigma_1\sigma_2 \dots \sigma_{T+1}$ with $|s| = T + 1$ (proofs for $|s| < T + 1$ and $|s| > T + 1$ can be easily covered by this case), where $\sigma_i \in \Sigma_o$ for $i \in \{1, 2, \dots, T + 1\}$. $\exists w \in P(L(G))$, such that $w = \sigma_1'\sigma_2' \dots \sigma_d'\sigma_{d+1} \dots \sigma_T\sigma'_{T+1}$, where $\sigma_j \in f(\sigma_j')$ for $j \in \{1, 2, \dots, d, T + 1\}$. $\exists w' \in L(G)$, such that $w' = t_1\sigma_1't_2 \dots \sigma_d't_{d+1}\sigma_{d+1} \dots t_{T+1}\sigma'_{T+1}$. According to steps 2-16 of *ConstructGa*, there should be a string in $L(G_A)$: $t_1\sigma_1t_2 \dots \sigma_dt_{d+1}\sigma_{d+1} \dots t_{T+1}\sigma_{T+1}$, where $t_i \in \Sigma_{\bar{o}}^*$. We have $P(t_1\sigma_1t_2 \dots \sigma_dt_{d+1}\sigma_{d+1} \dots t_{T+1}\sigma_{T+1}) = \sigma_1\sigma_2 \dots \sigma_{T+1} = s \Rightarrow s \in P(L(G_A))$.

As a result, we have $P(L(G_A)) = f(P(L(G)))$. ■

***Lemma* 1**: Let $s \in \tilde{L}$ and $z_A = \eta_A(z_{0,A}, s)$. If state $x_A \in z_A$ is certain w.r.t. $s \in \tilde{L}$, then $\forall x_A' \in z_A$, x_A' is certain w.r.t. s.

Proof: By contradiction, suppose that $\exists x_A' \in z_A$, x_A' is not certain w.r.t. s. If x_A' is fully ambiguous, then x_A is not certain based on C4, which contradicts the assumption that x_A is certain. If x_A' is uncertain, then $\nexists z \in Z$, such that $z \times \{q\} = z_A$. We cannot map z_A to a normal state in $Obs(G)$. No component of z_A can be certain, i.e., x_A is not certain w.r.t. s. ■

***Proposition* 3**: Let $s \in \tilde{L}$ and $z_A = \eta_A(z_{0,A}, s)$. If state $x_A \in z_A$ is certain w.r.t. s, then $|\hat{f}^{-1}(s) \cap P(L(G))| = 1$.

Proof: Let $s = \sigma_1\sigma_2 \ldots \sigma_{|s|}$, where $\sigma_i \in \Sigma_o$ and $i \in \{1, 2, \ldots, |s|\}$. We have a trajectory: $z_{0,A} \xrightarrow{\sigma_1} z_{1,A} \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_{|s|-1}} z_{|s|-1,A} \xrightarrow{\sigma_{|s|}} z_{|s|,A}$, where $z_{|s|,A} = z_A$. By Lemma 1, any component of $z_{|s|,A}$ is certain. We have that $\exists z \in Z$, such that $z \times \{q^{|s|}\} = z_{|s|,A}$, where $q^i \in Q$ is the second component of each state in $z_{i,A}$.

Let $I(z_{|s|-1,A}, \sigma_{|s|}) = \{x_A' \in z_{|s|-1,A} | \delta_A(x_A', \sigma_{|s|})!\}$. For each $x_A' \in I(z_{|s|-1,A}, \sigma_{|s|})$, x_A' is not fully ambiguous due to C3. We claim that for any $x_A' \in I(z_{|s|-1,A}, \sigma_{|s|})$, there exists only one state $z_{|s|-1} \in Z$, such that $x_A' \in z_{|s|-1} \times \{q^{|s|-1}\}$. Suppose that there exists another one $z' \in Z$, such that $z' \times \{q^{|s|-1}\} \cap I(z_{|s|-1,A}, \sigma_{|s|}) \neq \emptyset$. It implies that a state in $z' \times \{q^{|s|-1}\}$ can reach $z_{|s|,A}$ by enabling σ. It clearly results in case 2 discussed before. According to C3 and C4, a state in $z_{|s|,A}$ should be fully ambiguous, which contradicts the assumption that $x_A \in z_{|s|,A}$ is certain.

Then, we claim that $|\bigcup_{x_A' \in I(z_{|s|-1}, \sigma_{|s|})} \hat{f}^{-1}(\sigma_{|s|}) \cap \Gamma(x')| = 1$ if $q^{|s|-1} = q_0$, where x' is the first component of each x_A'. This claim can be easily proved by contradiction and is thus ignored. Let $e_{|s|} \in \Sigma_o$, such that $\eta(z_{|s|-1}, e_{|s|}) = z$. Then, we have $\{e_{|s|}\} = \hat{f}^{-1}(\sigma_{|s|})) \cap \Gamma(x')$ if $q^{|s|-1} = q_0$; and otherwise, $e_{|s|} = \sigma_{|s|}$.

The above two claims mean that we can determine the unique previous state of z as $z_{|s|-1}$ after observing $\sigma_{|s|}$, as well as the actual event has occurred at $z_{|s|-1}$. As $I(z_{|s|-1,A}, \sigma_{|s|}) \subseteq z_{|s|-1} \times \{q^{|s|-1}\}$ and none of states in $I(z_{|s|-1,A}, \sigma_{|s|})$ is fully ambiguous, the above two claims hold for any $x_A'' \in z_{|s|-2,A}$ if $\overline{R}(\delta_A(x_A'', \sigma_{|s|-1})) \subseteq I(z_{|s|-1,A}, \sigma_{|s|})$. We can find the unique previous state of $z_{|s|-1}$ and the actual event w.r.t. $\sigma_{|s|-1}$, denoted as $z_{|s|-2}$ and $e_{|s|-1}$. As a result, we can obtain a unique trajectory, which is the actual state evolution: $z_0 \xrightarrow{e_1} z_1 \xrightarrow{e_2} \ldots \xrightarrow{e_{|s|}} z_{|s|}$. Thus, we have $\hat{f}^{-1}(\sigma_1 \sigma_2 \ldots \sigma_{|s|}) \cap P(L(G)) = \{e_1e_2 \ldots e_{|s|}\}$ and $|\hat{f}^{-1}(\sigma_1\sigma_2 \ldots \sigma_{|s|}) \cap P(L(G))| = 1$. ■

***Corollary* 1**: Let $s \in \tilde{L}$ and $z_A = \eta_A(z_{0,A}, s)$. If a state $x_A \in z_A$ is certain w.r.t. s, we can find a unique trajectory $z_0 \xrightarrow{e_1} z_1 \xrightarrow{e_2} \ldots \xrightarrow{e_{|s|}} z_{|s|}$ in $Obs(G)$, such that $\hat{f}^{-1}(s) \cap P(L(G)) = \{e_1e_2 \ldots e_{|s|}\}$, where $e_i \in \Sigma_o$ and $i \in \{1, 2, \ldots, |s|\}$.

Proof: Directly from Proposition 2. ■

***Proposition* 4**: Let $s \in \tilde{L}$ and $z_A = \eta_A(z_{0,A}, s)$. If a state $x_A \in z_A$ is uncertain w.r.t. $s \in \tilde{L}$, we have $|\hat{f}^{-1}(s) \cap P(L(G))| \neq 1$.

Proof: It can be proved based on Lemma 1, Propositions 2 and 3, and the proof is omitted. ■

***Theorem* 1**: Given a plant G vulnerable to PSM attack $A = (T, f)$ with a known interval T, E is the A-estimator computed by Algorithm 1. Let a live language $\tilde{L} \subseteq L(E)$ be a set of observations, and $\tilde{E} \sqsubseteq E$ with $L(\tilde{E}) = \tilde{L}$. A is M-identifiable w.r.t. G and $\tilde{L}$ iff $\tilde{E}$ does not contain a loop where $\exists x_A l \in z_E$, $l \neq C$ for a state $z_E \in \tilde{Z}_E$ in the loop.

Proof: ($\Rightarrow$) By contradiction, assume that there exists a loop in $\tilde{E}$: $z_{1,E} \xrightarrow{\sigma_1} z_{2,E} \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_{m-1}} z_{m,E} \xrightarrow{\sigma_m} z_{1,E}$, where $\sigma_i \in \Sigma_o$, $z_{i,E} \in \tilde{Z}_E$, for all $i \in \{1, 2, \ldots, m\}$ and $m \in \mathbb{N}^+$. Let $s \in \tilde{L}$ and $\tilde{\eta}_E(z_{0,E}, s) = z_{j,E}$, where $\exists x_A l \in z_{j,E}$, $l \neq C$ and $j \in \{1, 2, \ldots, m\}$. We have $(\sigma_j\sigma_{j+1} \ldots \sigma_m\sigma_1 \ldots \sigma_{j-1})^* \subseteq \tilde{L}/s$. There always exists a string $t \in (\sigma_j\sigma_{j+1} \ldots \sigma_m\sigma_1 \ldots \sigma_{j-1})^*$ with $|t| > k$ for any arbitrarily large $k \in \mathbb{N}^+$, such that $\tilde{\eta}_E(z_{j,E}, t) = z_{j,E}$. By Proposition 1, $l \neq F$ for any $x_A l \in z_{j,E}$, and otherwise, A is not M-identifiable. Thus, $l = U$ for $x_A l \in z_{j,E}$, i.e., $\exists z_A \in Z_A$, such that $z_{j,E} = z_A \times \{U\}$. By Proposition 4, we have $|\hat{f}^{-1}(st) \cap P(L(G))| \neq 1$, i.e., $\exists w_1, w_2 \in P(L(G))$ with $w_1 \neq w_2$, such that $\{w_1, w_2\} \subseteq \hat{f}^{-1}(st) \cap P(L(G))$, which contradicts the assumption that A is M-identifiable w.r.t. G and $\tilde{L}$.

($\Leftarrow$) By contradiction, suppose that A is not M-identifiable w.r.t. G and $\tilde{L}$. Thus, $\exists s \in \tilde{L}$, $t_1 \in \tilde{L}/s$, and $|t_1| > k$ for any arbitrarily large $k \in \mathbb{N}$, such that $|\hat{f}^{-1}(st_1) \cap P(L(G))| \neq 1$. Let $\tilde{\eta}_E(z_{0,E}, s) = z_{1,E}$ and $\tilde{\eta}_E(z_{1,E}, t_1) = z_{2,E}$. Since L and $\tilde{L}$ are live and $\tilde{E}$ is a DFA, st_1 ultimately leads to a loop, i.e., $z_{2,E}$ can always reach a state in the loop via a sufficiently long string. Since $\tilde{E}$ does not contain a loop, where $\exists\, x_A l \in z_E$, $l \neq C$ for any $z_E \in \tilde{Z}_E$ in the loop. Thus, any loop in $\tilde{E}$ only involves certain states. It implies that $\forall t_2 \in L(\tilde{E})/st_1$ of sufficiently long length, $\tilde{\eta}_E(z_{2,E}, t_2)$ is a state in the loop. Let $t = t_1t_2$. We conclude that $\exists k \in \mathbb{N}$, $\forall t \in L(\tilde{E})/s$, $|t| > k \Rightarrow \forall x_A'l' \in \tilde{\eta}_E(z_{1,E}, t)$, $l' = C$. By Proposition 3, we have $|\hat{f}^{-1}(st) \cap P(L(G))| = 1$, which completes the contrapositive proof. ■

IV. Supplementary Contents on Examples

A. E in Example 4