Offline Opacity Enforcement and Online Mitigation Mechanism for Actuator-Enablement Attacks in Discrete Event Systems

João António da Silva Melo[1], Valeria Bonagura[2], Tiago Cruz[1], Paulo Simões[1], Federica Pascucci[2] and Graziana Cavone[2]

***Abstract*— Opacity is a fundamental confidentiality property in discrete-event systems, ensuring that an external, possibly malicious, observer cannot determine whether the system is in a secret state. A secret state generally represents a system vulnerability condition to be preserved. A suitable approach consists in opacity enforcement, mainly via supervisory control. While extensively studied under passive observation, opacity enforcement under active adversaries remains less explored, due to its inherent implementation complexity.**

This paper investigates current-state opacity for deterministic DES subject to actuator-enablement (AE) attacks, where an adversary may force disabled controllable events to occur. We propose a unified framework that combines supervisory opacity enforcement with a mitigation mechanism for defendable events, based on an observer-driven characterization of admissible control decisions.

The approach is preliminarily validated on a water distribution laboratory testbed.

I. Introduction

The evolution toward Industry 5.0 demands Cyber–Physical Systems (CPS) that are not only safe and efficient, but also secure and resilient. Many CPS can be modeled as Discrete Event Systems (DES) [1], for which Supervisory Control Theory (SCT) [2] provides a formal framework to regulate their behavior and enforce various properties.

Among the security properties studied for DES, opacity has emerged as a fundamental notion of confidentiality. Informally, opacity requires that an external passive observer cannot infer whether the system is in a secret state from their knowledge of the system and observable events [3]. Formal definitions and verification procedures for current-state opacity and related notions were established in [4], [5].

Modern CPS, however, are exposed to *active adversaries* capable of manipulating sensors and actuators [6]. In particular, actuator-enablement (AE) attacks, where disabled controllable events are maliciously forced, undermine supervisory authority and may compromise confidentiality, integrity, and availability, even when opacity holds nominally.

[1]J. A. da Silva Melo, T. Cruz, and P. Simões are with the Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal. joaomelo.iel@gmail.com, {tjcruz, psimoes}@dei.uc.pt

[2]V. Bonagura, F. Pascucci, and G. Cavone are with the Civil, Computer Science and Aeronautical Technologies Engineering of the University of Roma Tre, Rome, Italy {valeria.bonagura,federica.pascucci, graziana.cavone}@uniroma3.it

A. Related Work

Security in DES has been studied along three main directions: opacity analysis and enforcement, attack modeling and mitigation, and extensions toward adversarial confidentiality.

Opacity was introduced to formalize confidentiality under passive observation [7], [8], and several verification procedures for different opacity notions were later developed in [5]. Supervisory enforcement mechanisms were subsequently proposed to restrict system behavior in order to preserve secrecy [9], [10], under the assumption of passive observers.

Recent works have refined opacity notions and analysis techniques. In [11], current-state opacity is extended to state-output observation models. Strong variants of current-state opacity for time-labeled Petri nets are investigated in [12]. Stochastic and privacy-oriented approaches, such as almost k-step opacity enforcement via differential privacy, are explored in [13]. Parallel to opacity research, extensive work has addressed sensor and actuator attacks in CPS [6]. In the DES setting, actuator and sensor attack models and mitigation strategies have been developed in [14], [15], while attacker-centric formulations focus mainly on safety preservation and attack detection [16], [17]. More recently, actuator-enablement attacks have been studied from a resilience perspective. Robust recovery and supervisory control under actuator attacks are investigated in [18], while adversarial strategies aimed at violating opacity are constructed in [19]. Inferential properties such as detectability under active attacks are analyzed in [20]. However, these works do not integrate opacity enforcement with actuator-level mitigation.

Because opacity enforcement and attack-resilient control have largely evolved separately, a unified synthesis framework that guarantees current-state opacity under actuator-enablement attacks while preserving maximal permissiveness remains barely explored.

B. Paper Contribution

This paper bridges the gap between opacity enforcement and attack-resilient supervisory control by developing an opacity-preserving supervisory framework for deterministic DES under AE attacks. The approach combines offline supervisory synthesis with an online mitigation mechanism to preserve current-state opacity despite adversarial actuator manipulation. In particular, the main contributions of this paper can be summarized as follows:

- Use of the All-Inclusive Controller for Opacity (AIC - O) framework [21] to derive the maximal opacity-

preserving decision region under actuator-enablement attacks and adversarial loss of controllability.
- Definition of an online dynamic filtering mechanism that disables defendable vulnerable events only when their execution would violate opacity.
- Design of a supervisor that guarantees opacity while restricting behavior only when strictly necessary.

The approach is preliminarily validated on Hydra, i.e., a water distribution laboratory testbed [22].

II. Preliminaries

In this paper, a DES is modeled as a Deterministic Finite Automaton (DFA):

$$G = (X, A, \delta, x_0), \tag{1}$$

where X is the finite state set, A the event set, δ the transition function, and x_0 the initial state. The event set is partitioned as $A = A_o \cup A_{uo}$ (observable/unobservable) and $A = A_c \cup A_{uc}$ (controllable/uncontrollable). The generated language is

$$L(G) = \{w \in A^* \mid \delta(x_0, w) \text{ is defined}\}$$

where A^* denotes the set of all finite strings over A, including the empty string ε. The natural projection $P_o : A^* \to A_o^*$ is the mapping that removes all unobservable events from a string while preserving the order of observable ones.

Given an observation $t \in A_o^*$, the state estimate is

$$\hat{X}(t) = \{x \in X \mid \exists w \in L(G) : P_o(w) = t,\ \delta(x_0, w) = x\}.$$

A supervisor restricts G by disabling controllable events, while events in A_{uc} cannot be directly controlled.

Let $X_S \subseteq X$ be the secret-state set.

Definition 1 (Current-State Opacity)**.** *G is current-state opaque (CSO) w.r.t. X_S if* $\forall w \in L(G) : \quad \hat{X}(P_o(w)) \not\subseteq X_S$.

III. Problem Formulation

This section formalizes the closed-loop architecture under adversarial conditions and states the opacity-preserving control problem.

A. System Model

The plant is the deterministic finite automaton G introduced in Section II, and its behavior is described by $L(G)$. The event set is partitioned into controllable and uncontrollable events as in the Preliminaries.

The system operates under partial observation through the natural projection P_o, and current-state opacity is defined with respect to the secret set $X_S \subseteq X$ as in Definition 1.

A supervisor S maps each observation to a control decision that disables a subset of controllable events. The resulting nominal closed-loop behavior is denoted by S/G, with language $L(S/G) \subseteq L(G)$.

Under nominal conditions, opacity enforcement consists in restricting $L(G)$ so that the induced estimator never reveals the secret uniquely.

B. Attack Model

We consider an active adversary capable of performing actuator-enablement attacks.

Let $A_v \subseteq A_c$ denote the set of vulnerable controllable events. If an event $a \in A_v$ is disabled by the supervisor, the attacker may re-enable it, forcing its occurrence whenever it is physically defined in the plant.

Consequently, under AE attacks, events in A_v behave as *effectively uncontrollable* from the supervisor's perspective.

Formally, given a supervisor S, the attacked closed-loop language $L((S/G)^A)$ is defined as the set of strings $w \in L(G)$ such that, for every prefix u of w:

- if the next event σ after u belongs to A_{uc}, then σ must be enabled in G;
- if $\sigma \in A_c \setminus A_v$, then σ must be enabled by S after observation $P_o(u)$;
- if $\sigma \in A_v$, then σ may occur whenever it is defined in G, regardless of the supervisor's decision.

Thus, vulnerable events cannot be reliably disabled, and the supervisor loses effective authority over A_v.

The observation channel is assumed uncompromised.

C. Problem Statement

Given the plant G, the projection P_o, the secret set X_S, and the vulnerable event set A_v, the objective is to synthesize a supervisor S^* such that:

1) For all $w \in L((S^*/G)^A)$, the current-state estimate defined in Section II never becomes entirely contained in X_S.
2) The attacked closed-loop system admits infinite executions whenever the plant does.
3) Among all supervisors satisfying the above properties, S^* allows the largest attacked behavior, i.e., it is maximally permissive.

The key challenge lies in the fact that opacity must be preserved not only against passive inference, but also against adversarial loss of controllability. In particular, secrecy may be violated indirectly through the forced execution of vulnerable events, even if the nominal supervisor enforces opacity.

IV. Proposed Approach

The proposed approach integrates opacity enforcement and AE attack mitigation within a unified supervisory framework. The central objective is twofold: to guarantee CSO for a DFA-modeled DES even in the presence of AE attacks, and to restrict the plant behavior only to the extent strictly necessary to preserve confidentiality. In other words, the control strategy is designed to limit the behavior of the DES as little as possible while preserving the secret.

To achieve this objective, the methodology is structured into two complementary phases: an offline supervisory synthesis phase and an online mitigation phase. The offline phase computes the maximal set of control decisions that guarantee opacity under nominal conditions, while the online phase activates only when required, dynamically filtering defendable vulnerable events to preserve opacity under AE attacks.

A. Offline Phase: AIC-O Construction and Identification of the Safe Operating Region

The offline phase computes the All-Inclusive Controller for Opacity (AIC-O) [21], which represents all maximally permissive supervisory decisions preserving opacity under nominal conditions. The construction is based on a Bipartite Transition System (BTS) [21], whose nodes are divided into Y-states and Z-states. A Y-state is a decision state, i.e., an estimator information state where the supervisor selects a control decision $\gamma \in \Gamma(y)$, with $\gamma \subseteq A$ denoting the enabled events. Applying γ leads to a Z-state, which represents the corresponding post-decision information state, including possible unobservable evolutions.

The transition structure is described by the mappings

$$h_{YZ} : Y \times \Gamma(y) \to Z, \qquad h_{ZY} : Z \times A_o \to Y,$$

where h_{YZ} maps a decision state and a control decision to a post-decision state, while h_{ZY} maps a post-decision state and an observable event to the next decision state. The set H collects all transitions generated during the construction. By preserving the full admissible decision structure, the AIC-O avoids premature restrictions and keeps all opacity-preserving supervisory choices available for the subsequent resilience analysis.

The construction procedure is summarized in Algorithm 1. Starting from the initial decision state y_0, the algorithm explores admissible decisions $\gamma \in \Gamma(y)$, computes the corresponding Z-states, and then updates the decision state after each observable event $e \in A_o$. The output is the AIC-O structure $AIC - O(G) = (Y, Z, H)$.

Although the AIC-O contains all nominal opacity-preserving decisions, some of them may become unsafe under adversarial behavior. Therefore, the offline phase also computes the maximal safe decision region through the greatest fixed-point procedure in Algorithm 2. This procedure iteratively removes decision states that either reveal the secret with certainty or have no admissible continuation within the candidate safe set.

The resulting set X_{CS} is the maximal safe operating region. Since only decision states that inevitably lead to opacity violations are removed, the construction remains maximally permissive.

B. Online Phase: Mitigation under AE Attacks

We now consider actuator-enablement attacks. Let $A_v \subseteq A_c$ denote the set of vulnerable controllable events, and let $A_d \subseteq A_v$ denote the subset of defendable events.

Under AE attacks, events in $A_v \setminus A_d$ are effectively uncontrollable, since the adversary may re-enable them independently of the supervisor. Hence, the effective uncontrollable set becomes $A_{uc}^A = A_{uc} \cup (A_v \setminus A_d)$. Only events in A_d can be reliably disabled through the mitigation module.

The objective of the online phase is to enforce the invariance of the safe decision region X_{CS} computed offline. Given the current state estimate $\hat{X}$, the mitigation mechanism selects a control decision γ such that every event that may occur under AE semantics preserves the condition $\hat{X} \in X_{CS}$. The procedure, formalized in Algorithm 3, enables all events that are effectively uncontrollable under attack, and selectively enables defendable events only if their execution keeps the successor estimate within X_{CS}. This guarantees opacity preservation while maintaining maximal permissiveness.

Algorithm 1 Construction of the All-Inclusive Controller for Opacity (AIC-O)

Require: Plant $G = (X, A, \delta, x_0)$
Ensure: $AIC - O(G) = (Y, Z, H)$
Initialize $Y \leftarrow \{y_0\}$, $Z \leftarrow \emptyset$, $H \leftarrow \emptyset$
Initialize stack $\mathcal{S} \leftarrow \{y_0\}$
while $\mathcal{S} \neq \emptyset$ **do**
  Extract y from $\mathcal{S}$
  for each admissible control decision $\gamma \in \Gamma(y)$ **do**
    $z \leftarrow h_{YZ}(y, \gamma)$
    $H \leftarrow H \cup \{(y, \gamma, z)\}$
    if $z \notin Z$ **then**
      $Z \leftarrow Z \cup \{z\}$
      for each $e \in \gamma \cap A_o$ **do**
        $y' \leftarrow h_{ZY}(z, e)$
        $H \leftarrow H \cup \{(z, e, y')\}$
        if $y' \notin Y$ **then**
          $Y \leftarrow Y \cup \{y'\}$
          $\mathcal{S} \leftarrow \mathcal{S} \cup \{y'\}$
        end if
      end for
    end if
  end for
end while
return (Y, Z, H)

Algorithm 2 Greatest Fixed-Point Computation of Safe Decision States

Require: (Y, Z, H)
Ensure: Maximal safe decision set X_{CS}
$Y^{(0)} \leftarrow Y$
$k \leftarrow 0$
repeat
  $Y^{(k+1)} \leftarrow \left\{ y \in Y^{(k)} \;\middle|\; y \nsubseteq X_S \;\wedge\; \exists \gamma \in \Gamma(y) \text{ s.t. } h_{YZ}(y, \gamma) \in Z \text{ and } h_{ZY}(h_{YZ}(y, \gamma), e) \in Y^{(k)} \right\}$
  $k \leftarrow k + 1$
until $Y^{(k)} = Y^{(k-1)}$
return $X_{CS} \leftarrow Y^{(k)}$

C. Correctness and Minimal Restriction of Behavior

The following result establishes the correctness of the proposed framework under AE attacks and its maximal permissiveness.

Lemma 1. *Consider the plant G, the secret set X_S, the vulnerable event set A_v, and the defendable event set $A_d \subseteq A_v$. Assume that at least one admissible control decision can prevent inevitable opacity violation under AE semantics.*

Then, the supervisory strategy obtained by Algorithms 1, 2, and 3 enforces current-state opacity for all strings in $L((S/G)^A)$, keeps the safe decision region X_{CS} invariant under AE attacks, and is maximally permissive among all opacity-preserving supervisors under such attacks.

Algorithm 3 Online Mitigation under AE Attacks

Require: Current estimate $\hat{X}$, safe region X_{CS}, event sets A_{uc}, A_v, A_d
Ensure: Control decision γ
1: $\gamma \leftarrow A_{uc} \cup (A_v \setminus A_d)$
2: **for** each $\sigma \in A_d$ **do**
3: Compute $\hat{X}_\sigma = \hat{X}(\sigma)$
4: **if** $\hat{X}_\sigma \in X_{CS}$ **then**
5: $\gamma \leftarrow \gamma \cup \{\sigma\}$
6: **end if**
7: **end for**
8: **return** γ

Proof: Let (Y, Z, H) be the AIC-O structure computed by Algorithm 1. By construction, it contains all control decisions that preserve current-state opacity under nominal supervision. The proof follows from opacity preservation, invariance under AE semantics, nonblocking behavior, and maximal permissiveness.

Opacity preservation. Algorithm 2 computes $X_{CS} \subseteq Y$ as the greatest fixed point obtained by removing decision states that either reveal the secret with certainty or have no admissible continuation. Thus, every $y \in X_{CS}$ satisfies $y \nsubseteq X_S$ and admits a successor remaining in X_{CS}. Therefore, any trajectory whose estimator remains in X_{CS} preserves current-state opacity.

Invariance under AE attacks. Under AE attacks, events in $A_v \setminus A_d$ are effectively uncontrollable. Algorithm 3 always includes the events in $A_{uc} \cup (A_v \setminus A_d)$, while a defendable event $\sigma \in A_d$ is enabled only if the successor estimate $\hat{X}_\sigma$ remains in X_{CS}. Hence, every event that may occur under AE semantics either is unavoidable and accounted for in the fixed-point computation, or is filtered by the mitigation mechanism. Consequently, the attacked closed-loop evolution remains in X_{CS}, and opacity holds for all $w \in L((S/G)^A)$.

Nonblocking behavior. By hypothesis, at least one admissible decision prevents unavoidable disclosure under AE attacks. Since Algorithm 2 removes decision states without admissible continuations, each retained state admits a successor in X_{CS}. The resulting closed-loop system therefore admits continuing executions within the safe region.

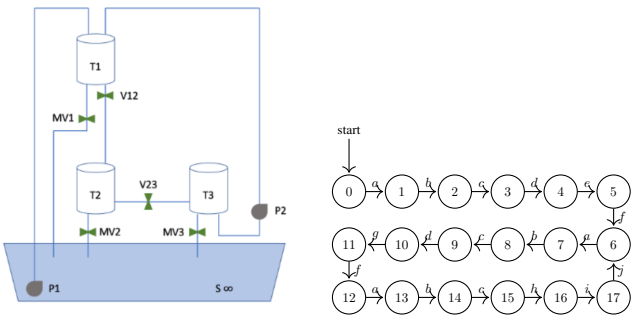
Maximal permissiveness. Assume, by contradiction, that another supervisor S' preserves opacity under AE attacks while allowing strictly larger behavior. Then S' must enable either a control decision not represented in the AIC-O, contradicting its all-inclusive property, or a transition leaving X_{CS}, contradicting the greatest fixed-point characterization of the safe region. Hence, no strictly more permissive supervisor can preserve current-state opacity under AE attacks. □

V. Case Study

To evaluate the proposed opacity-preserving supervisory framework in a realistic cyber–physical scenario, we consider a simplified abstraction of the Hydra testbed [22], a water distribution platform for cybersecurity experimentation.

Event	Actuators	Tanks	Obs.
a	$P_1 : 0 \to 1$	–	N
b	–	$T_1 : 0/10 \to 100$	Y
c	$P_1 : 1 \to 0$	–	N
d	$V_1 : 0 \to 1$	–	N
e	–	$T_1 : 100 \to 10,\ \ T_2 : 0 \to 45$	Y
f	$V_1 : 1 \to 0$	–	N
g	–	$T_1 : 100 \to 10,\ \ T_2 : 45 \to 90$	Y
h	$P_2 : 0 \to 1$	–	N
i	–	$T_2 : 90 \to 45$	N
j	$P_2 : 1 \to 0$	–	N

TABLE I: Event summary from the state-variable table. Events due only to tank filling/level evolution (T_1, T_2) are modeled as unobservable.



(a) Simplified physical configuration of the Hydra testbed.

(b) DES model of the simplified Hydra system.

Fig. 1: Physical system and corresponding DFA abstraction of the simplified Hydra case study.

As shown in Fig. 1a, the system consists of an infinite reservoir S and three interconnected tanks. Tank T_1 is filled by pumps P_1 and P_2 and discharges into T_2 by gravity through valve V_{12}. Since valve V_{23} is assumed permanently open, tanks T_2 and T_3 are modeled as a single equivalent tank, denoted by T_2, with doubled capacity. Level and pressure sensors provide the measurements used by the supervisory controller.

Under these assumptions, the system is abstracted as two interconnected tanks, T_1 and T_2, connected by a controllable valve V_1, where T_2 has twice the capacity of T_1. Fig. 1b shows the DFA model, while Table II reports the quantitative state description. Tank levels are expressed as capacity percentages, and pump/valve variables are Boolean, with 1 denoting active/open and 0 inactive/closed.

The plant combines passive hydraulic dynamics, due to gravity-driven flows, with active pump-assisted transfers. It is modeled as the deterministic finite automaton $G = (X, A, \delta, x_0)$, where $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$, $x_0 = \{0\}$, and $A = \{a, b, c, d, e, f, g, h, i, j\}$. Events represent either actuator commands, such as pump and valve switching, or discrete level changes associated with

State	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
T_1	0%	0%	100%	100%	100%	10%	10%	10%	100%	100%	100%	10%	10%	10%	100%	100%	100%	10%
T_2	0%	0%	0%	0%	0%	45	45%	45%	45%	45%	45%	90%	90%	90%	90%	90%	90%	45%
V_1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0
P_1	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0
P_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

TABLE II: State-variable configuration table.

threshold crossings. Table I summarizes the event set, while Table II maps each symbolic state to the corresponding physical configuration.

The controllable and observable event sets are $A_c = \{a, c, d, f, h, j\}$, $A_o = \{b, e, g, i\}$. Controllable events correspond to supervisory commands, whereas observable events represent externally detectable configuration changes. As shown in Fig. 1b, the plant follows a cyclic logic in which P_1 fills T_1, V_1 transfers water to T_2, and P_2 redistributes water to sustain periodic operation.

The secret-state set is defined as $X_S = \{5, 11\}$. These states correspond to critical depletion configurations where T_1 reaches its minimum level, 10%, while valve V_1 remains open. Revealing with certainty that the system is in either state would disclose sensitive information about internal resource levels and the current operating phase. The confidentiality requirement is therefore formalized as current-state opacity with respect to X_S.

Under nominal conditions, current-state opacity is enforced through Algorithms 1 and 2. The resulting safe estimator region X_{CS} contains all observer states that do not reveal the secret uniquely. For the Hydra case study, X_{CS} corresponds to the decision states highlighted in blue in Fig. 2.

A. Active Attacker Scenario

We consider an active attacker capable of re-enabling vulnerable controllable events disabled by the supervisor. In the Hydra case study, the vulnerable and defendable event sets are $A_v = \{a, c, d\}$, $A_d = \{d\}$. Thus, events a and c become effectively uncontrollable under AE attacks, whereas event d can still be physically blocked by the mitigation mechanism. The online mitigation module disables d only when its execution would drive the current estimator outside the safe region X_{CS}.

The effect of this strategy is illustrated in Fig. 2. Rectangular nodes denote Y-states, while oval nodes denote Z-states. Blue nodes remain reachable under the combined supervisor–mitigation architecture, whereas grey nodes become unreachable. The red states $\{5\}$ and $\{11\}$ correspond to disclosure configurations associated with the secret set X_S. By filtering defendable transitions leading to such configurations, the mitigation mechanism removes uniquely revealing states from the reachable region.

The remaining subgraph represents the maximal invariant opacity-preserving region contained in X_{CS}. Although the admissible behavior is reduced with respect to nominal operation, nonblocking cyclic behavior is preserved, showing that confidentiality can be maintained without forcing the plant into a degenerate configuration.

Indicator	Value
Plant states/events/transitions	18/10/18
Secret states	2
Vulnerable/defendable events	3/1
Vulnerability coverage ρ_D	1/3
Effective uncontrollability ρ_U^A	0.6
Reachable secret states after mitigation	0
Secret exposure ρ_S	0
Nonblocking cyclic behavior	Preserved

TABLE III: Quantitative indicators for the Hydra case study under AE attacks.

To complement this structural analysis, the following subsection introduces quantitative indicators that evaluate the impact of opacity enforcement and AE-attack mitigation on the admissible behavior of the Hydra model.

B. Quantitative Evaluation

The quantitative impact of the proposed mitigation strategy is assessed through three indicators: the secret-exposure ratio ρ_S, the vulnerability coverage ratio ρ_D, and the effective-uncontrollability ratio ρ_U^A:

$$\rho_S = \frac{|X_{\text{reach}} \cap X_S|}{|X_S|},\ \rho_D = \frac{|A_d|}{|A_v|},\ \rho_U^A = \frac{|A_{uc} \cup (A_v \setminus A_d)|}{|A|}. \tag{2}$$

Here, X_{reach} is the set of plant states reachable in the attacked closed loop. The indicators respectively measure residual secret exposure, the fraction of vulnerable events that can be physically blocked, and the loss of supervisory authority under AE attacks.

For the Hydra case study, $|X| = 18$, $|A| = 10$, $|X_S| = 2$, $A_v = \{a, c, d\}$, and $A_d = \{d\}$. Since $A_{uc} = \{b, e, g, i\}$, the effective uncontrollable set becomes $A_{uc}^A = \{a, b, c, e, g, i\}$. Thus, $\rho_D = 1/3$ and $\rho_U^A = 0.6$.

As reported in Table III, AE attacks make 60% of the event set effectively uncontrollable. Nevertheless, the proposed mitigation yields $\rho_S = 0$, preventing the reachability of uniquely revealing secret configurations while preserving nonblocking cyclic behavior.

VI. Conclusion

This paper presented a supervisory framework for enforcing current-state opacity in discrete-event cyber–physical systems under both passive observation and actuator-enablement attacks. The approach combines the AIC-O construction, a greatest fixed-point computation, and an online mitigation

mechanism that selectively disables defendable vulnerable events.

The methodology was validated on a simplified Hydra water distribution model, showing that opacity can be preserved under nominal conditions and maintained under AE attacks through targeted mitigation. The quantitative indicators show that uniquely revealing secret configurations are avoided while nonblocking cyclic behavior is preserved.

The results confirm that integrating opacity-enforcing supervision with attack-aware mitigation is relevant for resilient DES-based CPS. The main limitation is the offline computational cost of the AIC-O and fixed-point computations, which may suffer from state-space explosion under partial observation. Thus, direct application to large-scale CPS may require modular, compositional, or symbolic implementations. The Hydra validation is illustrative, and larger case studies are needed to further assess scalability and implementation constraints. Future work will address scalable synthesis methods, reduced conservativeness, and validation on larger DES models and additional attack scenarios.

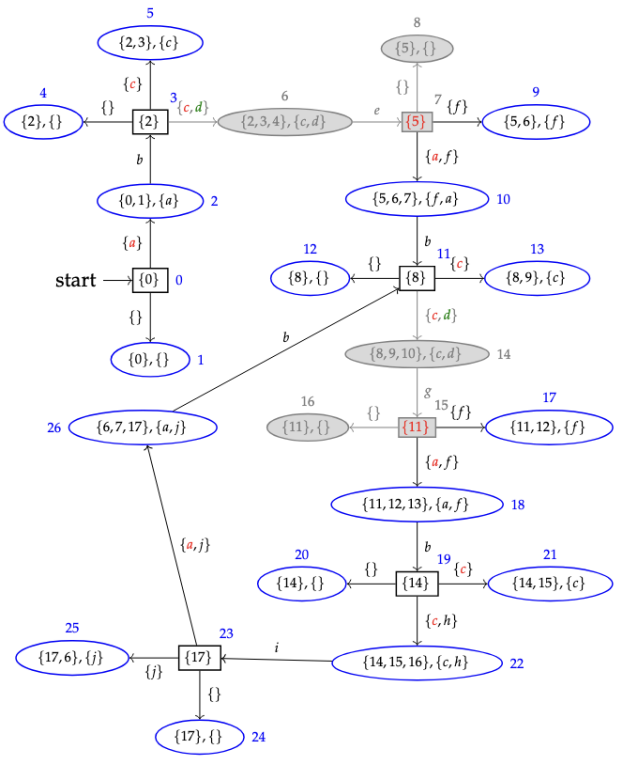


Fig. 2: AIC-O under actuator-enablement attack: unsafe estimator states are removed from the reachable region.

References

[1] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 01 2010, p. 800.

[2] P. Ramadge and W. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[3] S. Lafortune, F. Lin, and C. N. Hadjicostis, "On the history of diagnosability and opacity in discrete event systems," *Annual Reviews in Control*, vol. 45, pp. 257–266, 2018.

[4] A. Saboori and C. N. Hadjicostis, "Verification of k-step opacity and analysis of its complexity," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 549–559, 2011.

[5] ——, "Verification of initial-state opacity in security applications of discrete event systems," *Information Sciences*, vol. 246, pp. 115–132, 2013.

[6] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[7] L. Mazaré, "Using unification for opacity properties," *Proceedings of the 4th IFIP WG1*, vol. 7, pp. 165–176, 2004.

[8] E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau, "Concurrent secrets," *Discrete Event Dynamic Systems*, vol. 17, pp. 425–446, 2007.

[9] Y. Zhou, Z. Chen, and Z. Liu, "Verification and enforcement of current-state opacity based on a state space approach," *European Journal of Control*, vol. 71, p. 100795, 2023.

[10] Y. Tong, K. Cai, and A. Giua, "Decentralized opacity enforcement in discrete event systems using supervisory control," in *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2018, pp. 1053–1058.

[11] P. C. Mayer, F. G. Cabral, P. M. Lima, and M. V. Moreira, "Current-state opacity based on state outputs," *IFAC-PapersOnLine*, vol. 58, no. 1, pp. 19–23, 2024.

[12] T. Qin, L. Yin, G. Liu, N. Wu, and Z. Li, "Strong current-state opacity verification of discrete-event systems modeled with time labeled petri nets," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 1, pp. 54–68, 2025.

[13] R. Zhao, M. Uzam, and Z. Li, "Almost k-step opacity enforcement in stochastic discrete-event systems via differential privacy," *Mathematics*, vol. 13, no. 8, p. 1255, 2025.

[14] S. Zheng, S. Shu, and F. Lin, "Modeling and control of discrete event systems under joint sensor-actuator cyber attacks," *IEEE Transactions on Control of Network Systems*, 2023.

[15] L. K. Carvalho, Y.-C. Wu, R. Kwong, and S. Lafortune, "Detection and mitigation of classes of attacks in supervisory control systems," *Automatica*, vol. 97, pp. 121–133, 2018.

[16] J. Yao, S. Li, and X. Yin, "Sensor deception attacks against security in supervisory control systems," *Automatica*, vol. 159, 2024.

[17] R. Fritz and P. Zhang, "Detection and localization of stealthy cyber attacks in cyber-physical discrete event systems," *IEEE Transactions on Automatic Control*, 2023.

[18] S. Oliveira, M. T. Anbarani, G. Beal, I. Kovalenko, M. Teixeira, A. B. Leal, and R. Meira-Góes, "Robust recovery and control of cyber-physical discrete event systems under actuator attacks," pp. 1220–1226, 2025.

[19] X. Li and C. N. Hadjicostis, "Synthesis of state-attack strategies for anonymity and opacity violation in discrete event systems," *arXiv preprint arXiv:2510.22657*, 2025.

[20] K. Ritsuka, F. Lin, S. Lafortune, and C. Wang, "Detectability of discrete event systems under sensor attacks," *Journal of Systems Science and Complexity*, vol. 38, no. 1, pp. 150–177, 2025.

[21] X. Yin and S. Lafortune, "A new approach for synthesizing opacity-enforcing supervisors for partially-observed discrete-event systems," in *2015 American Control Conference (ACC)*, 2015, pp. 377–383.

[22] F. Battisti, G. Bernieri, M. Carli, M. Lopardo, and F. Pascucci, "Detecting integrity attacks in iot-based cyber physical systems: a case study on hydra testbed," in *2018 Global Internet of Things Summit (GIoTS)*, 2018, pp. 1–6.