Enforcing Opacity in Discrete Event Systems via Delayed Observations

Jiwei Wang, Simone Baldi, *Senior Member, IEEE*, Wenwu Yu, *Senior Member, IEEE*, and Xiang Yin, *Member, IEEE*

***Abstract*—Artificially introducing a delay in the observations of a system can be an effective mechanism to mask the system itself, with the goal to increase its opacity and thus its security. This letter investigates opacity in discrete event systems with delayed observations. We focus on two questions: how to verify opacity under delayed observations, and how to synthesize sensor activation policies that guarantee opacity under such delayed conditions. To address these questions, we first introduce the definition of opacity under delayed observation and develop a corresponding verification method. We then extend such analysis tool into a synthesis tool by proposing an optimization approach for designing sensor activation policies guaranteeing opacity under delayed observations. An example is used to illustrate the analysis and synthesis procedures.**

***Index Terms*—Discrete event systems, opacity, delayed observations, sensor activation, optimal policy.**

I. Introduction

MAINTAINING security and privacy of a system is important in many scenarios. In the context of partially-observed discrete event systems (DESs), opacity has been proposed to guarantee that some secret state of the system remains hidden to inference from outside [1]. A system is opaque if, for every execution that reaches a secret state, there exists another execution with an identical set of observable events that does not reach a secret state: this way, it is not possible to unambiguously determine if the system is in a secret state. Opacity of DESs has received significant attention [2], with potential applications spanning several domains [3]. Despite the various mechanisms to enforce opacity [4], [5], [6], [7], [8], [9], [10], none of these exploits the fact that artificially introducing delayed observations (e.g., on a system that originally is not opaque) can help to make the system opaque. In these contexts, a defender may artificially introduce a delay in the sensors of the system to safeguard some secret states to an eavesdropper.

This letter aims to cover this gap: we consider opacity from the perspective of a defender who can artificially introduce a finite delay K in the observations so as to mask the system to an eavesdropper. We investigate two problems: i) Analysis: how to verify opacity when a delay is introduced in its observations? ii) Synthesis: how to synthesize (and possibly optimize) a sensor activation policy that guarantees opacity? To analyze the effect of delayed observations, we extend the framework in [11], which is based on recording the delay of key events that help the verification of system properties [12]. To synthesize an opaque system, we extend the dynamic mask synthesis in [13], also referred to as dynamic sensor activation, and the framework of most permissive observer in [14]. The observation is dynamic when the sensors can be dynamically activated or not depending on the current observation [15], [16].

Given the above analysis and synthesis problems, this letter provides the following contributions. We address the analysis problem by introducing a definition of opacity under delayed observations, named K-delayed opacity. Building on this definition, we propose a delay observer structure to capture opacity under dynamic and delayed observations, and we develop a corresponding verification method. We address the synthesis problem by proposing a new container structure tailored to delayed observations. This container structure encompasses all sensor activation policies that guarantee K-delayed opacity, enabling the selection of an optimal policy according to a criterion defined in this letter. Let us mention that the proposed K-delayed opacity should not be confused with K-step opacity [1], a notion that cannot address delayed observations.

The remainder of this letter is organized as follows. Section II introduces partially-observed DESs under dynamic observation. Section III presents the definition of K-delayed

Received 28 July 2025; revised 22 September 2025; accepted 12 October 2025. Date of publication 20 October 2025; date of current version 27 October 2025. This work was supported in part by the Jiangsu Provincial Scientific Research Center of Applied Mathematics under Grant BK20233002, and in part by the National Natural Science Foundation of China under Grant 62150610499, Grant 62073074, Grant 62073076, Grant 62233004, and Grant 62173226. Recommended by Senior Editor A. P. Aguiar. *(Corresponding authors: Simone Baldi; Wenwu Yu.)*

Jiwei Wang is with the School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: jwwang@seu.edu.cn).

Simone Baldi is with the School of Mathematics, Southeast University, Nanjing 210096, China (e-mail: simonebaldi@seu.edu.cn).

Wenwu Yu is with the School of Mathematics, Southeast University, Nanjing 210096, China, and also with the School of Automation and Electrical Engineering, Linyi University, Linyi 276005, China (e-mail: wwyu@seu.edu.cn).

Xiang Yin is with the School of Automation and Sensing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: yinxiang@sjtu.edu.cn).

Digital Object Identifier 10.1109/LCSYS.2025.3623658

opacity and the delay observer for verification. Section IV addresses the synthesis problem. Section V concludes this letter.

II. Preliminaries

We start by recalling the basic formalism of automata. The finite event set E in DESs is modeled as an alphabet, so that a finite sequence of events in E can be described as the *concatenation* of a string of words in the alphabet. A *language* is a set of event strings formed from the alphabet E. Let E^* denote the set of all finite strings over E, also known as Kleene or star closure [17]. For any string $s \in E^*$, the length of s is denoted by $|s|$. The empty string is denoted by ϵ, with $|\epsilon| = 0$. The *prefix-closure* of a language L is defined as $\overline{L} = \{s \in E^* | \exists t \in E^*, \text{ s.t. } st \in L\}$. A language L is said to be *prefix-closed* if $L = \overline{L}$. We assume the language L to be *live*: $\forall s \in L, \exists e \in E$, s.t. $se \in L$. In other words, any string in L can be extended to arbitrary length.

Automata are a fundamental framework for manipulating languages. Consider a finite automaton defined as

$$G = (X, E, \alpha, X_0), \tag{1}$$

where X is the set of N finite system states, E is the set of finite events, $\alpha : X \times E^* \to 2^X$ is the transition function that describes the transition of an event string, and $X_0 \subseteq X$ is the set of initial states. The language generated by G is denoted by $\mathcal{L}(G) = \{s \in E^* | \alpha(x_0, s)! \ \exists x \in X_0\}$, where ! means that the function is defined. For any $s \in \mathcal{L}(G)$, let us denote for brevity $\alpha(s) := \{\alpha(x, s) \mid x \in X_0\}$. The *accessible part* of G is the set of states $\{x \in X \mid \exists s \in \mathcal{L}(G), \text{ s.t.} \, x \in \alpha(s)\}$.

A partially-observed DES is said to have *dynamic observation* when the observability of the events can be controlled by a *sensor activation policy* that depends on the current observation. We divide the event set E into the observable events E_o and the unobservable events E_{uo}. A sensor activation policy is defined by the pair $\Omega = (R, \Theta)$, where $R = (Q, E, \omega, q_0)$ is a deterministic automaton that satisfies $\mathcal{L}(R) = E^*$, with Q the set of policy states and $\omega : Q \times E^* \to Q$ the sensing transition function. For each $q \in Q$, $\Theta(q) \in 2^{E_o}$ specifies the sensing decision, which determines the set of events monitored at q. To ensure that Ω is feasible, the following feasibility condition [18] must be satisfied:

$$\forall q, q' \in Q, e \in E : \omega(q, e) = q', \left[q \neq q' \Rightarrow e \in \Theta(q)\right], \tag{2}$$

representing that sensing transitions in R occur only upon the observation of events. This aligns with practical situations in which unobservable events cannot trigger sensing transitions.

Let us introduce a projection operator to obtain the observation of an event string: $\forall se \in E^*$,

$$P_\Omega(\epsilon) = \epsilon, \ \ P_\Omega(se) = \begin{cases} P_\Omega(s)e, & \text{if } e \in \Theta(\omega(s)), \\ P_\Omega(s), & \text{if } e \notin \Theta(\omega(s)). \end{cases} \tag{3}$$

The projection operator P_Ω can also be applied to a language, that is, $\forall L \subseteq E^*$, $P_\Omega(L) = \{s \in E_o^* | \exists s' \in L, \text{ s.t. } s = P_\Omega(s')\}$. In other words, for any system trajectory $s \in \mathcal{L}(G)$, $\Theta(\omega(s))$ and $P_\Omega(s)$ return the current sensing decision and the observation under Ω, respectively. Using P_Ω, the feasibility condition (2) can also be stated as

$$\forall s, s' \in \mathcal{L}(G), \ P_\Omega(s) = P_\Omega\big(s'\big) \Rightarrow \omega(s) = \omega\big(s'\big), \tag{4}$$

implying that when two system trajectories are indistinguishable, they must result in the same sensing decision.

Given P_Ω, we consider the operation $\zeta_\Omega^n : \mathcal{L}(G) \to \mathcal{L}(G)$ to handle delayed information: $\forall s \in \mathcal{L}(G)$,

$$\zeta_\Omega^n(s) = \left\{s'' \in \bar{s} \mid \exists s' \in \bar{s} : |s'| \geq |s| - n, \text{ s.t.} P_\Omega\big(s''\big) = P_\Omega\big(s'\big)\right\}, \tag{5}$$

where $n \in \mathbb{N}$ is the number of delay steps. Intuitively, $\{P_\Omega(s') \mid s' \in \bar{s} \wedge |s'| \geq |s| - n\}$ represents the set of possible observations that eavesdroppers may receive with delay after the occurrence of s, and $\zeta_\Omega^n(s)$ collects all prefixes of s that yield the same observation under Ω as an element in this set.

Example 1: Throughout this letter, we consider the automaton G in Fig. 1(a) describing a simplified location-detection problem. The state set $X = \{0, 1, 2, 3\}$ corresponds to $N = 4$ locations, while events e_1 and e_2 denote detection signals received from sensor 1 and sensor 2, respectively, with $E = E_o = \{e_1, e_2\}$. Consider the sensor activation policy Ω shown in Fig. 1(b), describing that sensor 2 is kept continuously active. Given the string e_1e_2, we have $\Theta(0) = \{e_2\}$, $P_\Omega(e_1e_2) = e_2$, $\zeta_\Omega^0(e_1e_2) = \{e_1e_2\}$ and $\zeta_\Omega^1(e_1e_2) = \{\epsilon, e_1, e_1e_2\}$. ■

III. Opacity With Delayed Observations

In this section, we propose an opacity concept called K-delayed opacity, which incorporates artificially introduced delays as a mechanism for protecting the secret states.

A. K-Delayed Opacity

Let us first recall the notion of opacity.

Definition 1 (Opacity [1]): Given a system model G in (1) with a set of secret states $X_S \subseteq X$ and a sensor activation policy Ω, the live language $\mathcal{L}(G)$ is opaque w.r.t. X_S and P_Ω if $\forall s \in \mathcal{L}(G) : \alpha(s) \subseteq X_S$,

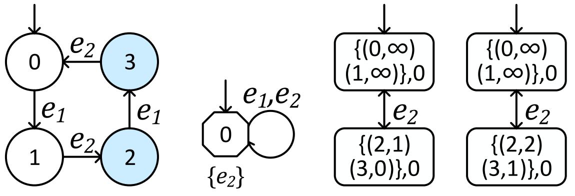
$$\exists s' \in \mathcal{L}(G) : P_\Omega\big(s'\big) = P_\Omega(s), \ \text{ s.t. } \alpha\big(s'\big) \not\subseteq X_S. \tag{6}$$

In other words, an opaque system is such that for any event string that leads to a secret state, there exists another string with the same observation leading to a non-secret state.

Example 2: Let the set of secret states of G in Fig. 1(a) be $X_S = \{2, 3\}$, representing that the defender wishes to protect the information of the current location being in states 2 or 3. Let the sensor activation policy Ω be as in Fig. 1(b). Since $\alpha(e_1e_2) = \{2\} \subseteq X_S, \alpha(e_1e_2e_1) = \{3\} \subseteq X_S$ and $\{s' \in \mathcal{L}(G) \mid P_\Omega(s') = P_\Omega(e_1e_2)\} = \{e_1e_2, e_1e_2e_1\}$, we have that $\mathcal{L}(G)$ is not opaque w.r.t. X_S and P_Ω. Note that the transition $(1, e_2, 2)$ is key to an eavesdropper to know that the system will be in secret states until the next e_2 is observed. ■

A system lacking opacity may become opaque when introducing an observation delay, motivating the following definition.

Definition 2 (K-Delayed Opacity): Given a system model G in (1) with a set of secret states $X_S \subseteq X$ and a sensor activation



(a) G (b) Ω (c) $\mathcal{O}^{K_1}_\Omega(G)$ (d) $\mathcal{O}^{K_2}_\Omega(G)$

Fig. 1. (a) System model G, with secret states $X_S = \{2, 3\}$ marked in blue; (b) Sensor activation policy Ω, maintaining sensor 2 continuously active; (c)-(d) Delay observers $\mathcal{O}^{K_1}_\Omega(G)$ and $\mathcal{O}^{K_2}_\Omega(G)$ with delay $K_1 = 1$ and $K_2 = 2$.

policy Ω, the live language $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and P_Ω if $\forall s \in \mathcal{L}(G) : |s| \geq K \wedge \alpha(s) \subseteq X_S$,

$$\exists s' \in \mathcal{L}(G) : P_\Omega\big(s'\big) \in P_\Omega\big(\zeta^K_\Omega(s)\big),\ \text{s.t.}\, \alpha\big(s'\big) \nsubseteq X_S. \tag{7}$$

If (7) holds, the string s cannot be distinguished from another string s' that does not lead to secret states under delayed observations. Thus, none of the events required to unambiguously determine if the system is in a secret state can be received timely by the eavesdropper.

As K increases, the strings that cannot be distinguished from the string $s : \alpha(s) \subseteq X_S$ become more. It can be verified from (7) that K'-delayed opacity $\Rightarrow$ K-delayed opacity when $K \geq K'$, i.e., larger delay increases opacity. Clearly, opacity implies K-delayed opacity

B. Delay Observer

Similar to opacity, verifying K-delayed opacity cannot be accomplished by examining all possible string of events of a system. It is essential to incorporate delay information in a verification method. We incorporate delay information via a delay observer: we stress on the fact that existing delay observers based on recording the delay of at most one observable event [12] cannot be applied to the scenario under consideration, where multiple delays within an observation may need to be recorded. Such difference requires to extend the delay observer structure.

The following deterministic observer structure is proposed

$$\mathcal{O}^K_\Omega(G) = (Y, E_o, \beta, y_0), \tag{8}$$

where $Y \in 2^{X\times(\mathbb{Z}\cup\{\infty\})} \times Q$ is the observer state space, with $\mathbb{Z}$ the set of integers. For notational convenience, let us decompose each $y \in Y$ as $y = (I(y), D(y))$. Namely, $I(y) \in 2^{X\times(\mathbb{Z}\cup\{\infty\})}$ represents the state estimation set: for any $(x, u) \in I(y)$, the system state x is assigned with a delay value u counting the maximum number of steps before receiving the key events (i.e., the observable events allowing to distinguish the states in X_S from other states). Meanwhile, $D(y) \in Q$ specifies the current sensing decision $\Theta(D(y)) \in 2^{E_o}$. For a state set $\iota \subseteq X$ and a sensing decision $\theta \subseteq E_o$, denote $\phi(\iota, \theta) = \{x \in X \mid \exists x' \in \iota, s \in (E \setminus \theta)^*, \text{s.t.}\, \alpha(x', s) = x\}$, representing the set of states that can be reached unobservably from a state in ι under the sensing decision θ. The transition function $\beta : Y \times E_o \to Y$ and the initial state y_0 in (8) are constructed as in Algorithm 1. Two recording procedures are proposed to handle the transitions involving unobservable and observable events, respectively. Specifically, REC-UNOBS(x, u, y) appends a delay value to each system state, storing the result in $I(y)$. Meanwhile, REC-OBS(y) identifies new transitions and observer states, where the delay values are appended to the next initial system states.

Algorithm 1: Construction of the Delay Observer $\mathcal{O}^K_\Omega(G)$

```
Input: $G = (X, E, \alpha, X_0)$, $R = (Q, E, \omega, q_0)$, $\Theta$, $K$, $E_o$, $X_S$;
Output: $\mathcal{O}^K_\Omega(G) = (Y, E_o, \beta, y_0)$;
$y_0 \leftarrow (\emptyset, q_0)$; $u_0 \leftarrow \infty$;
if $\phi(X_0, E_o) \subseteq X_S$ then
    $u_0 \leftarrow K$;
for $x_0 \in X_0$ do
    $I(y_0) \leftarrow I(y_0) \cup (x_0, u_0)$; REC-UNOBS$(x_0, u_0, y_0)$;
$Y \leftarrow \{y_0\}$; REC-OBS$(y_0)$;
Return $\mathcal{O}^K_\Omega(G)$;
Procedure REC-UNOBS$(x, u, y)$
for $e \in E \setminus \Theta(D(y)) : \alpha(x, e)!$ do
    for $x' \in \alpha(x, e)$ do
        if $0 < u < \infty$ then
            $u' \leftarrow u - 1$;
        else
            $u' \leftarrow u$;
        if $(x', u') \notin I(y)$ then
            $I(y) \leftarrow I(y) \cup \{(x', u')\}$;
            REC-UNOBS$(x', u', y)$;
Procedure REC-OBS$(y)$
$\iota \leftarrow \{x \in X \mid (x, u) \in I(y)\}$; $E^{\text{Rec}} \leftarrow \emptyset$;
for $e \in \Theta(D(y))$ do
    $\iota^* \leftarrow \{x \in X \mid \exists x' \in \iota, \text{s.t.}\, x \in \alpha(x', e)\}$;
    if $\phi(\iota^*, \Theta(\omega(D(y), e))) \subseteq X_S$ then
        $E^{\text{Rec}} \leftarrow E^{\text{Rec}} \cup \{e\}$;
    $y^e \leftarrow (\emptyset, \omega(D(y), e))$;
for $(x, u) \in I(y)$ do
    for $e \in \Theta(D(y)) : \alpha(x, e)!$ do
        if $u = \infty \wedge e \in E^{\text{Rec}}$ then
            $u' \leftarrow K$;
        else if $e \notin E^{\text{Rec}}$ then
            $u' \leftarrow \infty$;
        else
            $u' \leftarrow u - 1$;
        for $x' \in \alpha(x, e)$ do
            if $(x', u') \notin I(y^e)$ then
                $I(y^e) \leftarrow I(y^e) \cup \{(x', u')\}$;
                REC-UNOBS$(x', u', y^e)$;
for $e \in \Theta(D(y))$ do
    Add $\beta(y, e) = y^e$ to $\mathcal{O}_\Omega(G)$;
    if $y^e \notin Y$ then
        $Y \leftarrow Y \cup \{y^e\}$; REC-OBS$(y^e)$;
```

Note that for any $(x, u) \in I(y)$, there always exists $s \in \mathcal{L}(G) : x \in \alpha(s)$ such that $y = \beta(P_\Omega(s))$. If there exists another state $(x', u') \in I(y)$, then there exists $s' \in \mathcal{L}(G) : x' \in \alpha(s')$ such that $P_\Omega(s) = P_\Omega(s')$. For each $s \in \mathcal{L}(G) : \alpha(s) \subseteq X_S$, the delay observer $\mathcal{O}^K_\Omega(G)$ records the delay of the events in E_o that help to distinguish s from the system trajectories $s' \in \mathcal{L}(G) : \alpha(s') \nsubseteq X_S$. A delay value "0" indicates that the observation allowing to distinguish the states in X_S from other states has been received. In line with existing observers [17], also the observer in Algorithm 1 has worst-case complexity $O(2^{2^N})$, where N is the number of system states.

C. Verification of K-Delayed Opacity

For the delay observer $\mathcal{O}^K_\Omega(G) = (Y, E_o, \beta, y_0)$, define the verification function $\psi : Y \to \{O, T\}$ as follows: $\forall y \in Y$,

$$\psi(y) = \begin{cases} O, & \text{if } \forall (x,u) \in I(y), u > 0; \\ T, & \text{otherwise,} \end{cases} \tag{9}$$

where O stands for 'opacity' and T for 'transparency'. Verification of K-delayed opacity is along the following result.

Theorem 1: Given a system model G in (1) with secret state set $X_S \subseteq X$, let Ω be a sensor activation policy and $\mathcal{O}^K_\Omega(G)$ be the delay observer built via Algorithm 1. Then, $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and P_Ω if and only if

$$\forall y \in Y, \psi(y) = O. \tag{10}$$

Proof (Sufficiency): Suppose that (10) holds. Then, for any $s \in \mathcal{L}(G)$ such that $|s| \geq K$, $\psi(\beta(P_\Omega(s))) = O$, that is, $\forall (x,u) \in \beta(P_\Omega(s)), u > 0$. We claim that there exists $s_\infty \in \bar{s} : |s| - |s_\infty| \leq K$ such that

$$\exists (x,u) \in I(\beta(P_\Omega(s_\infty))), \text{ s.t. } u = \infty. \tag{11}$$

Suppose on the contrary that such a string s_∞ does not exist. Then, $\forall t \in \bar{s}: |s| - |t| \leq K$,

$$\forall (x,u) \in \beta(P_\Omega(t)), u \leq K.$$

According to the recording strategy in Algorithm 1 (cf. lines 11-14 and lines 27-32), the delay value $u < \infty$ will be updated to $u-1$ at each step and will become 0 within K steps, resulting in a contradiction.

Note that the existence of $s_\infty \in \bar{s}$ satisfying (11) implies that there exists a string $s' \in \mathcal{L}(G)$ such that $P_\Omega(s') = P_\Omega(s_\infty)$ and $\alpha(s') \nsubseteq X_S$ (cf. lines 19-23 and lines 29-30). By $|s| - |s_\infty| \leq K$, we have $s_\infty \in \zeta^K_\Omega(s)$, implying $P_\Omega(s') = P_\Omega(s_\infty) \in P_{E_o}(\zeta^K_\Omega(s))$. Hence, (7) holds for any $s \in \mathcal{L}(G) : |s| \geq K$. We conclude that $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and P_{E_o}.

(Necessity): Suppose that $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and P_Ω. We will show that for any $s \in \mathcal{L}(G)$, $x \in \alpha(s)$, the delay value of x will always be greater than 0.

We first consider the set of string $\{s \in \mathcal{L}(G) \mid |s| < K\}$. In Algorithm 1, the delay value of each initial state is ∞ or K. According to the strategy of delay record (cf. lines 11-14 and lines 27-32), the delay value of each state $x \in \alpha(s)$ will remain greater than 0 within $K-1$ steps. We now show that for any $s_\infty \in \mathcal{L}(G)$, if (6) holds for $s = s_\infty$, then the delay value of any $x \in \alpha(s_\infty)$ is ∞. Denote $s_\infty = s_0 e_0 s_1$ such that $P_\Omega(s_0 e_0) = P_\Omega(s_\infty)$ and $e_0 \in \Theta(\omega(s_0))$. According to the recording strategy for state estimation set not contained in X_S (cf. lines 19-23 and lines 29-30), the delay value of any $x \in \alpha(s_0 e_0)$ is ∞. Since $P_\Omega(s_0 e_0) = P_\Omega(s_\infty)$, we have that the delay value of any $x \in \alpha(s_\infty)$ remains ∞ (cf. lines 9-14). Hence, for any $s_\infty \in \mathcal{L}(G)$ such that (6) holds with $s = s_\infty$, the delay value of the states in $\alpha(s_\infty)$ is ∞. Finally, since $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and P_Ω, (7) holds for any $s \in \mathcal{L}(G) : |s| \geq K \wedge \alpha(s) \subseteq X_S$, implying that there exists $s_\infty \in \bar{s} : |s| - |s_\infty| \leq K$ such that (6) holds for $s = s_\infty$. Then, the delay value of any state $x_\infty \in \alpha(s_\infty)$ is ∞. According to the strategy of delay record (cf. lines 11-14 and lines 27-32), the delay value of the states in $\alpha(s)$ will remain greater than 0 starting from (x_∞, ∞) within K steps.Hence, for any $y \in Y$ and $(x,u) \in y$, we have $u > 0$, implying $\psi(y) = O$, i.e., (10) holds.

Example 3: Let us introduce delay $K_1 = 1$ or $K_2 = 2$ in the observations of system G in Fig. 1(a). Algorithm 1 returns the delay observers $\mathcal{O}^{K_1}_{E_o}(G)$ and $\mathcal{O}^{K_2}_{E_o}(G)$ shown in Fig. 1(c) and Fig. 1(d), where the delays of the key transition $(1, e_2, 2)$ are recorded. Using (9) in Theorem 1, we have $\psi(\{(2,1),(3,0)\}) = T$ for $\mathcal{O}^{K_1}_\Omega(G)$ and $\psi(y) = O, \forall y \in Y$, for $\mathcal{O}^{K_2}_\Omega(G)$. Hence, $\mathcal{L}(G)$ is not K-delayed opaque with $K = 1$, but it is K-delayed opaque with $K = 2$ w.r.t. X_S and P_Ω. Hence, system G originally lacking opacity (cf. Example 2) exhibits K-delayed opacity with $K = 2$. ■

IV. Synthesis and Optimization of Sensor Activation Policy

Instead of a given sensor activation policy, it might be desirable to design the sensor activation policy, e.g., balancing between acquiring information and preserving secrecy. In this section, we aim to design a policy that monitors as many events as possible while guaranteeing opacity of the system. To formalize this objective, for two sensor activation policies $\Omega_1 = (R_1, \Theta_1)$, $\Omega_2 = (R_2, \Theta_2)$, we write $\Omega_1 \subseteq \Omega_2$ if $\forall s \in E^*$, $\Theta_1(\omega_1(s)) \subseteq \Theta_2(\omega_2(s))$, and $\Omega_1 \subset \Omega_2$ if $\Omega_1 \neq \Omega_2$. We are now in a position to formulate the problem.

Problem 1: Given a system model G in (1) and a delay K, let $X_S \subseteq X$ be the set of secret states. Find a sensor activation policy $\bar{\Omega}$ such that

i) $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and $P_{\bar{\Omega}}$;

ii) there exists no other policy Ω such that $\bar{\Omega} \subset \Omega$ and $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and P_Ω.

To solve Problem 1, we leverage the verification function in (9), extending the delay observer structure to facilitate the application of Theorem 1 throughout the optimization process. For the system model G in (1) and the verification function ψ, let us define the following container automaton

$$\mathcal{O}^K_\psi(G) = \left(Y_\psi, E_o, \beta_\psi, Y_{\psi,0}\right), \tag{12}$$

with state space $Y_\psi \in 2^{X \times (\mathbb{Z} \cup \{\infty\})} \times 2^{E_o}$. Similar to (8), let us decompose each $\bar{y} \in Y_\psi$ as $\bar{y} = (I(\bar{y}), D_\psi(\bar{y}))$, where $I(\bar{y}) \in 2^{X \times (\mathbb{Z} \cup \{\infty\})}$ represents the state estimation set with delay value, and $D_\psi(\bar{y}) \in 2^{E_o}$ represents the current sensing decision. The transition function $\beta_\psi : Y_\psi \times E_o \to 2^{Y_\psi}$ and the set of initial states $Y_{\psi,0}$ in (12) are constructed in Algorithm 2. We stress that, instead of a sensor activation policy, the new structure is based on the verification function ψ, that is,

$$\forall \bar{y} \in Y_\psi, \psi(\bar{y}) = O, \tag{13}$$

where the domain of the verification function ψ is extended to include Y_ψ with a slight abuse of notation. Note that (13) is reflected in lines 9-11 and lines 39-42 of Algorithm 2. While REC-UNOBS is as in Algorithm 1, a new procedure REC-OBS$_\psi$ is proposed to traverse the state space and record

Algorithm 2: Construction of $\mathcal{O}_{\psi}^{K}(G)$

```
   Input: $G = (X, E, \alpha, X_0), \psi, K, E_o, X_S$
   Output: $\mathcal{O}_{\psi}^{K}(G) = (Y_{\psi}, E_o, \beta_{\psi}, Y_{\psi,0})$
1  $Y_{\psi,0} \leftarrow \emptyset$; $Y_{\psi} \leftarrow \emptyset$;
2  for $\theta \in 2^{E_o}$ do
3      $\bar{y}_0^{\theta} \leftarrow (\emptyset, \theta)$; $u_0 \leftarrow \infty$;
4      if $\phi(X_0, \theta) \subseteq X_S$ then
5          $u_0 \leftarrow K$;
6      for $x_0 \in X_0$ do
7          $I(\bar{y}_0^{\theta}) \leftarrow I(\bar{y}_0^{\theta}) \cup \{(x_0, u_0)\}$;
8          REC-UNOBS$(x_0, u_0, \bar{y}_0^{\theta})$;
9      if $\psi(\bar{y}_0^{\theta}) = O$ then
10         $Y_{\psi,0} \leftarrow Y_{\psi,0} \cup \{\bar{y}_0^{\theta}\}$; $Y_{\psi} \leftarrow Y_{\psi} \cup \{\bar{y}_0^{\theta}\}$;
11         REC-OBS$_{\psi}(\bar{y}_0^{\theta})$;
12 while $\exists \bar{y} \in Y_{\psi} : (x, u) \in I(\bar{y}), e \in D_{\psi}(y)$, s.t. [$\alpha(x, e)!$
   and $\beta_{\psi}(\bar{y}, e)$ is undefined] do
13     $Y_{\psi} \leftarrow Y_{\psi} \setminus \{\bar{y}\}$; Take the accessible part of $\mathcal{O}_{\psi}^{K}(G)$;
14 Return $\mathcal{O}_{\psi}^{K}(G)$;
15 Procedure REC-OBS$_{\psi}(\bar{y})$
16 $\iota \leftarrow \{x \in X \mid (x, u) \in I(\bar{y})\}$;
17 for $\theta \in 2^{E_o}$ do
18     $E_{\theta}^{\mathrm{Rec}} \leftarrow \emptyset$;
19 for $e \in D_{\psi}(\bar{y})$ do
20     $\iota^* \leftarrow \{x \in X \mid \exists x' \in \iota,\ x \in \alpha(x', e)\}$;
21     for $\theta \in 2^{E_o}$ do
22         if $\phi(\iota^*, \theta) \subseteq X_S$ then
23             $E_{\theta}^{\mathrm{Rec}} \leftarrow E_{\theta}^{\mathrm{Rec}} \cup \{e\}$;
24         $\bar{y}^{e,\theta} \leftarrow (\emptyset, \theta)$;
25 for $(x, u) \in I(\bar{y})$ do
26     for $e \in D_{\psi}(\bar{y}) : \alpha(x, e)!$ do
27         for $\theta \in 2^{E_o}$ do
28             if $u = \infty \wedge e \in E_{\theta}^{\mathrm{Rec}}$ then
29                 $u^{\theta} \leftarrow K$;
30             else if $e \notin E_{\theta}^{\mathrm{Rec}}$ then
31                 $u^{\theta} \leftarrow \infty$;
32             else
33                 $u^{\theta} \leftarrow u - 1$;
34             for $x' \in \alpha(x, e)$ do
35                 if $(x', u^{\theta}) \notin I(\bar{y}^{e,\theta})$ then
36                     $I(\bar{y}^{e,\theta}) \leftarrow I(\bar{y}^{e,\theta}) \cup \{(x', u^{\theta})\}$;
37                     REC-UNOBS$(x', u^{\theta}, \bar{y}^{e,\theta})$;
38 for $e \in D_{\psi}(\bar{y})$ do
39     for $\theta \in 2^{E_o} : \psi(\bar{y}^{e,\theta}) = O$ do
40         Add $\beta_{\psi}(\bar{y}, e) = \bar{y}^{e,\theta}$ to $\mathcal{O}_{\psi}^{K}(G)$;
41         if $\bar{y}^{e,\theta} \notin Y_{\psi}$ then
42             $Y_{\psi} \leftarrow Y_{\psi} \cup \{\bar{y}^{e,\theta}\}$; REC-OBS$_{\psi}(\bar{y}^{e,\theta})$;
```

the delays until state $\bar{y} \in Y_{\psi}$ is such that $\psi(\bar{y}) = T$. The final step of Algorithm 2 (cf. lines 12-13) removes states where no feasible sensing decision exists to enable further transitions.

Note that $\mathcal{O}_{\psi}^{K}(G)$ is a nondeterministic automaton. A deterministic automaton contained in $\mathcal{O}_{\psi}^{K}(G)$ can be obtained by choosing one initial state and by making one sensing decision at each transition. To formalize the reasoning above, we now establish that $\mathcal{O}_{\psi}^{K}(G)$ collects all possible sensor activation policies guaranteeing K-delayed opacity.

Algorithm 3: Sensor Activation Policy Optimization

```
   Input: $\mathcal{O}_{\psi}^{K}(G)$;
   Output: $\bar{\Omega} = (R, \Theta)$ with $R = (Q, E, \omega, q_0)$;
1  Find $\bar{y}_0 \in Y_{\psi,0}$ satifying $\forall \bar{y}'_0 \in Y_{\psi,0}, D_{\psi}(\bar{y}'_0) \not\subset D_{\psi}(\bar{y}_0)$;
2  $i \leftarrow 0$; $\psi_Q(\bar{y}_0) \leftarrow i$; $\Theta(i) \leftarrow D_{\psi}(\bar{y}_0)$; $q_0 \leftarrow i$;
3  ADD$(\bar{y}_0, \bar{\Omega}, \mathcal{O}_{\psi}^{K}(G))$;
4  for $q \in Q$ do
5      for $e \in E \backslash \{e \in E \mid \omega(q, e)!\}$ do
6          Add $\omega(q, e) = q$ to $R$;
7  Return $\bar{\Omega}$;
8  Procedure ADD$(\bar{y}, \bar{\Omega}, \mathcal{O}_{\psi}^{K}(G))$;
9  for $e \in E_o : \beta_{\psi}(\bar{y}, e)!$ do
10     Find $\bar{y}' \in \beta_{\psi}(\bar{y}, e)$ satisfying that
       $\forall \bar{y}'' \in \beta_{\psi}(\bar{y}, e), D_{\psi}(\bar{y}') \not\subset D_{\psi}(\bar{y}'')$;
11     if $\psi_Q(\bar{y}') \in \mathbb{N}$ then
12         Add $\omega(\psi_Q(\bar{y}), e) = \psi_Q(\bar{y}')$ to $R$;
13     else
14         $i \leftarrow i + 1$; $\psi_Q(\bar{y}') \leftarrow i$; $\Theta(i) \leftarrow D_{\psi}(\bar{y}')$;
15         $Q \leftarrow Q \cup \{i\}$; Add $\omega(\psi_Q(\bar{y}), e) = i$ to $R$;
16         ADD$(\bar{y}', \bar{\Omega}, \mathcal{O}_{\psi}^{K}(G))$;
```

Definition 3: Given a sensor activation policy Ω and a verification function ψ, the delay observer $\mathcal{O}_{\Omega}^{K}(G)$ is said to be contained in $\mathcal{O}_{\psi}^{K}(G)$ if $\forall s \in \mathcal{L}(G)$,

$$(I(\beta(P_{\Omega}(s))), \Theta(\omega(s))) \in \beta_{\psi}(P_{\Omega}(s)). \tag{14}$$

Lemma 1: The delay observer $\mathcal{O}_{\Omega}^{K}(G)$ is contained in $\mathcal{O}_{\psi}^{K}(G)$ if and only if (10) holds.

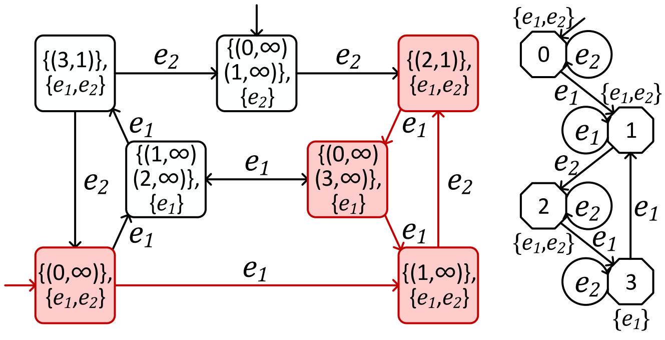
Proof (Necessity): Suppose that $\mathcal{O}_{\Omega}^{K}(G)$ is contained in $\mathcal{O}_{\psi}^{K}(G)$. Then, (14) holds for any $s \in \mathcal{L}(G)$. By (9) and (13), we have $\psi(\beta(P_{\Omega}(s))) = O$, implying that (10) holds.

(Sufficiency): Suppose that (10) holds. Since $\psi(y_0) = O$, there exists $\bar{y}_0 \in Y_{\psi,0}$ such that $I(y_0) = I(\bar{y}_0)$ and $\Theta(D(y_0)) = D_{\psi}(\bar{y}_0)$ (cf. lines 1-6 of Algorithm 1 and lines 2-11 of Algorithm 2). Suppose that (14) holds for a string $s = s' \in \mathcal{L}(G)$. We now consider an $e \in E$ such that $\alpha(s'e)!$. In case $e \notin \Theta(\omega(s'))$, we directly have that (14) holds for $s = s'e$ since $P_{\Omega}(s') = P_{\Omega}(s'e)$ and $\omega(s') = \omega(s'e)$. In case $e \in \Theta(\omega(s'))$, let us denote $\bar{y} = (I(\beta(P_{\Omega}(s'))), \Theta(\omega(s')))$ and evaluate REC-OBS$_{\psi}(\bar{y})$ in Algorithm 2. First, lines 16-24 return $(\emptyset, \Theta(\omega(s'e)))$. Then, lines 25-37 return $\bar{y}^e = (I(\beta(P_{\Omega}(s'e))), \Theta(\omega(s'e)))$ based on $\Theta(\omega(s'e))$ (cf. lines 25-36 of Algorithm 1). By (9), since $\psi(\beta(P_{\Omega}(s'))) = O$, we have $\psi(\bar{y}^e) = O$, implying that $\bar{y}^e$ is added in Y_{ψ} (cf. lines 38-42). Since $\bar{y}^e \in \beta_{\psi}(\bar{y}, e)$, $\bar{y}$ will not be eliminated (cf. lines 13-14). Hence, (14) holds for $s'e$. By induction, we conclude that $\mathcal{O}_{\Omega}^{K}(G)$ is contained in $\mathcal{O}_{\psi}^{K}(G)$.

To find an optimal policy $\bar{\Omega}$ based on $\mathcal{O}_{\psi}^{K}(G)$, Algorithm 3 applies a greedy strategy at each transition in $\mathcal{O}_{\psi}^{K}(G)$ to determine the sensing decision. The idea is to maximize the monitored events while guaranteeing K-delayed opacity. The following result shows that this greedy strategy returns a policy that solves Problem 1.

Theorem 2: Algorithm 2 and Algorithm 3 solve Problem 1.

Proof: First, we show that Algorithm 3 will produce a nonempty output whenever Problem 1 admits a solution. Suppose



(a) $\mathcal{O}_{\psi}^{K_1}(G)$ (b) $\bar{\Omega}$

Fig. 2. (a) Container $\mathcal{O}_{\psi}^{K_1}(G)$ (with greedy strategy marked in red), collecting all possible sensor activation policies guaranteeing K-delayed opacity with $K = K_1$; (b) Optimal sensor activation policy $\bar{\Omega}$ built with the marked container states.

that Problem 1 has a solution $\bar{\Omega}$. Then, $\mathcal{O}_{\bar{\Omega}}^{K}(G)$ is contained in $\mathcal{O}_{\psi}^{K}(G)$ by Theorem 1 and Lemma 1 since $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and $P_{\bar{\Omega}}$. Hence, Algorithm 2 will produce a non-empty $\mathcal{O}_{\psi}^{K}(G)$, which guarantees that Algorithm 3 will produce a non-empty output, as $\bar{\Omega}$ can be constructed directly from the states in $\mathcal{O}_{\psi}^{K}(G)$ (cf. lines 1 and 10 of Algorithm 3).

Let $\bar{\Omega}$ be the output of Algorithm 3. Next, we show that such $\bar{\Omega}$ indeed constitutes a solution to Problem 1.

We first check condition i) in Problem 1. Since $\bar{\Omega}$ is built directly with corresponding states in $\mathcal{O}_{\psi}^{K}(G)$ (cf. lines 1-2 and lines 10-15 of Algorithm 3), we have that $\mathcal{O}_{\bar{\Omega}}^{K}(G)$ is contained in $\mathcal{O}_{\psi}^{K}(G)$. Then, by Theorem 1 and Lemma 1, $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and $P_{\bar{\Omega}}$.

We then check condition ii) in Problem 1. By contradiction, suppose there exists $\Omega' = (R', \Theta')$ with $R' = (Q', E, \omega', q_0')$ such that $\bar{\Omega} \subset \Omega'$ and $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and $P_{\Omega'}$. Then, by Theorem 1 and Lemma 1, $\mathcal{O}_{\Omega'}^{K}(G) = (Y', E_o, \beta', y_0')$ is contained in $\mathcal{O}_{\psi}^{K}(G)$. Since $\bar{\Omega} \subset \Omega'$, there exist two cases: $\Theta(\omega(\epsilon)) \subset \Theta'(\omega'(\epsilon))$ or $\Theta(\omega(\epsilon)) = \Theta'(\omega'(\epsilon))$. In the first case, $\Theta(\omega(\epsilon))$ cannot be selected in line 1, contradicting the fact that $\bar{\Omega}$ is the output of Algorithm 3. In the second case, there exists $s \in \mathcal{L}(G)$ such that $\Theta(\omega(se)) \subset \Theta'(\omega'(se))$ and for any $s' \in \bar{s}$, $\Theta(\omega(s')) = \Theta'(\omega'(s'))$. Denote $\bar{y} = (I(\beta(P_{\bar{\Omega}}(s))), \Theta(\omega(s))) = (I(\beta'(P_{\Omega'}(s))), \Theta'(\omega'(s)))$. Since $\mathcal{O}_{\bar{\Omega}}^{K}(G)$ and $\mathcal{O}_{\Omega'}^{K}(G)$ are contained in $\mathcal{O}_{\psi}^{K}(G)$, we have $\bar{y} \in \beta_{\psi}(P_{\bar{\Omega}}(s))$. Then, since $\Theta(\omega(se)) \subset \Theta'(\omega'(se))$, we have that $(I(\beta(P_{\bar{\Omega}}(se))), \Theta(\omega(se))) \in \beta_{\psi}(\bar{y}, e)$ cannot be selected in line 10, contradicting the fact that $\bar{\Omega}$ is the output of Algorithm 3. Hence, there exists no other policy Ω such that $\bar{\Omega} \subset \Omega$ and $\mathcal{L}(G)$ is K-delayed opaque w.r.t. X_S and P_{Ω}. We conclude that $\bar{\Omega}$ is a solution to Problem 1. ■

Remark 1 (Non-Uniqueness of Solution): Note that Problem 1 does not exclude the existence of more $\bar{\Omega}$ satisfying i)-ii), meaning that the solution may be not unique. This is reflected in the fact that $\bar{y}_0$ and $\bar{y}^*$ in lines 1 and 10 of Algorithm 3 may be not unique.

Example 4: The sensor activation policy in Fig. 1(b) does not achieve K-delayed opacity with $K = K_1 = 1$ (cf. Example 3). To design a sensor activation policy achieving this, we obtain $\mathcal{O}_{\psi}^{K_1}(G)$ in Fig. 2(a) from Algorithm 3. Then, a greedy strategy is applied (marked in red) to obtain the sensor activation policy $\bar{\Omega}$ in Fig. 2(b). Note that in Fig. 2(a), redundant sensing decisions (representing that some events being monitored cannot happen at the current step) are omitted. By Theorem 3, $\bar{\Omega}$ in Fig. 2(b) solves Problem 1.

V. Conclusion

This letter has investigated opacity of discrete event systems under delayed observations. A novel notion of K-delayed opacity has been introduced along with a delay observer structure to enable verification. To synthesize and optimize the sensor activation policy, a new container structure has been constructed collecting all feasible sensor activation policies that preserve K-delayed opacity. The perspective adopted in this letter is the one of a defender trying to mask the system to an eavesdropper. An interesting point for future work is to adopt the perspective of the eavesdropper.

References

[1] A. Saboori and C. N. Hadjicostis, "Notions of security and opacity in discrete event systems," in *Proc. IEEE 46th Conf. Decis. Control*, 2007, pp. 5056–5061.

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

[3] R. Jacob, J.-J. Lesage, and J.-M. Faure, "Overview of discrete event systems opacity: Models, validation, and quantification," *Annu. Rev. Control*, vol. 41, pp. 135–146, Jun. 2016.

[4] Y. Ji, Y.-C. Wu, and S. Lafortune, "Enforcement of opacity by public and private insertion functions," *Automatica*, vol. 93, pp. 369–378, Jul. 2018.

[5] X. Yin and S. Li, "Synthesis of dynamic masks for infinite-step opacity," *IEEE Trans. Autom. Control*, vol. 65, no. 4, pp. 1429–1441, Apr. 2020.

[6] M. Mizoguchi and T. Ushio, "Abstraction-based control under quantized observation with approximate opacity using symbolic control barrier functions," *IEEE Control Syst. Lett.*, vol. 6, pp. 2222–2227, 2021.

[7] X. Li, C. N. Hadjicostis, and Z. Li, "Opacity enforcement in discrete event systems using extended insertion functions under inserted language constraints," *IEEE Trans. Autom. Control*, vol. 68, no. 11, pp. 6797–6803, Nov. 2023.

[8] W. Duan, R. Liu, M. P. Fanti, C. N. Hadjicostis, and Z. Li, "Edit mechanism synthesis for opacity enforcement under uncertain observations," *IEEE Control Syst. Lett.*, vol. 7, pp. 2041–2046, 2023.

[9] Y. Xie, S. Li, and X. Yin, "Optimal synthesis of opacity-enforcing supervisors for qualitative and quantitative specifications," *IEEE Trans. Autom. Control*, vol. 69, no. 8, pp. 4958–4973, Aug. 2024.

[10] X. Li, C. N. Hadjicostis, and Z. Li, "Opacity enforcement in discrete event systems using modification functions," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 3252–3264, 2025.

[11] J. Wang, S. Baldi, W. Yu, and X. Yin, "Distributed fault diagnosis in discrete event systems with transmission delay impairments," *IEEE Trans. Autom. Control*, vol. 69, no. 8, pp. 5508–5515, Aug. 2024.

[12] J. Wang, S. Baldi, W. Yu, and X. Yin, "Fault diagnosis and prognosis in partially-observed discrete event systems with delayed observations," in *Proc. IEEE 63rd Conf. Decis. Control (CDC)*, 2024, pp. 3675–3680.

[13] F. Cassez, J. Dubreil, and H. Marchand, "Synthesis of opaque systems with static and dynamic masks," *Formal Methods Syst. Design*, vol. 40, pp. 88–115, Jan. 2012.

[14] E. Dallal and S. Lafortune, "On most permissive observers in dynamic sensor activation problems," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 966–981, Apr. 2014.

[15] D. Sears and K. Rudie, "Minimal sensor activation and minimal communication in discrete-event systems," *Discr. Event Dyn. Syst.*, vol. 26, no. 2, pp. 295–349, 2016.

[16] X. Yin and S. Lafortune, "A general approach for optimizing dynamic sensor activation for discrete event systems," *Automatica*, vol. 105, pp. 376–383, Jul. 2019.

[17] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Cham, Switzerland: Springer, 2009.

[18] W. Wang, S. Lafortune, A. R. Girard, and F. Lin, "Optimal sensor activation for diagnosing discrete event systems," *Automatica*, vol. 46, no. 7, pp. 1165–1175, 2010.