2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Secure-by-Construction Controller Synthesis for Stochastic Systems under Linear Temporal Logic Specifications

Yifan Xie, Xiang Yin, Shaoyuan Li and Majid Zamani

Abstract— **In this paper, we investigate the problem of synthesizing optimal control policies for stochastic control systems to achieve high-level temporal logic specifications under security constraints. Specifically, we consider a stochastic control system modeled by a finite labeled Markov Decision Process (MDP). We consider a passive intruder (an eavesdropper) that can observe the external output behavior of the system. We assume the system has a secret, modeled as visiting of some secret states, that does not want to be revealed to the intruder. The security constraint is that the intruder can never determine for sure that the system is/was at a secret state for any specific instant of time. The overall objective is to maximize the probability of achieving the temporal logic task while ensuring the information-flow security of the system. An effective algorithm is proposed to solve this problem. Specifically, we show that the security constraints can be handled as a safety requirement over the information-state-space and the optimal control problem can be solved by leveraging existing results from probabilistic model checking. The proposed approach is also illustrated by a case study for robot task planning.**

I. Introduction

In recent years, there has been an increasing interest in using *formal methods* for the verification and design of complex engineering systems such as robots, autonomous vehicles and energy systems. Temporal logics such as linear temporal logic (LTL) provide rich and user-friendly languages for designer to describe the desired high-level specifications. Verification and synthesis of control strategies for LTL tasks have attracted considerable attentions in the control community in the past few years.

In many real-world applications, it is more practical to model the system using stochastic formal models such as Markov Decision Processes (MDPs). In the context of formal synthesis of stochastic systems, algorithms have been proposed to synthesize optimal control strategies for MDPs under general LTL specifications [3], co-safe LTL specifications [10] and CTL specifications [11]. In [6], the total cost of the prefix and suffix structure is minimized using linear programs. Recently, reinforcement learning has also been applied to s maximize the probability of satisfying LTL formulas; see, e.g., [7], [12].

The above mentioned works on formal synthesis of MDPs focus on the correctness of the system such as safety or liveness. Yet, security and privacy constraints, which are very important in cyber-physical systems, have not been fully considered [8], [9], [13]. Such issues arise naturally when agents exchange information among the group or transmit information to a third party. An outside malicious intruder may access the information-flow of the system and infer the secret of the system. Due to the importance of security and privacy concerns, recent works have started to consider security as additional constraints in temporal logic synthesis; see, e.g., [14], [17], [20], [21]. For example, the notion of differential privacy was used in [14], [20] to capture the privacy requirements in LTL synthesis for stochastic systems.

In this work, we study a security-aware optimal LTL synthesis problem for stochastic systems modeled as finite MDPs. Different from the existing works on security-aware formal synthesis, we consider an information-flow security property called the *infinite-step opacity* [15], [23], [24]. This security property requires that an outside eavesdropper can never infer for sure that the system is/was at a secret state at any specific instant of time based on its external observation. It is known that the expressiveness of opacity and differential privacy are different as discussed in detail in [25]. In this paper, we propose an effective approach that maximizes the probability of satisfying the LTL formula such that the information-flow of the system is infinite-step opaque. Our approach is based on the construction of a novel information structure that captures all delayed information of the system. Then by suitably augmenting the information space as well as the Rabin automaton capturing the LTL requirement, we show that the optimal security-aware control synthesis problem can be solved effectively by probabilistic model checking approaches.

In the context of security-aware formal design, our work is mostly related to [17], [21], where optimal paths satisfying both opacity and the LTL specification are generated. However, these works consider path planning for non-stochastic systems, while our work aims to *synthesize control policies* for stochastic systems. In the context of opacity-enforcing supervisory control of discrete-event systems [4], [16], [18], [19], [22], most of the existing works consider the case of non-stochastic systems modeled by finite-state automata except [2], which considers current-state opacity, a notion weaker than our notion of infinite-step opacity.

This work was supported by the National Natural Science Foundation of China (62061136004,6217020111, 61803259), the NSF under grant ECCS-2015403, and the German Research Foundatiion (DFG) under grant ZA 87317-1.

Yifan Xie, Xiang Yin and Shaoyuan Li are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. {xyfan1234,yinxiang,syli}@sjtu.edu.cn.

M. Zamani is with the Computer Science Department, University of Colorado Boulder, CO 80309, USA. M. Zamani is also with the Computer Science Department, Ludwig Maximilian University of Munich, 80539 Munich, Germany. majid.zamani@colorado.edu.

II. PRELIMINARY

A. System Model

Let S be a finite set, we use S^* and S^ω to denote the set of all finite and infinite sequences over S, respectively. We denote by 2^S the power set of S. Let $\tau = s_0 s_1 \dots s_n \in S^*$ be a finite string, we denote the last element of τ by $\texttt{Last}(\tau)$. For finite strings $\tau = s_0 \dots s_n, \tau' = s'_0 \dots s'_m \in S^*$, their concatenation is $\tau\tau' = s_0 \dots s_n s'_0 \dots s'_m \in S^*$. We write $\tau_1 \leq \tau_2$ if there exists $\tau' \in S^*$ such that $\tau_1\tau' = \tau_2$ and we write $\tau_1 < \tau_2$ if $\tau_1 \leq \tau_2$ and $\tau_1 \neq \tau_2$.

We model a stochastic system as a finite labeled MDP

$$\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, L),$$

where S is a finite set of states, $s_0 \in S$ is the initial state, A is a finite set of actions, $P : S \times A \times S \to [0, 1]$ describes the transition probability under each action, i.e., $P(s, a, s')$ is the probability of the transition from state s to state s' when action a is taken. For each state $s \in S$, we denote by $A(s) = \{a \in A : \exists s' \in S, P(s, a, s') > 0\}$ as the set of available actions at state s. Also, $\mathcal{AP}$ is a set of atomic propositions describing some basic properties of interest and $L : S \to 2^{\mathcal{AP}}$ is a labeling function that assigns to each state a set of atomic propositions that hold in that state.

Given a labeled MDP $\mathcal{M}$, a finite path in $\mathcal{M}$ is a finite sequence of states $\tau = s_0 s_1 \dots s_n$ such that $\forall i \geq 0, \exists a \in A(s_i) : P(s_i, a, s_{i+1}) > 0$. An infinite path in $\mathcal{M}$ is defined analogously. We denote by $\texttt{Path}(\mathcal{M})$ and $\texttt{Path}^\omega(\mathcal{M})$ the set of finite and infinite paths in $\mathcal{M}$, respectively. The trace of an infinite path $\tau = s_0 s_1 \cdots$ is denoted by $\texttt{trace}(\tau)$, which is an infinite sequence of atomic propositions, i.e. $\texttt{trace}(\tau) = L(s_0)L(s_1) \cdots \in (2^{\mathcal{AP}})^\omega$.

Given a labeled MDP $\mathcal{M}$, a control policy is a mapping $\Gamma : S^* \to A$. We denote by $\mathcal{M}_\Gamma$ the labeled MDP under control. Then we say a finite path $\tau = s_0 s_1 \dots s_n$ is in the controlled system $\mathcal{M}_\Gamma$ if for any $i \geq 0$, we have $P(s_i, \Gamma(s_0 s_1 \dots s_i), s_{i+1}) > 0$. An infinite path in $\mathcal{M}_\Gamma$ is defined analogously. We denote by $\texttt{Path}(\mathcal{M}_\Gamma)$ and $\texttt{Path}^\omega(\mathcal{M}_\Gamma)$ the set of finite and infinite paths in $\mathcal{M}_\Gamma$, respectively. Note that when Γ is a state-based policy, i.e., $\Gamma : S \to A$, then $\mathcal{M}_\Gamma$ is actually a Markov chain.

B. Linear Temporal Logic

An LTL formula consists of a finite set of atomic propositions $\mathcal{AP}$ and both logic and temporal operators according to the following syntax [1]:

$$\varphi := \text{True} \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc \varphi \mid \varphi_1 \,\mathcal{U}\, \varphi_2,$$

where $a \in \mathcal{AP}$, $\neg$ and $\wedge$ are logical negation and conjunction, respectively, $\bigcirc$ and $\mathcal{U}$ denote the "next" and "until" operators, respectively. True stands for the set $2^{\mathcal{AP}}$. Other temporal operators can be derived, e.g., "eventually" is $\Diamond\varphi =$ True $\mathcal{U}$ φ and "always" is $\Box\varphi = \neg\Diamond\neg\varphi$. LTL formulas are evaluated over infinite words. Given an infinite word $\sigma \in (2^{\mathcal{AP}})^\omega$, we write $\sigma \models \varphi$ if σ satisfies the LTL formula φ. The reader is referred to [1] for details on LTL semantics. We denote $\mathcal{L}_\varphi = \{\sigma \in (2^{\mathcal{AP}})^\omega : \sigma \models \varphi\}$ as the maximal set of words satisfying φ.

A deterministic Rabin automaton (DRA) is a tuple $R = (Q, \Sigma, \delta, q_0, Acc)$, where Q is a finite set of states, Σ is a finite set of alphabets, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the initial state, and $Acc = \{(B_1, G_1), (B_2, G_2), \dots, (B_n, G_n)\}$ is a finite set of Rabin pairs such that $B_i, G_i \subseteq Q$ for all $i = 1, 2, \dots, n$. Given an infinite word $\sigma = \sigma_1\sigma_2 \cdots \in \Sigma^\omega$, the infinite run of DRA R over σ is the sequence of infinite states $\rho = q_0 q_1 \cdots \in Q^\omega$ such that $q_i = \delta(q_{i-1}, \sigma_i)$ for any $i \geq 1$. An infinite run $\rho \in Q^\omega$ is said to be accepted if and only if there exists a rabin pair $(B_i, G_i) \in Acc$ such that $\texttt{inf}(\rho) \cap B_i = \emptyset$ and $\texttt{inf}(\rho) \cap G_i \neq \emptyset$, where $\texttt{inf}(\rho)$ is the set of states that occurs infinitely many times in ρ. Then an infinite word σ is said to be accepted if it induces an accepting infinite run. We denote by $\mathcal{L}^\omega(R) \subseteq \Sigma^\omega$ the accepted language of DRA R, which is the set of accepting words. Given an LTL formula φ over $\mathcal{AP}$, there exists a DRA over $2^{\mathcal{AP}}$ that (only) accepts all infinite words satisfying φ, i.e., $\mathcal{L}^\omega(R) = \mathcal{L}_\varphi$ [5].

III. PROBLEM FORMULATION

Given a labeled MDP $\mathcal{M}$, we assume that there is a malicious intruder that can observe the external behavior of the system. However, the internal states of the system are not available directly to the intruder and the intruder needs to infer the internal state based on its observation and its knowledge about the dynamic of the system. Formally, we model the intruder's observation as an output function

$$H : S \to Y,$$

where Y is the set of output symbols. The output function is extended to $H : S^* \cup S^\omega \to Y^* \cup Y^\omega$ naturally. Therefore, after executing a finite path $\tau = s_0 s_1 \dots s_n \in S^*$, the intruder will observe a sequence of observations $H(\tau) = H(s_0)H(s_1) \dots H(s_n) \in Y^*$.

We assume that the labeled MDP $\mathcal{M}$ has some "secrets", which are modeled as a set of secret states $S_{secret} \subseteq S$. The security requirement is that the intruder can never determine for sure that the system is/was at secret states for any *specific instant of time* based on observations. Such a requirement is called the *infinite-step opacity* [15]. Therefore, an infinite-step opaque system will always hold the plausible deniability for visiting secret states.

Definition 1: (Infinite-step Opacity) Given a labeled MDP $\mathcal{M}$ with a set of secret states $S_{secret} \subseteq S$ and a control policy Γ, the MDP under control $\mathcal{M}_\Gamma$ is said to be infinite-step opaque w.r.t. S_{secret} if

$$\begin{aligned} &(\forall \tau_1\tau_2 \in \texttt{Path}(\mathcal{M}_\Gamma) : \texttt{Last}(\tau_1) \in S_{secret}) \\ &(\exists \tau'_1\tau'_2 \in \texttt{Path}(\mathcal{M}) : \texttt{Last}(\tau'_1) \notin S_{secret}) \\ &[H(\tau_1) = H(\tau'_1) \wedge H(\tau_2) = H(\tau'_2)]. \end{aligned} \tag{1}$$

Remark 1: The above definition implicitly assumes that the intruder knows the system model $\mathcal{M}$ and it can observe the external observation $H(\tau)$. However, the intruder does not know the control policy Γ because we consider the non-secret path $\tau'_1\tau'_2$ in the open-loop system $\mathcal{M}$ rather than the

closed-loop system $\mathcal{M}_\Gamma$. Furthermore, we assume that the intruder cannot observe the input action $a \in A$ at each instant of time. This assumption, however, can actually be relaxed by augmenting the state-space so that each input a is encoded in the output of its successor state.

Infinite-step opacity can also be characterized in terms of the delayed state estimate. Formally, for any sequence of observations $\alpha\beta \in H(\texttt{Path}(\mathcal{M}_\Gamma)) \subseteq Y^*$ of the intruder, we denote by $\hat{E}(\alpha \mid \alpha\beta)$ the *delayed state estimate* that captures the set of all possible states the system could be in when α is observed given the entire observation $\alpha\beta$, i.e.,

$$\hat{E}(\alpha \mid \alpha\beta) := \left\{ \texttt{Last}(\tau_1) \in S : \begin{array}{c} \exists \tau_1\tau_2 \in \texttt{Path}(\mathcal{M}) \text{ s.t.} \\ H(\tau_1) = \alpha \wedge H(\tau_2) = \beta \end{array} \right\}.$$

For simplicity, we define $\hat{E}(\alpha) = \hat{E}(\alpha|\alpha)$ as the current state estimate given the observation α. Then infinite-step opacity in the labeled MDP under control defined in Definition 1 can be reformulated as follows

$$\forall \alpha\beta \in H(\texttt{Path}(\mathcal{M}_\Gamma)), \hat{E}(\alpha \mid \alpha\beta) \nsubseteq S_{secret}. \quad (2)$$

Given a labeled MDP $\mathcal{M}$ controlled by policy Γ and an LTL formula φ over $\mathcal{AP}$, we denote by $Pr(\mathcal{M}_\Gamma \vDash \varphi)$ the probability of satisfying φ in the labeled MDP under Γ, i.e.,

$$Pr(\mathcal{M}_\Gamma \vDash \varphi) = Pr\left(\{\tau \in \texttt{Path}(\mathcal{M}_\Gamma) : L(\tau) \vDash \varphi\}\right).$$

The reader is referred to [1] for more details on how to define the probability measure on infinite paths in MDPs under a policy. Intuitively, the probability of infinite paths satisfying LTL formula φ is calculated by generating a cylinder set that consists of all infinite paths that start with a finite path.

Our overall control objective is to synthesize a control policy Γ that maximizes the probability of satisfying a given LTL specification, while satisfying the security constraint described by infinite-step opacity as described next.

Problem 1: Given a labeled MDP $\mathcal{M}$ with a set of secret states $S_{secret} \subseteq S$ and an LTL specification φ, synthesize a control policy $\Gamma : S^* \to A$ such that:

(i) the MDP controlled by policy, i.e., $\mathcal{M}_\Gamma$, is infinite-step opaque; and

(ii) for any other control policy Γ' satisfying (i), the probability of $\mathcal{M}_\Gamma$ satisfying φ is no less than that of $\mathcal{M}_{\Gamma'}$, i.e., $Pr(\mathcal{M}_\Gamma \vDash \varphi) \geq Pr(\mathcal{M}_{\Gamma'} \vDash \varphi)$.

IV. Information-State Estimator

In order to capture the security constraint in Problem 1, one needs to think, from the intruder's point of view, what states the system may be at for some previous time instants. To this end, in this section, we study how intruder's information about the system evolves when delayed information is involved.

A. Information-state Estimator

First, we define some necessary operators. Let $s \in S$ be a state and $y \in Y$ be an output, we denote by $\texttt{Post}(s, y)$ the set of all possible successor states of s when the intruder observes y, i.e.,

$$\texttt{Post}(s, y) = \{s' : \exists a \in A \text{ s.t. } P(s, a, s') > 0 \wedge H(s') = y\}.$$

For a set of states $\eta \in 2^S$, we define $\texttt{Post}(\eta, y) = \cup_{s\in\eta}\texttt{Post}(s, y)$. Let $\theta \in 2^{S\times S}$ be a set of state pairs and $y \in Y$ be an observation, we define

$$\overline{\texttt{Post}}(\theta, y) = \{(s_1, s_2') : \exists (s_1, s_2) \in \theta \text{ s.t. } s_2' \in \texttt{Post}(s_2, y)\},$$

as the new set of state pairs that tracks the current states and where they come from. Also, for a set of states $\eta \in 2^S$, we define

$$\odot(\eta) = \{(s, s) \in S \times S : s \in \eta\},$$

that maps a set of states to a set of state pairs.

In order to enforce infinite-step opacity for the labeled MDP, we need to track the state estimate based on the external observation of the intruder. When a finite path $\tau = s_0 s_1 \dots s_n \in \texttt{Path}(\mathcal{M})$ is executed, the intruder observes a sequence of outputs $\alpha = H(\tau) = H(s_0)H(s_1)\dots H(s_n) = y_0 y_1 \dots y_n$. Note that the initial observation y_0 does not provide any information because we assume that the initial state s_0 is unique. Therefore, to capture infinite-step opacity, we need to compute all possible delayed state estimates for all time instants along the observation, i.e.,

$$\hat{E}(y_1 \mid \alpha), \hat{E}(y_1 y_2 \mid \alpha), \dots, \hat{E}(y_1 y_2 \dots y_n \mid \alpha).$$

This is done by the structure of *information-state estimator*, which is a new transition system defined as follows.

Definition 2: The *information-state estimator* is a transition system T w.r.t. the labeled MDP $\mathcal{M}$

$$T = (X, x_0, Y, \zeta),$$

where

- $X \subseteq 2^S \times 2^{2^{S\times S}}$ is the set of states. For any state $x \in X$, it is in the form of $x = (C(x), D(x))$, where the first component $C(x)$ represents the current state estimate of the system and the second component $D(x)$ is a set of state pairs sets that captures all possible delayed state estimates in history.
- $x_0 = (\{s_0\}, \{\{(s_0, s_0)\}\}) \in X$ is the initial state.
- Y is the set of inputs, which is the intruder's observation by the output function H.
- $\zeta : X \times Y \to X$ is a transition function, where $\zeta(x, y) = x'$ means that there is a transition labeled by input y from state x to state x', which is obtained as follows:

$$\begin{cases} C(x') = \texttt{Post}(C(x), y), \\ D(x') = \{\overline{\texttt{Post}}(\theta, y) \in 2^{S\times S} : \theta \in D(x)\} \\ \qquad\quad \cup \{\odot(C(x'))\}. \end{cases} \quad (3)$$

The information-state estimator tracks information as follows. The initial state x_0 is the initial information-state estimate from the intruder's point of view, which is s_0 because we assume the initial-state is unique. When a new observation y is observed, the intruder will update its information by equation (3). Intuitively, the first equation updates the current state estimate of the system, and the second equation updates the delayed state estimate of the system and adds the current state estimate to the history. For each $\alpha = y_0 y_1 \dots y_n$, we denote by $\zeta(\alpha) = \zeta(x_0, y_1 \dots y_n)$

the information state reached by α. Note that for $\alpha = y_0$, we have $\zeta(\alpha) = x_0$ because y_0 does not provide any additional information.

B. Properties of the Information-state Estimator

The following result shows that the proposed information updating rule in equation (3) indeed yields the desired delayed state estimate in the labeled MDP.

Proposition 1: Given a labeled MDP $\mathcal{M}$ and a control policy, let $\tau \in \texttt{Path}(\mathcal{M}_\Gamma)$ be a finite path, $\alpha = H(\tau)$ be an observation of the intruder and $\zeta(\alpha)$ be the information-state reached. We have

(i) $C(\zeta(\alpha)) = \hat{E}(\alpha)$; and

(ii) $D(\zeta(\alpha)) = \{\theta_{\alpha',\alpha} \in 2^{S\times S} : \alpha' \leq \alpha\}$, where

$$\theta_{\alpha',\alpha} = \left\{ (\texttt{Last}(\tau_1), \texttt{Last}(\tau_2)) : \begin{matrix} \tau_1, \tau_2 \in \texttt{Path}(\mathcal{M}), \tau_1 \leq \tau_2 \\ H(\tau_1) = \alpha', H(\tau_2) = \alpha \end{matrix} \right\}.$$

The above result suggests an approach for computing all possible delayed state estimates along an observation. Specifically, for any information-state $x = (C(x), D(x)) \in 2^S \times 2^{2^{S\times S}}$, we define the class of state sets

$$D_1(x) := \{\{s \in S : (s, s') \in \theta\} : \theta \in D(x)\}.$$

Then for any observation $\alpha = y_0 y_1 \dots y_n \in H(\texttt{Path}(\mathcal{M}_\Gamma))$, by Proposition 1, all possible delayed-state estimates of the intruder can be computed by

$$D_1(\zeta(\alpha)) = \{\hat{E}(y_0 \dots y_i \mid \alpha) \in 2^S : i = 0, 1, \dots, n\}. \quad (4)$$

V. Synthesis Procedure

In this section, we present an algorithm that solves Problem 1. Our approach is to compose the MDP with the DRA and the information-state estimator in order to capture both the LTL specification and the security requirement. Then we solve the problem by applying safety game as well as probabilistic model checking techniques over the product state-space.

A. Product MDP

Let $\mathcal{M}$ be the labeled MDP, R be a DRA accepting $\mathcal{L}_\varphi$ and T be the information-state estimator. We construct a *product MDP* in order to compute all infinite paths satisfying φ and to find all finite paths that reveal the secret.

Definition 3: Given a labeled MDP $\mathcal{M} = (S, s_0, A, P, \mathcal{AP}, L)$, a DRA $R = (Q, 2^{\mathcal{AP}}, \delta, q_0, Acc)$ accepting φ and the information-state estimator $T = (X, x_0, Y, \zeta)$, the product MDP is a tuple

$$\tilde{\mathcal{M}} = (\tilde{S}, \tilde{s}_0, A, \tilde{P}, \widetilde{Acc}),$$

where

- $\tilde{S} \subseteq S \times Q \times X$ is a finite set of states. A state $\tilde{s} \in \tilde{S}$ is in the form of $\tilde{s} = (S(\tilde{s}), Q(\tilde{s}), X(\tilde{s}))$.
- $\tilde{s}_0 = (s_0, q, x_0)$ is the initial state such that $q = \delta(q_0, L(s_0))$.
- A is a finite set of actions and $A(\tilde{s})$ denotes the set of actions available at state $\tilde{s}$.
- $\tilde{P} : \tilde{S} \times A \times \tilde{S} \to [0, 1]$ is the transition probability defined by: $\tilde{P}((s, q, x), a, (s', q', x')) = P(s, a, s')$ if $q' = \delta(q, L(s'))$ and $x' = \zeta(x, H(s'))$; otherwise, $\tilde{P}((s, q, x), a, (s', q', x')) = 0$.
- $\widetilde{Acc} = \{(\tilde{B}_1, \tilde{G}_1), (\tilde{B}_2, \tilde{G}_2), \dots, (\tilde{B}_n, \tilde{G}_n)\}$ is a finite set of Rabin pairs. For every $(\tilde{B}_i, \tilde{G}_i) \in \widetilde{Acc}$, a state $(s, q, x) \in \tilde{B}_i$ if $q \in B_i$ and a state $(s, q, x) \in \tilde{G}_i$ if $q \in G_i$.

For a state $(s_i, q_{i+1}, x_i) \in \tilde{S}$, the first component s_i is the state that the system is currently at. The second component is the state corresponding to the satisfication of the LTL formula. The third component is the state estimate from the intruder's point of view. The product MDP restricts the transition of the labeled MDP such that each transition not only satisfy the LTL specification and, at the same time, the information-state update is consistent with its current observation. If a transition is feasible, then the transition probability of the product MDP $\tilde{\mathcal{M}}$ is determined by the original labeled MDP $\mathcal{M}$.

Without loss of generality, we consider a control policy on the product MDP as a mapping $\tilde{\Gamma} : \tilde{S} \to A$ that maps a state to a specific action. An action at state $\tilde{s}$ is denoted by $\tilde{\Gamma}(\tilde{s})$ and there exists a state $\tilde{s}'$ such that $\tilde{P}(\tilde{s}, \tilde{\Gamma}(\tilde{s}), \tilde{s}') > 0$. We denote by $\tau^+ = \tilde{s}_0 \tilde{s}_1 \dots \tilde{s}_n$ a path on the product MDP under a policy $\tilde{\Gamma}$ such that for all $i \geq 0$, $\tilde{P}(\tilde{s}_i, \tilde{\Gamma}(\tilde{s}_i), \tilde{s}_{i+1}) > 0$. There is a one-to-one correspondence between the path $\tau = s_0 s_1 \dots$ in the MDP $\mathcal{M}$ and the path $\tau^+ = (s_0, q_1, x_0)(s_1, q_2, x_1) \dots$ in the product MDP $\tilde{\mathcal{M}}$. Given a path τ^+ on the product MDP, the correspondence path τ on the labeled MDP $\mathcal{M}$ satisfies φ if and only if τ^+ is accepting.

Note that the set of actions for $\tilde{\mathcal{M}}$ is the same as the one for $\mathcal{M}$. Due to the one-to-one correspondence relationship, given a control policy $\tilde{\Gamma} : \tilde{S} \to A$ on the product MDP $\tilde{\mathcal{M}}$, one can obtain a control policy $\Gamma : S^* \to A$ on the labeled MDP $\mathcal{M}$ by keeping track of sequences of states on the product MDP. Therefore, we can induce a policy Γ for $\mathcal{M}$ from a policy $\tilde{\Gamma}$ for $\tilde{\mathcal{M}}$ as follows.

Definition 4: (Induced Policy) Given a product MDP and a policy $\tilde{\Gamma} : \tilde{S} \to A$, $\tau^+ = (s_0, q_1, x_0)(s_1, q_2, x_1) \dots (s_n, q_{n+1}, x_n)$ is a finite path on the product MDP $\tilde{\mathcal{M}}$ that ends in $\tilde{s}_n = (s_n, q_{n+1}, x_n)$, then the control policy Γ on the labeled MDP maps a sequence of states $s_0 s_1 \dots s_n$ to an action $\tilde{\Gamma}(\tilde{s}_n)$, i.e., $\Gamma(s_0 s_1 \dots s_n) = \tilde{\Gamma}(\tilde{s}_n)$.

Note that although policy $\tilde{\Gamma}$ is state-based in $\tilde{\mathcal{M}}$, its induced policy Γ may not be state-based in $\mathcal{M}$. This is because the state-space of $\tilde{\mathcal{M}}$ further contains of the state-spaces of the DRA and the information-state estimator into the state-space of $\mathcal{M}$.

B. Enforcement of Infinite-step Opacity

To enforce the security constraint, we need to make sure that the labeled MDP under control does not reveal secrets at the current and in the future time. By equations (2) and (4), it suffices to avoid those states in the product MDP at which the secret was revealed for some previous time instants. Formally, we define

$$S_{rev} = \{\tilde{s} \in \tilde{S} : \exists \eta \in D_1(X(\tilde{s})) \text{ s.t. } \eta \subseteq S_{secret}\}$$

as the set of secret-revealing states. We need to guarantee that all reachable states in the MDP under control are not secret-revealing. Given an MDP $\mathcal{M}$, we define $\mathcal{M}|_{S'}$ as the restriction of $\mathcal{M}$ to state-space $S' \subseteq S$, which is obtained by removing states not in S' and changing the related transition probability to 0.

We remove those secret-revealing states and the corresponding transitions from the product MDP $\tilde{\mathcal{M}}$ and get a new product MDP $\tilde{\mathcal{M}}_0 = \tilde{\mathcal{M}}|_{\tilde{S}\setminus S_{rev}}$. Furthermore, for any state in the product MDP, we need to make sure that

(i) there is at least one action defined at each state; and
(ii) the overall probability of transitions labeled with an action is one, i.e., $\sum_{\tilde{s}' \in \tilde{S}} \tilde{P}(\tilde{s}, a, \tilde{s}') = 1$.

We call those states satisfying the above two conditions as consistent states $S_{consist}$. Therefore, we need to iteratively removing those inconsistent states from the product MDP. We define an operater $F : \tilde{\mathcal{M}} \to \tilde{\mathcal{M}}$, as follows: for any $\tilde{\mathcal{M}}$, we have $F(\tilde{\mathcal{M}}) = \tilde{M}|_{S_{consist}}$. Removing some inconsistent states may create new inconsistent states. We define $\tilde{\mathcal{M}}^* = \lim_{k\to\infty} F^k(\tilde{\mathcal{M}}_0)$, as the resulting product MDP, which is the fixed-point of operator F. Essentially, $\tilde{\mathcal{M}}^*$ is the winning region if one wants to avoid reaching states S_{rev}.

C. Optimal Control Policy

Once we obtain MDP $\tilde{\mathcal{M}}^*$ that does not reveal any secret, we need to find the optimal control policy to obtain a solution to satisfy the second condition of Problem 1. Our approach for this part is similar to the standard probabilistic synthesis for maximizing the satisfication of an LTL formula [1]. To this end, we introduce the definition of sub-MDP, maximal end component (MEC) and accepting maximal end component (AMEC) in the following.

Definition 5: (sub-MDP) A sub-MDP of a MDP is a pair of states and action sets (M, N) where i) $\emptyset \neq M \subseteq S$ is a set of states and $N : M \to 2^A$ is a mapping such that $N(s) \subseteq A(s)$ for all $s \in M$ and ii) for all $s \in M$, we have $\{s' | P(s, a, s') > 0\} \subseteq M$.

Definition 6: (MEC) An end component is a sub-MDP (M, N) such that the underlying digraph $G_{(M,N)}$ is strongly-connected. An end component (M, N) is maximal if there exists no other $(M', N') \neq (M, N)$ and $M \subseteq M'$ such that $N(s) \subseteq N'(s)$ for all $s \in M$.

Definition 7: (AMEC) Given a product MDP $\tilde{\mathcal{M}}$, an accepting maximal end component (AMEC) of the product MDP is a maximal end component (M, N) such that there exists some $(\tilde{B}_i, \tilde{G}_i) \in \widetilde{Acc}$ such that $\tilde{B}_i \cap M = \emptyset$ and $\tilde{G}_i \subseteq M$. Given the set of AMECs $\{(M_1, N_1), \ldots, (M_K, N_K)\}$, the set of accepting states $\mathcal{E}$ can be computed as $\mathcal{E} = \cup_{i=1}^{K} M_i$.

Intuitively, an AMEC on product MDP is a set of states, in which once a state $(\tilde{B}_i, \tilde{G}_i)$ in AMEC is reached, the system remains in $\tilde{G}_i$ forever and the LTL specification is always satisfied. An AMEC fulfills the acceptance condition $\widetilde{Acc}$ which is determined by the DRA R. Therefore, once an AMEC is reached in the product MDP, the acceptance criterion for the DRA holds. The control policy $\tilde{\Gamma}$ on the product MDP that maximizes the probability of satisfying φ is the policy that maximizes the probability of reaching the set of accepting states $\mathcal{E}$. Besides, there is a one-to-one correspondence relationship between $\tilde{\Gamma}$ and Γ. We have the following Proposition.

Proposition 2: For any control policy Γ, the maximal probability satisfying the LTL formula is equal to the maximal probability of reaching $\mathcal{E}$:

$$\max_{\Gamma} Pr(\mathcal{M}_\Gamma \vDash \varphi) = \max_{\Gamma} Pr(\text{reach } \mathcal{E}). \tag{5}$$

The procedure to obtain all AMECs of an MDP can be found in [1]. The second condition of Problem 1 can be converted to find a control policy that maximizes the probability of reaching accepting states $\mathcal{E}$ in the product MDP. Such maximum probability of reaching accepting states $\mathcal{E}$ can be computed by value iteration. We denote the set of states that cannot reach $\mathcal{E}$ under any policy as $\mathcal{E}_N$. For any state $\tilde{s} \in \mathcal{E}$, the value remains 1. For any state $\tilde{s} \in \mathcal{E}_N$, the value remains 0. For the remaining state $\tilde{s}_k \in S\setminus(\mathcal{E}\cup\mathcal{E}_N)$, the value is computed by the iteration procedure as follows

$$v^{i+1}(k) = \max \left\{ \sum_{\tilde{s}_t \in \tilde{S}} \tilde{P}(\tilde{s}_k, a, \tilde{s}_t) v^i(t) | a \in A(\tilde{s}_k) \right\}, \tag{6}$$

where vector $v = (v(1), v(2), \ldots, v(|\tilde{S}|))$ is the probability of reaching $\mathcal{E}$ from each state. The initial value function is

$$v^0(k) = \begin{cases} 1 & \text{if } \tilde{s}_k \in \mathcal{E}, \\ 0 & \text{if } \tilde{s}_k \notin \mathcal{E}. \end{cases} \tag{7}$$

The value iteration will converge to the value function denoted by v^*. Obviously, $v^*(0)$ is the maximum probability of reaching $\mathcal{E}$ in the product MDP $\tilde{\mathcal{M}}$ under a policy $\tilde{\Gamma}$. Once v^* is obtained, we can compute a control policy $\tilde{\Gamma} : \tilde{S} \to A$ for each state $\tilde{s} \in \tilde{S} \setminus (\mathcal{E} \cup \mathcal{E}_N)$ by

$$v^*(k) = \sum_{\tilde{s}_t \in \tilde{S}} \tilde{P}(\tilde{s}_k, a, \tilde{s}_t) v^*(t). \tag{8}$$

If value $v^*(k)$ satisfies equation (8), then a is the chosen action at state $\tilde{s}_k$. For states in $\mathcal{E}$ or $\mathcal{E}_N$, the actions are chosen arbitrarily since it is irrelevant with the probability of reaching $\mathcal{E}$. We can get a state-based optimal control policy $\tilde{\Gamma}$ that maximizes the probability of reaching the set of accepting states $\mathcal{E}$ on the safety product MDP $\tilde{\mathcal{M}}^*$ by following the above procedures. Given an optimal control policy $\tilde{\Gamma}$ on the product MDP $\tilde{M}$, by Proposition 2, the control policy Γ on the labeled MDP $\mathcal{M}$ that solves Problem 1 can be obtained by Definition 4.

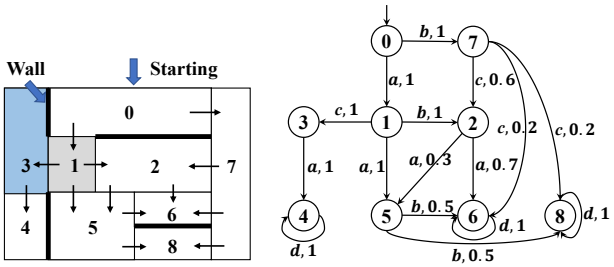
Our overall approach is summarized in Algorithm 1. Lines 1-3 construct a DRA R, the information-state estimator T and the product MDP $\tilde{\mathcal{M}}$. Lines 4-5 aim to delete all secret-revealing states and inconsistent states and to get a safety product MDP. Then we find all AMECs of the safety product MDP and compute the optimal control policy by value iteration (Lines 6-7). Finally, we can induce control policy Γ from $\tilde{\Gamma}$.

Algorithm 1: Compute the optimal control strategy Γ given $\mathcal{M}$, φ and H

input : Labeled MDP $\mathcal{M}$ with S_{secret}, output function H, LTL formula φ

output: Optimal control strategy Γ

1 Translate the LTL formula φ to a DRA R;
2 Construct the information-state estimator T;
3 Generate the product MDP $\tilde{\mathcal{M}} = \mathcal{M} \times R \times T$;
4 Delete all secret-revealing states in S_{rev} from $\tilde{\mathcal{M}}$ and get a new product MDP $\tilde{\mathcal{M}}_0 = \tilde{\mathcal{M}}|_{S \setminus S_{rev}}$;
5 Remove all inconsistent states iteratively from $\tilde{\mathcal{M}}_0$ and get a product MDP $\tilde{\mathcal{M}}^* = \lim_{k\to\infty} F^k(\tilde{\mathcal{M}}_0)$;
6 Find all AMECs (M, N) for product MDP $\tilde{\mathcal{M}}^*$ and compute the set of accepting states $\mathcal{E}$;
7 Compute the maximum probability of reaching states set $\mathcal{E}$ from $\tilde{s}_0$ and generate the optimal control policy $\tilde{\Gamma}$ for $\tilde{\mathcal{M}}$ by value iteration;
8 Induce the control policy Γ for $\mathcal{M}$ from $\tilde{\Gamma}$;



(a) (b)

Fig. 1: (a) Work space of the robot. (b) Labeled MDP $\mathcal{M}$ representing the mobility of the robot.

VI. Case Study: Security-Aware Motion Planning

Let us consider a robot moving in a workspace, which is partitioned into eight regions as shown in Fig.1(a). We assume that the robot can only move to the adjacent region without crossing a wall and the arrows represent allowable transitions between regions. The robot can take control actions such as move southwards, move eastwards and move westwards and stay, which are represented by actions a, b, c and d, respectively. Due to uncertainty, the robot may move to different regions from its current region when a control action is taken, e.g., the robot can move to region 5 and region 6 from region 2 when it moves southwards. The probability of the transition from a region to another region is related to the length of the overlap region. Therefore, the mobility as well as the transition probability of the robot can be represented by MDP $\mathcal{M}$ shown in Fig. 1(b).

The goal of the robot is to first go to region 1 to send a message to a station and then visit in region 4 or region 6 infinitely often for surveillance. Therefore, we choose atomic propositions as $\mathcal{AP} = \{P_1, P_2\}$ and define a labeling function $L : S \to 2^{\mathcal{AP}}$ by $L(1) = \{P_1\}$, $L(4) = L(6) = \{P_2\}$ and $L(s) = \emptyset$ for other states. Then task of the robot

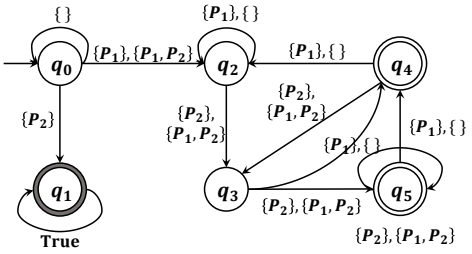


Fig. 2: A DRA R translated from $\varphi = (\neg P_2 \mathcal{U} P_1) \wedge (\square\lozenge P_2)$.

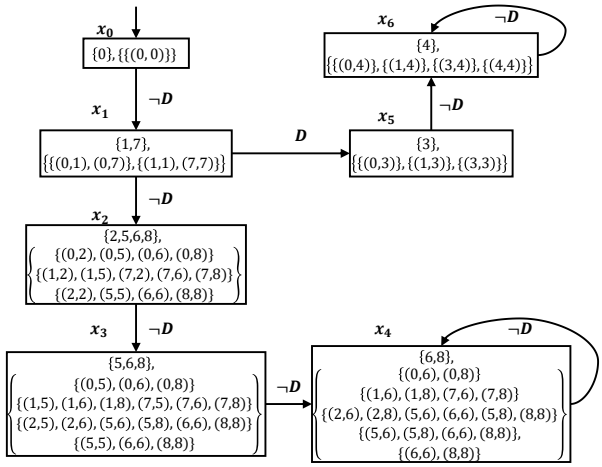


Fig. 3: The information-state estimator T.

can be expressed by the LTL formula

$$\varphi = (\neg P_2 \mathcal{U} P_1) \wedge (\square\lozenge P_2).$$

Now, suppose that there is an outside intruder that can detect whether the robot is moving from one region to another. In addition, it has a sensor to detect whether or not the robot is in region 3. Therefore, we can simply consider a binary output $Y = \{D, \neg D\}$ and the output function is given by $H(3) = D$ and $H(s) = \neg D$ for other states.

On the other hand, we assume that the robot does not want the intruder to know exactly when it visits region 1 to send the information. Otherwise, the intruder may use the time-log to check what information the robot sends to the station. Such a security requirement can be captured by infinite-step opacity by considering secret-state $S_{secret} = \{1\}$.

In order to solve the security-aware LTL synthesis problem, first we translate the LTL specification φ to a DRA R shown in Fig.2. The acceptance set of Rabin pairs is $Acc = \{(q_1, q_4), (q_1, q_5)\}$. Now we construct the information-state estimator shown in Fig.3. We generate the product MDP $\tilde{\mathcal{M}} = \mathcal{M} \times R \times T$ shown in Fig.4(a). Since $D_1(x_5) = \{\{0\}, \{1\}, \{3\}\}$, $D_1(x_6) = \{\{0\}, \{1\}, \{3\}, \{4\}\}$, and $\{1\} \subseteq S_{secret}$, x_6 and x_7 are secret-revealing states. Therefore, we have $S_{rev} = \{\tilde{s}_{19}, \tilde{s}_{20}, \tilde{s}_{21}\}$. Product MDP $\tilde{\mathcal{M}}_0$ is obtained by restricting $\tilde{\mathcal{M}}$ to $\tilde{S} \setminus S_{rev}$. After we delete secret-revealing states in the product MDP, there does not exist an inconsistent state. The resulting product MDP $\mathcal{M}^*$ after deleting all secret-revealing states is secure in the sense that all states in $\tilde{\mathcal{M}}^*$ will not reveal the secrets.

Next, we find an optimal control policy that maximizes the probability of satisfying φ in $\tilde{\mathcal{M}}^*$. An infinite run

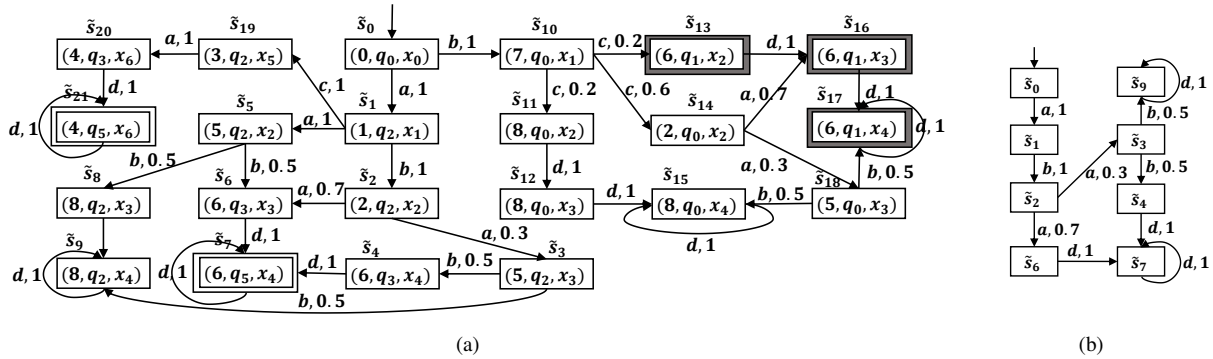


Fig. 4: (a) The product MDP $\tilde{\mathcal{M}}$ generated from $\mathcal{M}$, R and T. (b) Solution $\tilde{\Gamma}$ on the product MDP.

τ^+ of product MDP $\tilde{\mathcal{M}}^*$ is accepted if and only if τ^+ intersects with $\tilde{s}_7$ infinitely often and with $\tilde{s}_{13}$, $\tilde{s}_{16}$ and $\tilde{s}_{17}$ finitely often. The AMECs of $\tilde{\mathcal{M}}^*$ is $\{(M, N)\}$, where $M = \{\tilde{s}_4, \tilde{s}_6, \tilde{s}_7\}$. The maximal probability of satisfying φ is equal to the maximal probability of reaching M. Then, we iteratively update the value of each state. The optimal control policy in the product MDP can be computed using equation (8), e.g., since $v^*(0) = v^*(1)$ and $v^*(1) = v^*(2)$, the policy at states $\tilde{s}_0$ and $\tilde{s}_1$ are a and b, respectively. The function of the optimal control policy $\tilde{\Gamma}$ on the product MDP is shown in Fig.4(b). We can induce the control policy on the labeled MDP $\mathcal{M}$ through $\tilde{\Gamma}$ by Definition 4.

VII. Conclusion

In this paper, we formulated and solved a security-aware LTL synthesis problem for MDPs. The security constraint is captured by the notion of infinite-step opacity. A new type of information-state estimator was proposed to effectively handle the security constraint. By taking the product of the MDP, the DRA and the information-state estimator, the synthesis problem can be effectively solved by first solving a safety game w.r.t. the information-state estimator and then maximizing the probability of reaching accepting regions. In the future, we plan to further investigate the quantitative tradeoff between the probability of being secure and the probability of satisfying the LTL specification.

References

[1] C. Baier and J.-P. Katoen. *Principles of model checking*. MIT press, 2008.

[2] B. Bérard, K. Chatterjee, and N. Sznajder. Probabilistic opacity for Markov decision processes. *Information Processing Letters*, 115(1):52–59, 2015.

[3] X. Ding, S. Smith, C. Belta, and D. Rus. Optimal control of Markov decision processes with linear temporal logic constraints. *IEEE Trans. Automatic Control*, 59(5):1244–1257, 2014.

[4] J. Dubreil, P. Darondeau, and H. Marchand. Supervisory control for opacity. *IEEE Trans. Automatic Control*, 55(5):1089–1100, 2010.

[5] J. Esparza and J. Křetínskỳ. From LTL to deterministic automata: A safraless compositional approach. In *International Conference on Computer Aided Verification*, pages 192–208. Springer, 2014.

[6] M. Guo and M. Zavlanos. Probabilistic motion planning under temporal tasks and soft constraints. *IEEE Trans. Automatic Control*, 63(12):4051–4066, 2018.

[7] E.M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak. Omega-regular objectives in model-free reinforcement learning. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 395–412, 2019.

[8] Z. He and Z. Ma. Performance safety enforcement in stochastic event graphs against boost and slow attacks. *Nonlinear Analysis: Hybrid Systems*, 41:101057, 2021.

[9] Y. Ji, X. Yin, and S. Lafortune. Enforcing opacity by insertion functions under multiple energy constraints. *Automatica*, 108:108476, 2019.

[10] B. Lacerda, D. Parker, and N. Hawes. Optimal and dynamic planning for Markov decision processes with co-safe ltl specifications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1511–1516, 2014.

[11] M. Lahijanian, S. Andersson, and C. Belta. Temporal logic motion planning and control with probabilistic satisfaction guarantees. *IEEE Trans. Robotics*, 28(2):396–409, 2011.

[12] A. Lavaei, F. Somenzi, S. Soudjani, A. Trivedi, and M. Zamani. Formal controller synthesis for continuous-space mdps via model-free reinforcement learning. In *11th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 98–107, 2020.

[13] Z. Ma and K. Cai. Optimal secret protections in discrete-event systems. *IEEE Transactions on Automatic Control*, 2021.

[14] B. Ramasubramanian, L. Niu, A. Clark, L. Bushnell, and R. Poovendran. Privacy-preserving resilience of cyber-physical systems to adversaries. In *59th IEEE CDC*, pages 3785–3792, 2020.

[15] A. Saboori and C. N. Hadjicostis. Verification of infinite-step opacity and complexity considerations. *IEEE Trans. Automatic Control*, 57(5):1265–1269, 2011.

[16] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Current-state opacity enforcement in discrete event systems under incomparable observations. *Discrete Event Dynamic Systems*, 28(2):161–182, 2018.

[17] Y. Wang, S. Nalluri, and M. Pajic. Hyperproperties for robotics: Planning via hyperltl. In *IEEE International Conference on Robotics and Automation*, pages 8462–8468, 2020.

[18] Y. Xie and X. Yin. Supervisory control of discrete-event systems for infinite-step opacity. In *American Control Conference*, pages 3665–3671, 2020.

[19] Y. Xie, X. Yin, and S. Li. Opacity enforcing supervisory control using non-deterministic supervisors. *IFAC-PapersOnLine*, 53(2):1763–1769, 2020.

[20] Z. Xu, K. Yazdani, M. T. Hale, and U. Topcu. Differentially private controller synthesis with metric temporal logic specifications. In *American Control Conference*, pages 4745–4750, 2020.

[21] S. Yang, X. Yin, S. Li, and M. Zamani. Secure-by-construction optimal path planning for linear temporal logic tasks. In *59th IEEE CDC*, pages 4460–4466, 2020.

[22] X. Yin and S. Lafortune. A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Trans. Automatic Control*, 61(8):2140–2154, 2015.

[23] X. Yin and S. Lafortune. A new approach for the verification of infinite-step and k-step opacity using two-way observers. *Automatica*, 80:162–171, 2017.

[24] X. Yin, Z. Li, W. Wang, and S. Li. Infinite-step opacity and K-step opacity of stochastic discrete-event systems. *Automatica*, 99:266–274, 2019.

[25] X Yin, M Zamani, and S Liu. On approximate opacity of cyber-physical systems. *IEEE Trans. Automatic Control*, 66(4):1630–1645, 2021.