2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Security-Aware Reinforcement Learning under Linear Temporal Logic Specifications

Bohan Cui, Keyi Zhu, Shaoyuan Li and Xiang Yin

Abstract— **In this paper, we investigate the problem of reinforcement learning under linear temporal logic (LTL) specifications for Markov decision processes (MDPs) with security constraints. We consider an outside passive intruder (observer) that can observe the external output behavior of the system through an output projection. We assume that the secret of the system is a subset of the initial states. The security constraint requires that the observer can never infer for sure that the agent was initiated from a secret state. Our objective is to learn a control policy that achieves the LTL task while ensuring security. To solve the problem of shaping the reward for reinforcement learning, we propose an approach based on the initial-state estimator and the limit deterministic Büchi automata. We illustrate the proposed approach by a case study of mobile robot example.**

I. INTRODUCTION

Designing controllers for complex tasks in real world applications such as autonomous vehicles and energy systems is admittedly hard due to the difficulty of formally describing complex tasks and the inability to obtain all information of the systems. The stochastic dynamic of many systems can be modeled as Markov Decision Processes (MDPs), which suitably capture the inherent unknown dynamic and uncertainties as probabilistic transition functions. In the context of task description, Linear Temporal Logic (LTL) is one of most widely used user-friendly formal languages which can represent many important properties such as safety, liveness, and priority [1]. For controller synthesis of MDPs under LTL specification, existing methods for probabilistic model checking such as [2] have been well studied. However, probabilistic model checking approach requires that the knowledge of the transition probabilities to compute the probability measure of each path. For a more general scenario where the transition probabilities are unavailable, a popular method is Reinforcement Learning (RL).

Reinforcement learning, which has shown great potentials in various applications, focuses on finding a policy that maximizes the expected average reward or the long-term discounted reward. In reinforcement learning, the agent improves its policy by observing the state, taking an action, and receiving a reward repeatedly [3]. For the recent works on reinforcement learning for complex tasks, there are two main research directions in the past few years: formal reward shaping according to the tasks and strictly safe guarantee during learning. Algorithms have been proposed to synthesize policies that maximize the probability of satisfying the given LTL specifications; see, e.g., [4], [5], [6]. For safe reinforcement learning, recent works have presented algorithms to provide strictly safe guarantees such as obstacle avoidance during learning. Most of these algorithms modify policy before the policy is executed actually using physical [7], [8] or logical [9], [10] methods.

While the above works extensively investigated LTL-based learning and learning with safe constraints, security and privacy, which are also very important in many applications, are not fully considered yet. For example, let us consider the following scenario: A mobile robot needs to complete a moving task in an unknown environment and to send information back to the cloud. However, an outside malicious intruder may access the information flow of the system and infer the critical information which the robot does not intend to transmit. Due to the important of security and privacy in nowadays applications, this direction has been attracting attention in recent works; see, e.g., [11]–[17]. But most of these works mainly consider the secure *planning problem*, while for secure learning, the problem considered in this paper, there are few works on it.

In this paper, we investigate a security-aware learning problem under LTL specifications for systems modeled as finite MDPs. To capture the security constraints, we consider a notion of information-flow security property called *initial-state opacity* [18], [19]. This security property requires that for any paths starting from a secret initial state, there exists at least one path starting from a non-secret initial state such that these two paths are observationally equivalent from the outside intruder's point of view. In [19], authors proposed the notion of almost initial-state opacity to capture the security requirements, and here we consider the more strict initial-state opacity, i.e., sure initial-state opacity [18]. To solve the security-aware LTL synthesis problem for unknown MDPs, we present an approach that shapes the reward according to the LTL formula for reinforcement learning. Meanwhile, it can keep the secret initial-state from being leaked. The approach we presented is mostly related to the standard initial-state opacity verification [18] and the LDBA-based reward shaping [20], which is sound and complete.

In the context of the opacity verification and synthesis of discrete-event systems, several works considered the stochastic systems opacity [21], [19]. But most of existing literature still work on opacity for deterministic systems modeled by

This work was supported by the National Natural Science Foundation of China (62061136004, 62173226, 61803259) and by the National Key Research and Development Program of China (2018AAA0101700).)

B. Cui, K. Zhu, X. Yin and S. Li are with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the Key Laboratory of System Control and Information Processing, the Ministry of Education of China, Shanghai 200240, China. E-mail: {bohan_cui,bai12wp,yinxiang,syli}@sjtu.edu.cn
(Corresponding Author: Xiang Yin)

labeled transition systems; see, e.g., [22], [23], [24], [25], [26]. The strict initial-state opacity in stochastic systems and its application in reinforcement learning have not been fully studied yet.

The rest of the paper is organized as follows. The problem formulation and some necessary preliminaries are presented in Section II. In Section III, a motivating example which we consider throughout the paper is presented. We also propose an approach for the reward shaping of reinforcement learning under LTL while considering the initial-state opacity. We present the complete algorithm in Section IV and in Section V we present a case study of the motivating example and give the simulation result for our algorithm. Finally, we conclude the paper in Section VI.

II. PRELIMINARIES

A. System Model

Consider a labeled MDP which can be modeled as :

$$\mathcal{M} = (X, A, X_0, f, AP, L)$$

where x is a finite set of states; A is a finite set of actions; $x_0 \subset S$ is a finite set of initial states; $f : X \times A \times X \to [0, 1]$ is a transition function; AP is a set of atomic propositions and $L : X \to 2^{AP}$ is a labeling function. In MDP, an agent repeatedly choose actions based on a policy $\pi : Hist \to A$ which depends on the history, where $Hist \in (XA)^*X$ is the history of the system.

Given an MDP $\mathcal{M}$, a path from $x_0 \in X_0$ is a sequence $p^{\pi}_{x_0} = x[1]x[2]\cdots \in X^{\omega}$, such that $x[1] = x_0$ and policy π satisfies $f(x[i], \pi(x[1]a[1]...x[i]), x[i+1]) > 0$ for all $i \in \mathbb{N}^+$, where X^{ω} the set of all infinite sequences over X. We denote the set of all paths produced by MDP $\mathcal{M}$ as $\mathtt{Path}_{\mathcal{M}}$. For the sake of simplicity, we will omit the superscript when $\mathcal{M}$ is clear.

B. Logical Task Model

A LTL formula ϕ over a atomic propositions set AP is formed according to the following syntax:

$$\phi ::= true \,|\, a \,|\, \phi_1 \wedge \phi_2 \,|\, \neg\phi \,|\, \bigcirc \phi \,|\, \phi_1 U \phi_2$$

where $a \in AP$, $\bigcirc$ and U denote "next" and "until" respectively. Using "U" we can derive temporal modalities $\Diamond$ ("eventually") and $\Box$ ("always") as follows:

$$\Diamond\phi := true U \phi \qquad \Box\phi := \neg\Diamond\neg\phi$$

For an MDP $\mathcal{M}$, a trace is a sequence $t = L(p) = L(x[0])L(x[1])\cdots \in L(X)^{\omega}$ such that there exists a policy π satisfies $f(x[i], \pi(x[1]a[1]...x[i]), x[i+1]) > 0$ for all $i \in \mathbb{N}^+$. The set of all traces produced by MDP $\mathcal{M}$ is $\mathtt{Trace}_{\mathcal{M}}$. We denote by $t \models \phi$ if an infinite sequence t over 2^{AP} satisfies the LTL formula ϕ, for more details about the LTL syntax and semantics, readers are referred to [1].

Given an LTL task specification ϕ, if it can be achieved by a given MDP $\mathcal{M}$, then there must exist a policy $\pi^* : Hist \to A$ such that the probability that the trace generated by this policy satisfies ϕ is maximized, i.e., $\Pr(L(p^{\pi^*}_{x_0}) \models \phi) = \max(\Pr(L(p^{\pi}_{x_0}) \models \phi))$. When the transition probability is unknown, we can find this policy using reinforcement learning.

An LTL formula can be converted to a limit deterministic Büchi automata (LDBA). An LDBA is a tuple

$$\mathcal{N} = (Q, \Sigma, \delta, Q_0, F)$$

where $Q = Q_N \cup Q_D$ is the finite set of states; $\Sigma = 2^{AP} \cup \{\epsilon\}$ is the finite input alphabet; $\delta : Q \times \Sigma \to 2^Q$ is transition function; $Q_0 \subseteq Q_N$ is the set of initial states and $F = \{F_1, F_2, \ldots, F_n\}$ is the set of acceptance conditions where $F_i \subseteq Q_D$ for all $i \in \mathbb{N}^+, i \leq n$. For technique reason, LDBA should satisfy the following conditions:

- $Q_N \cap Q_D = \emptyset$;
- $\delta(q, \alpha) \subseteq Q_D$ and $|\delta(q, \alpha)| = 1$ for every state $q \in Q_D$ and for every $\alpha \in \Sigma$;
- For every $q \in Q_D$, $\delta(q, \epsilon) = \emptyset$;
- For every $q \in Q_N$, if it has ϵ-transition, then $\delta(q, \epsilon) \in Q_D$.

An infinite run of LDBA $\mathcal{N}$ is an infinite sequence $\rho = q[1]q[2]\cdots$ such that $q[1] \in Q_0$ and for any $i \in \mathbb{N}^+$, $\exists \sigma \in \Sigma \cup \{\epsilon\}$, $q[i+1] \in \delta(q[i], \sigma)$. We denote by $\mathtt{inf}(\rho)$ the set of states be visited infinite number of times in ρ. If for all $i \in \mathbb{N}^+, i \leq n$, $\mathtt{inf}(\rho) \cap F_i \neq \emptyset$, we say that the run ρ is accepted by LDBA $\mathcal{N}$. Given an LTL formula ϕ, we denote the corresponding LDBA as $\mathcal{N}_{\phi}$.

C. Intruder Model and Initial-state Opacity

In this paper, we consider a more concrete type of security the notion of initial-state opacity to capture the security requirement. Given an MDP $\mathcal{M} = (X, A, X_0, f, AP, L)$, assume that there is a set of secret states $X_s \subset X_0$ and the intruder can observe the output sequence of the system, which is produced by an output function $H : X \to Y$, where Y is the set of outputs, H can be extended to $H : X^{\omega} \to Y^{\omega}$ recursively defined by $\forall p = x[1]x[2]\cdots \in X^{\omega}, H(p) = H(x[1])H(x[2])\cdots$.

In the existing notions of initial-state opacity, the system is said to be initial-state opaque if for any path starting from $x[1] \in X_s$, there exists another path starting from $x'[1] \in X_0 \setminus X_s$ such that $H(p) = H(p')$. In this paper, we have the following definition:

Definition 1: An MDP $\mathcal{M}$ is said to be initial-state opaque from $x_0 \in X_S$ under policy π if:

$$\begin{aligned} &(\forall p = x[1]x[2]\cdots \in \mathtt{Path} : \\ &x[1] = x_0, f(x[i], \pi(x[1]a[1]...x[i]), x[i+1]) > 0) \\ &(\exists t = x'[1]x'[2]\cdots \in \mathtt{Path} : x'[1] \in X_0 \setminus X_s) \\ &[H(s) = H(t)] \end{aligned} \tag{1}$$

D. Problem Formulation

After giving the necessary preliminaries, we are now ready to formulate the problem we solve in this work as follows:

Problem 1: (*Security-Aware Reinforcement Learning*) Given an MDP $\mathcal{M}$ with secret states $X_S \subset X_0$, the initial state $x_0 \in X_S$ and an LTL task ϕ, learning an secure optimal strategy π^* such that:

- $\mathcal{M}$ is initial-state opaque from x_0 under policy π^*.
- $\texttt{Pr}(L(p_{x_0}^{\pi^*}) \models \phi)$ is maximized.

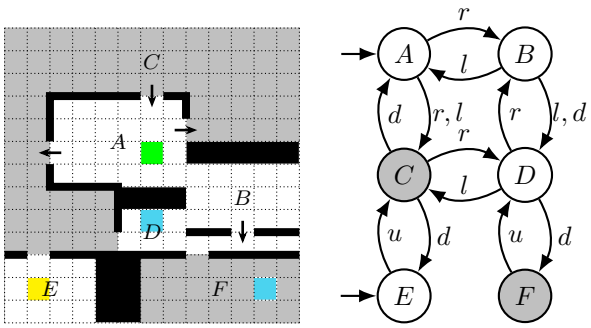
III. Security-aware Reinforcement Learning

A. Motivating Example

Consider a robot moving in a house as shown in Figure 1(a). The robot is released from the green region and it needs to visit the two blue regions infinitely often, meanwhile, the biggest room should only be visited finite number of times for the consideration of utility. There are one-way doors (the feasible direction is shown as the arrow) and two-way corridors between each rooms. The robot has four actions to take, which we denote as "r (right)", "l (left)", "u (up)" and "d (down)". Suppose that the robot does not know its precise location in each room and there are tiny motion errors when it takes action. As an example, when the robot is in the initial room A and choose to move right, there are two possible following rooms it may enter in, the dark room C (gray) or the light room B (white). Since the robot only knows the rough location (which room it is in) and the uncertainty caused by mechanical structure or information loss, it is reasonable to train it first by simulation and then conduct the final policy in the real world.

Assume that there is an outside passive intruder who wants to determine from which room (the yellow state or the green one) the robot is released. There are two kind of sensors in each room, when the robot enter, sensors send message to the intruder. The sensors in dark room are only able to send message "g", and the sensors in light rooms can only send "w". Based on the output of the system, the intruder can infer the possible initial states the robot released from. It requires that the final policy should also keep initial-state opaque.

To solve this problem, We can translate the workspace to an MDP as shown in Figure 1(b), the robot is released from room A. The user wants it to visit state D and F infinitely often and only visit C finite number of times. For example, on state A, the robot can choose action r or l, if it chooses a, then the robot may go to B or C, and the transition probability is unknown for the robot. The intruder knows that the robot may be released from state A or E and the user do not want him to know that the robot started from A. Now, suppose that the robot is released from A. One possible path in this system is $(A \xrightarrow{r} B \xrightarrow{d} D \xrightarrow{l} C)^\omega$. However, this path is not secure because the intruder can infer from the output $(w \to w \to w \to g)^\omega$ that the robot is started from A. The reason is that there is no feasible way to generate the same observation sequence from state E. On the other hand, one possible secure path is $(A \xrightarrow{l} C \xrightarrow{r} D \xrightarrow{r} B)^\omega$ because there is another possible path $E \xrightarrow{u} (C \xrightarrow{r} D \xrightarrow{r} B \xrightarrow{l} A)^\omega$ starting from state E that has the same output sequence $(w \to g \to w \to w)^\omega$ as above. By analogy, one possible final path which not only satisfies LTL specification but also initial-state opaque is $(A \xrightarrow{l} C)^* \xrightarrow{r} (D \xrightarrow{d} F)^\omega$ where the superscript $*$ means the finite repetition of the transition.



(a) Workspace of the robot. (b) MDP model of the robot.

Fig. 1. A motivating example.

B. Security Perception

As we discussed above, the final policy should satisfy the security specification which is modeled as initial-state opacity. One feasible way to meet this requirement is to find all of the paths in this system which may leak the secret to the intruder. After that, we can disable all of the actions which may lead the system to these unsecure paths. To this end, one of the key ingredients of the algorithm in this paper is to track the information flow of the intruder, one complete approach is to construct the whole initial state estimator (ISE) of the MDP.

Given MDP $\mathcal{M} = (X, A, X_0, T, AP, L)$, given the set of secret states $X_s \subset X_0$ and the output function $H : X \to Y$, the initial-state estimate after observe string $H(p) \in Y^*$ is defined as:

$$\hat{X}_0(H(p)) = \left\{ x[1] \in X_0 : \begin{array}{c} \exists p' = x[1]\cdots x[N] \in \overline{\texttt{Path}_\mathcal{M}} \text{ s.t.} \\ H(p') = H(p) \end{array} \right\}$$

where $\texttt{Path}_\mathcal{M}$ is the set of paths that can be generated by MDP $\mathcal{M}$ and $\overline{\texttt{Path}_\mathcal{M}}$ is the set of all the prefix of $\texttt{Path}_\mathcal{M}$. Note that the path p' can be any path that the MDP can produce without considering the specific policy since the observer does not know the robot's policy.

However, straightly storing all of the observation sequences is impossible since the numbers of the observation can be very large, so we have to map the observation sequences to a finite structure. For a finite set of states X, we define the operator $\odot : X \to X^2$ to represent the replication of the states $X \odot X := \{(x, x) | x \in X\}$. To define the state transition in initial-state estimator, we define the composition operator $\circ : 2^{X^2} \times 2^{X^2} \to 2^{X^2}$ for $m_1, m_2 \in 2^{X^2}$ as $m_1 \circ m_2 := \{(x_1, x_3) | \exists x_2 \in X, (x_1, x_2) \in m_1, (x_2, x_3) \in m_2\}$, where m is the state mapping relation. Finally, we define the mapping $O : Y^* \to 2^{X^2}$ as

$$O(\alpha) = \left\{ (x[1], x[N]) \in X^2 : \begin{array}{c} \exists p' = x[1]\cdots x[N] \in \overline{\texttt{Path}_\mathcal{M}} \text{ s.t.} \\ H(p') = \alpha \end{array} \right\}$$

$O(\alpha)$ represents the set of state mapping of the head and

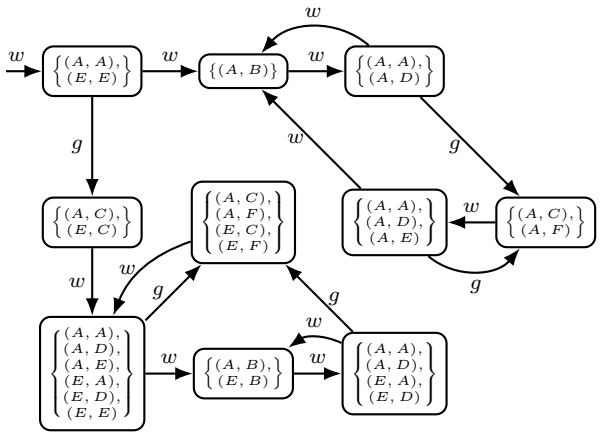


Fig. 2. The ISE of system in Figure 1

tail of the paths that produce the observation sequence $\alpha \in H(\overline{\texttt{Path}_{\mathcal{M}}})$.

Definition 2: (*Initial-State Estimator*) Given an MDP $\mathcal{M} = (X, A, X_0, T, AP, L)$, secret states $X_s \subset X_0$ and output function $H : X \to Y$, we define the initial-state estimator as:

$$\text{ISE}(\mathcal{M}) = (M, Y, \delta_{\text{ise}}, M_0)$$

where

- $M : 2^{X^2}$ is the finite set of states;
- Y is the finite set of outputs;
- $M_0 : 2^{X_0 \odot X_0}$ is the set of initial states, for any $m_0 \in M_0$, if $(x_{10}, x_{10}) \in m_0$ and $(x_{20}, x_{20}) \in m_0$, then $H(x_{10}) = H(x_{20})$ and vise verse;
- $\delta_{\text{ise}} : M \times Y \to M$ is the transition function defined by: for any $m, m' \in M, y \in Y$, denote that $m = (x_1, x_2)$,

  $$m' = \delta_{\text{ise}}(m, y) := (m \circ O(H(x_2)y))$$

  for $m \in M_0$, we define that

  $$\delta_{\text{ise}}(\varepsilon, y) = \{(x_0, x_0) \in X_0 \odot X_0 : H(x_0) = y\},$$

  where ε represents the empty state.

Intuitively, for all the states $m \in M$ of ISE($\mathcal{M}$), the set of first components of m is the initial state estimation $\hat{X}_0$, and the set of second components is the current state estimation.

Example 1: Consider the MDP shown in Figure 1(b), We show the initial-state estimator for this MDP in Figure 2. We assume the initial uncertainty is equal to the initial state space so $m_0 = \delta_{\text{ise}}(\varepsilon, w) = \{(A, A), (E, E)\}$. After observing g, the state goes to $m_1 = \delta_{\text{ise}}(m_0, g) = (m_0 \circ O(w \to g)) = \{(A, C), (E, C)\}$. Analogously on the other branch, after observing w, the state goes to $m_2 = \delta_{\text{ise}}(m_0, w) = (m_0 \circ O(w \to w)) = \{(A, B)\}$. Continuing this process, we can completely construct the ISE of system in Figure 1 which is shown in Figure 2.

C. Secure Learning under LTL

Remark that we only discuss the verification of initial-state opacity above. However, in this paper, the requirement is that preserve the secret against the intruder and complete the LTL task, i.e., synthesis the controller for MDP under both secure constraint and LTL task.

To this end, after the construction of initial-state estimator ISE($\mathcal{M}$) $= (M, Y, \delta_{\text{ise}}, M_0)$, we need to compose it with the original MDP $\mathcal{M} = (X, A, X_0, f, AP, L)$ and the LDBA $\mathcal{N}_\phi = (Q, \Sigma, \delta, Q_0, F)$, which is defined as follows:

Definition 3: (*Product MDP*) Given MDP $\mathcal{M} = (X, A, X_0, f, AP, L)$, initial-state estimator ISE($\mathcal{M}$) $= (M, Y, \delta_{\text{ise}}, M_0)$ and LDBA $\mathcal{N}_\phi = (Q, \Sigma, \delta, Q_0, F)$, the product MDP is

$$\mathcal{P} = \mathcal{M} \times \text{ISE}(\mathcal{M}) \times \mathcal{N}_\phi = (X_p, A_p, X_{p,0}, f_p, AP, L_p, Acc)$$

where

- $X_p = \{(m, x, q) \in M \times X \times Q\}$ is the finite set of states;
- $A_p = A \cup \{\eta\}$ is the finite set of actions;
- $X_{p,0} = \{(m_0, x_0, q_0) \in M_0 \times X_0 \times Q_0 : m_0 = \delta_{\text{ise}}(\varepsilon, H(x_0))\}$ is the set of initial states;
- $f_p : X_p \times A_p \times X_p \to [0, 1]$ is the transition function defined by: For any $(m, x, q), (m', x', q') \in X_p$ and $a \in A$, if $m' = \delta_{\text{ise}}(m, H(xx')), q' \in \delta(q, L(x'))$, we have

  $$f_p((m, x, q), a, (m', x', q')) = f(x, a, x')$$

  For $a = \eta$, if $m' = m, x' = x, q' \in \delta(q, \epsilon)$, we have

  $$f_p((m, x, q), \eta, (m', x', q')) = 1,$$

  Otherwise

  $$f_p((m, x, q), a, (m', x', q')) = 0;$$

- AP is the set of atomic propositions;
- $L_p((m, x, q)) = L(x)$ is the labeling function;
- $Acc = \{Acc_1, Acc_2, ..., Acc_n\}$ is the set of acceptance conditions where $Acc_i = M \times X \times F_i$ for all $i \in \mathbb{N}^+, i \leq n$.

To make sure the system is initial-state opaque, first, we define the set of secret-revealing states as following:

$$X_{p,\text{rev}} = \{(m, x, q) \in X_p : \forall (x_1, x_2) \in m, x_1 \in X_S\}.$$

We have the following proposition:

Proposition 1: $\mathcal{P}$ is initial state opaque (w.r.t. X_S and H) iff $X_{p,rev} = \emptyset$.

Proof: The proof is omitted for the sake of page limit. ■

Intuitively, if the initial state estimation is the subset of X_S, then the intruder can determine for sure that the system is started from the secret states. Thus, the states in $X_{p,\text{rev}}$ should not be visited in the final policy. We define secure product MDP (SPMDP) $\mathcal{P}_{\text{safe}} = (X_{\text{safe}}, A_{\text{safe}}, X_{\text{safe},0}, f_{\text{safe}}, AP, L_p, Acc)$ obtained from $\mathcal{P}$ by removing the states in $X_{p,\text{rev}}$, disabling the corresponding actions on the former states and rewriting the transition function for each state to make the sum equals to 1. However, some states in $\mathcal{P}_{\text{safe}}$ may have no outgoing transition after removing states in $X_{p,\text{rev}}$, we call them inconsistent states. Therefore, we need to iteratively remove the inconsistent

states and disable the corresponding actions until all states are consistent.

To meet the requirement of LTL, the final policy should visit the states in Acc infinitely often. To this end, we define the augmented secure product MDP as follows:

Definition 4: (*Augmented SPMDP*) Given a secure product MDP $\mathcal{P}_{\text{safe}} = (X_{\text{safe}}, A_{\text{safe}}, X_{\text{safe},0}, f_{\text{safe}}, AP, L_p, Acc)$ and a constant $\theta \in [0,1]$, the augmented secure product MDP is

$$\mathcal{P}^{\theta}_{\text{safe}} = (X^{\theta}_{\text{safe}}, A_{\text{safe}}, X_{\text{safe},0}, f^{\theta}_{\text{safe}}, Acc, W)$$

where

- $X^{\theta}_{\text{safe}} = X_{\text{safe}} \cup \{t\}$ is the finite set of states, t is an augmented absorbing state with a self-loop.
- $f^{\theta}_{\text{safe}} : X^{\theta}_{\text{safe}} \times A_{\text{safe}} \times X^{\theta}_{\text{safe}} \to [0,1]$ is the transition function defined by: For any $(m,x,q),(m',x',q') \in X^{\theta}_{\text{safe}}$ and $a \in A_{\text{safe}}$, if $(m,x,q) \in Acc$, we have

$$f^{\theta}_{\text{safe}}((m,x,q),a,(m',x',q')) = \theta \times f_{\text{safe}}((m,x,q),a,(m',x',q'))$$
$$f^{\theta}_{\text{safe}}((m,x,q),a,t) = 1-\theta$$

  Otherwise,

$$f^{\theta}_{\text{safe}}((m,x,q),a,(m',x',q')) = f_{\text{safe}}((m,x,q),a,(m',x',q'))$$

- $W : X^{\theta}_{\text{safe}} \to \{0,1\}$ is a reward function which is defined as follows:

$$W(x^{\theta}_{\text{safe}}) = \begin{cases} 1 & if\ x^{\theta}_{\text{safe}} \in t \\ 0 & otherwise \end{cases}$$

Since $\mathcal{P}^{\theta}_{\text{safe}}$ contains all the information about the history we concern for the security-aware reinforcement learning problem, the optimal strategy π^* should be stationary on $\mathcal{P}^{\theta}_{\text{safe}}$, which means that we can reduce the policy from $\pi : Hist \to A$ to $\pi : X^{\theta}_{\text{safe}} \to A_{\text{safe}}$. Let $q(x_{\text{safe},0},\pi) = lim_{N\to\infty}\frac{1}{N}\mathbb{E}\{\Sigma_{0\leqslant i\leqslant N} W(x^{\theta}_{\text{safe},i})\}$ be the expected average reward from $x_{\text{safe},0} \in X_{\text{safe},0}$ by using policy π, we have the following main results:

Theorem 1: There exists a threshold $\theta^* \in [0,1]$ such that for all $\theta > \theta^*$, the policy π^* that maximize the expected average reward $q(x_{\text{safe},0},\pi)$ is an secure optimal policy for the original MDP $\mathcal{M}$.

Proof: In terms of security concern, $\mathcal{M}$ is initial-state opaque under any policy π in $\mathcal{P}_{\text{safe}}$ since we have already deleted the secret-revealing states and disabled the corresponding actions when constructing $\mathcal{P}_{\text{safe}}$. Thus, the secure optimal policy for the original MDP $\mathcal{M}$ is the optimal policy for $\mathcal{P}_{\text{safe}}$.

As for the optimality, we have that the expected average reward $q(x_{\text{safe},0},\pi)$ in $\mathcal{P}^{\theta}_{\text{safe}}$ is equal to the probability of reaching t since the unit reward is 1 or 0. Thus, policy π^* also maximize the probability of reaching t. We note that for any $\theta \in [0,1]$ the probability of reaching t is greater than the probability that the trace is accepting because the unaccepting trace still have a positive probability of reaching t if it contains accepting states. Thus, we have $q(x_{\text{safe},0},\pi) \geq \Pr(L(p^{\pi}_{x_{\text{safe},0}}) \models \phi)$. Moreover, we claim that $\Pr(L(p^{\pi}_{x_{\text{safe},0}}) \models \phi) \geq q(x_{\text{safe},0},\pi) - (1-\theta)\times n(x_{\text{safe},0},\pi)$, where $n : X_{\text{safe},0} \times \Pi \to \mathbb{N}$ is the expected number of visiting accepting state before reaching an strong connect component(SCC). It is because that the probability that an unaccepting trace reaching t without reaching any SCC is at most $(1-\theta)\times n(x_{\text{safe},0},\pi)$, and then the probability that the trace is rejected is $1 - \Pr(L(p^{\pi}_{x_{\text{safe},0}}) \models \phi)$, which is at most $1 - q(x_{\text{safe},0},\pi) + (1-\theta)\times n(x_{\text{safe},0},\pi)$. Thus, we have $q(x_{\text{safe},0},\pi) \geq \Pr(L(p^{\pi}_{x_{\text{safe},0}}) \models \phi) \geq q(x_{\text{safe},0},\pi) - (1-\theta)\times n(x_{\text{safe},0},\pi)$. Then, let π^1 be the optimal policy of secure product MDP $\mathcal{P}_{\text{safe}}$, let π^2 be the optimal one except π^1. We claim that if we pick θ^* such that $(1-\theta^*)\times \max_{\pi} n(x_{\text{safe},0},\pi) < \Pr(L(p^{\pi^1}_{x_{\text{safe},0}}) \models \phi) - \Pr(L(p^{\pi^2}_{x_{\text{safe},0}}) \models \phi)$ then π^* is optimal for $\mathcal{P}_{\text{safe}}$.

By contradiction, we suppose that π^* is optimal for $\mathcal{P}^{\theta}_{\text{safe}}$ but not optimal for $\mathcal{P}_{\text{safe}}$. From the above discussion, we have that

$$\Pr(L(p^{\pi^*}_{x_{\text{safe},0}}) \models \phi) \leq q(x_{\text{safe},0},\pi^*) \leq$$
$$\Pr(L(p^{\pi^*}_{x_{\text{safe},0}}) \models \phi) + (1-\theta)\times n(x_{\text{safe},0},\pi^*) <$$
$$\Pr(L(p^{\pi^*}_{x_{\text{safe},0}}) \models \phi) + (1-\theta^*)\times \max_{\pi} n(x_{\text{safe},0},\pi) <$$
$$\Pr(L(p^{\pi^*}_{x_{\text{safe},0}}) \models \phi) + \Pr(L(p^{\pi^1}_{x_{\text{safe},0}}) \models \phi) -$$
$$\Pr(L(p^{\pi^2}_{x_{\text{safe},0}}) \models \phi) \leq$$
$$\Pr(L(p^{\pi^1}_{x_{\text{safe},0}}) \models \phi) \leq q(x_{\text{safe},0},\pi^1)$$

Thus we have that $q(x_{\text{safe},0},\pi^*) < q(x_{\text{safe},0},\pi^1)$, which is a contradiction. Then we get that any policy π^* that is optimal in $\mathcal{P}^{\theta}_{\text{safe}}$ with $\theta > \theta^*$ is also optimal in $\mathcal{P}_{\text{safe}}$. The proof is now complete.

■

IV. Algorithm Overview

In this section, we conclude the steps mentioned above into Algorithm 1. In the beginning, we construct the augmented SPMDP from the original MDP, the ISE and LDBA. Then we apply Q-Learning the this augmented SPMDP. It should be noted that, while training, an episode will not always end when firstly reaching one of the accepting states, compared with the situation in ScLTL, because LTL requires repeatedly visiting the accepting states. So, an episode will only end when reaching the state set t in the augmented SPMDP, which ensures repeatedly visiting the accepting states in the LDBA.

In Algorithm 1, we introduced the whole algorithm. To explain, we have $Q : X^{\theta}_{\text{safe}} \times A_{\text{safe}} \to R$, which measures the potential future reward the agent will obtain when choosing action a at state x. During training, selecting the action $a \in A_{\text{safe}}(x^{\theta}_{\text{safe}})$ is biased by $q(x^{\theta}_{\text{safe}},:)$, which means all q value available at state x^{θ}_{safe}, i.e. $q(x^{\theta}_{\text{safe}},:) = \{q(x^{\theta}_{\text{safe}},a), a \in A_{\text{safe}}(x)\}$ (Line 14) The q value will be updated incrementally in one episode (Line 16), where $\gamma \in [0,1]$ is a discount factor, the greater γ is, the more attention to "long term interest" the agent will pay. Note that in practice, we can not mimic the case that the robot gets infinite reward. Instead,

Algorithm 1 Security-Aware Q-Learning

Input: Augmented SPMDP $\mathcal{P}_{\text{safe}}$, learning rate α, discount factor γ, max episode $episode_{max}$, max iteration per episode i_{max}

Output: An optimal positional strategy $\pi(x^{\theta}_{\text{safe}})$ which maximizes the probability of satisfying the LTL specification while preserve the initial-state opacity

for all $x^{\theta}_{\text{safe}} \in X^{\theta}_{\text{safe}}$ and $a \in A_{\text{safe}}$ **do**
  if $x^{\theta}_{\text{safe}} \in t$ **then**
    $q(x^{\theta}_{\text{safe}}, a) = M$
  else if $x^{\theta}_{\text{safe}} \in Acc$ **then**
    $q(x^{\theta}_{\text{safe}}, a) = M \times (1-\theta)$
  else
    $q(x^{\theta}_{\text{safe}}, a) = 0$
  end if
end for
$episode = 0$
while $episode \leq episode_{max}$ **do**
  $x^{\theta}_{\text{safe}} = (m_0, x_0, q_0)$, $i = 0$
  while $i \leq i_{max}$ **do**
    $a = Sample(A_{\text{safe}}(x^{\theta}_{\text{safe}}), q(x^{\theta}_{\text{safe}}, :))$
    $x^{\theta}_{\text{safe}}{}' = Transition(x^{\theta}_{\text{safe}}, a)$
    $q(x^{\theta}_{\text{safe}}, a) = (1-\alpha)q(x^{\theta}_{\text{safe}}, a) + \alpha[W(x^{\theta}_{\text{safe}}{}') + \gamma z]$
    where $z = \max_{a' \in A_{\text{safe}}(x^{\theta}_{\text{safe}}{}')} q(x^{\theta}_{\text{safe}}{}', a')$
    if $x^{\theta}_{\text{safe}}{}' \in t$ **then**
      $break$
    end if
    $x^{\theta}_{\text{safe}} = x^{\theta}_{\text{safe}}{}'$, $i = i + 1$
  end while
  $episode = episode + 1$
end while
for all $x^{\theta}_{\text{safe}} \in X^{\theta}_{\text{safe}}$ **do**
  $\pi(x^{\theta}_{\text{safe}}) = \underset{a \in A_{\text{safe}}(x^{\theta}_{\text{safe}})}{\arg\max}\ q(x^{\theta}_{\text{safe}}, a)$
end for

we set a very large reward in state t and renew the episode. As the result, the robot tends to complete the task with θ tends to 1. When the maximum episode is accomplished, the optimal strategy at state x^{θ}_{safe} is the action a which maximizes $q(x^{\theta}_{\text{safe}}, a)$, and thus, maximize $\Pr(L(p^{\pi^*}_{\mathcal{M}}) \models \phi)$.

V. CASE STUDY

In this section, we come back to the illustrative example in Section III, to shape the reward function according to the LTL specification, let the set of atomic propositions be $AP = \{\text{RoomD}, \text{RoomF}, \text{RoomC}\}$ and define that $L(D) = \text{RoomD}$, $L(F) = \text{RoomF}$, $L(C) = \text{RoomC}$ and $L(x) = \emptyset$ for any other states. Then the task of the robot can be expressed as

$$\phi = \Box\Diamond\text{RoomD} \wedge \Box\Diamond\text{RoomF} \wedge \Diamond\Box\neg\text{RoomC}$$

Then we can get an LDBA translated from this LTL formula as shown in Figure 3.For the sake of simplicity, we let "D", "F" and "C" denote "RoomD", "RoomF" and "RoomC" respectively. Note that we omit the absorbing trap state q_4 with the ingoing transitions from q_0, q_1, q_2 under the events $\{C\}, \{D, C\}, \{F, C\}, \{D, F, C\}$. The acceptance condition is $Acc = \{q_0\}$. Since the robot starts from A, then in state $(\{(A, A), (E, E)\}, A, q_3)$, only b are allowed since action a may lead to the secret leak and therefore is disabled. Then we can find a strategy that maximizes the. We have implemented our algorithm in robot simulator V-REP[1]. As shown in Figure 4, to ensure the initial-state opacity, the robot have to go to room C at the first step. After that, it can move arbitrarily, since it has a little possibility to get infinite reward by visiting room D and room F, it will tend to visit room D and room F infinitely often in the final policy.

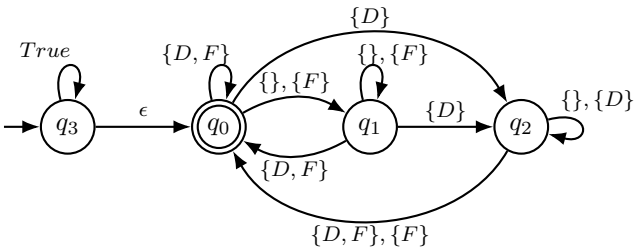


Fig. 3. LDBA translated from $\Box\Diamond$RoomD $\wedge$ $\Box\Diamond$RoomF $\wedge$ $\Diamond\Box\neg$RoomC.

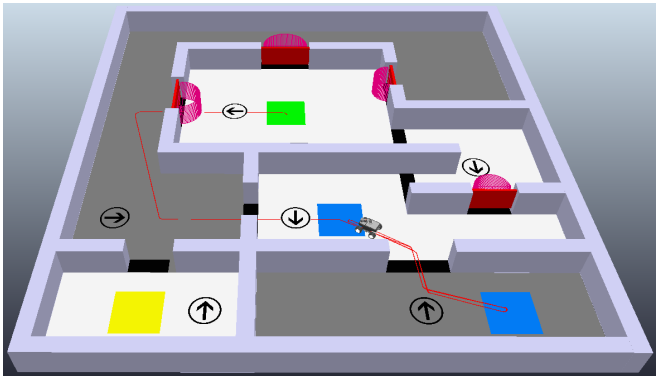


Fig. 4. Partial of the secure path generated by Algorithm 1.

VI. CONCLUSION

In this paper, we formulated and solved a security-aware reinforcement learning problem under LTL. The security constraint is captured by the notion of initial-state opacity. A standard approach was proposed to solve this problem, which is based on the construction of ISE and then take the product with LDBA to effectively solve the reward shaping problem. Eventually, it is proved by simulation that we can find a secure optimal policy with standard MDP algorithm using this structure. Note that in this paper we considered initial-state opacity for MDPs. In future work, we plan to further investigate other types of security properties (e.g., anonymity and detectability) using some new structures.

[1]Videos are available at https://youtu.be/kHYls8qBHIc

REFERENCES

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

[2] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of markov decision processes with linear temporal logic constraints," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1244–1257, 2014.

[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[4] A. K. Bozkurt, Y. Wang, M. M. Zavlanos, and M. Pajic, "Control synthesis from linear temporal logic specifications using model-free reinforcement learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 349–10 355.

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

[6] M. Wen, R. Ehlers, and U. Topcu, "Correct-by-synthesis reinforcement learning with temporal logic constraints," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 4983–4990.

[7] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3387–3395.

[8] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Transactions on Automatic Control*, 2020.

[9] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[11] S. Yang, X. Yin, S. Li, and M. Zamani, "Secure-by-construction optimal path planning for linear temporal logic tasks," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 4460–4466.

[12] L. Li, A. Bayuelo, L. Bobadilla, T. Alam, and D. A. Shell, "Coordinated multi-robot planning while preserving individual privacy," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2188–2194.

[13] Y. Wang, S. Nalluri, and M. Pajic, "Hyperproperties for robotics: Planning via hyperltl," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8462–8468.

[14] Z. Xu, K. Yazdani, M. T. Hale, and U. Topcu, "Differentially private controller synthesis with metric temporal logic specifications," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 4745–4750.

[15] Y. Xie, X. Yin, S. Li, and M. Zamani, "Secure-by-construction controller synthesis for stochastic systems under linear temporal logic specifications," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 7015–7021.

[16] X. Yu, X. Yin, S. Li, and Z. Li, "Security-preserving multi-agent coordination for complex temporal logic tasks," *Control Engineering Practice*, vol. 123, p. 105130, 2022.

[17] S. Liu, A. Trivedi, X. Yin, and M. Zamani, "Secure-by-construction synthesis of cyber-physical systems," *Annual Reviews in Control*, 2022.

[18] A. Saboori and C. N. Hadjicostis, "Verification of initial-state opacity in security applications of discrete event systems," *Information Sciences*, vol. 246, pp. 115–132, 2013.

[19] C. Keroglou and C. N. Hadjicostis, "Initial state opacity in stochastic des," in *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2013, pp. 1–8.

[20] 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*. Springer, 2019, pp. 395–412.

[21] B. Bérard, K. Chatterjee, and N. Sznajder, "Probabilistic opacity for markov decision processes," *Information Processing Letters*, vol. 115, no. 1, pp. 52–59, 2015.

[22] J. Dubreil, P. Darondeau, and H. Marchand, "Supervisory control for opacity," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1089–1100, 2010.

[23] A. Saboori and C. N. Hadjicostis, "Opacity-enforcing supervisory strategies via state estimator constructions," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1155–1165, 2011.

[24] X. Yin and S. Lafortune, "A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2140–2154, 2015.

[25] Y. Xie and X. Yin, "Supervisory control of discrete-event systems for infinite-step opacity," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 3665–3671.

[26] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Current-state opacity enforcement in discrete event systems under incomparable observations," *Discrete Event Dynamic Systems*, vol. 28, no. 2, pp. 161–182, 2018.