Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.079684

ARTICLE

A Deception Defense Timing Selection Method Based on Time-Delayed FlipIt Game in Cloud-Edge Collaborative Networks

Jinchuan Pei[1], **Yuxiang Hu**[1,2,3,4,*], **Hongtao Yu**[1], **Zihao Wang**[1] and **Menglong Li**[1,2,3,4]

[1]Information Engineering University, Zhengzhou, China
[2]Key Laboratory of Cyberspace Endogenous Security of Henan Province, Zhengzhou, China
[3]Key Laboratory of Cyberspace Security Ministry of Education of China, Zhengzhou, China
[4]National Key Laboratory of Advanced Communication Networks, Shijiazhuang, China

*Corresponding Author: Yuxiang Hu. Email: huyuxiangchn@163.com

Received: 26 January 2026; Accepted: 25 March 2026; Published: 08 May 2026

ABSTRACT: In the cloud-edge collaborative network, advanced persistent threats (APTs) pose a serious security risk to critical network assets. Although network deception defense can mislead attackers' cognition, its effectiveness depends on dynamically selecting appropriate rotation timings of the deception defense. However, the deployment of deception resources and state updates is not completed instantaneously, and existing methods ignore the state transition delay and the dynamic interaction between the attackers and defenders during the real attack and defense process. To address this, we propose a deception defense timing selection method based on the time-delayed FlipIt game. Firstly, a network state evolution model integrating state transition delay is constructed, and the dynamic transfer process between node states is characterized by a set of delay differential equations. Secondly, a cloud-edge collaborative defense architecture is designed. On this basis, a time-delayed FlipIt game model (TD-FlipIt) is established, and the gate control mechanism is introduced to formalize the defense cooling period as a constraint for the rotation action of deception resources. Subsequently, we use the multi-agent deep deterministic policy gradient (MADDPG) algorithm to solve the rotation strategy for deception defense timing. Experimental results show that the proposed method can effectively optimize the selection of defense timing, ensuring defense effectiveness while reducing resource consumption, and providing effective support for defense in the cloud-edge collaborative environment.

KEYWORDS: Cloud-edge collaborative network; deception defense timing; FlipIt game; multi-agent reinforcement learning

1 Introduction

With the deep integration of cloud computing and edge computing, cloud-edge collaboration [1] has become an important infrastructure supporting key fields such as the Internet of Things and Industrial Internet. By distributing computing, storage, and processing capabilities to the network edge, it effectively improves service response efficiency and overall system flexibility [2]. However, this also expands the attack surface [3], creating opportunities for highly visible, multi-stage attacks such as Advanced Persistent Threats (APT). Attackers usually have the characteristics of long-term lurking and continuous reconnaissance, and seize control after obtaining information about the target system [4]. This poses a serious security threat to critical assets under the cloud-edge collaboration network.

In the face of such increasingly complex and intelligent cyber attacks, traditional defense mechanisms based on static rules or immediate responses have been found inadequate to effectively counterattack

behaviors such as APT, which have strong concealment and staged evolution characteristics [5]. Against this backdrop, network deception defense technologies, such as honeypots and decoy services [6], by actively interfering with the attackers' cognition and delaying their infiltration process, have become the core components of a dynamic active defense system.

However, the effectiveness of deception defense relies heavily on the timing of defensive actions [7]. Especially in cloud-edge collaborative networks, edge nodes have limited resources, and the network status changes dynamically. The rationality of the defense timing selection directly affects the overall security effectiveness and resource consumption. If the rotation timing strategy of deception defense remains unchanged for a long time, attackers can gradually identify the deceptive assets through continuous reconnaissance, reducing the effectiveness of defense. However, frequent rotations may lead to excessive resource consumption.

The current research still has deficiencies in the decision-making of deception defense timing. On one hand, existing cybersecurity dynamic models such as SIS and SIR [8,9] generally assume that the transition of security states is an instantaneous process, ignoring the time-delay effects that are ubiquitous in real attack and defense operations [10]. For example, the deployment of deceptive assets, the penetration and spread of attacks, etc., all require a certain amount of time to complete. These time-delay factors not only affect the timeliness of the defense response but may also be exploited by attackers, such as launching precise attacks during the defense reset or strategy update window. On the other hand, the time decision-making methods of proactive defense that adopt fixed cycles or random triggering mechanisms lack the modeling of the dynamic game relationship between the attacker and the defender [11], resulting in the defense logic becoming static and difficult to adapt to the real constraints of resource limitations and dynamic state changes in the cloud-edge collaborative environment.

To achieve the dynamic optimization of the rotation timing for deception defense in the cloud-edge collaborative network, a game framework with temporal modeling capabilities is required. The temporal characteristics of the FlipIt game [12] can effectively model the selection of deception defense timing. It abstracts the competition for control rights over resources between the attacker and the defender into discrete seizing actions, and the seizing timing determines the defense benefit. The time delay factor serves as a key variable in designing attack-defense strategies. Ignoring it leads to incorrect strategic assessments and fails to accurately capture the temporal characteristics of state transitions. As a result, the adaptability of the defense strategy in a highly dynamic environment will be insufficient.

Therefore, defenders need to select the appropriate timing for the rotation of deception defense resources in the time dimension while considering the limited resources and the time delay constraints of state transitions. The goal is to comprehensively consider the benefits of deception defense and the costs of deception timing transitions, and to find the optimal strategy for the rotation of deception defense timing. Based on this, we propose a deception defense timing selection method based on the time-delay FlipIt game, TD-FlipIt-MADDPG. Our main contributions are as follows:

(1) We deeply analyze the network attack and defense timing, improve the network security dynamics model by combining the time-delay factor, and construct the time-delay differential equations of coupled state evolution.
(2) We design the cloud-edge collaborative network defense architecture and describe the overall defense process.
(3) We design a time-delay FlipIt game model, modeling the deceptive attack and defense process as a game model with temporal constraints, and formally describe the optimal rotation decision of the defending side under the cooling period constraint by using a time gate control mechanism, in order to reduce the defense cost.

(4) We use the multi-agent reinforcement learning algorithm to solve the deception defense timing rotation strategy. The experimental results show that the method can be effectively applied to the attack and defense process in the cloud-edge collaborative network and improve the efficiency of the defender.

The follow-up content of this paper is as follows: Section 2 gives the related work, Section 3 systematically analyzes the network attack and defense timing, and reveals the impact of time delay on the evolution of network security state. In Section 4, we construct the system model based on the time-delay FlipIt game, elaborate the model design, and introduce the time gating mechanism. In Section 5, the solution method of deception defense strategy is proposed, and the multi-agent reinforcement learning algorithm is used to solve the strategy. In Section 6, the effectiveness of the proposed method is verified by experiments. Finally, we make a conclusion.

2 Related Work

Currently, Moving Target Defense (MTD) and cyber deception defense technologies have been extensively studied. References [13,14] indicate that MTD dynamically changes the available attack surface through system reconfiguration, serving as an effective method for cloud computing security defense. Soussi et al. [15] proposed a multi-objective deep reinforcement learning approach in the Edge-to-Cloud Continuum to learn optimal MTD strategies. Casola et al. [16] presented a novel MTD framework that implements MTD techniques in cloud-edge systems according to predefined policies. In terms of deception defense, Anwar et al. [17] addressed malicious reconnaissance attacks by employing hypergame theory to construct a mixed honeypot deployment model, optimizing the collaborative configuration strategies of low-interaction and high-interaction honeypots under information asymmetry conditions to enhance deception effectiveness. Li et al. [18] proposed an optimal deception defense framework for container clouds based on System Risk Graph (SRG) modeling and Deep Reinforcement Learning (DRL). Through SRG-based dynamic adversarial modeling, they trained DRL agents to generate optimal deployment strategies for decoys and deceptive routing, and defined deception coefficients to quantitatively evaluate defense effectiveness.

Regarding deception defense architectures, Khoa et al. [19] proposed an SDN-based proactive defense framework that achieves attack trapping and dynamic security policy updates through honeypot deployment combined with cyber threat intelligence. Subsequently, Qin et al. [20] addressed reconnaissance attacks in industrial control systems by proposing a hybrid adaptive defense framework integrating network obfuscation and deception techniques. Through heterogeneous redundancy and regenerative dual-heterogeneous subnet design, this framework enhances defense performance while ensuring system availability. Furthermore, reference [21] explored the integration of MTD and deception technologies, proposing a hybrid defense effectiveness evaluation method that integrates queuing and evolutionary game theories. The aforementioned studies have enriched proactive defense means from different perspectives; however, most works focus on the implementation of defense technologies or architecture optimization, with insufficient attention to dynamic game-theoretic decision-making under defense timing and delay constraints.

However, at the same time, the effectiveness of deception defense highly depends on the timing of the rotation of deceptive assets. Mann [22] emphasized the importance of time factor for network defense, reasoned on the time factor of network attack and defense process, and proposed a formal model based on time dimension. Farhang and Grossklags [23] considered the time-based attack-defense scenario, and combined the protection time, detection time and reaction time to construct a time security game model. Under the assumption of periodic strategies of both attackers and defenders, the payoff function was derived, which provided a theoretical basis and numerical method for the defender to calculate the optimal defense reset time. By fusing the time characteristics of differential games and the state transition mechanism of Markov decision, Zhang et al. [24] characterized the continuous-time randomness of strategy triggering to

construct a multi-stage attack-defense confrontation model. However, the time decision methods of such active defense mostly adopt fixed cycle or random trigger mechanism [23,24], which leads to the defense logic tends to be static, and it is difficult to adapt to the realistic constraints of limited resources and dynamic state changes in the cloud-edge collaborative environment.

In order to enhance the defense effect of deception, it is necessary to model the dynamic game relationship between attack and defense, and determine the appropriate time to dynamically rotate deception assets to dynamically adjust the defense strategy. FlipIt game has become a powerful tool for modeling security timing decision because of its exquisite description of preemption timing [25]. Merlevede et al. [26] described the timing decision problem in network security model in detail by introducing time exponential discount mechanism to extend the attack-defense model. Tan et al. [27] used the SIRM epidemic model to construct the state transition model of the attack surface, and then established the FlipIt game driven attack-defense timing decision-making framework FG-MTD, and designed the optimal timing selection algorithm to verify the timing law of different attack-defense strategies. He et al. [28] proposed a deception strategy selection method based on multi-stage FlipIt game and proximal Strategy optimization (PPO) algorithm. By constructing a mobile deception attack surface model and introducing discount factors and state transition probabilities, they achieved the solution of the optimal defense strategy in the spatiotemporal dimension. Zhu and Zhou [29] integrated SIR propagation model and FlipIt game to solve the optimal defense action timing, and verified that different attack periods and defense periods correspond to different defense effects.

However, the above classical models and most of their extended models assume that attack-defense actions take effect instantaneously, ignoring the state transition delay [30] that is ubiquitous in real attack-defense operations. The uncertainty of the time delay plays a key role in the game between the attacker and the defender, which can make the defender shift from passive response to active prediction, such as adjusting the rotation rhythm of the deception strategy by predicting the attacker's action window.

Reinforcement learning (RL) methods are widely used to solve complex dynamic security games. Among them, the single-agent RL method [28] regards the attacker as a passive environment and ignores his individual rationality. However, the existing work on multi-agent RL applied to attack-defense games [31] fails to effectively embed key constraints such as state transition delay and cooling-off period of defense actions into the decision-making process of agents, which limits the effectiveness and robustness of strategies in real environments.

To sum up, the existing research on deception defense timing decision still has shortcomings. On the one hand, the attack-defense game strategy lacks consideration of the attacker's individual rationality. On the other hand, it ignores the time-delay of state transition and the cooling-off period constraints of defense actions in real attack-defense, which is difficult to guide the optimal rotation frequency under limited resources. To solve this problem, we propose a time-delay FlipIt game. By constructing a network state evolution model with time-delay, introducing a formal time gate control mechanism, and using multi-agent reinforcement learning to solve the problem, it aims to realize efficient, low-cost and robust deception defense timing optimization in the cloud-edge collaborative environment.

3 Network Attack and Defense Analysis

In the cloud-edge collaborative network, the asymmetry in timing and the delay interval in the behavior of network attack and defense directly affect the ownership of control and defense effectiveness. This section systematically analyzes the time characteristics of network attack and defense behaviors, points out the key role of defense timing in dynamic deception defense, and reveals the impact of time delay on the evolution of network security state.

3.1 Attack and Defense Timing Analysis

In cloud-edge collaborative networks, attackers usually use edge nodes as springboards to gradually infiltrate into the cloud. Defenders need to dynamically adjust the deployment of deception defense resources, such as honeypots and decoy services, to change the attack surface, so as to interfere with the attacker's reconnaissance and penetration behavior.

The attacker's action timing is unknown, requiring the defender to rotate deception resources, thereby changing the attack surface and prolonging control over the target system. Therefore, the effectiveness of the deception defense action is highly dependent on its execution timing, which is constrained by two types of critical timing constraints. The first is the defense cooling period, which is endogenously determined by the defense resource cost, that is, each defense action requires a resource cost, and the defense cooling period must be experienced after the completion of a defense action, during which the next defense action cannot be started. The second is the network state transition delay, which is the time from the deployment of deception defense assets and the launch of network attacks to the actual update of network node states. These two types of delay constrain the choice of defense timing and redefine the strategy interaction mode of both sides. The time delay factor makes the defense timing decision no longer a simple frequency optimization, but a game process of time series window prediction and preemption.

Ignoring these constraints and performing high-frequency rotation will not only greatly increase the resource overhead, but also lead to defense failure due to the overlapping configuration of deception resources or the delay of network node state transition. On the contrary, too low rotation frequency will make the attack surface fixed for a long time, which makes it easy to be penetrated by attackers, so as to obtain longer control time and attack benefits.

It can be seen that the alternation of resource control in the process of network attack and defense has significant temporal dependence on the behaviors of both sides. Time delay as a key factor restricts the offensive and defensive actions, and reshapes the decision-making structure of defense rotation timing. Under the condition of limited resources, the defender needs to dynamically select the best rotation time of deception defense resources to maximize the long-term defense utility through the complex game of time delay window prediction and preemption, so as to effectively deal with advanced persistent threats in the cloud-edge collaborative environment.

3.2 Network Security State Evolution

In the attack-defense game of cloud-edge collaborative network, the security state of network nodes is not an instantaneous transition, but a dynamic evolution process with significant time delay. The traditional network security propagation dynamics model usually assumes that the node state transition is completed immediately, ignoring the factors such as operation or response delay that are common in the real network attack and defense environment, and it is difficult to accurately describe the evolution of the security state.

In order to more realistically describe the transition of network security state in cloud-edge collaborative networks, this paper introduces a time delay term based on the classical propagation dynamics framework, and constructs a security state transition model with time delay. Let the set of network nodes be V, and the Node Evolution State of each node $\mathrm{NES} \in \{\mathrm{NS},\ \mathrm{PS}, \mathrm{IS},\ \mathrm{DS}\}$, denoted as normal, protected, infected and damaged states, respectively. At any time t in the process of attack and defense, the proportions of nodes in normal state, protected state, infected state and damaged state in the cloud-edge collaborative network are $S_{\mathrm{NS}}(t)$, $S_{\mathrm{PS}}(t)$, $S_{\mathrm{IS}}(t)$ and $S_{\mathrm{DS}}(t)$, respectively, and the sum of the proportions of four node states is 1.

The evolution process of the network state needs to consider the time delay factor. For example, the attacker's penetration of the node needs to go through reconnaissance, empowerment and other stages, so

that the normal node will not immediately change into the infected or damaged state. Similarly, after the defender deploys honeypots, decoys and other deception assets, it also needs a certain time to complete the configuration and activation before the node can enter the protection state. Nodes in different network states can be transformed into each other, and the network security state transition relationship is shown in Fig. 1. There are six state transition paths, which are normal state to protected state, protected state to normal state, protected state to infected state, normal state to infected state, infected state to protected state, and infected state to damaged state. Each transition path has the corresponding transition coefficients λ_{NP}, λ_{PN}, λ_{PI}, λ_{NI}, λ_{IP} and λ_{ID}. The influence of the behavior of the attack and defense sides on the network state transition has a non-negligible time delay, and the size of the time delay is uncertain. The corresponding delay parameters are introduced as τ_{NP}, τ_{PN}, τ_{PI}, τ_{NI}, τ_{IP} and τ_{ID}, and the state transition paths are as follows.

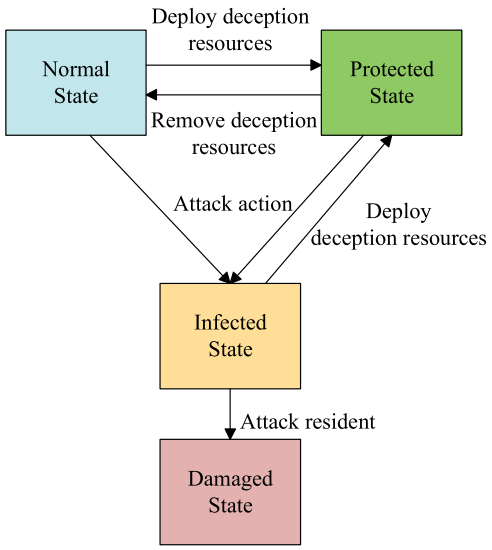


Figure 1: Diagram of network security state evolution.

NS→PS: After the defender deploys the deception resource, some normal nodes transition to the protected state, and the transition needs to experience a certain deployment delay τ_{NP}, that is, there is a time delay from the time the deception resource deployment instruction is issued to the actual effect. Once deployed, these nodes can interfere with attacker reconnaissance and penetration, thereby improving overall defense capabilities.

NS→IS: When a network node is in a normal state, if it is subjected to the penetration behavior initiated by the attacker, such as vulnerability exploitation, lateral movement, etc., and is not intercepted in time, the node will change into the infected state after a time delay τ_{NI}.

PS→NS: Nodes in the protected state may lose their protection due to the failure of the deception strategy, such as when the honeypot is detected, when resources are limited, or when the defense rotation occurs and the deception defense resources are removed. The removal of the deception resources deployed for protecting the nodes also has a response delay. After delay τ_{PN}, it returns to the normal state.

PS→IS: If the attacker identifies the deception resources deployed by the defender and successfully attacks the protected state node, the node transforms from the protected state to the infected state after time delay τ_{PI}.

IS→PS: After the defender recognizes that the node has been infected, the system resets and deploys the deception defense resources, and turns it into a protected state after the response delay τ_{IP}.

IS→DS: If the infected node fails to recover in time, the attacker can further perform penetration operations, resulting in the node transitioning to damaged state after time delay τ_{ID}, that is, unable to provide normal services.

The transition between states is affected by both historical states and time delay. In order to describe the influence of time delay on the evolution of security states, we introduce a time delay term into the traditional propagation dynamics model, and construct a set of differential equations that can reflect the timing characteristics of state transition. Based on the above state transition path, a delay differential equation system of the following form can be established to describe the evolution of the network node state, as shown in Eq. (1):

$$\begin{aligned}
\frac{dS_{\mathrm{NS}}(t)}{dt} &= -\lambda_{NP}S_{\mathrm{NS}}(t-\tau_{NP})e^{-\tau_{NP}} - \lambda_{NI}S_{\mathrm{NS}}(t-\tau_{NI})e^{-\tau_{NI}} + \lambda_{PN}S_{\mathrm{PS}}(t-\tau_{PN})e^{-\tau_{PN}} \\
\frac{dS_{\mathrm{PS}}(t)}{dt} &= \lambda_{NP}S_{\mathrm{NS}}(t-\tau_{NP})e^{-\tau_{NP}} + \lambda_{IP}S_{\mathrm{IS}}(t-\tau_{IP})e^{-\tau_{IP}} - \lambda_{PN}S_{\mathrm{PS}}(t-\tau_{PN})e^{-\tau_{PN}} - \lambda_{PI}S_{\mathrm{PS}}(t-\tau_{PI})e^{-\tau_{PI}} \\
\frac{dS_{\mathrm{IS}}(t)}{dt} &= \lambda_{PI}S_{\mathrm{PS}}(t-\tau_{PI})e^{-\tau_{PI}} + \lambda_{NI}S_{\mathrm{NS}}(t-\tau_{NI})e^{-\tau_{NI}} - \lambda_{IP}S_{\mathrm{IS}}(t-\tau_{IP})e^{-\tau_{IP}} - \lambda_{ID}S_{\mathrm{IS}}(t-\tau_{ID})e^{-\tau_{ID}} \\
\frac{dS_{\mathrm{DS}}(t)}{dt} &= \lambda_{ID}S_{\mathrm{IS}}(t-\tau_{ID})e^{-\tau_{ID}}
\end{aligned} \tag{1}$$

The delay differential equations show that the effective deception defense must consider the delay factor. The network state transition at the current time t is not determined by the immediate attack-defense behavior, but is driven by the historical action delay before $t-\tau$. Therefore, the event-based real-time response strategy has become invalid, and it is necessary to pre-defend based on the attack prediction within the time delay window. Therefore, the timing selection becomes the core of defense effectiveness, that is, by predicting the attack window, the deployment and rotation of spoofing resources are completed before the attack takes effect, so as to preempt the initiative of state evolution.

It should be noted that the exponential term $e^{-\tau}$ in the formula is not intended to describe the exponential decay of the delay, but rather to characterize the objective law that the effective probability of nodes completing state transitions decreases with increasing delay duration τ, due to factors such as dynamic interactions between attack and defense behaviors, competition for system resources, or environmental noise during the delay interval. This modeling concept originates from the "effective transition rate" commonly employed in delay differential equations, where state changes depend not only on historical states but also on the system evolution process during the delay period. In reliability engineering and queuing theory, the assumption that task success probability decreases exponentially with waiting time is classical; similarly, in network attack-defense scenarios, attackers may be disrupted, shift targets, or encounter defensive resets during the delay period, causing their actual success rate to attenuate with increasing delay. Therefore, adopting the exponential relaxation factor $e^{-\tau}$ can reasonably approximate the inhibitory effect of such uncertainty on state transitions. This form has been widely applied in time-delayed system dynamics, epidemic propagation modeling, and related fields, and has been validated in existing research on dynamic games in network security [32].

In summary, the state evolution model with time delay factor is not only closer to the actual operation mechanism of cloud-edge collaborative network, but also provides a dynamic state space for subsequent defense timing optimization based on FlipIt game. The goal of the defender is to dynamically plan the rotation time of deception actions under the constraint of limited resources and the delay characteristics of state transition. To suppress the attacker's control window and maximize the defense utility.

4 System Model

To effectively implement the deception defense timing rotation strategy, this section proposes a cloud-edge collaborative network defense architecture, and presents a multi-stage deception defense model based on the delayed FlipIt game, as well as a time gate control mechanism.

4.1 Cloud Edge Collaborative Network Defense Architecture

The cloud-edge collaborative network collaborative defense architecture proposed in this section is shown in Fig. 2. The overall architecture adopts a hierarchical design, which can be divided into cloud layer and edge layer. Its network architecture can be formally defined as an undirected graph $G = (V, E)$, the node set is $V = \{V^c, V^e\}$, and the link set $E \subseteq V_c \times V_e$ represents the cross-domain connection relationship.

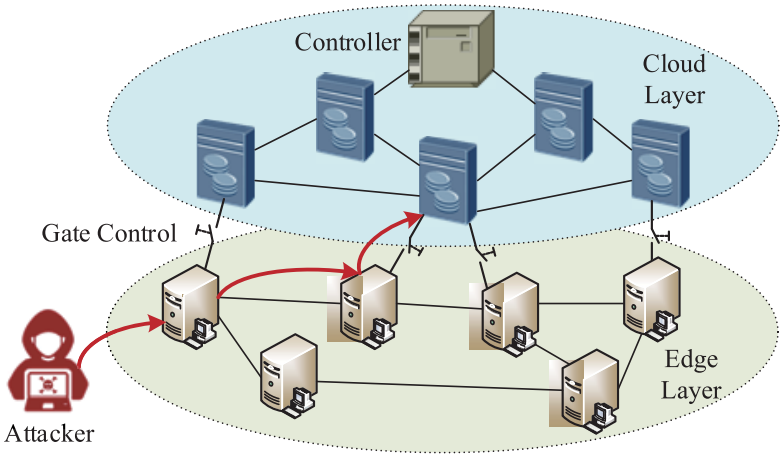


Figure 2: Diagram of cloud-edge collaborative network defense architecture.

In the process of attack-defense interaction, the attacker represents the external threat entity and launches the attack from the edge layer. The attack usually scans and intrudes the edge nodes, uses the relatively exposed attack surface and possible vulnerabilities as a springboard, and then moves laterally, and penetrates the core services of key nodes in the cloud layer by analyzing the cross-domain connection $E \subseteq V_c \times V_e$. To steal data or gain long-term control. In this architecture, the attacker behavior is the external input and game opponent that drives the whole defense system to make dynamic adjustment and timing decisions.

In the defense architecture, the cloud layer is responsible for the generation and timing planning of the defense rotation timing strategy, and there is an inevitable time delay between the issuance and implementation of the strategy. It can be seen that in the process of network attack and defense, as the defender, the cloud-edge collaborative network responds to the actions of the attacker, and there is a corresponding delay and uncertainty after issuing the deception time rotation strategy. Based on the historical attack and defense trajectories and the current network state, the controller in the cloud layer used the time-delay FlipIt game model and multi-agent reinforcement learning algorithm to solve the game strategy, dynamically calculated the optimal deception resource rotation time, distributed the strategy to the edge layer device through the gate control mechanism, and received the environment state feedback from the edge layer. By updating the parameters of the game model and the reinforcement learning strategy, the deception timing decision was updated in the continuous attack-defense game, which provided an overall framework for the construction of the subsequent multi-stage deception defense model.

4.2 Deception Defense Model Based on Time-Delay FlipIt Game

The FlipIt game is a continuous-time framework involving three entities: the attacker, the defender, and the system resources. The attack and defense sides try to obtain or maintain the control through discrete "flip" actions, which can effectively simulate the dynamic competition process of the attack and defense sides for the control of the system resources. The longer one of the parties controls the resource, the higher its payoff. However, the classical model assumes that the action is effective and completed instantly. In real scenarios, there is often a time delay from the issuance of offensive and defensive strategies to the state response. In order to describe the timing dependence and response delay characteristics of attack and defense in the control struggle process in cloud-edge cooperative network, we introduce time delay on the basis of the classical FlipIt game model, and construct a time-delayed FlipIt game model (TD-FlipIt) to be closer to the actual attack-defense scenario.

Fig. 3 shows the diagram of the TD-FlipIt game, the attacker and defender control the system resources alternately, the arrow pointed by the circle represents the time to execute the attack and defense actions, the blue area represents the time when the defender controls the system resources, the red area represents the time when the attacker has penetrated and controlled the resources, and each attack and defense game action does not change the network state instantly. It takes time delay τ for both sides to gain control after making response actions. Therefore, the state transition delay in the defense model can affect the evolution of the network state, and this delay mechanism makes the selection of defense time no longer a reactive preemption problem to deal with attack events, but a predictive preemption problem to the attacker's attack window under the constraint of state evolution delay. To formalize the attacker's strategy space, assuming that the time interval for the attacker to gain control of the target follows an exponential distribution with parameter $\lambda > 0$, the probability distribution of the time interval for the attacker to gain control of the target is $p(t) = \lambda e^{-\lambda t}\ (t > 0)$. This assumption stems from the modeling approach of attacker behavioral randomness in the classical FlipIt game framework [12], where the memoryless property of the exponential distribution can reasonably describe the attacker's independent decision-making behavior pattern in continuous time.

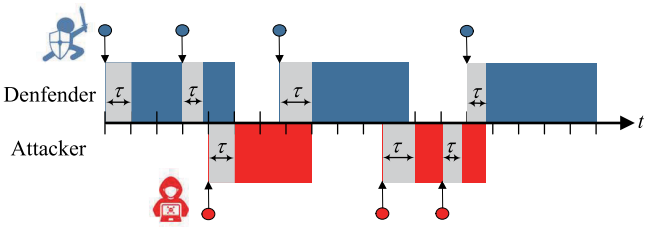


Figure 3: Schematic diagram of TD-FlipIt game.

The TD-FlipIt game model can be expressed as a tuple $FG = \{N, S, \Pi, \tau, U\}$ as follows.

$N = (N_D(t), N_A(t))$ represents the players of the game, which are the defender and the attacker, respectively. When $N_D(t) = 1$ and $N_A(t) = 0$, it means that the system resources are controlled by the defender at the current time, otherwise they are controlled by the attacker.

$S = (S_{\mathrm{NS}}(t), S_{\mathrm{PS}}(t), S_{\mathrm{IS}}(t), S_{\mathrm{DS}}(t))$ is the system state vector, which represents the proportion of four kinds of state nodes at time t. The state vector evolves dynamically from the delay differential equations, and its change is affected by the strategies of both attack and defense sides.

$\Pi = (\Pi_D, \Pi_A)$ is the set of participant strategies, and the defender redeploys the deception assets by dynamically selecting the rotation opportunity to reset the attack surface, that is, whether to launch an action at the current time t. For example, the defender's strategy Π_D at time t can be mapped to an action sequence

$\pi_D(t) \in \{0,1\}$. Where $\pi_D(t) = 1$ means that the defender initiates an action at time t; In the attack-defense game at time t, the attacker actively adjusts the probability distribution strategy $\pi_A(t)$ of the control time interval of the system, that is, adjusts the parameter λ to obtain the target control. When λ is larger, the attacker has higher attack frequency and faster penetration speed, but usually with higher attack cost. The parameter λ can regulate the transition coefficients of the evolution of node states to infected states and damaged states, which can be respectively expressed as:

$$\begin{aligned} \lambda_{NI} &= \frac{\lambda S_{\mathrm{NS}}(t)}{1 + S_{\mathrm{IS}}(t)} \\ \lambda_{PI} &= \frac{\lambda S_{\mathrm{PS}}(t)}{1 + S_{\mathrm{IS}}(t)} \\ \lambda_{ID} &= \frac{\lambda S_{\mathrm{IS}}(t)}{1 + S_{\mathrm{DS}}(t)} \end{aligned} \tag{2}$$

$\tau = (\tau_1, \tau_2, \cdots, \tau_T)$ represents the time delay set of the action executed by the participant, and describes the response delay from the initiation of the action to the effect of the action in the game process of the attacker and defender, where τ_i contains the transition delay between the four states.

$U = (U_D, U_A)$ is the utility function of the participants, the time delay is dynamic uncertain, and the operations of the attack and defense sides in the FlipIt game may not occur at the same time, this asynchronous game will affect the revenue of both sides, so the utility function of the defender and the attacker is constructed based on the long-term cumulative revenue, considering the holding time of the resource control and the action cost. The defender utility function U_D can be expressed as

$$U_D = \int_0^T (r_{\mathrm{NS}} \cdot S_{\mathrm{NS}}(t) + r_{\mathrm{PS}} \cdot S_{\mathrm{PS}}(t))dt - c_d \cdot K_{\Pi_D}, \tag{3}$$

where r_{NS} and r_{PS} are the defender's payoff of controlling the normal or protected state node in unit time, c_d is the resource cost of a single deception defense rotation, and K_{Π_D} is the number of rotation actions executed by the defender's policy Π_D in $[0, T]$. The attacker utility function U_A can be expressed as

$$U_A = \int_0^T (r_{\mathrm{IS}} \cdot S_{\mathrm{IS}}(t) + r_{\mathrm{DS}} \cdot S_{\mathrm{DS}}(t))dt - c_a \cdot \lambda T, \tag{4}$$

where r_{IS} and r_{DS} are the attacker's payoff per unit time of controlling the infected or infected state node, c_a is the cost of the attacker's single attack action to control the target resource, and λT is the number of times the attacker controls the target resource in $[0, T]$.

According to the TD-FlipIt model, the strategy space of both sides is compact, and their utility functions are bounded and upper semicontinuous, and the network state has continuous dependence on the strategy perturbation. According to Glicksberg's theorem [33], there is a mixed strategy Nash equilibrium in this kind of non-cooperative game. In this equilibrium strategy, the offensive and defensive sides cannot unilaterally adjust their strategies to obtain better utility values, and this equilibrium strategy will be approached by multi-agent reinforcement learning method.

In the TD-FlipIt game, both attackers and defenders are rational players, and they each pursue the maximization of long-term utility, so their optimization objectives are to maximize their respective utility functions U_A and U_D.

4.3 Time Gate Control Mechanism

Due to the limited defense resources in the cloud-edge collaborative network, each rotation operation needs to consume resources. In order to reflect the principle of individual rationality of the defender and effectively balance the defense benefit and resource cost, we introduce time gate control mechanism before the decision execution of the deception defense rotation strategy. In this mechanism, after the defender completes a deception asset rotation action, the system will enter the defense cooling period, during which the defender cannot initiate the next rotation action. The gate control mechanism is implemented by time-gating function, which can be expressed as

$$g(t) = \begin{cases} 1, & \text{if } r_{\mathrm{NS}} \cdot S_{\mathrm{NS}}(t) + r_{\mathrm{PS}} \cdot S_{\mathrm{PS}}(t) > c_d \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The function reflects the individual rationality of the defender, and its value depends on the comparison between the immediate defense benefit and the rotation resource cost. Only when the benefit exceeds the cost, the gating signal is set to 1, and the rotation instruction is allowed to be issued. Otherwise $g(t) = 0$, the system is in the cooling period, and the rotation instruction cannot be issued. It should be noted that the opening of gating only means that the execution condition is satisfied, and does not necessarily mean that the rotation is triggered, so as to realize the active arrangement and restriction of the rotation time.

The time gate control mechanism is tightly coupled with the time delay FlipIt game model. In the game model, the defender's utility function U_D includes the action cost $c_d \cdot K_{\Pi_D}$, and the time gating mechanism can limit the maximum possible value of K_{Π_D}, which ensures the rationality of the cost term in the utility function from the strategy execution level. Therefore, the gating mechanism can guide the defender to seek the "right time" instead of the "immediate response" to defend under the resource constraint, and maximize the defense utility through accurate predictive preemption.

5 Deception Defense Strategy Solution

In the process of attack and defense in cloud-edge collaborative network, both attackers and defenders are agents with individual rationality. Due to the interdependence of attack and defense strategies and the continuous evolution of environmental states, and the uncertainty and dynamics of actions of both attackers and defenders, traditional strategy optimization methods cannot meet the needs of defenders for strategy optimization. To solve this problem, we propose a solution method based on Multi-Agent Deep Deterministic Policy Gradient (MADDPG). The TD-FlipIt game model is embedded into the multi-agent reinforcement learning framework to realize the co-evolution and adaptive optimization of attack and defense strategies.

The MADDPG algorithm sets two agents, the defender and the attacker, the system state is $s \in S$, the observation of each agent i is $o_i \in O$, and the action is $a_i \in A$. Each agent selects an action a_i according to its policy $\pi_i(o_i)$, gets an immediate reward r_i and the next state s', and gets an experience set (s, a_i, r_i, s') to store in buffer D. The MARL interface of state, action and reward of MADDPG algorithm is designed as follows.

The state space includes the state proportions of the four types of nodes in the current network, denoted as $S_{\mathrm{NS}}(t)$, $S_{\mathrm{PS}}(t)$, $S_{\mathrm{IS}}(t)$, and $S_{\mathrm{DS}}(t)$, as well as the action time during which the attacking and defending sides can observe the current network state and execute the flipping actions. **The action space** takes into account the strategies of both the defenders and the attackers. The defender's action a_D is to adjust the timing of the deceptive defense rotation, and the attacker's action a_A is also to adjust the timing of the control of the target system, which is represented by the adjustment amplitude of the parameter λ. **The reward function**

serves as the utility function for the participants in the game model. The defender's reward is the defender's utility function U_D, and the attacker's reward is the attacker's utility function U_A.

Each agent in MADDPG adopts the actor-critic framework. The critic network takes the global state s_t and the joint agent action (a_D, a_A) as input, and outputs the Q value to evaluate the long-term value of the joint action to agent i. The target Q value of the critic network is expressed as follows:

$$y_i = r_i + \gamma Q_i^*(s', a'_D, a'_A; \varphi_i^*), \tag{6}$$

where r_i is the immediate reward, γ is the discount factor, φ_i^* is the parameter of the target Q network, a'_D and a'_A are respectively generated for the target actor network of the defender agent and the attacker agent, and noise is added to the target action a'_i to realize the exploration goal strategy. The target action a'_i is as follows:

$$a'_i = \pi_i^*(o'_i) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma), \tag{7}$$

where π_i^* is the target actor network policy function, o'_i is the observation of agent i in the next state s', ε is the noise term following the normal distribution $\mathcal{N}(0, \sigma)$.

The loss function of the critic network can be defined as follows by minimizing the Bellman error.

$$L(\varphi_i) = \mathbb{E}_{s,a,r,s'}\left[\left(Q_i(s, a_D, a_A; \varphi_i) - y_i\right)^2\right] \tag{8}$$

φ_i are the parameters of the critic network, and the actor network maximizes the Q-value through the policy gradient, whose gradient is calculated as follows.

$$\nabla_{\theta_i} J(\pi_i) = \mathbb{E}_{s,a}\left[\nabla_{a_i} Q_i(s, a_D, a_A; \varphi_i) \cdot \nabla_{\theta_i} \pi_i(o_i; \theta_i)\right], \tag{9}$$

where $\nabla_{a_i} Q_i$ is the gradient of the critic network Q_i to the action a_i, and $\nabla_{\theta_i} \pi_i$ is the gradient of the policy function π_i to the parameter θ_i in the actor network. In addition, in order to improve the training stability, the parameters θ_i^* and φ_i^* of the target actor network and the target critic network are synchronized by soft update, and $\upsilon \in (0,1)$ is the rate parameter of soft update.

$$\begin{aligned} \theta_i^* &\leftarrow \upsilon\theta_i + (1-\upsilon)\theta_i^* \\ \varphi_i^* &\leftarrow \upsilon\varphi_i + (1-\upsilon)\varphi_i^* \end{aligned} \tag{10}$$

The deception defense timing selection algorithm based on MADDPG is implemented by centralized training and decentralized execution, and the algorithm process is shown in Algorithm 1.

Algorithm 1: Deception defense timing selection algorithm based on MADDPG

Input: The MADDPG hyperparameters, states, and rewards
Output: The actions and training parameters of each agent
1: Create defender and attacker agents in the cloud-edge collaborative networks.
2: Initialize the parameters θ and ϕ of the actor network and critic network.
3: Initialize the replay buffer D of each agent.
4: **for** episode = 1, 2, ..., N_e **do**
5: Reset the cloud-edge collaborative environment and obtain the states s.
6: **for** time step = 1, 2, ..., T **do**

(Continued)

Algorithm 1 (continued)

if gate control $g(t) = 1$ **then**
  Defender actor network executes action $a_D(t)$ based on observation o_D.
  Obtain the defender agent reward $r_D(t)$ and the next state s'.
else
  Defender does not perform any action.
end if
Attacker actor network executes action $a_A(t)$ based on observation o_A.
Obtain the attacker agent reward $r_A(t)$ and the next state s'.
Store the experience group (s, a, r, s') into buffer D.
for each agent i **do**
  Randomly extract mini-batch data from the buffer D.
  Updated critic network parameter φ_{ji} with gradient descent.
  Updated actor network parameter θ_i with gradient ascent.
  Update the target network parameters φ_{ji}^* and ${\theta_i}^*$.
end for
end for
end for

Through continuous interaction with the environment, the agent learns its own optimal policy, evaluates the value of each state given by the environment, and selects actions accordingly. The update of the state information of the attack-defense game also affects the action selection of the attacker and the defender. After multiple strategy-value iterative updates, it finally converges to a stable interactive decision-making strategy. Therefore, the defender can dynamically adjust the trigger time of deception rotation according to the attacker's behavior pattern, so as to achieve more efficient active defense.

6 Experimental and Analysis

6.1 Experimental Setup

To systematically evaluate the performance of the proposed deception defense timing selection method TD-FlipIt-MADDPG based on the time-delay FlipIt game, this section constructs an attack-defense scenario in the cloud-edge collaborative network. The experimental hardware configuration environment is an Intel(R) Xeon(R) Gold 5218 CPU with 128 GB RAM, and the operating system is Ubuntu 18.04. The experimental environment is programmed and set up based on Python 3.8.20 and PyTorch 1.13.0 deep learning framework. Additionally, we use the Mininet network simulator to create the attack scenario of the cloud-edge collaborative network. The Ryu controller manages 20 OpenFlow switches and deploys 500 terminal hosts. The attacker employs a port scanning script as the actuator of attack behavior. The scanning interval of this script is determined by parameter λ, which is decided by the attacker agent. The attack frequency is adaptively adjusted through the policy evolution of the MADDPG algorithm across episodes, while the defender uses IP address dynamic transformation to simulate the rotation of deceptive assets, thereby constructing a realistic confrontation scenario. We referenced the parameter settings from the literature [28] for the indicators, the defender's unit-time revenue from controlling normal/protected nodes ranges from 0.5 to 2.0, while the attacker's revenue from controlling infected/compromised nodes ranges from 1.0 to 2.5. This setting reflects the realistic assumption that attackers typically obtain higher unit revenues from controlling nodes, while maintaining relative comparability of revenues. And the time delay parameter of the network state transition in the attack-defense game is regarded as the physical characteristic

of the network environment, which is uncertain. We randomly sample delay values from preset ranges at the beginning of each experimental round to simulate delay characteristics under different network conditions. This treatment draws upon the approach for modeling delay uncertainty in the literature [32], ensuring that the model can demonstrate diverse attack-defense scenarios ranging from low-intensity to high-intensity. The time delay and transfer rate coefficients of the network state transition are set within the corresponding range. In the proposed method, delay values are randomly selected within the parameter range for each simulation experiment. Specifically, at the beginning of each independent experimental run, delay values are randomly sampled from the preset parameter range to simulate the inherent delay characteristics under different network environments. The specific parameter settings are shown in Table 1.

Table 1: Simulation parameter settings.

Simulation Parameter	Value
Revenue of the defender control nodes per unit time	0.5–2.0
Revenue of the attacker control nodes per unit time	1.0–2.5
Network state transition delay	0.5–1.2
Transition coefficient	0.1–1.0
Cost of the attacker's single control target resource c_a	0.4
Resource cost of single deception defense rotation c_d	0.2
Experience replay buffer D	100000
Batch size	128
Noise standard deviation σ	0.1
Discount factor γ	0.95
Episodes	2500
Time step	512
Soft update rate ν	0.1

To demonstrate the performance of the TD-FlipIt-MADDPG method proposed in this paper in selecting the timing for deception defense rotations, the following methods are selected as the comparison methods, all comparison methods employ the same state space.

(1) FP: This method serves as a classic trigger mechanism for moving target defense, performing the rotation of deception defense actions at a fixed period.

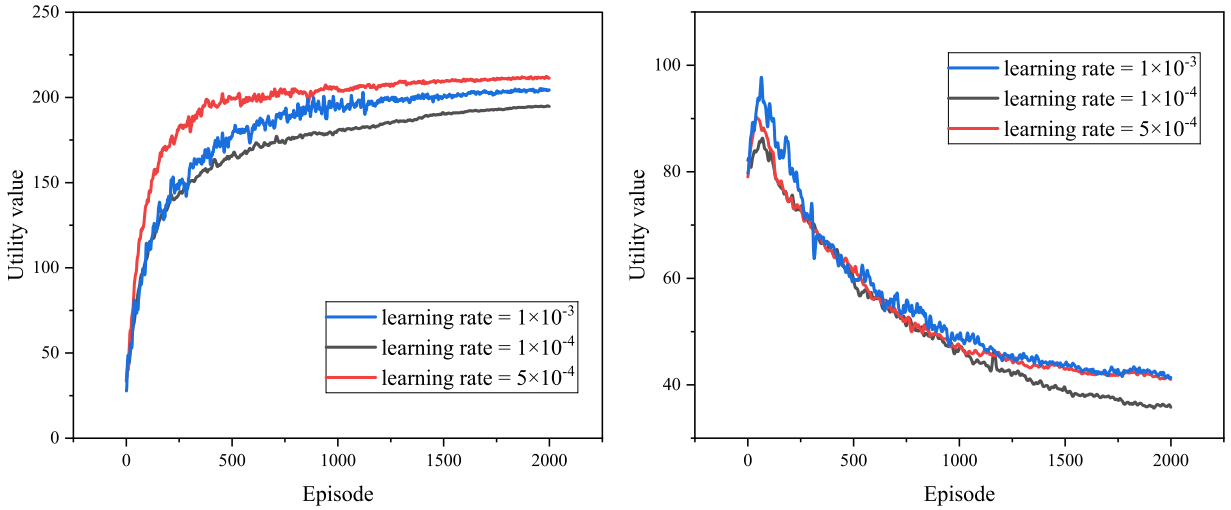
(2) RP: This method adopts a random triggering mechanism. Compared with the FP methods, RP introduces randomness in selecting rotation periods.

(3) MFD-PPO [28]: This method models the attack and defense process as a multi-stage FlipIt game, and characterizes the dynamic evolution of the system by introducing a discount factor and stage transition probability. This method uses a single-agent proximal policy optimization (PPO) algorithm to solve the defender's strategy, treating the attacker as part of the environment rather than an independent strategy decision-maker.

(4) SF-MAWP [31]: The attack and defense process is modeled as a Stackelberg-FlipIt game, and the multi-agent WoLF-PHC algorithm is used to solve it. This method takes into account the information asymmetry between the attacker and the defender and sequential decision-making, but does not explicitly model the time delay during state transitions.

6.2 Convergence Analysis

In order to evaluate the influence of the learning rate of the actor network in MADDPG algorithm on the convergence of the algorithm, this section sets the learning rate as 1×10^{-4}, 5×10^{-4} and 1×10^{-3} respectively for training. Fig. 4 shows the utility function convergence curves of the defender agent and the attacker agent, respectively. The three different learning rates selected can make the training converge, and the defender's utility shows a good upward trend under the three learning rates, but there is a significant shock in the training process with a higher learning rate. When the learning rate is 5×10^{-4}, the defender's final convergence value is higher, and the initial convergence is rapid and stable, and the attacker's agent's utility value finally converges stably under this learning rate. Therefore, we choose the learning rate of 5×10^{-4} to achieve a balance between learning efficiency and stability.



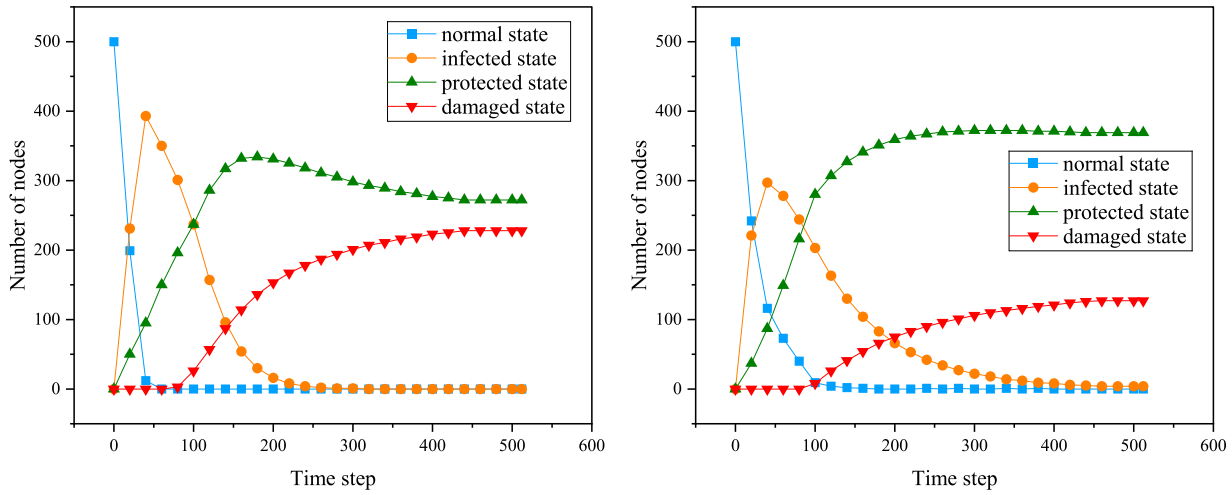
a Convergence curve of defender utility function **b** Convergence curve of attacker utility function

Figure 4: Convergence curves of utility functions at different learning rates.

6.3 Validity Analysis

In order to verify the key role of the proposed modeling of state transition delay in deception defense timing decision-making, we conducted ablation experiments. The TD-FlipIt-MADDPG method and the No-Delay MADDPG method are trained, respectively. TD-FlipIt-MADDPG is the state transition delay explicitly modeled, while the node state evolution process in the No-Delay MADDPG method does not consider the delay, assuming that all state transitions are completed instantaneously, that is, $\tau = 0$. The node state change curves of the two methods are shown in Fig. 5.

Fig. 5a shows the network state change curve of nodes under the No-Delay MADDPG method. In the early stage of the game, the number of nodes in the normal state immediately began to decline, while the number of nodes in the infected state increased rapidly. This immediate response ignores the delay required to deploy defense resources and penetrate attacks in reality. In addition, the proportion of nodes in the protected state could not be maintained in the late stage of the attack defense game, and the proportion of nodes in the protected state only converged to 54.4%.



a Curve of node state change without time delay

b Curve of node state change with time delay

Figure 5: The node state change curves.

In contrast, Fig. 5b shows the network node state change curve of TD-FlipIt-MADDPG method. Due to the introduction of time delay factor, the number of nodes in normal state decreases slowly, making the defense strategy more accurately capture the evolution process of network node state, avoiding invalid early rotation, and finally the proportion of nodes in protected state can be stabilized at 73.8%. To sum up, this experiment shows that ignoring the delay factor will lead to the model quickly learning the suboptimal strategy, and the explicit modeling of state transition delay can effectively improve the effectiveness of the deception defense rotation strategy.

6.4 Defense Cost Analysis

In order to verify the effectiveness of the time gate control mechanism proposed in this paper in reducing the defense cost, this section compares the IP hopping frequency of different methods under the condition of fixed proportion of protected state nodes. In deception defense, nodes with deployed deception defense resources are considered to be in a protected state. In the experiments, it is assumed that hosts endowed with IP hopping capabilities, which remain undetected when facing scanning attacks and are not included in the attack list, are regarded as protected state nodes. The proportion of protected nodes is set to 0.1, 0.2, 0.3, 0.4 and 0.5, respectively. The transformation frequency of IP addresses reflects the defense cost. Under the same protection status node conditions, the lower the IP address hopping frequency, the lower the defense cost.

The experimental results are shown in Table 2, and the proposed method with time gate control mechanism achieves the lowest IP hopping frequency for all the proportions of protected state nodes. It can be seen that by introducing the time gate control mechanism, the IP hopping frequency is greatly reduced while maintaining high security utility, which effectively balances security and resource overhead. After the node is invaded, the defender will not immediately counterattack, so its IP hopping frequency is the lowest. Therefore, reducing the rotation frequency of deception defense time is also a part of the defender's strategy, which can reduce the cost of deception defense.

Table 2: IP hopping frequency of different methods.

Proportion of Protected Nodes	FP	RP	MFD-PPO	SF-MAWP	Without Time Gate Control	With Time Gate Control
0.1	0.033	0.035	0.025	0.028	0.032	0.021
0.2	0.075	0.071	0.048	0.052	0.056	0.035
0.3	0.098	0.092	0.072	0.087	0.085	0.064
0.4	0.135	0.119	0.096	0.104	0.112	0.095
0.5	0.182	0.171	0.120	0.161	0.153	0.126

6.5 Utility Value Analysis

In order to evaluate the performance of the proposed method in the attack-defense game, we conduct a quantitative analysis by comparing the final utility values of the defender and the attacker after the training convergence of different methods. As shown in Fig. 6, the FP method has the lowest defender utility value of 127.1, which indicates that the static rotation mechanism cannot effectively cope with the dynamically changing attack behavior.

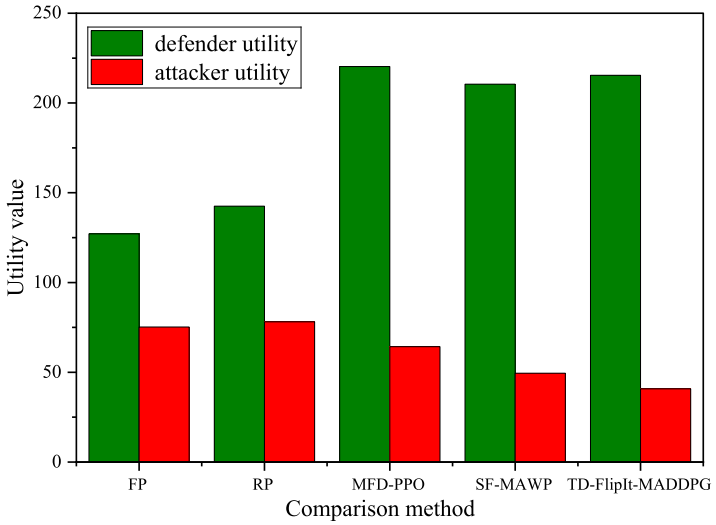


Figure 6: Comparison of utility values of different methods.

In contrast, the defense effectiveness of the other three reinforcement learning based methods is significantly improved. The defense utilities of SF-MAWP and the proposed method are slightly lower than that of MFD-PPO. This is because MFD-PPO adopts a single-agent framework, which fails to adequately characterize the adversarial behavior between the attacker and defender as independent decision-makers. By neglecting the individual rationality of the attacker, MFD-PPO limits the attacker's gains, resulting in higher defender utility values. As a result, it performs better in specific indicators but lacks model authenticity. Due to the explicit modeling of the state transition time delay, the attacker utility value of the proposed method is the lowest, which is 40.77, which is 36.59% lower than that of the MFD-PPO method. Moreover, due to the time gate control mechanism set in the proposed method, the strategy execution cost can be effectively constrained, and the defense utility of the proposed method is close to that of the MFD-PPO method.

In summary, although the defender utility value of the proposed method is slightly lower than that of the MFD-PPO method in absolute values, it shows significant advantages in terms of comprehensive performance and model rationality, which further verifies its feasibility as a rotation timing strategy for deception defense.

7 Conclusion

Aiming at the problem that it is difficult to dynamically optimize the rotation time of deception defense in a cloud-edge collaborative network, we propose a deception defense time selection method based on the time-delay FlipIt game. Based on the analysis of the dynamic evolution characteristics of attack and defense states in the cloud-edge environment, the physical time delay in the process of node state transition is explicitly modeled, and the network state evolution model fused with a delay differential equation is constructed. Then, the cloud-edge collaborative defense architecture is designed, and on this basis, TD-FlipIt game model is established, and the time gate control mechanism is introduced to formalize the defense cooling period as the execution interval constraint of the rotation action, so as to suppress invalid high-frequency operations. Finally, the MADDPG algorithm is used to solve the optimal deception defense rotation timing strategy. Experimental results show that the proposed method is superior to baseline methods in key indicators such as the proportion of protected state nodes and defense cost control. It effectively balances security and resource overhead and provides a feasible technical path for active and efficient deception defense time rotation in cloud-edge collaborative networks. We conduct experiments based on the Mininet simulation of cloud-edge collaborative networks, deploying 500 terminal nodes and employing real-world scanning scripts to simulate attack behaviors,to a certain extent reproducing the attack-defense interaction process in real-world environments. In particular, the modeling of state transition delays using delay differential equations directly correspond to physical delay mechanisms in real networks, such as firewall rule updates and virtual machine migrations, thereby enhancing the consistency between the model and reality. In future work, we will devote ourselves to constructing a real attack-defense exercise environment encompassing multi-stage attack behaviors, introducing more concrete attack chain models and traffic characteristics, thereby enabling direct measurement of security metrics such as attack success rates, so as to further validate the robustness of the proposed method in real-world scenarios against complex threats.

Acknowledgement: None.

Funding Statement: This work was supported in part by the National Key Research and Development Program of China under Grants 2024YFB2906704 and 2023YFB2903902; and in part by the State Key Laboratory of Advanced Communication Networks under Grant FFX24641X028; and in part by the Science and Technology Innovation Leading Talents Subsidy Project of Central Plains under Grant 244200510038.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jinchuan Pei and Yuxiang Hu; methodology, Jinchuan Pei and Yuxiang Hu; formal analysis, Yuxiang Hu; investigation, Zihao Wang; writing—original draft preparation, Jinchuan Pei; writing—review and editing, Jinchuan Pei, Zihao Wang and Menglong Li; supervision, Hongtao Yu; funding acquisition, Yuxiang Hu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support this study are available from authors.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Souza P, Ferreto T, Calheiros R. Maintenance operations on cloud, edge, and IoT environments: taxonomy, survey, and research challenges. ACM Comput Surv. 2024;56(10):1–38. doi:10.1145/3659097.
2. Li Q, Li L, Liu Z, Sun W, Li W, Li J, et al. Cloud-edge collaboration for industrial internet of things: scalable neurocomputing and rolling-horizon optimization. IEEE Internet Things J. 2025;12(12):19929–43. doi:10.1109/jiot.2025.3542428.
3. Devarajan MV, Yallamelli ARG, Kanta Yalla RKM, Mamidala V, Ganesan T, Sambas A. Attacks classification and data privacy protection in cloud-edge collaborative computing systems. Int J Parall Emerg Distrib Syst. 2024:1–20. doi:10.1080/17445760.2024.2417875.
4. Laurent S. AI-driven collaborative security protection for cloud-edge computing ecosystems: architecture design and performance evaluation. Int J Cybersp Secur. 2025;1(1):14–24. doi:10.22399/ijcesen.4994.
5. Ferdous J, Islam R, Mahboubi A, Islam MZ. A review of state-of-the-art malware attack trends and defense mechanisms. IEEE Access. 2023;11:121118–41. doi:10.1109/ACCESS.2023.3328351.
6. Rehman Z, Gondal I, Ge M, Dong H, Gregory M, Tari Z. Proactive defense mechanism: enhancing IoT security through diversity-based moving target defense and cyber deception. Comput Secur. 2024;139:103685. doi:10.1016/j.cose.2023.103685.
7. Aminu M, Akinsanya A, Dako DA, Oyedokun O. Enhancing cyber threat detection through real-time threat intelligence and adaptive defense mechanisms. Int J Comput Appl Technol Res. 2024;13(8):11–27. doi:10.7753/ijcatr1308.1002.
8. Zheng Y, Na Z, Ji W, Lu Y. An adaptive fuzzy SIR model for real-time malware spread prediction in industrial internet of things networks. IEEE Internet Things J. 2025;12(13):22875–88. doi:10.1109/jiot.2025.3550671.
9. Qi J. Loss and premium calculation of network nodes under the spread of SIS virus. J Intell Fuzzy Syst. 2023;44(5):7919–33. doi:10.3233/JIFS-222308.
10. Zhai W, Liu L, Ding Y, Sun S, Gu Y. ETD: an efficient time delay attack detection framework for UAV networks. IEEE Trans Inform Foren Secur. 2023;18:2913–28. doi:10.1109/tifs.2023.3272862.
11. Feng Y, Zhang W, Feng Z, Zhong X, Liu F. An MTD-driven hybrid defense method against DDoS based on Markov game in multi-controller SDN-enabled IoT networks. In: Proceedings of the 2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS); 2024 Jun 19–21; Guangzhou, China. p. 1–6.
12. van M, Juels A, Oprea A, Rivest RL. FlipIt: the game of "Stealthy Takeover". J Cryptol. 2012;26:655–713.
13. Torquato M, Vieira M. Moving target defense in cloud computing: a systematic mapping study. Comput Secur. 2020;92(4):101742. doi:10.1016/j.cose.2020.101742.
14. Cho JH, Sharma DP, Alavizadeh H, Yoon S, Ben-Asher N, Moore TJ, et al. Toward proactive, adaptive defense: a survey on moving target defense. IEEE Commun Surv Tutor. 2020;22(1):709–45. doi:10.1109/COMST.2019.2963791.
15. Soussi W, Gür G, Stiller B. Moving target defense (MTD) for 6G edge-to-cloud continuum: a cognitive perspective. IEEE Network. 2025;39(1):149–56. doi:10.1109/mnet.2024.3483302.
16. Casola V, De Benedictis A, Iorio D, Migliaccio S. A moving target defense framework to improve resilience of cloud-edge systems. In: International Conference on Advanced Information Networking and Applications. Cham, Switzerland: Springer; 2025. p. 243–52.
17. Anwar AH, Zhu M, Wan Z, Cho JH, Kamhoua CA, Singh MP. Honeypot-based cyber deception against malicious reconnaissance via hypergame theory. In: Proceedings of the GLOBECOM 2022-2022 IEEE Global Communications Conference; 2022 Dec 4–8; Rio de Janeiro, Brazil. p. 3393–8.
18. Li H, Guo Y, Sun P, Wang Y, Huo S. An optimal defensive deception framework for the container-based cloud with deep reinforcement learning. IET Inform Secur. 2022;16(3):178–92. doi:10.1049/ise2.12050.
19. Khoa NH, Do Hoang H, Ngo-Khanh K, Duy PT, Pham VH. Sdn-based cyber deception deployment for proactive defense strategy using honey of things and cyber threat intelligence. In: International Conference on Intelligence of Things. Cham, Switzerland: Springer; 2023. p. 269–78.
20. Qin X, Jiang F, Dong C, Doss R. A hybrid cyber defense framework for reconnaissance attack in industrial control systems. Comput Secur. 2024;136(4):103506. doi:10.1016/j.cose.2023.103506.

21. Hou F, Hou F, Zang X, Hua Z, Liu Z, Wu Z. Effectiveness evaluation method for hybrid defense of moving target defense and cyber deception. Computers. 2025;14(12):513. doi:10.3390/computers14120513.
22. Mann ZÁ. Time is money: a temporal model of cybersecurity. In: Nemec Zlatolas L, Rannenberg K, Welzer T, Garcia-Alfaro J, editors. ICT systems security and privacy protection. Cham, Switzerland: Springer; 2025. p. 82–96.
23. Farhang S, Grossklags J. When to invest in security? Empirical evidence and a game-theoretic approach for time-based security. arXiv:1706.00302. 2017.
24. Zhang H, Tan J, Liu X, Wang J. Moving target defense decision-making method: a dynamic Markov differential game model. In: Proceedings of the 7th ACM Workshop on Moving Target Defense; 2020 Nov 9; Virtual Event. p. 21–9.
25. Chen X, Cao W, Chen L, Han J, Yang M, Wang Z, et al. iCyberGuard: a flipit game for enhanced cybersecurity in IIoT. IEEE Trans Comput Soc Syst. 2024;11(6):8005–14.
26. Merlevede J, Johnson B, Grossklags J, Holvoet T. Time-dependent strategies in games of timing. In: Alpcan T, Vorobeychik Y, Baras JS, Dán G, editors. Decision and game theory for security. Cham, Switzerland: Springer; 2019. p. 310–30. doi:10.1007/978-3-030-32430-8_19.
27. Tan J-L, Zhang H-W, Zhang H-Q, Lei C, Jin H, Li B-W, et al. Optimal timing selection approach to moving target defense: a flipit attack-defense game model. Secur Commun Netw. 2020;2020(1):3151495–12. doi:10.1155/2020/3151495.
28. He W, Tan J, Guo Y, Shang K, Kong G. Flipit game deception strategy selection method based on deep reinforcement learning. Int J Intell Syst. 2023;2023(1):5560416. doi:10.1155/2023/5560416.
29. Zhu Z, Zhou L. Application of complex network attack and defense time game model in network security defense decision. J Cyber Secur Mobility. 2025;14(2):311–37. doi:10.13052/jcsm2245-1439.1423.
30. Qiu L, Xiang C, Wen Y, Najariyan M, Liu C, Wu Z. Predictive output feedback control of networked control system with Markov DoS attack and time delay. Int J Rob Nonlin Cont. 2023;33(5):3376–95. doi:10.1002/rnc.6572.
31. Sun R, Fei J, Zhu Y, Guo Z. Multi-agent reinforcement learning for moving target defense temporal decision-making approach based on stackelberg-flipit games. Comput Mater Contin. 2025;84(2):3765–86. doi:10.32604/cmc.2025.064849.
32. He W, Tan J, Wang R, Liu Z, Luo X, Hu H, et al. A deep reinforcement learning approach to time delay differential game deception resource deployment. IEEE Trans Depend Secure Comput. 2026;23(1):1655–70. doi:10.1109/tdsc.2025.3620151.
33. Glicksberg IL. A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points. Proc Am Math Soc. 1952;3(1):170–4. doi:10.2307/2032478.