This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

An SDN-enabled Proactive Defense Framework for DDoS Mitigation in IoT Networks

Yuyang Zhou, Guang Cheng, *Member, IEEE*, and Shui Yu, *Senior Member, IEEE*

***Abstract*—The Internet of Things (IoT) is becoming truly ubiquitous in every domain of human lives, and a large number of objects can be connected and enabled to communicate with cloud servers at any time. However, complex connections and vulnerabilities of IoT devices introduce inevitable security threats, in which distributed denial-of-service (DDoS) attacks usually incur catastrophic results. Unfortunately, the existing DDoS mitigation methods cannot provide effective protection. Moreover, the amplifying complexity and increasing delay incurred by defense greatly affect the stability of IoT networks. To tackle these problems, we present a novel framework that can proactively adapt the attack surface of IoT networks, dynamically optimize defense strategies, and rapidly deploy the corresponding defense mechanisms. In particular, we establish hybrid proactive defense mechanisms combining Moving Target Defense (MTD) techniques with cyber deception to spread camouflage information to confuse attackers. Based on these mechanisms, we introduce a defender-led signaling game model to formalize defense scenarios and depict the interactions between the defender and the attacker. Besides, we present an optimal algorithm to solve decision problems and optimize defense implementation in a cost-effective manner. Our extensive experiments demonstrate that the proposed approach can effectively mitigate DDoS attacks and maintain a high level of performance in IoT networks with acceptable overhead.**

***Index Terms*—Internet of Things, DDoS Attacks, Moving Target Defense, Cyber Deception, Signaling Game.**

I. Introduction

The Internet of Things (IoT) [1] is considered as an emerging paradigm because of its enormous capabilities for ubiquitously intelligent connectivity and applications in many domains (e.g., wearables, smartphones, and smart grids). As a result, a large number of objects with different service requirements can be connected at any time and any place in distributed networks. The high scalability is accompanied by increased complexity in management, as well as raising unavoidable challenges in ensuring cybersecurity [2]. To deal with the ever-increasing IoT applications, software-defined network (SDN) provides a novel way to tackle the challenges through the fundamental idea of decoupling the control plane and the data plane in IoT networks [3]. This decoupling provides the SDN controller a global view of the network, promoting flexible traffic engineering and alleviating the IoT security concerns at runtime [4].

Y. Zhou and G. Cheng are with School of Cyber Science and Engineering, Southeast University, International Governance Research Base of Cyberspace, Jiangsu Ubiquitous Cyber Security Engineering Research Center, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Nanjing, 211189, China. E-mail: yyzhou@njnet.edu.cn, chengguang@seu.edu.cn. S. Yu is with School of Computer Science, University of Technology Sydney, Australia. E-mail: Shui.Yu@uts.edu.au. The corresponding author is Guang Cheng.

While SDN can flexibly manage IoT by ensuring network and data security, SDN itself introduces vulnerabilities due to its architecture, inevitably increasing risk in the environment [5]. First, cloud servers are under the SDN architecture, but numerous devices cannot be managed by the SDN controller. Because of lacking necessary security protocols, an attacker can easily infect IoT devices and spread the infection to form a botnet [6] (e.g., Bashlite and Mirai botnet), and then launch distributed denial-of-service (DDoS) attacks. Meanwhile, many IoT applications rely on real-time inputs, and network unavailability causes catastrophic results in some cases [7]. For example, if an autonomous vehicle is on road with the existence of DDoS attacks on the cloud, the vehicle may lose control without receiving inputs. Moreover, the SDN controller is easy to be a single point of failure [8]. According to the baseline network flow policy, DDoS attackers can inject a lot of attack packets from spoofed IPs, and the controller will be saturated with attack traffic because of no matched rules in local caches [9]. Once the controller is compromised, the attacker can further destroy other cloud servers, resulting in a large area of network paralysis.

Unfortunately, the traditional DDoS protection methods (e.g., blackhole routing, intrusion detection system, etc.) have shown poor performance with high overhead [10] when facing attacks from a large-scale botnet. Since network configurations are typically deterministic, static, and homogeneous, the defenders are always passive and their countermeasures are costly and only effective in a short time [11]. In recent years, Moving Target Defense (MTD) [12] has gained ever-growing attention in the field of cybersecurity. It aims to create uncertainty on the attacker side and reverse the asymmetry between attackers and defenders [13], [14]. In detail, MTD continuously and randomly modifies the attack surfaces to interrupt Cyber Kill Chain (CKC) [15]. In addition, cyber deception technique [16] also increases attackers' uncertainty when they make the decision of launching attacks, and shows advantages in protection capability and resource occupation. Nevertheless, cyber deception adopts a more radical strategy than MTD in terms of deliberately providing false information (e.g., decoys, honeypots) to mislead attackers [17].

Although many solutions based on MTD and deception techniques have been proposed to deal with attacks in IoT networks [18], [19], there exist several drawbacks in these studies. Firstly, prior work was usually driven by one single technology and ignored the possibility of a combination of multiple mechanisms. Once one of defense technologies failed, attackers could bypass the defense and strike the core applications or sensitive data, which may cause serious results

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

in terms of economy and privacy. Secondly, existing research may not work for advanced attackers. They could passively monitor the traffic, actively perform network reconnaissance, and use the information to lock real attack targets. Thirdly, the frequent adaptation (e.g., IP shuffling [20], [21] or VM migration [22], [23]) may incur extra costs and have negative effects by increasing the response delay and reducing the quality of service (QoS). Therefore, it is essential to combine proactive techniques seamlessly to achieve security goals while reaching a balance between performance and cost.

In this paper, a new proactive and automatic defense framework is proposed to provide secure services in IoT networks. The key insight is to build dynamic and adaptive network properties for shifting the existing attack surface to adversaries. On the basis of the previous research of MTD and deception-based methods, we extend the idea into building a hybrid architecture that establishes combined defense over clouds, to resist external attacks while satisfying the constraints of delay and resources in IoT systems. Meanwhile, in order to mitigate DDoS attacks cost-effectively, we adopt the multistage signaling game model in which the defender thwarts a strategic attacker using proactive defense techniques, and design an optimal strategy algorithm to keep the proposed framework efficient and robust. Moreover, compared to other defense strategies, the proposed approach enables an excellent trade-off between performance and overhead in real-world experiments, while achieving meaningful security goals. The contributions of this paper are as follows.

- We propose an SDN-enabled lightweight defense framework, which can be easily deployed without adjusting the network architecture. Moreover, it can resist DDoS attacks proactively with negligible performance degradation and delay variation in the IoT environment.
- We deploy a hybrid defense that combines MTD with cyber deception seamlessly to improve the efficiency and robustness of the framework, proactively spreading misinformation to manipulate the attacker's perception.
- To further model the interactions in the proactive defense scenarios, we apply a defender-led multistage signaling game model for comprehensive analysis, which fully benefits the defense decision-making. Then, an optimal strategy selection algorithm is presented for helping solve the optimization problem and mitigate DDoS attacks in a cost-effective manner.
- We design and implement an experimental testbed to for performance evaluation. Compared with other defense strategies, simulation and experimental results have shown that the proposed framework can run with less overhead and perform better in DDoS mitigation in IoT networks.

The remainder of this paper is organized as follows. We discuss the related work in Section II and propose the threat model in Section III. Section IV demonstrates the framework design and elaborates the signaling game model and strategy algorithm. Section V presents the evaluation results. Moreover, we discuss the limitations and compare with previous works in Section VI. Finally, we conclude in Section VII.

II. Related Work

Providing security in IoT networks is a challenging task not only owing to the limited resources and heterogeneity of the end-devices, but also due to the novel protocols and networking technologies [24]. In the literature, research in IoT security is being conducted widely at a rapid pace [25], and solutions to defend against security attacks mainly include prevention [26], detection [27], and mitigation [7]. Besides, solutions employing SDN and proactive defense technologies [28], [29], [30] mitigate attacks against IoT in a more proactive manner and have attracted more attention due to its flexibility and dynamicity. Therefore, we provide a brief discussion of the related techniques, including MTD and cyber deception. We also briefly review the signaling games for IoT security in this section.

A. MTD Techniques

MTD is a proactive defense mechanism which aims to break the static nature of the protected system by dynamically changing the attack surface. MTD techniques can be categorized as:
Shuffling-based MTD techniques (e.g., IP/port/topology shuffling) confuse attackers in the reconnaissance stage and can be deployed in various network domains. VM migration is considered as another shuffling technique that changes the location of VMs or the service deployment to increase the dynamics and reduce the likelihood of attack exploitation.
Diversity-based MTD techniques provide equivalent applications with different implementations [31]. Code diversification realizes defense by running functionally-equivalent variants [32]. In software diversity, a program can be diversified at different stages of the life cycle by just-in-time compiler [33]. Besides, instruction sets randomization and address space layout randomization diversify in codes to resist injection attacks [34].
Redundancy-based MTD techniques construct replicas of applications or services with the same functionality. For instance, an FPGA-oriented MTD method [35] is proposed to defeat hardware Trojan through randomly selecting one replica from multiple replicas for operation at runtime. Kanellopoulos et al. [36] introduced an MTD-based framework that employs redundancy to resist sensor attacks in cyber-physical systems.

In addition, there are hybrid MTD techniques that combine multiple MTD mechanisms to work in cooperation. Nevertheless, frequent MTD mutations inevitably have negative effects on the performance and additional deployments may not be suitable for IoT devices due to lack of computation capabilities. However, the proposed lightweight framework can be easily deployed as applications on top of SDN controller, and the proposed algorithm can effectively avoid unnecessary defense cost and decrease the overhead incurred by MTD mechanisms.

B. Deception Techniques

Cyber deception is another proactive defense technique that adds decoys into the network to attract adversaries and prevent them away from their potential targets. In cybersecurity research, deception techniques were first proposed in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

the late 1980s to trace intruders and the concept of honeypot followed [16]. The basic idea of traditional honeypots is to disguise themselves as regular hosts and mislead potential attackers, then defenders can record information when attackers attempt to strike decoys. Subsequently, in order to further entrap adversaries and prevent them from eliminating vulnerabilities, honeynets and honeyfarms were proposed to form a network of high-interaction honeypots [37].

Cyber deception has shown great promise in disrupting the CKC at its early stage, however, the complexity of the honeypots and the programming required in a large system generally lead to a reduction in the attackers' beliefs and inability to maintain their attention [38]. Meanwhile, due to the cost of implementation and maintenance, it is difficult to scale high-interaction honeypots that are usually deployed to gain in-depth information about attacks and attackers' behaviors [39], [40]. Therefore, cyber deception [41] is recently accompanied by visualization and automation technologies in the IoT environment, which facilitates the deployment and boosts the update of decoys. Nevertheless, due to the limitation of computing and storage capacity of IoT devices, attackers can identify decoys according to longer response time. Moreover, it is a challenge to develop deception techniques in real environments, which requires significant effort in continuous network monitoring and safe deployment without compromising system integrity [42]. In contrary to traditional deception-based methods, the proposed approach disguises the production systems and decoys through flexible adjustments on network properties instead of deploying a large number of honeypots, which will save a lot of defense resources.

C. Signaling Games for IoT security

Game theory has been introduced to model the interaction between defender and attacker, and select options to produce an optimizing outcome for both players. The defender cannot precisely discover whether the obtained messages are coming from legitimate users or malicious adversaries in many security scenarios [43]. Therefore, signaling game arises as an interesting subclass of games when both players have incomplete information concerning the strategies and payoffs of each other, and has been utilized to model specific issues in the cybersecurity field.

In the study of IoT security, signaling games supply a formal framework that takes the impacts of deception in games into consideration [44]. Despite the fact that we also apply signaling game to cyber deception, a hybrid defense framework with a multistage signaling game has not been researched. Furthermore, the proposed framework is not only applied to DDoS mitigation but also suitable for delay-sensitive IoT environments. Different from existing approaches using signaling game model, MTD and deception techniques in this study can be strategically utilized to send deceptive signals proactively in the defender's perspective, reversing the information-asymmetric situation and creating more advantages for defenders. Therefore, thanks to the proposed defender-led signaling game model, we can accurately depict the dynamic behaviors of both players, and effectively defend against DDoS attacks in a more proactive manner.

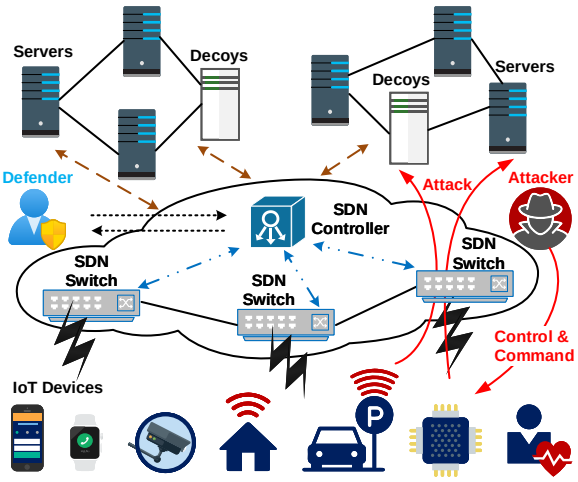


Fig. 1. Proactive defense and DDoS attacks in the IoT environment.

III. Threat Model

The IoT environment in which exists a couple of SDN-based cloud servers is vulnerable to DDoS attacks. We assume that the SDN controller and the key infrastructure are trusted, while the attackers come from an external network. The intention of DDoS attackers is flooding the cloud servers to prevent legitimate access. We first display an instance of proactive defense deployment and DDoS attacks in the IoT environment in Fig. 1. Infected IoT devices are controlled by attackers to send malicious traffic simultaneously to compromise cloud servers, while defenders can implement proactive defense to guide attackers to flood wrong targets. Then, we define the attacker's characteristics as follows:

Attack Targets: In the IoT environment, cloud nodes contain different applications, thus, they become the attack targets. Once one or several cloud servers are compromised by DDoS attackers, these controlled servers may spread the attack and send malicious packets to other servers at high speed. Consequently, IoT devices could not receive responses from these servers, and meanwhile, the spread of attacks may further cause network paralysis or even interruption. Therefore, we assume that every server in the IoT network can be a potential attack target.

Attack Process: To formally depict the multistage attack process, we assume that the attacker performs a multi-round attack based on CKC. In order to identify real assets and honeypots in the network, attackers first perform reconnaissance actions (e.g., IP and port scanning) on the protected system. Due to the existence of proactive defense framework, attackers can only obtain a part of information from current reconnaissance. Further, they will analyze the authenticity of discovered servers and select an appropriate target to compromise. In other words, these potential targets are the servers recognized by attackers as real ones based on their existing knowledge. Next, attackers can use DDoS attack tools for flooding the target. Finally, after the attackers initiate malicious attacks against selected targets, they will evaluate their payoffs incurred by compromising real production systems, and then

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

enter the next round of reconnaissance until the attack payoffs reach the maximum.

Attackers' Capability: Since DDoS attackers in the IoT environment usually launch attacks via the botnet, thus, we assume they can continually strike multiple cloud servers at the same time until congesting the entire IoT network. In addition, attackers are rational and have some knowledge of the defense in our assumption, and we scope their capabilities as follows.

- We assume that attackers know there exist some proactive defense techniques in the network, but they do not know the detailed deployments.
- Given the information collected during the reconnaissance stage, attackers distinguish between real servers and decoys based on it. Specifically, attackers will not launch an attack until obtaining enough information.
- Once attack targets are confirmed, attackers are able to utilize the compromised IoT devices to perform DDoS attacks on the chosen servers to maximize their payoffs.

Especially, the attackers' capabilities also have the following limitations. In our assumption, attackers cannot obtain the details of defense configurations, including the running mechanisms of MTD and the deployment of decoys among cloud servers. As well, attackers need to judge the authenticity of the information from their perspectives, thus, they may be fooled by deliberately released false information.

IV. Framework Overview

We first show the framework and its design details in this section, and further describe the signaling game model for analyzing the interactions between both players. At last, we propose an algorithm for designing the strategies to rapidly optimize the cost and effectiveness of the proactive defense deployments and efficiently resist DDoS attacks.

A. Framework Design

In order to explain the proposed method in detail, we provide an overview of the framework design. The proposed framework can not only evaluate the comprehensive status of the network via implementing agents into resources, but also optimize the cost and effectiveness based on the signaling game model and proactive defense mechanisms. Fig. 2 presents the overall framework architecture which includes three modules, Monitor Agents, Decision Module, and Deployment Module.

The Monitor Agents are embedded into the cloud servers and SDN switches for monitoring security events and real-time system performance. In general, the security events cover various aspects containing security logs, alerts, and malware detection, etc. For efficiency, we focus on the events on IP/port scanning and the security logs of each server. While these events occurred, the Monitor Agents collect them and notify Status Collector. Then, the security events on a given server will be recorded as input parameters that help make decisions. With the framework running, these parameters will be updated continually. Moreover, based on the change in service performance caused by defense implementation, Monitor Agents will have a preliminary assessment of the defense status and overhead, and send the analysis data to Status Collector.

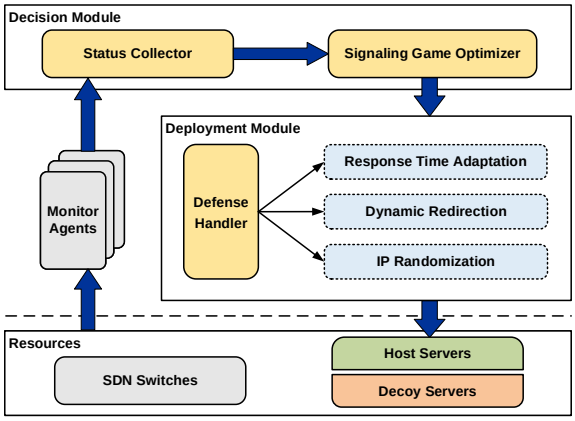


Fig. 2. Overview of the Proposed Framework.

Further, based on the received security events, Status Collector will analyze and construct the actions triggered by attackers when the defense mechanisms of the framework work. On the basis of attackers' actions, Signaling Game Optimizer can evaluate the effectiveness of defense strategies and select appropriate mechanisms to deploy. Specifically, to maximize the security, proactive defense should be utilized to confuse attackers and make them as far as possible to have a completely wrong perception about the network, meanwhile, keep total performance costs acceptable. Thus, we optimize the signaling strategies that disclosure information to attackers and further affect their behaviors for the defender's advantage. More details on the optimal strategy selection will be shown in Sections IV-B and IV-C. Next, according to the strategy results obtained from Decision Module, Defense Handler will deploy different MTD mechanisms on specified cloud servers.

B. Multistage Signaling Game Model

According to the threat model, we propose a defender-led game model in which the defender plays as the leader and adopts proactive defense mechanisms to thwart a strategic attacker by false information disclosure. Since MTD and deception techniques can create uncertainty or unreality to attackers, attackers will have incomplete information about the dynamic environment and interact with the proactive defenders in the multistage defense-attack process. To this end, we will construct the multistage signaling game model to help select the optimal strategy over defense. In the following, we first provide the overview of the multistage signaling game and formally define the message and action spaces, belief functions, and payoff functions of both players. Then, we analyze the game processes between them, and explore optimal strategies defending the IoT networks by the computation of Perfect Bayesian Nash Equilibria (PBNE) in this game.

1) Game Overview: The strategic attacker can observe the defender's actions (e.g., defense deployment or undeployment) through the reconnaissance efforts, and then commit attacks in a rational way. Both sides have their own goals, in which the DDoS attacker is to maximize the number of comprising

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

TABLE I
NOTATION TABLE.

Notation	Definition
S, R	Sender and Receiver
$\theta \in \Theta$	Types
$m \in M, a \in A$	Messages and Actions
N	Number of servers in the network
$\zeta_n(t)$	Belief of R for the n-th S at time step t
$p(\theta)$	Prior Probability of Type θ
$\lambda^S(m\|\theta)$	Strategy of S with Type θ
$\lambda^R(a\|m)$	Strategy of R given Message m
$U^X(\theta, m, a)$	Payoff Function of Player X, $X \in \{S, R\}$
g_p, c_p	Gain and cost of S for sending $m = 1$
g_h, c_h	Gain and cost of S for sending $m = 0$
g_a, c_a	Gain and cost of R for taking action $a = 0$

servers while the goal of the defender is to protect them cost-effectively. To simplify the analysis, we only consider binary information and actions. The notations used in this paper are summarized in Table I, and the necessary definitions in this game model are listed as follows.

Players and Types: The interplay between them forms a noncooperative game, and the defender plays as the sender whereas the attacker receives the messages. In this game, the defender adopts MTD and deception techniques to send deceptive signals, which makes it difficult to identify the real assets of the clouds. Therefore, the attackers cannot know the type of sender but need to receive the defender's messages by reconnaissance and monitoring network traffic for judgment. The proactive defense signaling game $\mathcal{G}^{pd}$ is played individually by each server/sender along with its type, $\theta \in \Theta = \{0, 1\}$, where $\theta = 0$ represents a real server whereas $\theta = 1$ denotes a honeypot.

Messages: We consider a multistage game where the defender sends a message from a set of possible actions once ($m \in M = \{0, 1\}$) and the attacker receives it and takes one corresponding action. In general, different configurations constitute various messages, and they can manipulate the attacker's perspective about whether a server is a real cloud server or a honeypot. Since the space of possible messages is large, we focus on one specific message: whether the service of the system can be requested at low-speed or high-speed. Let $m = 1$ represent that a system allows only slow response times, and $m = 0$ can be denoted that a system can respond to a request with quick response time.

For example, the defender deploys a series of decoy servers ($\theta = 1$) in the system. Typically, honeypots respond to requests slowly ($m = 1$), so they have $\theta = m$ that may reveal these servers are decoys. On the contrary, honeypots with quick response times ($m = 0$) can release deceptive signals in order to make themselves appear to be real servers. Similarly, the defender could disguise real servers as honeypots for the attacker by decrease the response times of the cloud servers.

Actions: Let $a = 0$ denotes attack, and let $a = 1$ denotes withdraw. After receiving the message m, the attacker would select an action $a \in A = \{0, 1\}$.

2) Belief model: In order to launch successful attacks, strategic attackers usually conduct reconnaissance on target servers, however, they do not know whether the received messages from a real server or a honeypot. Therefore, the attacker receives these messages and maintains a belief function ζ for the sender's type according to the reconnaissance and messages. Let $\zeta : \Theta \to [0, 1]$ denotes the belief function for the receiver R, where ζ represents the likelihood with which the attacker believes that the sender is of type θ. In practice, the defender and the attacker will continue to interact until the horizon of game comes. Especially, in the multistage signaling game, this kind of defense-attack interactions will be repeated continuously at every time step t, where $t = 0, 1, ..., T$ ($T \in Z^+$represents the time horizon). Therefore, the attacker in this game will have a specific belief of each server at each time step: $\zeta_n(t)$, where $0 < n \leqslant N$. Furthermore, the belief will consist of two parts, including the initial belief and dynamic belief.

Initial Belief: In our assumption, there is no information available about any server before conducting reconnaissance, and all servers are at the same level of suspicion for the attacker in the beginning. At time step $t = 0$, we assume that the attacker believes each server is a production system initially. Thus, the initial belief for each cloud server at the beginning of the game does not need to be calculated and can be formally represented as:

$$\zeta_n(0) = \begin{cases} 1, \theta = 0 \\ 0, \theta = 1 \end{cases}, 0 < n \leqslant N \tag{1}$$

Dynamic Belief: The dynamic belief for a specific sender is a key parameter to evaluate the expected payoffs of both two players. Therefore, we should give a formal definition to the dynamic belief $\zeta_n(t)$, where $1 \leqslant t \leqslant T$. Contrast to the initial belief, the dynamic belief is not only adjusted to the game process dynamically but also updated according to the sender's messages. With the development of the game process, $\zeta_n(t)$ will be calculated according to the given message m each time, which can be denoted as follows:

$$\zeta_n(t) = \begin{cases} (1 - \frac{1}{t+1})\zeta(\theta|m), & \theta = 0 \\ 1 - \frac{t}{t+1}(1 - \zeta(\theta|m)), & \theta = 1 \end{cases}, 0 < n \leqslant N, t > 0 \tag{2}$$

where $\zeta(\theta|m)$ represents the posterior probability with which R believes that S is of type θ when given message m, and $\zeta_n(t)$ is the basis for the receiver R to make a decision about which action to take at this time. For the binary type of the sender, we can conclude that the posterior probability for a given message satisfies $\sum_{\theta \in \Theta} \zeta(\theta|m) = 1$, and Eq. 2 can promise that the sum of dynamic belief values for different types of senders also equals 1, such that $\sum_{\theta \in \Theta} \zeta_n(t) = 1$. Due to the awareness of the existence of decoys, it is cautious for attackers not to expose themselves until collecting enough knowledge. Therefore, a lower belief value for a real system will be given at the beginning, which means that the receiver prefers to believe that it is a honeypot. As the role of receiver, the attacker will perform more rounds of reconnaissance as the game progresses and gain more but not full knowledge of the target, so the belief is dynamically updated according to Eq. 2 and continues to be close to the reality. To be specific, both of the belief values constantly tend to the posterior probability and will be equal to $\zeta(\theta|m)$ when t is infinite.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

TABLE II
PAYOFF MATRIX FOR DIFFERENT ACTIONS FOR THE TWO PLAYERS.

Type & Message	Player	Action	
		a=0	a=1
$\theta=0, m=0$	S R	$-g_p$ g_a-c_a	0 $-g_p$
$\theta=0, m=1$	S R	$-g_p-c_p$ $g_a-g_p-c_a$	g_p-c_p $-g_p$
$\theta=1, m=0$	S R	g_h-c_h $-c_a$	$-c_h$ 0
$\theta=1, m=1$	S R	g_h $-g_h-c_a$	0 0

3) Payoff model: In this section, we define the payoff functions of both players considering their possible strategies. The purpose of proactive defense is to make the identification of the real servers difficult and guide the attacker to launch attacks on decoys. In the proposed framework, the defender takes necessary system resources to apply defensive actions (i.e., IP randomization), sending deceptive messages to attackers and attempting to induce them to strike the wrong targets (i.e., R plays $a=0$ in response to S with type $\theta=1$). Thus, the defender can obtain the defensive payoff by successfully protecting cloud servers against the attacks. On the other hand, it certainly consumes the attackers' resources to perform reconnaissance operations and launch DDoS attacks. Once the target servers have been compromised, then the attacker achieves the purpose of flooding the network and benefits from it.

Let $U^X: \Theta \times M \times A \rightarrow \mathbb{R}$ denotes the payoff function of the player X, where $X \in \{S, R\}$. Thus, $U^S(\theta, m, a)$ is given to represent the sender's payoff value when S of type θ sends m while R responds with action a. Likewise, $U^R(\theta, m, a)$ gives the receiver's payoff under the same scenario. Then, we can illustrate various payoffs $U^X(\theta, m, a)$ of different action profiles based on the parameters defined in Table I. There are four type and message profiles between the two players as well as two action profiles, as presented in Table II. Since there is not a significant impact on the security when the attacks have been withdrawn from honeypots, we consider $U^R(1,0,1), U^S(1,1,1)$, and $U^R(1,1,1)$ equal to zero. When the production system does not deploy any active defense mechanism, even if no cost is consumed, the defender will not gain benefits when the attacker chooses not to attack.

Furthermore, both players have their game strategies against each other, so we define the expected payoff of them as well. For the sender S, it is worth noting that the expected payoff not only relies on the sender's strategy of releasing signals but also depends on the receiver's reaction. As a result, let $\overline{U^S}(\lambda^S, \lambda^R|\theta)$ represent the expected payoff to S when a given sender of type θ plays λ^S as follows:

$$\overline{U^S}(\lambda^S, \lambda^R|\theta) = \sum_{m \in M} \sum_{a \in A} \lambda^R(a|m) U^S(\theta, m, a) \tag{3}$$

where $\lambda^R(a|m)$ gives the probability of responding with action a depends on the message m that R receives, and $\forall m \in M, \sum_{a \in A} \lambda^R(a|m) = 1$. Moreover, $\lambda^S(m|\theta)$ denotes that S sends message m with a certain probability when the sender is of type θ, and $\sum_{m \in M} \lambda^S(m|\theta) = 1$.

Different from the expected payoff of the sender, owing to the type of sender and message are both given for the receiver R when taking actions, the expected payoff function to R is related to the receiver's strategy. Based on the given sender's type θ and message m, we can define $\overline{U^R}(\lambda^R|\theta, m)$ as:

$$\overline{U^R}(\lambda^R|\theta, m) = \sum_{\theta \in \Theta} \zeta(\theta|m) U^R(\theta, m, a) \tag{4}$$

4) Game analysis: Generally, players involved in the signaling games would like to maximize expected payoffs from their respective perspectives, given their beliefs about the other players. In this section, we study the existence and properties of PBNE for the proposed game $\mathcal{G}^{pd}$. A PBNE can be characterized to meet all the following requirements as:

Requirement 1: The receiver will have a belief $\zeta(\theta|m)$ about the type of sender when receiving m from S such that

$$\forall \theta \in \Theta, \zeta(\theta|m) \geqslant 0 \ and \ \sum_{\theta \in \Theta} \zeta(\theta|m) = 1 \tag{5}$$

Requirement 2: For a given belief $\zeta(\theta|m)$, the receiver's strategy must maximize the expected payoff $\overline{U^R}$ such that

$$\forall m \in M, \lambda^R \in \arg\max \sum_{\theta \in \Theta} \zeta(\theta|m) U^R(\theta, m, a) \tag{6}$$

Requirement 3: For a given sender's type θ, each sender's strategy must maximize the payoff given the λ^R, such that

$$\forall \theta \in \Theta, \lambda^S \in \arg\max U^S(\theta, m, a) \tag{7}$$

Requirement 4: If there exists a type θ that satisfies the sender's strategy is $\lambda^S = m^*$, where $m^* \in M$. Then, the receiver's belief about the sender's type corresponding to m^* must satisfy Bayesian rules, such that

$$\forall m \in M, \zeta(\theta|m) = \frac{p(\theta)}{\sum\limits_{\theta \in \Theta^*} p(\theta)} \tag{8}$$

where Θ^* presents the set of types for those senders who send m^*, and $p(\theta)$ denotes the prior probability of type θ.

Definition 1. *A PBNE in a signaling game is both strategies and the receiver's belief that satisfy Requirement 1-4.*

Then, based on the analysis of the multistage signaling game between both two players, we use PBNE to explore optimal strategies resisting DDoS attacks. In this scenario, the defender controls the servers to send different messages to the attacker. With the posterior belief $\zeta(\theta|m)$, the attacker would like to maximize the expected payoff according to the strategy of selecting actions at different time steps. Although there is a time step t defined in the game process, all the parameters except beliefs do not depend on it. Therefore, for simplicity, we also suppose that the payoffs of the signaling game at different time steps are the same under the same conditions. In other words, there is no payoff discounted with the continuous defense-attack scenarios that involve both two players. Then, we propose a theorem about PBNE computation and release this relationship.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

Theorem 1. *There are two mixed-strategy PBNE at each time step of the multistage signaling game $\mathcal{G}^{pd}$.*

Proof. There are two possible mixed-strategy PBNE in this signaling game. First, let the mixed strategy $\lambda_t^{S*}(1|0)$ for the sender of type $\theta = 0$ such that S sends message $m = 1$ with probability λ^{S*} at time step t. Similarly, for the receiver R, let the mixed strategy be $\lambda_t^{R*}(0|1)$. Since we consider the existence of the PBNE, the receiver's beliefs must satisfy Requirement 4, that is, determined by Bayesian rules and the sender's strategy. Let $p_t = \{\zeta_0(t), \zeta_1(t), ..., \zeta_N(t)|\theta = 0\}$. We then compute the expected payoff of R based on its belief. The receiver's expected payoffs when taking $a = 0$ and $a = 1$ at each time step are:

$$\begin{aligned}\overline{U_{a=0}^{R}} =&\lambda_t^{S*}p_t(g_a-c_a)+(1-\lambda_t^{S*})p_t(g_a-g_p-c_a)\\ &+\lambda_t^{S*}(1-p_t)(-g_h-c_a)+(1-\lambda_t^{S*})(1-p_t)(-c_a)\end{aligned} \tag{9}$$

and

$$\overline{U_{a=1}^{R}} = \lambda_t^{S*}p_t(-g_p) + (1-\lambda_t^{S*})p_t(-g_p) \tag{10}$$

According to the indifference principle, R can play either $a = 0$ or $a = 1$ with the equilibrium strategy, and the expected payoff is considered as the optimal one when the sender S optimizes its strategy. Let $\overline{U_{a=0}^{R}} = \overline{U_{a=1}^{R}}$, and we can get the optimal strategy that S with type $\theta = 0$ would send message $m = 1$ with the probability of λ_t^{S*}, such that

$$\lambda_t^{S*} = \frac{c_a - p_t g_a}{p_t g_p - (1-p_t)g_h} \tag{11}$$

On the other hand, from the payoff matrix in Table II, the sender S's expected payoffs of sending messages $m = 1$ and $m = 0$ can be respectively represented as:

$$\overline{U_{m=1}^{S}}=\lambda_t^{R*}p_t(-g_p-c_p)+(1-\lambda_t^{R*})p_t(g_p-c_p)+\lambda_t^{R*}(1-p_t)g_h \tag{12}$$

and

$$\begin{aligned}\overline{U_{m=0}^{S}} =&\lambda_t^{R*}p_t(-g_p) + (1-\lambda_t^{R*})(1-p_t)(g_h-c_h)\\ &+\lambda_t^{R*}(1-p_t)(-c_h)\end{aligned} \tag{13}$$

Similarly, the sender S at the t-th stage can reach its equilibrium by sending either m or 1–m according to the indifference principle, and make its expected payoff optimal when the receiver R plays the optimal strategy. Let $\overline{U_{m=1}^{S}} = \overline{U_{m=0}^{S}}$, and we can calculate the optimal probability λ_t^{R*} of R playing action $a = 0$ as follows:

$$\lambda_t^{R*} = \frac{(1-p_t)(g_h-c_h) - p_t(g_p-c_p)}{2(1-p_t)g_h - p_t g_p} \tag{14}$$

As a matter of fact, this is a mixed-strategy PBNE in this multistage signaling game, which can be represented as strategy pairs $(\lambda_t^{S*}, \lambda_t^{R*})$. That is, the sender of type $\theta = 0$ will send message $m = 1$ with probability λ_t^{S*}, and R will response with action $a=0$ of probability λ_t^{R*}, respectively.

The other mixed-strategy PBNE is for another type of S with $\theta=1$ to send $m=0$ to R with probability $\lambda_t^{S\dagger}(0|1)$. Then, R will take action $a=1$ with probability $\lambda_t^{R\dagger}(1|0)$. Similarly, the optimal probabilities $\lambda_t^{S\dagger}$ and $\lambda_t^{R\dagger}$ can be attained as:

$$\lambda_t^{S\dagger} = \frac{p_t g_a + (1-p_t)g_h + (1-2p_t)c_a}{(1-p_t)g_h + 2(1-p_t)c_a - p_t g_p} \tag{15}$$

Algorithm 1: Optimal Strategy Selection Algorithm

Input: Sets of server types $\theta(t) = \{\theta_1, \theta_2, ...\theta_N\}$;
Status logs of servers $l(t) = \{l_1, l_2, ...l_N\}$;
game parameters: $g_p, c_p, g_h, c_h, g_a, c_a$;
Output: Sets of decisions with probabilities each time:
$D(t) = \{(d_1, \sigma_1), (d_2, \sigma_2), ...(d_N, \sigma_N)\}$;

```
Initialize the signaling game with input at t = 1;
Formulate the payoff matrix (cf. Table II);
for time step t = 1 to T do
    ζ(t) = (1 − 1/(t+1)) · p(θ=0) / Σ_{θ∈θ(t)} p(θ=0)  (cf. Eqs. 2 and 8);
    for server i = 1 to N do
        if 0 ⩽ Σ_{i=1}^{N} (1−θ_i) l_i ⩽ N p(θ_i) and Σ_{i=1}^{N} θ_i l_i = 0 then
            d_i ← Perform IP Randomization, σ_i = 1;
        else
            if | Σ_{i=1}^{N} θ_i l_i − Σ_{i=1}^{N} l_i | ≥ ε and θ_i = 0 then
                d_i ← Perform Dynamic Redirection;
                σ_i = (c_a − ζ(t) g_a) / (ζ(t) g_p − (1−ζ(t)) g_h)  (cf. Eq. 11);
            else if | Σ_{i=1}^{N} θ_i l_i − Σ_{i=1}^{N} l_i | ≥ ε and θ_i = 1 then
                d_i ← Perform Response Time Adaptation;
                σ_i = (ζ(t) g_a + (1−ζ(t)) g_h + (1−2ζ(t)) c_a) / ((1−ζ(t)) g_h + 2(1−ζ(t)) c_a − ζ(t) g_p)  (cf. Eq. 15);
            else
                D(t) = D(t − 1)
Return D(t) = {(d_1, σ_1), (d_2, σ_2), ...(d_n, σ_N)}
```

$$\lambda_t^{R\dagger} = \frac{p_t c_p - (1-p_t)c_h}{p_t g_p} \tag{16}$$

In summary, the multistage signaling game $\mathcal{G}^{pd}$ has two kinds of mixed-strategy PBNE at each time step, which can be expressed with strategy pairs $(\lambda_t^{S*}, \lambda_t^{R*})$ and $(\lambda_t^{S\dagger}, \lambda_t^{R\dagger})$. These strategy pairs mean that a real server in the IoT environment will deliberately send a deceptive signal with probability λ_t^{S*} and the DDoS attacker may strike it with probability λ_t^{R*}, while a honeypot can try to deceive the attacker with probability $\lambda_t^{S\dagger}$ but the attacker will withdraw with probability $\lambda_t^{R\dagger}$ to avoid unnecessary costs at time step t. Based on above equilibria, we can now guide the defender to develop the optimal defense strategy considering the attacker's belief of the proactive framework and payoffs under various strategies. □

C. Optimal Strategy Selection Algorithm

According to the signaling game and its PBNE calculation procession proposed previously, the optimal strategy profile indicates that two players in the signaling game should perform an optimal strategy pair respectively to maximize their expected payoffs. In the optimization problem, our objective is to maximize the number of decoys accessed by the attackers,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

thereby reducing the number of compromised production systems. The objective function can be expressed as:

$$\begin{aligned} \textbf{Problem:} \max_{D(t)} \ & \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\theta_i l_i, \\ \textit{s.t.}\ \textbf{C1:}\ & 0 \leqslant \sum_{i=1}^{N} l_i \leqslant \sum_{i=1}^{N}\theta_i, \quad \forall t \in [1,T], \\ \textbf{C2:}\ & 0 \leqslant \sum_{i=1}^{N}(1-\theta_i) l_i \leqslant N p(\theta_i), \quad \forall t \in [1,T]. \end{aligned} \tag{17}$$

where θ_i represents the type of the ith server, and l_i is the security log recorded by the ith server. Let $l_i = 0$ denotes no attack has been logged while $l_i = 1$ means that the ith server is under attack. In the above, the objective function is to find the maximum difference between the numbers of decoys targeted and compromised production systems. The first constraint condition indicates the number of servers that DDoS attackers can flood at the same time, and the other constraint ensures that the number of compromised servers will not exceed its preset quantity.

To maximize the objective function, the optimization algorithm is usually used to solve this problem. However, classical algorithms have some shortcomings of high computational complexity and low convergence rate [45], and are incapable of satisfying the needs for quick decision-making in this attack-defense scenario. In this paper, the probabilities that represent dynamic beliefs affect attackers' actions in each iteration, which is eventually reflected in both players' payoffs and the equilibria of the signaling game model. Based on the analysis of the game model and the solution to the PBNE, therefore, we design an algorithm to help reshape the defense-attack interplay and provide guidance for the defender to resist DDoS attacks in a cost-effective manner. The optimal strategy selection algorithm (OSA) for the defender is presented as Algorithm 1. It is important to note that the strategy is represented as decision pair where d_i indicates the defensive action and σ_i denotes the probability.

First, there exists a holistic view of the servers and initializes the multistage signaling game in Line 1, and Line 2 formulates the payoff matrix. Then, the defender estimates the attacker's belief in Line 4. Subsequently, if there exists an extreme case in which attackers execute a completely reverse strategy or all the decoys have failed to attract DDoS attackers (shown in Line 6), IP randomization is performed to all servers with a certain probability $\sigma_i = 1$ in Line 7. In other cases, dynamic redirection and response time adaptation will be used to separately disguise two types of servers when the objective function has not reached the optimal value ($|\sum_{i=1}^{N}\theta_i l_i - \sum_{i=1}^{N} l_i| \geq \varepsilon$, where ε is a very small value). In detail, the decision of dynamic redirection is given in Line 10 and 11, in which a part of real servers are randomly selected to send deceptive signals through dynamic redirection and the ratio of selection is based on the average probability of all real servers. Similarly, Line 13 and 14 show that some honeypots will be configured to act like a real server by response time adaptation with its probability that can be calculated based on Eq. 15. Moreover, Line 16 indicates that the strategy will not be updated when the defense payoffs have been maximized. Finally, the optimal strategy of each server at this time step is returned in Line 17.

The complexity of our work is analyzed by individually considering the complexities of three parts: payoff formulation, belief estimation, and strategy generation. Here, we use $\mathcal{N}$ and $\mathcal{M}$ to denote the number of states and the maximum number of actions per state, respectively. First, the payoff matrix is calculated based on some game parameters in Line 2, and the complexity is $\mathcal{O}(1)$. Then, in the belief estimation stage, as Line 4 needs to consider all states of senders, the complexity is $\mathcal{O}(\mathcal{N})$. Moreover, the strategy generation relies on conditional judgment which takes not only all states but also possible actions into consideration, hence, the associated complexity is given by $\mathcal{O}(\mathcal{MN})$. Based on the above analysis, we can conclude that the computational complexity of the proposed algorithm is $\mathcal{O}(\mathcal{MN})$.

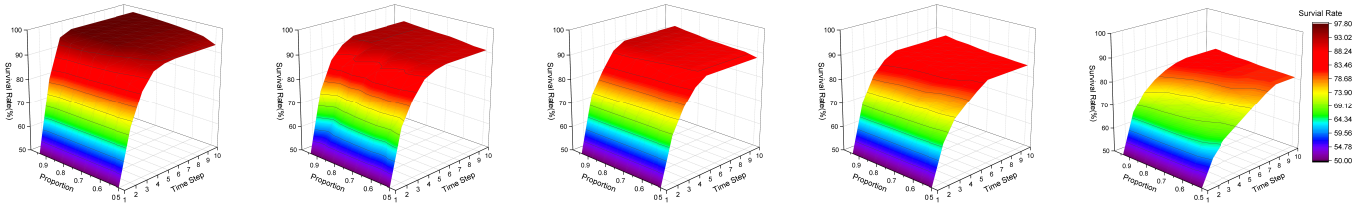
It should be noted that IP randomization in traditional MTD techniques may fail due to the possible prediction of IP address allocation patterns [46], [47], but Algorithm 1 does not rely on the fact that IP addresses cannot be discovered. First, the key insight of this paper is driving the attackers to mistakenly recognize the decoys as their targets through a combination of three kinds of MTD mechanisms, rather than continuously moving the target to distract attackers only by IP randomization. Second, aggressive attackers may not accurately recognize the honeypot from the beginning, but they can launch attacks on a large number of targets at the same time regardless of the beliefs based on their obtained knowledge, which increases the probability of real servers being attacked. Thus, the goal of IP randomization in this paper is to refresh all servers' IP addresses when necessary (e.g., attackers no longer access most of the decoys) and rebuild a new game model. Third, even if rational attackers have the ability to predict the next round of IP addresses, the IP address allocation for each type of server is unknown, therefore, they will have to study and estimate the types of newly-discovered servers. In general, Algorithm 1 in the proposed defense framework will guide the defender to continuously send deceptive signals to confuse attackers, and repeatedly refresh IP addresses to reconstruct the game model and regain the advantage in the interaction.

D. Feasibility Analysis

In this subsection, we discuss the rationality and feasibility of the proposed framework. We regularly consider that the switches can connect to IoT devices via protocols such as 6LowPAN, Bluetooth, etc., or wired [48]. The SDN controller can utilize southbound protocols (e.g., OpenFlow) to connect to switches while communicating with clouds via northbound APIs [49]. It is impractical to embed security applications in devices due to limited resources. Hence, the decision and deployment modules in the proposed method run as applications on top of the controller, which can be flexibly deployed and adapted in real-time without service interruption.

To compromise the target servers, rational attackers perform reconnaissance to discover their potential targets. As discussed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security



(a) 64 Servers (b) 128 Servers (c) 256 Servers (d) 512 Servers (e) 1024 Servers

Fig. 3. The average survival rate of production systems under different proportions of honeypots and scales.

in [50], the uncertainty generated by the proactive defense can affect the attacker's behavior for the benefit of the defender. Strategic information disclosure in the defender-led signaling game model is an effective method to reverse information asymmetry. In the proposed framework, we improve the signaling game model by changing from one stage game to a multi-stage one to optimize the attacker's belief function. Furthermore, we introduce several defense mechanisms to transmit signals to confuse attackers, so that the model can be used in realistic scenarios. As we show in Sections IV-A and IV-B, the proactive defense boils down to adapting the time or the priority of service response while the operations can be easily realized by the SDN controller and the deceptive message can be conveyed to the attacker at a very low cost.

However, in some attack scenarios, aggressive attackers can invade multiple targets or even all targets simultaneously, and launch DDoS attacks continually without inspection and analysis. And we observe that it is extremely difficult to resist aggressive attacks under the traditional signaling game model, that is, attackers can eventually hit their targets when there is no limitation of attack capability. Thus, in the proposed framework, another proactive defense mechanism has been added as a solution to prevent attackers who rarely follow the beliefs. To deal with uncontrolled behaviors and include all attack scenarios, we provide the IP randomization that revises the parameters of protected systems to reconstruct the game model when most production systems have been compromised. Based on the above analysis, the proposed framework can be suitable for all kinds of attack scenarios and helpful for effective strategy generation.

V. Evaluations

In this section, we analyze and discuss the performance of the defensive framework in DDoS mitigation through simulation and experiments. We first introduce the simulation configurations and determine the parameters by comparing the results under different conditions. Then, we deploy the proposed method on a real-world testbed, and finally, the effectiveness and cost of the proposed OSA algorithm are compared with other strategies.

A. Evaluation Metrics

To simplify the analysis, the quantitative measures of game parameters are introduced as follows. In detail, gain for a production system g_p is set to a normalized value 1, and g_h equals ν, where ν is a ratio coefficient of resources between a honeypot and a production system. Especially, g_a can be formulated by $1-\omega$, where ω represents the ratio of idle system resources of the target server. In addition, c_p and c_h can be calculated by the ratio of the flow entries of defense implementation to all current flow entries, and c_a is defined as the ratio of attack traffic rate to the maximum bandwidth.

Definition 2. *We define the survival rate $P_S(t)$ as the percentage of production systems which are free from attacks several time steps after the start of the multistage signaling game.*

In this paper, we focus on whether the cloud servers are under DDoS attacks. Thus, the survival rate $P_S(t)$ is the first sign of successful defense when the attacker hits more honeypots rather than real servers. Moreover, we define the following metrics from the aspects of time overhead and iterations of the game to directly evaluate the efficiency.

Definition 3. *The temporal metric T_D is the time needed by the defense framework, which can be divided into two parts: decision time and deployment time.*

Definition 4. *The metric N_R is the number of game reconstructions caused by the optimal strategy selection algorithm.*

In our scenario, according to the difference between the time when the framework starts to make a decision and the time when the defense has been implemented, we can quantitatively measure the time cost T_D. Besides, the reconstruction times N_R can intuitively show the effectiveness in a period. Moreover, the defense implementation would occupy a small part of resources and slightly affect the quality of original services. Thus, we utilize the metrics of throughput, packet loss rate, CPU load, and Round-Trip Time (RTT) to evaluate the performance degradation caused by the framework.

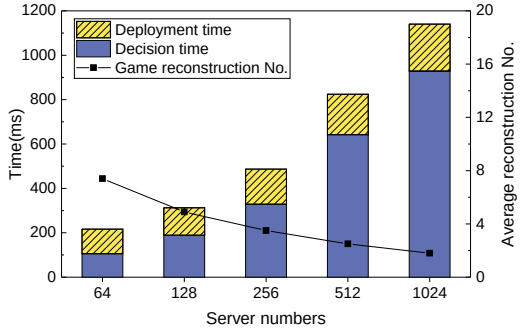
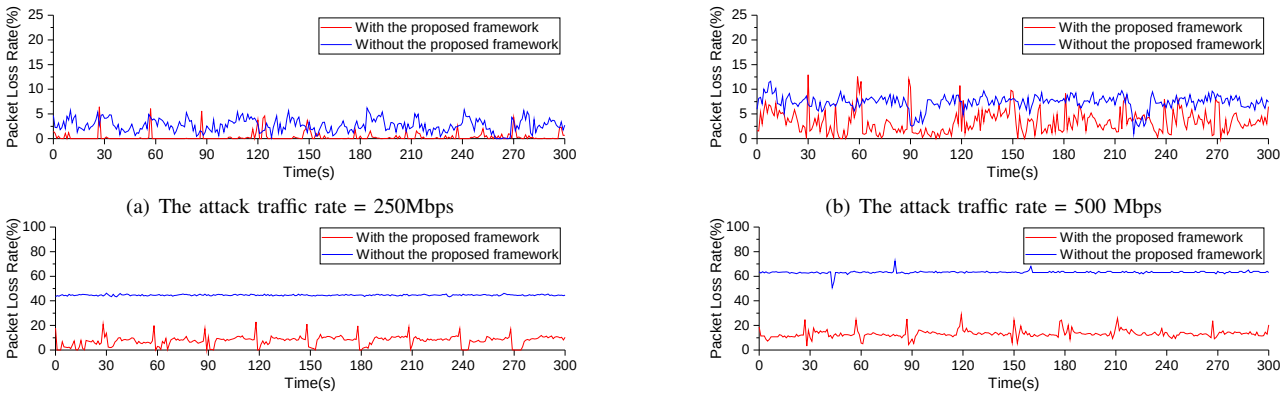


Fig. 4. The comparison of time spent and reconstruction numbers while deploying defense on different scales.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

TABLE III
AVERAGE THROUGHPUT OF THE SDN-BASED IOT CLOUD WHEN IMPLEMENTING DEFENSE WITH DIFFERENT SCALES AND INTERVALS.

Time interval	64 Servers		128 Servers		256 Servers		512 Servers		1024 Servers	
	Avg.(Mbps)	StdDev.	Avg.(Mbps)	StdDev.	Avg.(Mbps)	StdDev.	Avg.(Mbps)	StdDev.	Avg.(Mbps)	StdDev.
5s	947.44	23.34	944.23	42.88	937.12	39.44	931.52	46.89	927.84	51.89
10s	951.63	23.89	951.37	21.33	947.59	41.36	942.32	35.96	939.20	46.98
15s	954.30	10.73	954.13	12.93	950.49	38.39	945.56	35.79	941.96	37.46
30s	955.51	5.66	955.43	8.34	952.35	6.60	949.32	13.13	944.72	15.59
60s	956.06	4.03	955.56	12.44	954.63	22.23	951.60	30.59	947.68	14.58



(a) The attack traffic rate = 250Mbps

(b) The attack traffic rate = 500 Mbps

(c) The attack traffic rate = 750 Mbps

(d) The attack traffic rate = 1 Gbps

Fig. 5. The average packet loss rate under different rates of attack traffic and defense deployments.

B. Simulation

In the simulation, we use Mininet to build the SDN-based IoT network on our testbed containing multiple instances of OpenFlow switches and hosts equipped with Intel Xeon Processor E5-2680 of 16 cores, 64 GB RAM, and 2 TB hard disk. Furthermore, we implement proactive defense applications and the OSA algorithm on the Ryu controller that runs on Ubuntu 18.04 with 4 cores, 16 GB of RAM, and 120 GB hard disk.

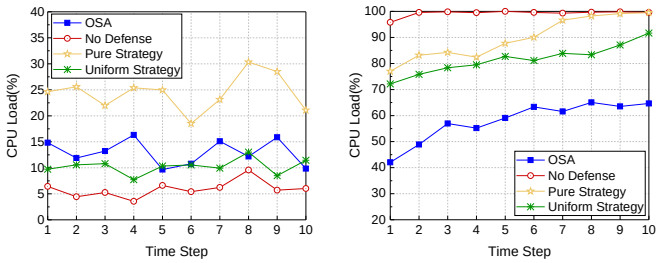
We deploy a set of honeypots among all hosts, and these honeypots are allocated fewer system resources. Each host is equipped with a monitor agent to log the status, and we set the bandwidth of each link as 1Gbps. Moreover, we update the flow tables and enable proactive defense via the SDN controller. Then, we determine the optimal parameters according to the simulation results under different configurations.

We first implement defense with the proportion of honeypots ranging from 50% to 93.75% (there is no network full of honeypots), in which the attacker can attack half of all servers at each time. Fig. 3 denotes the survival rate with a different scale under the framework. In these figures, the x-axis presents the proportion of honeypots, and the y-axis denotes the time step in the signaling game. Then, the decision frequency is set from 5s to 60s. Table III displays the average and standard deviation of throughput in different scales and intervals. The proportion of honeypots and the decision interval have an inessential influence on the survival rate and throughput, but they are greatly affected by the scale of the system. However, the higher proportion of honeypots means the higher the cost, and high-frequency decision-making also brings a greater burden on the controller. Thus, we set the frequency of decision as 30s and 50% of servers implemented as decoys.

Then, we measure the time costs and number metrics defined in Section V-A to assess the efficiency of the framework. Fig. 4 displays the average time needed by strategy generation and defense deployment under a different scale. The decision time has significantly increased as the scale increases, and there is an unapparent increase in the deployment time depicted by the shadow area. Nevertheless, the total time incurred by the framework would not exceed 1,200 milliseconds, and it ensures timely decisions and rapid defense deployment. Moreover, the broken line denotes the number for reconstructing the multistage signaling game during the attack process. It can be seen that the attacker's ability to analyze defense is inversely proportional to the implementation scale, and the proposed method regains the initiative of the signaling game by randomly assigning the IP addresses of all servers.

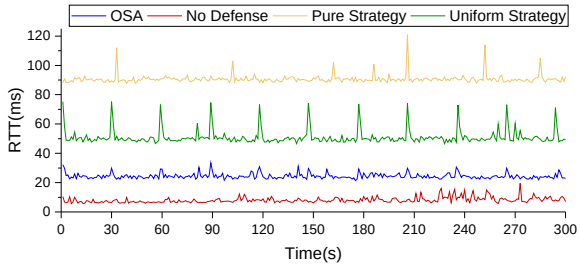
Finally, to verify the effectiveness of the defensive framework against attacks, a variety of DDoS attacks including SYN flood, UDP flood, and ICMP flood, are launched simultaneously using the hping3 tool for collectively flooding half of the network that contains 256 servers. As seen in Fig. 5, the packet loss rates increase with the upgrading of the attack rate in both scenarios. More specifically, in the case of no defense, the packet loss rate will remain at a higher level, and it stays almost similar throughout the attacks. However, the proposed method is effective in reducing the packet loss rate and keep it below 30% under 1Gbps of attack traffic. An important observation is that there is a peak value of the packet loss rate every 30 seconds under the proposed framework, which meets the time interval of decision-making. Therefore, the proposed framework can effectively defend against DDoS attacks within limited time costs and performance degradation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

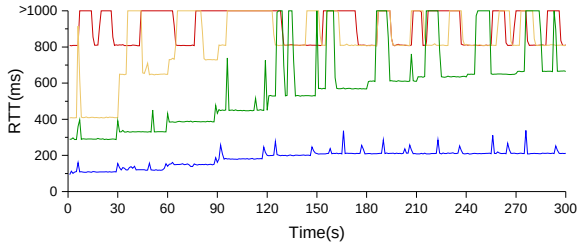


(a) Without attacks (b) Under DDoS attacks

Fig. 6. CPU utilization of SDN controller with different defense strategies.



(a) Without attacks



(b) Under DDoS attacks

Fig. 7. The comparison of average RTT with different defense strategies.

C. Experiments

1) Experimental setups: In our experiments, the framework is implemented in a real-world testbed, which is composed of 2 Dell PowerEdge R630 servers and a Dell PowerEdge R430 server. Each Dell PowerEdge R630 has a 24-core CPU, 64 GB of RAM, and 4 TB hard disk storage, and Dell PowerEdge R430 has a 4-core CPU, 16 GB RAM, and 120 GB disk storage.

The control platform is deployed with Ryu based SDN controller on the Dell PowerEdge R430 server, in which we implement the proposed strategy algorithm and MTD techniques as modules. To create the cloud environment, we utilize OpenStack to manage the virtualized computing and storage resources of the other two servers. According to the parameters obtained in Sections V-B, we create 32 one-core VMs (half of them are initialized as honeypots), and each VM is managed by SDN controller via Open vSwitch.

Usually, honeypots have fewer computing resources than real servers, thus, a production system in the experiments comes with 1 GB of RAM while a VM-based honeypot has 512 MB of RAM. To ensure its effectiveness in the real world, we deploy real services, including Apache HTTP server, Nginx server, Redis server, MySQL database server, and Mosquitto broker in both types of VMs. Similarly, we strike half servers within the network via the hping3 tool at each time step.

TABLE IV
SUMMARY OF MEMORY OVERHEAD WITH DIFFERENT SERVICES.

Service Type	Memory Consumption (KB)
Apache HTTP	4384
Nginx	6300
Redis	3796
MySQL	40484
Mosquitto	6064

Furthermore, we compare the proposed OSA algorithm with some baseline strategies. First, no defense means there exists no mechanism against DDoS attacks. Then, a pure strategy is commonly known as the fully reverse strategy. The defender adopts a determined configuration that all honeypots and production systems act like each other, and maintains this configuration as the game process continues. In addition, a uniform strategy modifies the configurations of all servers with equal probability at each time step. In detail, we randomly select 50% from each type of VMs to send deceptive signals to the attacker.

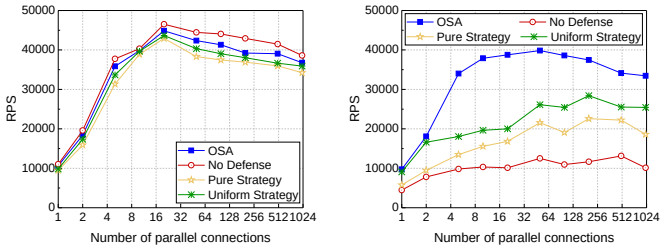
2) CPU utilization of SDN controller: To compare the processing overhead of defense strategies in different attack scenarios, we measure the CPU utilization of the SDN controller when deploying different strategies, which can be seen in Fig. 6. The CPU load of no defense strategy never exceeds 10% when there is no attack, while it surges to almost 100% at the beginning of DDoS attacks. Pure strategy selects totally reverse configurations, which inevitably results in the highest CPU utilization without attacks, ranging from 18.52% to 30.34%. However, it cannot resist DDoS attacks when the attacker has enough time to analyze the strategy. Although the overhead of uniform strategy is acceptable, its CPU load at the time of attacks still reaches about 90%. In our study, the overhead of the OSA strategy is similar to the uniform strategy. However, the proposed algorithm has excellent results that effectively limit the CPU load under DDoS attacks within 65% and ensure the normal operation of the IoT network.

3) RTT for production systems: Generally, DDoS attacks will lead to a long response time and even cause the service to fail to respond to requests in the real world. Accordingly, we conduct a set of experiments to compare the RTT of production systems under specified strategies in the normal and attack scenarios. In different defense strategies, we send ICMP packets to all production systems every second and measure the average RTT in 300 seconds. In terms of the normal scenario in Fig. 7, the RTT of no defense strategy is constant at about 10 milliseconds, while the pure strategy has the highest RTT, fluctuating around 90 milliseconds. For the attack scenario, an interesting observation is that pure strategy, uniform strategy, and OSA all show a gradual upward trend in RTT over time. However, the RTT of no defense exceeds 1,000 milliseconds when DDoS attacks occur. Another important observation is that the OSA algorithm outperforms other strategies under attacks, in which the average RTT of OSA is almost 200 milliseconds that can maintain normal communications.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

TABLE V
AVERAGE PERFORMANCE OVERHEAD OF WEB SERVICES WHEN IMPLEMENTING DEFENSE WITH DIFFERENT STRATEGIES.

Defense strategy	Apache HTTP server		Nginx server	
(a) Without attacks	RPS	TPR (ms)	RPS	TPR (ms)
OSA	1747.44	286.13	4814.80	103.85
No Defense	1855.37	269.49	4931.19	101.39
Pure Strategy	1473.06	339.43	4130.70	121.05
Uniform Strategy	1534.55	315.71	4586.90	109.01
(b) Under attacks	RPS	TPR (ms)	RPS	TPR (ms)
OSA	1406.08	355.59	4280.45	111.59
No Defense	361.55	1382.95	1103.72	584.01
Pure Strategy	504.58	990.93	1712.29	453.01
Uniform Strategy	1073.50	465.77	2700.93	185.12



(a) Without attacks (b) Under DDoS attacks

Fig. 8. The average RPS of Redis with different connections and strategies.

4) Performance and overhead for real services: We also measure the memory overhead introduced by different services. As shown in Table IV, most application processes in the experiments consume thousands of kilobytes of memory, which is moderately acceptable for real services.

In order to evaluate the performance overhead, we first create 10,000 requests originating from 500 connections (that represent multiple IoT devices) with the Apache Benchmark tool. Then, we measure the requests per second (RPS) and time per request (TPR) of two kinds of web services. According to the results displayed in Table V, the defense strategy has an insignificant influence on the performance of web services under normal circumstances. This indicates that the deployment of OSA does not affect the normal service. In the case of a DDoS attack scenario, all strategies except OSA will cause a dramatic drop in the performance of both web services. However, the proposed OSA strategy can effectively resist DDoS attacks, meanwhile, the performance degradation of web services is less than 20%.

We then evaluate the performance of the Redis server in the conditions of different parallel connections and strategies. Similarly, the RPS can be measured with the Redis Benchmark tool by simulating 10,000 requests from several IoT devices. As summarized in Fig. 8, the RPS of all strategies can reach almost 45,000 when connecting with 20 devices parallelly in the normal case, and then gradually decline as more IoT devices request services from clouds. In the case of DDoS attacks, there is a certain degree of decline in the RPS of each strategy. Compared with other strategies, the OSA strategy can maintain a higher level of performance and in the meanwhile achieve a better defense against DDoS attacks.

TABLE VI
TIME OVERHEAD OF MYSQL SERVER TO RUN ALL QUERIES WITH DIFFERENT DEFENSE STRATEGIES.

Defense strategy	Without attacks		Under attacks	
	Avg.(ms)	StdDev.	Avg.(ms)	StdDev.
OSA	1239.62	67.06	1731.15	65.21
No Defense	1183.52	41.09	8019.25	366.22
Pure Strategy	1825.40	83.24	4197.65	200.79
Uniform Strategy	1415.45	84.04	2666.50	49.52

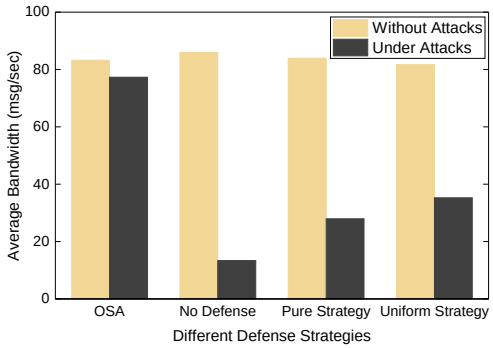


Fig. 9. The average bandwidth of Mosquitto server with different strategies in different attack scenarios.

Furthermore, we also initiate 10,000 requests from 50 connections and measure the time overhead of the MySQL database server. Table VI displays the average and standard deviation of the time to run all queries in different conditions. It can be seen that no defense deployment has the lowest time overhead in the normal case and the proposed OSA algorithm almost needs extra 50 milliseconds which is acceptable compared to other strategies. When suffering from attacks, no defense means no resistance to DDoS attacks, resulting in an extremely high response time that exceeds 8,000 milliseconds. Due to the resource consumption caused by DDoS attacks, the pure strategy and uniform strategy have also resulted in exponential growth in time overhead. Nevertheless, it takes only 1731.15 milliseconds when running 10,000 queries with OSA under attacks. This indicates that the proposed algorithm can obtain a good balance between effectiveness and overhead when defending against DDoS attacks.

Finally, a Message Queuing Telemetry Transport (MQTT) broker is benchmarked in the IoT environment. We implement Mosquitto [51], which provides standard implementations of server and client under the MQTT protocol, and evaluate the bandwidth when each of 100 clients sends 100 messages with 100 bytes of payload. Fig. 9 shows the average received messages by the Mosquitto broker every second among different strategies. We can find that the MQTT broker can reach an average bandwidth of about 83 msg/sec when adopting any kind of strategy in normal circumstances. However, the broker cannot maintain high performance without defense deployed when the server is saturated with DDoS attack traffic, and the average bandwidth is acutely dropped to 13.385 msg/sec. Even if the pure or uniform strategy has been adopted, there is a negative effect on the performance while the bandwidth is still up to 77.306 msg/sec under the proposed OSA strategy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIFS.2021.3127009, IEEE Transactions on Information Forensics and Security

TABLE VII
COMPARISON BETWEEN FASTMOVE AND THE PROPOSED METHOD.

Aspects	FastMove	The proposed method
Main assumption	Attack targets will be lost as the proxies change	Attackers determine attack targets based on their beliefs
Key method	High-frequency IP address shuffling mechanism	Combination of MTD and cyber deception with signaling game
Working scenario	Web services	Any service in the SDN-enabled IoT networks
Survival rate	Range from 2% to 98%	Range from 80% to 98%
CPU utilization	N/A	Additional CPU utilization $< 8\%$
Round-Trip Time	N/A	RTT increase $\approx 10ms$
Service performance	N/A	Performance degradation $\leq 6.2\%$
Evaluation method	NS-3 simulation	Mininet simulation and real-world testbed

VI. LIMITATIONS AND DISCUSSIONS

A. Limitations

Defense against powerful attackers. In this work, we have only taken into account attackers who own resources to launch attacks towards a part of servers. However, cyber deception will fail in the face of very powerful attackers because they choose to strike all scanned network resources regardless of cost limitation. To this limitation, we will incorporate other proactive defense techniques (e.g., VM/container migration) into the framework and redesign the defense strategy to resist DDoS attacks caused by extremely powerful attackers. Due to the complexity and overhead of live migration, there is also a challenge on how to effectively deploy it without affecting the performance of systems.

Resisting against insider attacks. The proposed framework only supports effective defense against external attacks on the IoT network. However, internal servers sometimes may be hijacked. For instance, an attacker usually seizes control of servers by Trojan or code injection. However, it is difficult to defeat insider attacks, as legitimate users and attackers cannot be distinguished at the beginning of attacks. Another reason is that the infected servers may spread malicious applications to compromise other servers, and the attacker can obtain the characteristics of the proactive defense in silence. These problems will be studied and solved in our future works.

B. Compared with Previous Works

Comparison with FastMove. Previous research [21] proposed an MTD-based DDoS mitigation method named FastMove. The main insight of it is implementing a new proxy layer that covers servers, and creating an IP shuffling mechanism through fast-flux DNS technique to separate attackers from benign users and defeat DDoS attacks. By contrast, the proposed method is not limited to deploying IP shuffling to reconstruct the game when the attacker penetrates the previous defense, but also involves camouflaging the properties via response time adaptation and dynamic redirection to manipulate the attacker's perspectives among real servers and decoys.

FastMove mainly changes the IP addresses to prevent attack traffic from reaching the original web servers, the simulation result could achieve the highest survival rate of 98% with 200 clients and only one bot-master, but the result will drop to 2% when facing 50 bot-masters among 700 clients. By contrast, the proposed method could have a higher survival rate ($> 80\%$) even in the face of attacks in a large-scale network environment. In addition, the shuffling of multiple proxies in FastMove requires additional resources, but its defense cost and QoS degradation are not quantitatively evaluated in their paper. According to the evaluation results, the proposed method is able to enhance security and maintain performance at the same time. The performance degradation and additional CPU utilization are both less than 8%, and the RTT is only increased by about 10 milliseconds, which is acceptable when considering the effectiveness of DDoS mitigation. We summarize the differences on Table VII, it can be found that while both methods are based on proactive defense mechanisms, their application scenarios are quite different. The proposed framework is at a more mature level than FastMove and more suitable for the IoT environments.

Comparison with deception methods. Several deception-based DDoS mitigation methods have also been proposed, such as Honeyman [41] and Pseudo-Honeypot [5]. These methods attempt to create confusion to adversaries only through deploying honeypots. However, in this study, we do not propose to use traditional honeypots for cyber deception, instead, the proposed method aims to utilize MTD mechanisms to modify the characteristics of servers and increase the difficulty of distinguishing between real systems and honeypots. Since there is no evaluation result in Honeyman, we make a quantitative comparison between Pseudo-Honeypot and the proposed method, which is shown in Table VIII. Pseudo-Honeypot mitigates DDoS attacks by deploying low-interaction honeypots and high-interaction pseudo-honeypots simultaneously and generating strategy with game theory. The attack detection rate via honeypots is almost 60% in the study of Pseudo-Honeypot, while the proposed method can capture more than 80% of attacks because it can change the characteristics of systems through MTD techniques to avoid being recognized by attackers. In the case where all types of honeypots account for 50%, the average energy consumption of Pseudo-Honeypot is about 1J. However, the proposed method does not need to deploy high-interaction honeypots, and it will consume less energy when deploying the same number of decoys.

In addition, due to the limitation of traditional honeypots (e.g., low-interaction honeypot cannot obtain enough information from adversaries), collecting attackers' comprehensive information without being discovered in the honeypots is not easy. For instance, Honeyman captures data through recurrent neural networks and feed the data back to update existing honeypots [41], which takes a large amount of resources. So

TABLE VIII
COMPARISON BETWEEN PSEUDO-HONEYPOT AND THE PROPOSED METHOD.

Aspects	Pseudo-Honeypot	The proposed method
Main assumption	Attackers recognize the types of systems before attacks	Attackers determine attack targets based on their beliefs
Key method	Honeypots and pseudo-honeypots with game theory	Combination of MTD and cyber deception with signaling game
Working scenario	Industrial IoT networks	Any service in the SDN-enabled IoT networks
Detection rate	$\approx 60\%$	$> 80\%$
Energy consumption	$\approx 1J$	$< 1J$
Evaluation method	SDN testbed	Mininet simulation and real-world testbed

instead of studying high-interaction honeypots, the technical contribution of the proposed method is strategically adapting the network properties to manipulate the attacker's judgment on production systems and honeypots.

VII. CONCLUSION AND FUTURE WORK

We present a lightweight hybrid framework for automatically generating, deploying, and adjusting proactive defense in the IoT environment, in which all defense mechanisms can be flexibly deployed and easily integrated by leveraging the SDN architecture. Based on MTD techniques and cyber deception, we establish hybrid defense mechanisms to induce attackers to take wrong actions and drain their resources within limited performance degradation. To comprehensively depict the interactions between the defender and the attacker, we introduce a defender-led multistage signaling game model to formalize the attack-defense processes and analyze the payoffs of both players. Further, we present an optimal strategy selection algorithm to rapidly reach a balance between effectiveness and cost, and optimize the defense implementation. The evaluation results indicate that the proposed method can effectively provide excellent performance in resisting DDoS attacks while still providing high-quality and low-delay network services in IoT networks.

In our future work, we will first incorporate more security mechanisms for providing real-world products in a more flexible manner. Second, designing an efficient defense scheduling method for providing stable service for IoT devices is also a necessity. Third, previous works focus on defeating external attackers, however, insider attacks can hardly be defended. Therefore, exploiting proactive adaptation to distinguish malicious attackers from legitimate users will be another significant research direction.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Program of China (Grant No. 2020YFB1804604) and in part by the General Program of the National Natural Science Foundation of China (Grant No. 62172093).

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] O. Salman, I. H. Elhajj, A. Chehab, and A. Kayssi, "Iot survey: An sdn and fog computing perspective," *Computer Networks*, vol. 143, pp. 221–246, 2018.

[3] X. Guo, H. Lin, Z. Li, and M. Peng, "Deep-reinforcement-learning-based qos-aware secure routing for sdn-iot," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6242–6251, 2019.

[4] I. Farris, T. Taleb, Y. Khettab, and J. Song, "A survey on emerging sdn and nfv security mechanisms for iot systems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 812–837, 2018.

[5] M. Du and K. Wang, "An sdn-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 648–657, 2019.

[6] S. Yu, S. Guo, and I. Stojmenovic, "Fool me if you can: Mimicking attacks and anti-attacks in cyberspace," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 139–151, 2013.

[7] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of ddos attack in iot via sdn-cloud architecture," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3559–3570, 2020.

[8] Z. Abou El Houda, A. Hafid, and L. Khoukhi, "Co-iot: a collaborative ddos mitigation scheme in iot environment based on blockchain using sdn," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[9] J. Zheng and A. S. Namin, "Defending sdn-based iot networks against ddos attacks using markov decision process," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 4589–4592.

[10] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat ddos attacks in clouds?" *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2245–2254, 2013.

[11] Z. Zhou, C. Xu, X. Kuang, T. Zhang, and L. Sun, "An efficient and agile spatio-temporal route mutation moving target defense mechanism," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[12] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 3, pp. 1909–1941, 2020.

[13] H. Jin, Z. Li, D. Zou, and B. Yuan, "Dseom: A framework for dynamic security evaluation and optimization of mtd in container-based cloud," *IEEE Transactions on Dependable and Secure Computing*, 2019.

[14] Y. Zhou, G. Cheng, S. Jiang, Y. Zhao, and Z. Chen, "Cost-effective moving target defense against ddos attacks using trilateral game and multi-objective markov decision processes," *Computers & Security*, p. 101976, 2020.

[15] M. Khosravi-Farmad, A. A. Ramaki, and A. G. Bafghi, "Moving target defense against advanced persistent threats for cybersecurity enhancement," in *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*. IEEE, 2018, pp. 280–285.

[16] C. Wang and Z. Lu, "Cyber deception: Overview and the road ahead," *IEEE Security & Privacy*, vol. 16, no. 2, pp. 80–85, 2018.

[17] J. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Benasher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[18] W. K. K. Oo and H. Koide, "A framework of moving target defenses for the internet of things," *Bulletin of Networking, Computing, Systems, and Software*, vol. 8, no. 2, pp. 104–107, 2019.

[19] W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An iot honeynet based on multiport honeypots for capturing iot attacks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3991–3999, 2020.

[20] S. Yao, Z. Li, J. Guan, and Y. Liu, "Stochastic cost minimization mechanism based on identifier network for iot security," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3923–3934, 2019.

[21] N. Bandi, H. Tajbakhsh, and M. Analoui, "Fastmove: Fast ip switching moving target defense to mitigate ddos attacks," in *2021 IEEE Conference on Dependable and Secure Computing (DSC)*. IEEE, pp. 1–7.

[22] H. Alavizadeh, D. S. Kim, and J. Jang-Jaccard, "Model-based evaluation of combinations of shuffle and diversity mtd techniques on the cloud," *Future Generation Computer Systems*, vol. 111, pp. 507–522, 2020.

[23] M. Torquato, P. Maciel, and M. Vieira, "Security and availability modeling of vm migration as moving target defense," in *2020 IEEE 25th Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 2020, pp. 50–59.

[24] I. Butun, P. Österberg, and H. Song, "Security of the internet of things: Vulnerabilities, attacks, and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 616–644, 2019.

[25] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[26] K. Haseeb, N. Islam, A. Almogren, and I. U. Din, "Intrusion prevention framework for secure routing in wsn-based mobile internet of things," *Ieee Access*, vol. 7, pp. 185 496–185 505, 2019.

[27] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[28] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in iot using sdn," in *2017 27th International telecommunication networks and applications conference (ITNAC)*. IEEE, 2017, pp. 1–6.

[29] W. Liu, M. Ge, and D. S. Kim, "Integrated proactive defense for software defined internet of things under multi-target attacks," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 2020, pp. 767–774.

[30] Y. Zhou, G. Cheng, Y. Zhao, Z. Chen, and S. Jiang, "Towards proactive and efficient ddos mitigation in iiot systems: A moving target defense approach," *IEEE Transactions on Industrial Informatics*, 2021.

[31] K. Mahmood and D. M. Shila, "Moving target defense for internet of things using context aware code partitioning and code diversification," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE, 2016, pp. 329–330.

[32] H. Koo, Y. Chen, L. Lu, V. P. Kemerlis, and M. Polychronakis, "Compiler-assisted code randomization," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 461–477.

[33] X. Wang, S. Yeoh, R. Lyerly, P. Olivier, S.-H. Kim, and B. Ravindran, "A framework for software diversification with {ISA} heterogeneity," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020)*, 2020, pp. 427–442.

[34] D. Evans, A. Nguyen-Tuong, and J. Knight, "Effectiveness of moving target defenses," in *Moving target defense*. Springer, 2011, pp. 29–48.

[35] Z. Zhang, L. Njilla, C. A. Kamhoua, and Q. Yu, "Thwarting security threats from malicious fpga tools with novel fpga-oriented moving target defense," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 3, pp. 665–678, 2018.

[36] A. Kanellopoulos and K. G. Vamvoudakis, "A moving target defense control framework for cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1029–1043, 2019.

[37] C.-Y. J. Chiang, S. Venkatesan, S. Sugrim, J. A. Youzwak, R. Chadha, E. I. Colbert, H. Cam, and M. Albanese, "On defensive cyber deception: A case study using sdn," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*. IEEE, 2018, pp. 110–115.

[38] M. Winn, M. Rice, S. Dunlap, J. Lopez, and B. Mullins, "Constructing cost-effective and targetable industrial control system honeypots for production networks," *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 47–58, 2015.

[39] J. D. Guarnizo, A. Tambe, S. S. Bhunia, M. Ochoa, N. O. Tippenhauer, A. Shabtai, and Y. Elovici, "Siphon: Towards scalable high-interaction physical honeypots," in *Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security*, 2017, pp. 57–68.

[40] A. Tambe, Y. L. Aung, R. Sridharan, M. Ochoa, N. O. Tippenhauer, A. Shabtai, and Y. Elovici, "Detection of threats to iot devices using scalable vpn-forwarded honeypots," in *Proceedings of the Ninth ACM Conference on Data and Application Security and Privacy*, 2019, pp. 85–96.

[41] P. J. Hanson, L. Truax, and D. D. Saranchak, "Iot honeynet for military deception and indications and warnings," in *Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything*, vol. 10643, 2018.

[42] M. M. Islam and E. Al-Shaer, "Active deception framework: an extensible development environment for adaptive cyber deception," in *2020 IEEE Secure Development (SecDev)*. IEEE, 2020, pp. 41–48.

[43] A. Aydeger, M. H. Manshaei, M. A. Rahman, and K. Akkaya, "Strategic defense against stealthy link flooding attacks: A signaling game approach," *IEEE Transactions on Network Science and Engineering*, 2021.

[44] J. Pawlick, E. Colbert, and Q. Zhu, "Modeling and analysis of leaky deception using signaling games with evidence," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1871–1886, 2019.

[45] W. Deng, J. Xu, and H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," *IEEE access*, vol. 7, pp. 20 281–20 292, 2019.

[46] J. Ullrich, P. Kieseberg, K. Krombholz, and E. Weippl, "On reconnaissance with ipv6: a pattern-based scanning approach," in *2015 10th International Conference on Availability, Reliability and Security*. IEEE, 2015, pp. 186–192.

[47] H. M. Almohri, L. T. Watson, and D. Evans, "Predictability of ip address allocations for cloud computing platforms," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 500–511, 2019.

[48] M. Ojo, D. Adami, and S. Giordano, "A sdn-iot architecture with nfv implementation," in *2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2016, pp. 1–6.

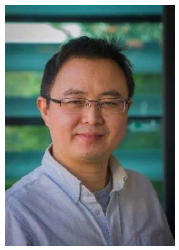
[49] T. G. Nguyen, T. V. Phan, B. T. Nguyen, C. So-In, Z. A. Baig, and S. Sanguanpong, "Search: A collaborative and intelligent nids architecture for sdn-based cloud iot networks," *IEEE access*, vol. 7, pp. 107 678–107 694, 2019.

[50] X. Feng, Z. Zheng, D. Cansever, A. Swami, and P. Mohapatra, "A signaling game model for moving target defense," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[51] R. A. Light, "Mosquitto: server and client implementation of the mqtt protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.

Yuyang Zhou is currently pursuing the Ph.D. degree in the School of Cyber Science and Engineering, Southeast University. His research interests include moving target defense, security modeling, and application of software defined networking.

Guang Cheng received the B.S. degree in Traffic Engineering from Southeast University in 1994, the M.S. degree in Computer Application from Hefei University of Technology in 2000, and the Ph.D. degree in Computer Network from Southeast University in 2003. He is a full professor in the School of Cyber Science and Engineering, Southeast University, Nanjing, China. Dr Cheng's research interests include network security, network measurement and traffic behavior analysis. He has published seven monographs and more than 100 technical papers. He is a Member of IEEE and a Senior Member of CCF.



Shui Yu obtained his PhD from Deakin University, Australia, in 2004. He currently is a Professor of School of Computer Science, University of Technology Sydney, Australia. Dr Yu's research interest includes Big Data, Security and Privacy, Networking, and Mathematical Modelling. He has published three monographs and edited two books, more than 400 technical papers, including top journals and top conferences, such as IEEE TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. His h-index is 53. Dr Yu initiated the research field of networking for big data in 2013, and his research outputs have been widely adopted by industrial systems, such as Amazon cloud security. He is currently serving a number of prestigious editorial boards, including IEEE Communications Surveys and Tutorials (Area Editor), IEEE Communications Magazine, IEEE Internet of Things Journal, and so on. He is a Senior Member of IEEE, a member of AAAS and ACM, and a Distinguished Lecturer of IEEE Communications Society.