An SDN-Enabled Pseudo-Honeypot Strategy for Distributed Denial of Service Attacks in Industrial Internet of Things

Miao Du and Kun Wang, *Senior Member, IEEE*

***Abstract*—Leveraging high-performance software-defined networks (SDNs) to manage industrial Internet of Things (IIoT) devices has become a promising trend; the SDN is expected to be the next generation as a unified and virtualized network platform that provides unprecedented automation, flexibility, and efficiency. As the core of business applications and sensitive data storage, the SDN is vulnerable to distributed denial-of-service (DDoS) attacks in IIoT environment that numerous requests are sent to the SDN to interrupt its services. In the traditional defense systems, honeypots have shown great promises in resisting DDoS attacks. In this paper, we reveal a new attack that can identify honeypots to invalidate their protection. In addition, we analyze the optimal strategies of attackers, so that they can find the best time to carry on attacks. To protect SDN from such a kind of anti-honeypot attacks, we propose a pseudo-honeypot game (PHG) strategy with theoretical performance guarantee. We prove several groups of Bayesian–Nash Equilibrium in the PHG strategy. Moreover, we show that these strategies can achieve the optimal equilibrium between legitimate users and attackers. The proposed honeypot strategies can provide dynamic protection for SDN. Hence, malicious attacks under our strategies can be effectively controlled. Finally, we evaluate our proposals on a testbed, and experimental results show that our proposals can effectively resist DDoS attacks with lower energy consumption compared with the existing methods.**

***Index Terms*—Anti-honeypot, distributed denial of service attacks, game theory, industrial Internet of Things (IIoT), pseudo-honeypot, software-defined networks (SDN).**

I. Introduction

SOFTWARE-DEFINED network (SDN) is an automatic and virtualized network with high flexibility, efficiency, and reliability. It carries the core business and stores confidential data of the users, while providing business interaction and data exchange for internal, external, and other customers [1]. Therefore, SDN can effectively manage industrial Internet of Things (IIoT) devices, protect the data they generate, and quickly analyze the data to provide the visibility that enterprises need. SDN provides a viable, cost-effective way to manage IIoT to maximize application and analytics performance. Due to the explosive growth of IIoT and the chaotic nature of the public Internet, traffic in this area needs to be migrated to private dedicated channels, otherwise critical communication services and applications will experience latency issues.

On the other hand, SDN enables the network to be automated and centrally managed so that it can quickly configure IIoT devices around the world. The inherent scalability of SDN allows for the rapid addition of new IIoT devices, and the dynamic response system greatly reduces the risk of IIoT. In addition, virtualization of SDN components enables dynamic reconfiguration of network devices and traffic, automatic bandwidth provisioning, and de-provisioning of bandwidth. Therefore, as IIoT traffic grows, high-traffic instantaneous bandwidth or traffic involving health and security applications will be prioritized.

However, the centralized control of SDN puts the controller at risk of a single point of failure. First, the centralized control of the controller makes the controller easy to be the target of attack. Once the attacker successfully implements the attack on the controller, it will cause a large area of network service and affect the entire network coverage covered by the controller. Second, the centralized control makes the controller vulnerable to resource exhaustion attacks such as distributed denial-of-service (DDoS). DDoS attacks injects a large number of packets with forged source addresses into the SDN servers [2]. Since no matched rules can be found in local caches, switches forward these malicious packets to the network controller, which will be congested to interrupt services. More seriously, once the DDoS attack succeeds, it is likely that the attacked SDN server will be used as a puppet to further attack IIoT devices managed by SDN, resulting in huge industrial losses. In recent years, IIoT environments have been targeted by hackers using different approaches. For instance, Ukrainian power distribution system was subjected to cyber attacks, resulting in a continuous blackout.

Although many approaches (e.g., intrusion detection systems (IDS), firewalls, and upstream filtering) have been proposed to deal with DDoS attacks, honeypots show great advantages in protection capability and resource occupation [3]. For example, since an IDS may have to monitor a large number of network activities with trillions of bytes per second, its

Manuscript received March 30, 2019; accepted April 29, 2019. Date of publication May 20, 2019; date of current version January 4, 2020. This work was supported in part by the NSFC under Grant 61872195 and Grant 61572262. Paper no. TII-19-1138. *(Corresponding author: Miao Du.)*

M. Du is with the Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: dumiao0118@163.com).

K. Wang is with the Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: wangk@ucla.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2917912

cache will be quickly depleted. On the contrary, honeypots only capture and monitor a small portion of network activities, without the problem of resource depletion. By pretending to be normal servers to attract attackers, honeypots can consume attackers' resources and time. They can also influence and interfere with the choice of intruders, and further detect the intruders' attack intention. However, the traditional single-honeypot strategy is vulnerable to be identified by attackers because of low decoy performance. It has a limited monitoring scope and is unable to protect the whole network. Furthermore, most of the existing honeypot strategies are static defense and insufficient to deal with dynamic attacks [4].

In this paper, we reveal an anti-honeypot attack targeting on honeypots deployed in SDN. Specifically, this attack first determines whether there is a honeypot in the SDN servers. Then, it identifies the type of honeypots, and further finds out the optimal strategy for attacking. To protect the SDN from the anti-honeypot attack, we propose a PHG strategy that deploys different kinds of honeypots in SDN. It can improve the decoy performance for server protection and reinforcement. Our proposals not only are effective in ensuring secure data transmission but also have a low energy consumption compared with existing mechanisms.

The main contributions of this paper are summarized as follows.

1) We are the first to study the problem of deploying honeypot for DDoS attacks in SDN. The proposed honeypot strategies in SDN can provide dynamic protection for SDN, and further protect the security of the IIoT devices. Hence, malicious attacks under our strategies can be effectively controlled.
2) We reveal a two-step anti-honeypot based attack in SDN. The optimal strategy of attackers is analyzed, and an insight about such attacks provided.
3) We propose a PHG strategy to protect the SDN from anti-honeypot based attacks. We prove that there are several groups of different Bayesian–Nash Equilibrium (BNE) in the PHG strategy. In addition, we analyze the optimal equilibrium between legitimate users and attackers.
4) We conduct experiments on an SDN testbed to evaluate the performance of our proposals. The results show that our strategy outperforms the existing work in both the energy consumption and detection rate.

The rest of this paper are organized as follows. In Section II, we propose an anti-honeypot attack, and discuss the optimal strategies in this attack. Then, we design the pseudo-honeypot game strategy to protect the system in Section III. Experimental results are given in Section IV. Related work is shown in Section V. Finally, Section VI concludes this paper.

II. Anti-Honeypot Attacks

In this section, we first present the anti-honeypot attacks (AHA), and then the optimal strategies are discussed.

A. DDoS Attack and Honeypot Defense in SDN

Compared with the traditional architecture, as shown in Fig. 1, the SDN controller conducts the centralized management of

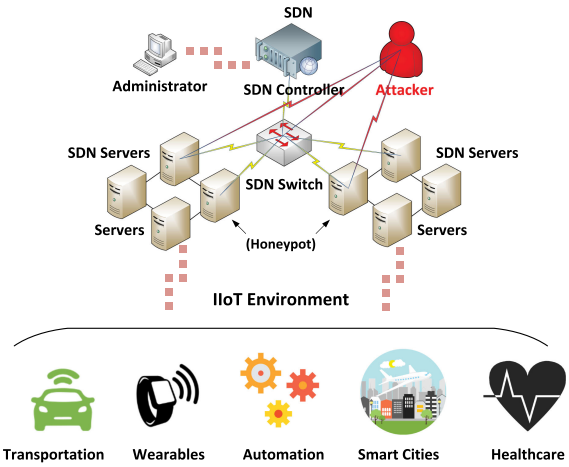

Fig. 1. SDN-based honeypot defense and DDoS attacks in IIoT environment.

IIoT devices [5], which provides more flexibility for the utilization and control of network resources. In the DDoS attacks to SDN, a large number of packets with forged source addresses are injected into the SDN servers. Since switches cannot find matches in their rule tables, these packets will be forwarded to the network controller of SDN. The legitimate and DDoS forged packets can bind the controller's resources into a collection of continuous processing to their exhausted points [6]. It is a major reason that the SDN controller is unreachable to the newly arrived legitimate data packets. Once the DDoS attack is successfully invaded and occupies one or several host servers in the network, these controlled host servers may send a large number of packets to the SDN controller at high speed. As a result, the SDN controller may crash or fail to respond to normal requests, which may further lead to paralysis of the entire SDN. What is more serious is that hackers can further invade IIoT nodes by attacking SDN, thus affecting the normal operation of IIoT devices.

Under these circumstances, honeypots are designed to attract attackers, and they can collect evidence and help hide the real servers. If we embed honeypots into the real servers, the real servers can serve as an internal network on the honeypots' network port mapping, which can increase the safety of real servers. Even if the attackers penetrate the external "servers", they cannot obtain any valuable information, as they attack honeypots instead [7].

B. Attacker's Optimal Strategy

As shown in the left side of Fig. 2, we propose an anti-honeypot attack that can identify honeypots in the defense system, such that attackers can bypass honeypots to intrude SDN servers. There are two steps in the AHA. In the first step, attackers recognize the existence of honeypots in defense system. Then in the second step, attackers recognize the types of honeypots deployed in the defense system. There are many types of honeypots, e.g., high-interaction honeypot and low-interaction honeypot. The interaction indicates that the attacker can collect information on the honeypots, as well as their decoy performance. Since low-interaction honeypot is easy to deploy

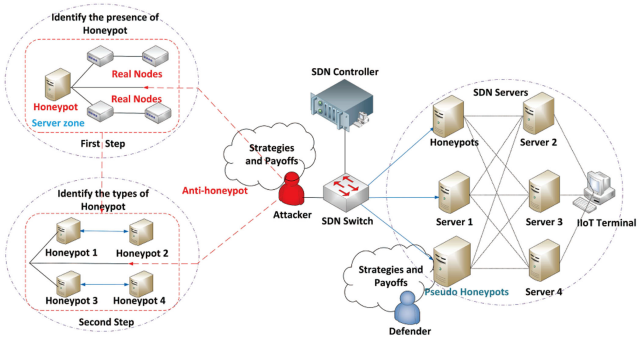

Fig. 2. System model.

and with less risk, attackers cannot carry out effective attacks easily in a low-interaction honeypot, so that the low-interaction honeypot can collect limited information about attackers. In high-interaction honeypot, we can obtain many attackers' information. At the same time, high-interaction honeypots enhance the activity freedom of attackers, naturally increase the complexity and risk of deployment and maintenance. In addition, as a rational attacker, the possibility of attacks on different types of honeypots is not the same. Thus, it is necessary for attackers to recognize the types of honeypots.

Step One. Honeypot Recognition: At this stage, attackers recognize the defense system as a honeypot or not. On the other hand, defenders can be a honeypot or not as two alternative strategies. Thus, there are four possible results under AHA and they are denoted by: (recognize is honeypot), (recognize, not honeypot), (not recognize, is honeypot), and (not recognize, not honeypot).

The probability of the targeted system as a honeypot is defined as $0 \leq \alpha \leq 1$. The recognition cost is denoted by $Ł_\kappa^n \geq 0$, regardless of whether the targeted system exists honeypots. When a honeypot is deployed, the cost of successful recognition is denoted as $Ł_\mu^n \geq 0$, and the successful recognition's payoff of the attacker is $K_h^i \geq 0$. We further let $K_\tau^i \geq 0$ represent the payoff gained by attacker if it cannot recognize the defense system. This payoff can also be achieved when attacker carries on a recognition. In addition, attacker can achieve the extra payoff $K_r^s \geq 0$ if the targeted system is a honeypot. The probability of attacker to recognize the targeted system is denoted as β, $0 \leq \beta \leq 1$. Finally, let $Ł_i \geq 0$ denote the whole cost when attacker recognizes in the targeted system. The detailed list of notations is provided in the Table I.

Lemma 1: The priority strategy of the anti-honeypot attacker and defender is recognizing and deploying the honeypot in the defense system when $\alpha \geq Ł_i / K_r^s$ and $\beta \geq K_h^i / Ł_\mu^n + K_h^i$, respectively.

Proof: For attackers, α is unknown but fixed. The expected payoff of attackers by successfully recognizing honeypots is calculated by

$$P(1,\alpha) = (K_r^s + K_\tau^i - Ł_i)\alpha + (K_\tau^i - Ł_i)(1-\alpha). \quad (1)$$

If the honeypots are not recognized, the corresponding payoff is

$$P(0,\alpha) = K_\tau^i \alpha + K_\tau^i (1-\alpha) = K_\tau^i. \quad (2)$$

TABLE I
LIST OF SYMBOLS IN THE PAPER

Symbols	Descriptions
α	probability of the targeted system as a honeypot
$Ł_\kappa^n$	recognition cost
$Ł_\mu^n$	succeessful recognition cost
K_h^i	succeessful recognition's payoff
K_τ^i	unsucceessful recognition's payoff
K_r^s	extra payoff
β	probability of recognizing the targeted system
$Ł_i$	the whole recognition cost
$Ł_t^l$	recognition cost at system layer
$Ł_r^l$	recognition cost at network layer
K_τ^s	recognition's payoff at system layer
K_h^s	recognition's payoff at network layer
$Ł_v^n$	cost for deploying low-interaction honeypot
$Ł_\mu^n$	cost for deploying high-interaction honeypot
$Ł_m^n$	payoffs for the targeted system as a low-interaction honeypot
K_h^i	payoffs for the targeted system as a high-interaction honeypot

If attackers recognize the honeypot in targeted system, it should meet $P(1,\alpha) \geq P(0,\alpha)$

$$(K_r^s + K_\tau^i - Ł_i)\alpha + (K_\tau^i - Ł_i)(1-\alpha) \geq K_\tau^i. \quad (3)$$

Thus, when $\alpha \geq Ł_i / K_r^s$, the attacker's strategy is recognizing the honeypot in the defense system.

Analogously, the priority strategy of defenders is analysed as follows. For defenders, β is unknown but fixed. If honeypots are deployed, the expected payoff of defenders is computed as

$$P(1,\beta) = (-Ł_\mu^n - Ł_\kappa^n)\beta + (K_h^i - Ł_\kappa^n)(1-\beta). \quad (4)$$

Otherwise, the payoff is

$$P(0,\beta) = -Ł_\kappa^n \beta - Ł_\kappa^n (1-\beta) = -Ł_\kappa^n. \quad (5)$$

If defenders deploy honeypots in targeted system, it should meet $P(1,\beta) \geq P(0,\beta)$, i.e.,

$$(-Ł_\mu^n - Ł_\kappa^n)\beta + (K_h^i - Ł_\kappa^n)(1-\beta) \geq -Ł_\kappa^n. \quad (6)$$

Thus, when $\beta \geq K_h^i / Ł_\mu^n + K_h^i$, the defender's priority strategy is deploying the honeypot in the targeted system.

In the first step, we analyze when the attackers should recognize the honeypot in the targeted system. In this case, there are two options for attackers. First, they may give up carrying attacks in the targeted system if they know the targeted system owns honeypot. Second, they may continue to attack if they have identified a target (e.g., a critical server) to be attacked in the defense system. Thus, under this circumstance, the attacker needs to recognize the type of the honeypot, to further help intrude the target in the defense system.

Step Two. Honeypot Type Recognition: At this stage, attackers will recognize the different types of the honeypot. It is an imperfect information game since attackers have little knowledge about the targeted server.

Attackers own alternative strategies that recognize at system layer or network layer. In the defense system, the security requirements of the system layer are higher than that of the

network layer. Therefore, we deploy more low-interaction honeypots in order to reduce the risks in the system layer. Information gathering and deception is the main task of the network layer, and thus we need to deploy more high-interaction honeypots. We further assume that an attacker can identify the target through an anti-honeypot. Once the recognition is successful, the attacker has mastered the method of bypassing the honeypot and attacking the target, but identifying the target requires a cost. The cost for attackers to recognize at system layer is defined as $Ł_t^l$, with $Ł_t^l \geq 0$, and $Ł_r^l \geq 0$ denotes the cost for attackers to recognize at network layer. The payoffs for attackers after recognizing the honeypot at system layer and network layer are $K_\tau^s \geq 0$ and $K_h^s \geq 0$, respectively. Let $\xi\alpha$ denote the probability of the targeted system as a low-interaction honeypot and $(1-\xi)\alpha$ denote the probability of high-interaction honeypot. The cost for deploying a low-interaction honeypot and a high-interaction honeypot at the targeted system is $Ł_v^n \geq 0$ and $Ł_\mu^n \geq 0$, respectively. $Ł_m^n$ and K_h^i represent the payoffs for the targeted system as a low-interaction or a high-interaction honeypot, respectively, with $K_h^i \geq Ł_m^n \geq 0$. Finally, let V_{t1} and V_{t2} $(t = 1, ..., 6)$ be the expected payoff of attackers and defenders, respectively.

The payoff for attackers at system layer and network layer is

$$\begin{aligned} V_s &= \xi\alpha(K_\tau^s - Ł_t^l) + (1-\xi)\alpha(K_\tau^s - Ł_t^l) + (1-\alpha)(-Ł_t^l) \\ &= \alpha K_\tau^s - Ł_t^l, \end{aligned} \tag{7}$$

and

$$\begin{aligned} V_n &= \xi\alpha(K_h^s - Ł_r^l) + (1-\xi)\alpha(K_h^s - Ł_r^l) + (1-\alpha)(-Ł_r^l) \\ &= \alpha K_h^s - Ł_r^l. \end{aligned} \tag{8}$$

Then the difference between these two payoffs can determine the optimal strategy as follows:

$$\begin{aligned} V' &= V_n - V_s = (\alpha K_h^s - Ł_r^l) - (\alpha K_\tau^s - Ł_t^l) \\ &= \alpha(K_h^s - K_\tau^s) + Ł_t^l - Ł_r^l. \end{aligned} \tag{9}$$

According to (9), it is obvious that the optimal strategy is relevant to α. We need further discussion on the layout of the honeypots in the SDN server based on the different range of α, aiming at finding the best strategy to attack.

C. Optimal Strategy Analysis in AHA

As shown in Fig. 3, we can obtain the payoffs for attackers and defenders as follows:

1) When $\alpha = 1$, i.e., targeted system is a honeypot, then,
   a) If a target has been recognized by attacker, the expected payoff is fixed, i.e., $K_h^s - K_\tau^s$ is a constant. Then, $V' = C - (Ł_r^l - Ł_t^l)$ where C is a constant. At this time, the target is recognized, and the attack is unavoidable. Under this circumstance, the attacker's optimal strategy is to attack the recognized target.
   b) If a target has not been recognized by attacker, i.e., $V' = K_h^s - K_\tau^s + C$, where C is a constant. In this case, the payoff of the attacker is unknown, the attacker's optimal strategy is to wait until the target is determined to reconsider whether to launch attacks.
2) When $0 \leq \alpha \leq 1$, targeted system may be a honeypot, then
$$V' = \alpha(K_h^s - K_\tau^s) + Ł_t^l - Ł_r^l. \tag{10}$$
   a) If attacker has recognized a target, we have $V' = \alpha C + Ł_t^l - Ł_r^l$ where C is a constant. Since $\alpha(K_h^s - K_\tau^s) > 0$, V' increases when the cost for recognition decreases. In this case, the optimal strategy is to attack the recognized target preferentially at a lower recognition cost.
   b) If attacker has not recognized a target, the optimal strategy is the same as 1)b).
3) When $\alpha = 0$, targeted system is not a honeypot. Thus
$$V' = Ł_t^l - Ł_r^l = -(Ł_r^l - Ł_t^l). \tag{11}$$
   At this time, no matter whether attacker has recognized a target, the optimal strategy is to carry on an attack.

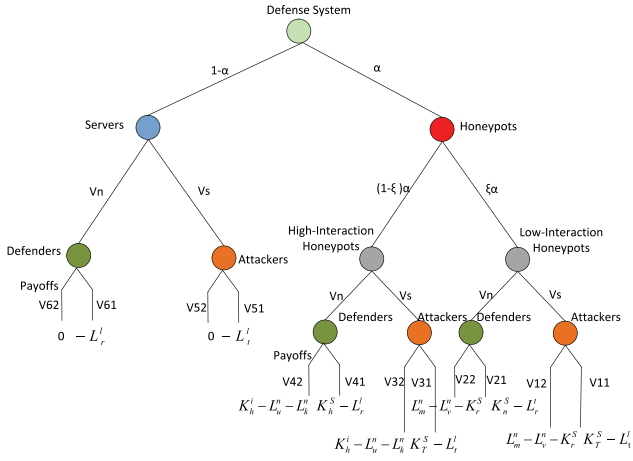

Fig. 3. Optimal attack strategy based on anti-honeypot.

In order to protect the system from AHA, we present a solution in the next section.

III. Protection Based on Pseudo-Honeypot Game

In this section, we present a solution based on the pseudo-honeypot strategy from the defender's perspective.

A. Defender's Optimal Strategy

Our proposed PHG strategy is shown on the right side of Fig. 2. We define the game $\mathcal{G}$ as $\mathcal{G} \triangleq \{\{\mathcal{U}, \mathcal{K}\}, \{\mathcal{C}_\mathcal{U}, \mathcal{C}_\mathcal{K}\}, \mathcal{S}_\mathcal{U}, \mathcal{S}_\mathcal{K}\}$. Here, we regard the SDN as a service provider (SP). $\{\mathcal{U}, \mathcal{K}\}$ is the finite collection of players, $\mathcal{U} \triangleq \{U_1, U_2, U_3\}$ represents different services: real communications, honeypot service, and pseudo-honeypot service, provided by the SP respectively. $\mathcal{K} \triangleq \{K_1, K_2\}$ represents the set of different visitors: legitimate users and attackers, respectively. $\{\mathcal{C}_\mathcal{U}, \mathcal{C}_\mathcal{K}\}$ is the set of strategies of the attackers and that of the honeypots, respectively. $\mathcal{C}_\mathcal{U} \triangleq \{\Omega_1, \Omega_2\}$ is a binary variable, and Ω_1 indicates providing service. $\mathcal{C}_\mathcal{K} \triangleq \{\Lambda_1, \Lambda_2\}$ is also a binary

variable, and Λ_1 indicates providing access. Finally, we let $\mathcal{S}_{\mathcal{U}}$ and $\mathcal{S}_{\mathcal{K}}$ denote the payoffs of the real servers and the visitors, respectively. Different from the previous work [8], we present a multistage game in this model. We first reveal a two-step anti-honeypot based attack in SDN, and then we propose a PHG strategy to protect the SDN from anti-honeypot based attacks.

The payoffs are discussed for three different cases as follows:

Case 1: Real soft-defined network communication is provided by the SP.

If legitimate users receive normal service, the payoff is κ ($\kappa > 0$) for both sides; otherwise it is $-\kappa$. If providing services to attackers, which will deteriorate service quality, the SP's payoff is $-\mu\kappa$ and attackers' payoff is $\mu\kappa$ ($\mu \geq 1$). Otherwise, both services' payoff and attackers' payoff are 0.

Case 2: Honeypot service is provided by the SP.

For legitimate users, no matter whether the service-side provides honeypot service, they are unable to obtain normal service, and consequently the payoff is $-\kappa$. If the services provide effective honeypot service to decoy attackers successfully, the service-side's payoff is $\delta_1\varsigma$ ($\varsigma > 0$ and $\delta_1 \geq 1$), and the attackers' payoff is $-\delta_1\varsigma$.

Case 3: Pseudo-honeypot service is provided by SP.

For legitimate users, pseudo-honeypot service is similar to normal service, so that they are able to obtain normal service, and consequently the payoff is κ. If providing service for attackers, the service performance will deteriorate, and the payoffs of service-side and attackers are $-\delta\kappa$ and $\mu\kappa - \delta_2\varsigma$, respectively, in which δ_2 represents pseudo-honeypot decoy factor and $\delta_2 \geq 1$. If the services provide effective pseudo-honeypot service to decoy attackers successfully, the service-side's payoff is $\delta_2\varsigma$.

In our model, the SP does not know the type of the visitors in advance, but it has a priori information about certain statistical metrics regarding the visitors, for instance, the distribution of the type of visitors. We assume $\{P(K_1) = 1-\tau, P(K_2) = \tau\}$. Similarly, we consider that the visitors also know the probability distributions of the type of services provided, $\{P(U_1) = 1-\alpha-\sigma, P(U_2) = \alpha, P(U_3) = \sigma\}$. Since the players are conscious of the strategies of the adversaries, we utilize Bayesian rules to obtain the posterior probability of the players in the game and calculate the expected maximum payoffs for all the players.

Theorem 1: A BNE strategy $\{(\Omega_1, \Omega_1, \Omega_1), (\Lambda_1, \Lambda_1)\}$ exists in the PHG provided

$$\tau < \frac{\mu}{2+\mu}, \alpha + \sigma < \frac{1}{2}, \alpha + \sigma < \frac{\delta\varsigma}{\mu\kappa + \delta\varsigma}.$$

Proof: The payoff of the real services for the strategy Ω_1 is denoted as $S_{U_1}(\Omega_1)$

$$\begin{aligned} S_{U_1}(\Omega_1) &= P(K_1 \mid \Lambda_1) * (-\mu\kappa) + P(K_2 \mid \Lambda_1) * (\kappa) \\ &= -\mu\kappa\tau + (1-\tau)\kappa \\ &= (1-\tau-\tau\mu)\kappa. \end{aligned} \tag{12}$$

Since the average payoff $S_{U_1}(\Omega_2)$ is obtained by real services' strategy Ω_2, we have

$$\begin{aligned} S_{U_1}(\Omega_2) &= P(K_1 \mid \Lambda_1) * 0 + P(K_2 \mid \Lambda_1) * (-\kappa) \\ &= -\tau\kappa. \end{aligned} \tag{13}$$

Similarly, we can obtain the average payoff $S_{U_3}(\Omega_1)$ and $S_{U_3}(\Omega_2)$ by pseudo-honeypot services' strategies Ω_1 and Ω_2, respectively, as follows:

$$\begin{aligned} S_{U_3}(\Omega_1) &= P(K_1 \mid \Lambda_1) * (-\mu\kappa) + P(K_2 \mid \Lambda_1) * (\kappa) \\ &= (1-\tau)\mu\kappa + \tau\kappa \\ &= (\mu + \tau - \tau\mu)\kappa, \end{aligned} \tag{14}$$

and

$$\begin{aligned} S_{U_3}(\Omega_2) &= P(K_1 \mid \Lambda_1) * 0 + P(K_2 \mid \Lambda_1) * (-\kappa) \\ &= -\tau\kappa. \end{aligned} \tag{15}$$

We first assume that $S_{U_1}(\Omega_1) = S_{U_1}(\Omega_2)$, and $S_{U_3}(\Omega_1) = S_{U_3}(\Omega_2)$. Then, we have

$$\tau = \frac{\mu}{2+\mu}. \tag{16}$$

According to (16), the optimal strategy for both the real communications and the pseudo-honeypots is Ω_1 when $\tau < \mu/(2+\mu)$. Otherwise the optimal strategy is Ω_2. Therefore, we conclude that the dominant strategy for the visitors' strategy (Λ_1, Λ_1) is $(\Omega_1, \Omega_1, \Omega_1)$ when $\tau < \mu/(2+\mu)$, and $(\Omega_2, \Omega_1, \Omega_2)$ otherwise.

Thus, for pseudo-honeypot services, strategy Ω_1 is the strictly dominant strategy. Consequently, the pseudo-honeypot services invariably select strategy Ω_1 for any visitor. However, for real communications, they can make a choice between strategy Ω_1 and Ω_2. The payoff obtained with strategy Λ_1 of legitimate users is given by

$$\begin{aligned} S_{K_1}(\Lambda_1) &= P(U_1 \mid \Omega_1) * (-\kappa) + P(U_2 \mid \Omega_2) * \kappa \\ &\quad + P(U_3 \mid \Omega_2) * \kappa \\ &= -(1-\alpha-\sigma)\kappa + \alpha\kappa + \sigma\kappa \\ &= [2(\alpha+\sigma) - 1]\kappa. \end{aligned} \tag{17}$$

The payoff obtained with legitimate users' strategy Λ_2 can be computed as $S_{K_1}(\Lambda_2) = P(U_1 \mid \Omega_1) * 0 + P(U_2 \mid \Omega_1) * 0 + P(U_3 \mid \Omega_1) * 0 = 0$.

Similarly, we can obtain the average payoff $S_{K_2}(\Lambda_1)$ and $S_{K_2}(\Lambda_2)$ for the attackers as follows:

$$\begin{aligned} S_{K_2}(\Lambda_1) &= P(U_1 \mid \Omega_1) * (-\delta\varsigma) + P(U_2 \mid \Omega_1) * \mu\kappa \\ &\quad P(U_3 \mid \Omega_2) * \mu\kappa \\ &= (1-\alpha-\sigma)(-\delta\varsigma) + \alpha\mu\kappa + \sigma\mu\kappa \\ &= (\alpha+\sigma)(\mu\kappa + \delta\varsigma) - \delta\varsigma, \end{aligned} \tag{18}$$

and similarly the payoff of attackers with strategy (Λ_2) can be computed as $S_{K_2}(\Lambda_2) = P(U_1 \mid \Omega_1) * 0 + P(U_2 \mid \Omega_1) * 0 + P(U_3 \mid \Omega_1) * 0 = 0$.

By assuming $S_{K_1}(\Lambda_1) = S_{K_1}(\Lambda_2)$, and $S_{K_2}(\Lambda_1) = S_{K_2}(\Lambda_2)$, we have

$$\alpha + \sigma = \frac{1}{2} \tag{19}$$

and

$$\alpha + \sigma = \frac{\delta_\varsigma}{\mu\kappa + \delta_\varsigma}. \tag{20}$$

According to (19) and (20), if the visitors are legitimate users, Λ_1 is the dominant strategy for portfolio strategy $\{(\Omega_1, \Omega_1, \Omega_1)\}$ under the condition $\alpha + \sigma < 1/2$. On the contrary, if the visitors are attackers, Λ_1 is the SP's dominant strategy for portfolio strategy $\{(\Omega_1, \Omega_1, \Omega_1)\}$ under the condition $\alpha + \sigma < \delta_\varsigma/(\mu\kappa + \delta_\varsigma)$. Consequently, we have

$$\tau < \frac{\mu}{2+\mu}, \alpha + \sigma < \frac{1}{2}, \alpha + \sigma < \frac{\delta_\varsigma}{\mu\kappa + \delta_\varsigma}. \tag{21}$$

Therefore, we can obtain a BNE strategy $\{(\Omega_1, \Omega_1, \Omega_1), (\Lambda_1, \Lambda_1))\}$ in the game when (21) is true. ■

Analogously, we can obtain other two BNE strategies $\{(\Omega_1, \Omega_1, \Omega_1), (\Lambda_1, \Lambda_2)\}$ and $\{(\Omega_2, \Omega_1, \Omega_2), (\Lambda_2, \Lambda_2)\}$ in the game under the conditions (22) and (23), respectively, as follows:

$$\tau < \frac{\mu}{2+\mu}, \frac{\delta_\varsigma}{\mu\kappa + \delta_\varsigma} < \alpha + \sigma < \frac{1}{2} \tag{22}$$

and

$$\tau < \frac{\mu}{2+\mu}, \alpha + \sigma > \frac{1}{2}, \alpha + \sigma > \frac{\delta_\varsigma}{\mu\kappa + \delta_\varsigma}. \tag{23}$$

In this section, Nash Equilibriums (NEs) will be analyzed in the HHG model. We first analyse the NEs in the PHG and AHA model compared to the traditional game in terms of equilibrium strategies. Then, we analyse the payoffs of legitimate users and attackers via game trees. Importantly, we further discuss the τ in the AHA model to find out the optimal strategies for attackers.

B. Optimal Strategy Analysis in PHG

In the PHG strategy, the equilibrium conditions are related to α and σ, which makes the BNE conditions more diversified. We design the optimal strategies algorithm for PHG strategy in Algorithm 1.

Figs. 4 and 5 are the game tree from the legitimate user's perspective and from the attacker's perspective, respectively. It is clear that the game equilibrium conditions relate to the probabilities of the honeypot and the pseudo-honeypot, α and σ, as well as decoy factors δ_1 and δ_2. Therefore, the result of the game is completely controlled by the services. Furthermore, the pseudo-honeypot is more deceptive compared to the honeypot, because the attackers cannot be certain about the existence of the pseudo-honeypot when they launch the attack, which makes attackers cautious. The figures clearly indicate that the PHG strategy is the most favorable for defenders. The PHG strategy provides a variety of strategies so that the defenders can adjust α and σ, or δ_1 and δ_2 to improve the defense mechanism and achieve different BNEs according to different portfolios.

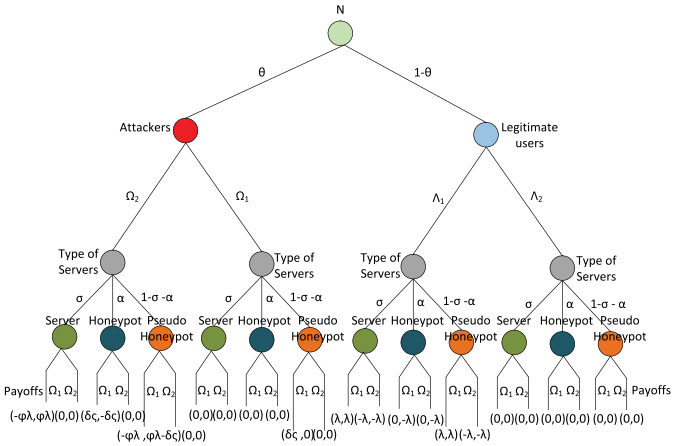

Fig. 4. The game tree from legitimate user' perspective.

Algorithm 1: Optimal Strategies for Pseudo-Honeypot Game.

Input: τ, α, σ, κ, χ, μ, δ_1, δ_2 and ς
Output: Optimal strategies $\{(\Omega_{ii},\Omega_{jj},\Omega_{kk}),(\Lambda_{ii},\Lambda_{jj})\}$
/* Initialize the strategies, $\{\Omega_i,\Omega_j,\Omega_k\}$ */
/* Find the stable state */
if $\tau < \mu/2 + \mu$ **then**
  if $\alpha + \sigma < \delta_\varsigma/(\mu\kappa + \delta_\varsigma) \wedge \alpha + \sigma < 1/2$ **then**
    choose optimal portfolio strategy $\{(\Omega_1,\Omega_1,\Omega_1),(\Lambda_1,\Lambda_1)\}$.
  end
  else
    cannot achieve a BNE.
  end
  if $\delta_\varsigma/(\mu\kappa + \delta_\varsigma) < \alpha + \sigma < 1/2$ **then**
    choose optimal portfolio strategy $\{(\Omega_1,\Omega_1,\Omega_1),(\Lambda_1,\Lambda_2)\}$.
  end
  else
    cannot achieve a BNE.
  end
  if $\alpha + \sigma > \delta_\varsigma/(\mu\kappa + \delta_\varsigma) \wedge \alpha + \sigma > 1/2$ **then**
    choose optimal portfolio strategy $\{(\Omega_2,\Omega_1,\Omega_2),(\Lambda_2,\Lambda_2)\}$.
  end
  else
    cannot achieve a BNE.
  end
end
else
  cannot achieve a BNE.
end

We conclude that the AHA can effectively improve the attack performance compared with single honeypot game (SHG) model [8], and the PHG strategy can effectively solve the attacks based on AHA. The proposed system model (AHA + PHG) can achieve the optimal strategies for both defenders and attackers when the performance of energy consumption and detection rate reaches a dynamic balance. We will simulate and confirm our deductions in the next section.

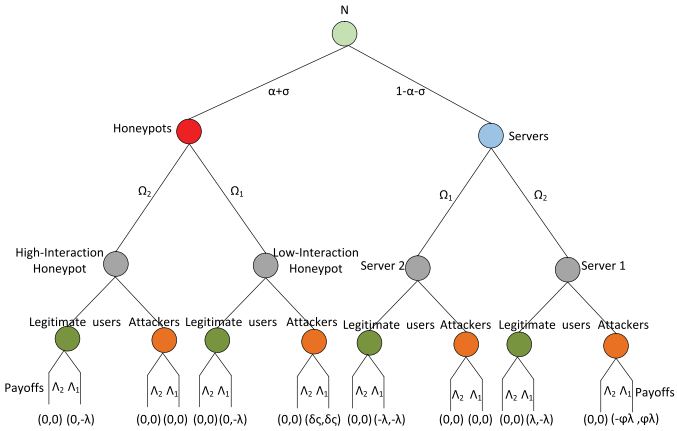

Fig. 5. The game tree from attackers' perspective.

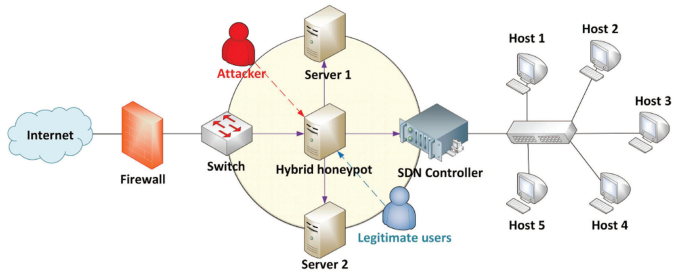

Fig. 6. SDN testbed.

IV. Performance Evaluations

In this section, we construct an SDN testbed to evaluate the performance of our proposals. The experiment settings are first presented, followed by detailed experiment results.

A. Experiment Settings

As shown in Fig. 6, we conduct an SDN testbed experiment in the OpenFlow network. First, all forwarding decisions are transferred from each switch to the controller, and the controller centrally manages the forwarding policy of the packets. Second, the OpenFlow protocol is responsible for interacting with all network switches and configuring data forwarding paths. The transmission path of all data packets in the network is determined by the unified control of the SDN controller, and the OpenFlow switch is only responsible for forwarding data packets. The controller implements the above process by sending a flow to the switch. An OpenFlow switch usually contains multiple flow tables, each table contains multiple flow table entries, each entry consists of a matching field, counters, and instructions.

The controller and the OpenFlow switch implement information interaction through OpenFlow messages. When a packet arrives at the switch, if a certain flow in the flow table matches, the operation of the flow instruction is sequentially executed. If there is no matching flow, the entire packet is cached in the switch, and the packet is assigned with a Buffer ID, which is sent to the controller as a Packet In message along with the packet. Finally, the controller redefines a new flow according to the header of the packet to determine the processing strategy for the packet.

According to the above SDN flow matching strategy, we construct a small-scale testbed consisting of 12 servers and 10 honeypots. We use five groups of different service deployments of servers and hybrid honeypots (i.e., 12 servers, four honeypots, and six pseudo-hoenypots), which can simulate attack flows on services and service flows of legitimate users by using synchronize sequence numbers (SYN)-Flood attack and file transfer protocol (FTP) flow, respectively. In our experiments, we take two different honeypot scenarios into consideration.

1) AHA: We assume that attackers and defenders may have their own strategies in the SDN. Attackers apply the anti-honeypot based attack strategy to identify and detect the honeypots in the defense systems.
2) PHG: We assume that there are five groups of different honeypot deployments linked by SDN controller. Under normal circumstances, the SDN controller sends data flows to each node. However, due to the possible presence of an attacker in the SDN, we deploy twofold honeypots (i.e., 12 servers, two honeypots, and six pseudo-hoenypots) to protect SDN against DDoS attacks. In addition, we can determine which is the most reasonable deployment through the experimental analysis of data flows between the service flow and the attack flow by using FTP flow and SYN-Flood attack, respectively.

Accordingly, we consider the following comparisons of our system model under the performance of energy consumption and detection rate. We first put forward the AHA from the attacker's perspective compared with the existing SHG model. Then we further add the PHG to improve the decoy performance. Under these circumstances, the performance comparisons are over the SHG model, the AHA model, and the AHA + PHG model.

We fix τ and deploy the PHG with $\alpha+\sigma=\{0.2, 0.5, 0.8\}$ and the AHA with $\alpha=0$, $0<\alpha<1$, and $\alpha=1$, respectively. We assume the pseudo-honeypot's decoy factor $\delta_1=\delta_2=5$ and set up six groups of different service deployments, i.e., $\{S=12, H=2, PH=6\}$, where S, H, and PH represent number of servers, honeypots and pseudo-honeypots under different preconditions $\mu=4$ and $\mu=1$, respectively.

B. Experiment Results

As shown in Fig. 7, the slopes of the energy consumption curves of the PHG are relatively smooth, which means the change in the energy consumption is slow. For instance, when $\alpha+\sigma=0.2$, the proposed model has an advantage in energy consumption performance, which can save almost 1 and 2 J of energy, respectively, compared with that of the existing models AHA and SHG, respectively. However, the PHG will consume energy with the frequency increase of honeypot and pseudo-honeypot, prompting that we need to adjust and deploy pseudo-honeypot more reasonably. Otherwise, it may result in unnecessary waste of resources. Therefore, if we deploy the pseudo-honeypot methodology, our proposed model can incorporate both intrusion detection and cyber resource efficiency.

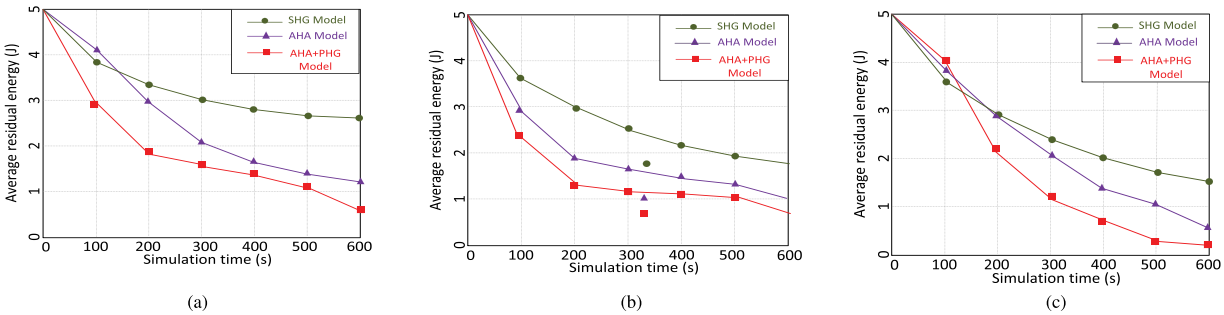

Fig. 7. Performance for energy consumptions in deploying a twofold honeypot. (a) $\alpha + \sigma = 0.2$. (b) $\alpha + \sigma = 0.5$. (c) $\alpha + \sigma = 0.8$.

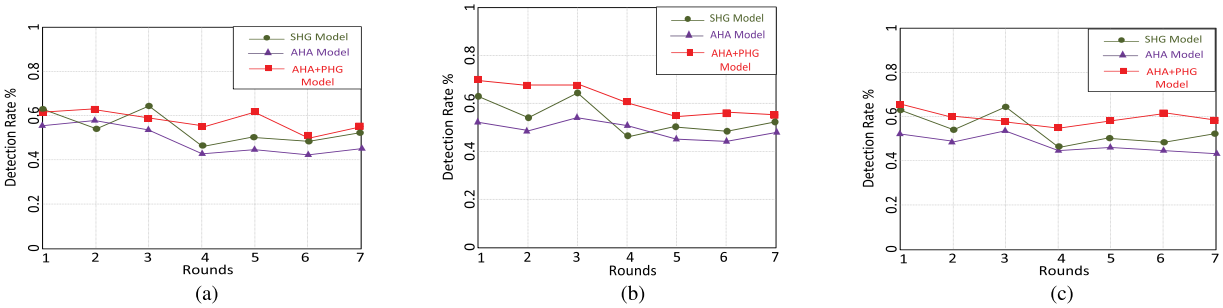

Fig. 8. Performance for detection rate in deploying a twofold honeypot. (a) $\alpha + \sigma = 0.2$. (b) $\alpha + \sigma = 0.5$. (c) $\alpha + \sigma = 0.8$.

Second, Fig. 8 shows that the AHA greatly reduces the detection rate compared to the SHG model. Accordingly, the anti-honeypot based attack strategy can effectively help the attacker to choose the strategy to avoid the tracking of the defense system. In addition, when we add the PHG to the system model to solve this problem, the AHA + PHG model has a lower detection rate compared with the AHA, which means the PHG can be a good solution to the AHA.

Finally, Fig. 9 shows that as the attack damage factor μ increases, the intruders are likely to launch attacks. In terms of the overall trend, when the value of μ is fixed, the attack flow of the SHG model and the proposed model shows a gradual upward trend as the probability of attack increases, while the service flow shows a downward trend. Specifically, for example, when $\mu = 4$ and the attack probability is 0.2, the number of attack flows rises to over 40 000, and the service flow drops sharply to only about 50 000. In contrast, the proposed model for rational deployment of honeypots (S = 12, H = 2, P = 6) has an attack flow of about 30 000, and the number of service flows exceeds 70 000 under the same conditions. Consequently, we can conclude that reasonable deploying of honeypots and pseudo-honeypots ($\alpha + \sigma < 1/2$) is of paramount importance. Therefore, the PHG model can be an effective defense mechanism against DDoS attacks.

V. Related Work

In this section, we present the existing literatures on DDoS detection in SDN, honeypot for DDoS, and game theory in DDoS.

A. DDoS Detection in SDN

The detection of DDoS attacks has been extensively studied. Nevertheless, little research is about the security issues with respect to SDN environments. We describe several intrusion detection techniques related to the SDN environments. Sun *et al.* [9] utilized OpenFlow and sFlow to deal with anomaly detection problems on SDN. The performance of traffic collection of this method was compared with sFlow. Sinclair *et al.* [10] presented a novel method to address the DDoS detection by self-organizing mapsmachine learning technique. Lara *et al.* [11] discussed the alarm flow specification language, aiming at controlling a large amount of traffic sent to the controller. Wang *et al.* [12] investigated a survey on the solutions for DDoS attacks in SDNs, and further analyzed the advantages and disadvantages of several mechanisms in defending against DDoS attacks. Zhu *et al.* [13] proposed Predis to address SDN privacy protection and cross-domain attacks. In addition, the authors utilized the kNN algorithm to detect attacks efficiently and accurately.

B. Honeypot for DDoS

Zhan *et al.* [14] proposed an IHoneycol approach to address the DDoS problem efficiently. Jiang *et al.* [15] presented an architecture using virtual machine to capture attacks called Collapsar, which played a significant role in supervising high-interaction virtual honeypots in the heart of a black hole. Walfish *et al.* [16] presented an effective hop tracking mechanism called honeypot back-propagation, which is a new roaming honeypot scheme leverages to achieve accurate attack signatures.

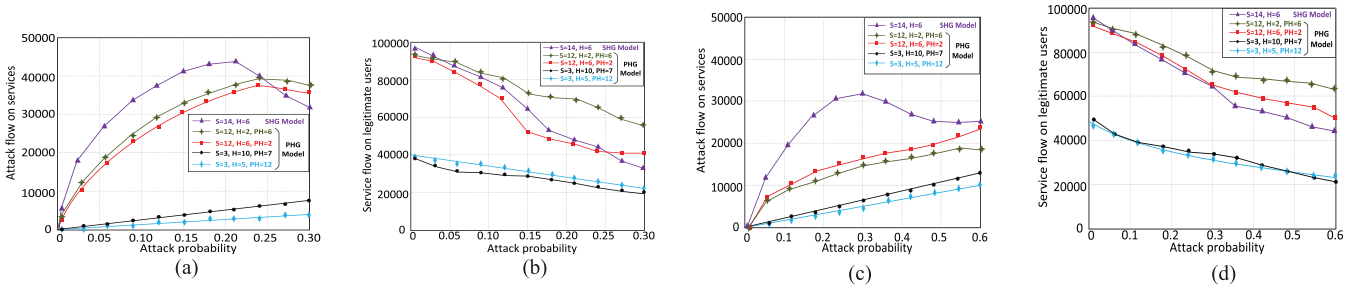

Fig. 9. Attack flow on services and service flow on legitimate users when $\mu = 4$ and $\mu = 1$, respectively. (a) $\mu = 4$. (b) $\mu = 4$. (c) $\mu = 1$. (d) $\mu = 1$.

Wang *et al.* [17] proposed a honeypot detection to address botnet attacks. The attacker can be detected in their honeypot botnet. Hayatle *et al.* [18] presented how a technique can allow botmasters to determine the performance of compromised machines with a relatively high certainty.

C. Game Theory in DDoS

B. Rashidi *et al.* [19] proposed a DDoS defense mechanism called CoFence that promotes a "domain-help-domain" collaborative network between network function virtualization (NFV)-based domain networks. In addition, the authors designed a dynamic resource allocation mechanism for the domain and establishes a game model to find an effective, incentive-compatible, fair and reciprocal resource allocation method by analyzing Nash equilibrium. Luo *et al.* [20] presented a taxonomy of DDoS attack and defense system, and Wang *et al.* [21] considered the confrontation between defenders and attackers and developed a game theory analysis framework for collaborative security detection. Anwar and Malik [22] proposed a simulation to study the feasibility of DDoS attack, the experience of such a security event in a real SDN server.

Yan and Yu [23] discussed the new trends and characteristics of DDoS attacks in cloud computing, and conducted a comprehensive investigation of the DDoS attack defense mechanism using SDN. By reviewing existing researches on launching DDoS attacks on SDN, and methods for DDoS attacks in SDN. The authors found that the contradictory relationship between SDN and DDoS attacks had not been well resolved in previous work. This work helped to understand how to take advantage of SDN to protect against DDoS attacks in cloud computing environments.Yan *et al.* [24] further investigated DDoS attacks in SDN-based cloud computing environments. SDN was easy to detect and respond to DDoS attacks due to its software-based traffic analysis, centralized control, and dynamic update of forwarding rules. In detail, the author analyzed and explored the following issues: how to use SDN to defend against application-level DDoS attacks, how to use SDN to defend against mobile DDoS attacks, how to achieve multiple location defenses, how to use cross-layer traffic analysis, how to collaborate between key defensive points, and how to conduct a DDoS attack tolerant system using SDN.

Nevertheless, none of these state-of-the-art researches introduces the twofold honeypot strategy into SDN for resisting DDoS attacks. Our proposal can prevent the DDoS attacks by the decoy performance of the pseudo-honeypots, and utilize a game approach to analyze and prove the effectiveness of honeypots, which can help us deploy the honeypots more effectively in the real SDN scenario. Moreover, we employ a two-step anti-honeypot attack to analyze defense systems deployed with honeypots from the perspective of attackers. Therefore, we can comprehensively resist against DDoS attacks in SDN.

VI. Conclusion

In this paper, we introduced the honeypot strategies into SDN to solve DDoS attacks in IIoT environment. In our honeypot strategies, they can provide dynamic protection for SDN. In addition, the honeypot strategies can offer optimal defensive strategies when facing DDoS attacks. Specifically, we presented an anti-honeypot based attack to help attackers find out their optimal strategies. Moreover, we proposed a PHG strategy to analyze the strategic interaction between attackers and defenders. The experiment results showed that the energy consumption and the defense efficiency can be improved with the proposed strategies. Furthermore, the anti-honeypot based attack strategy can effectively help the attackers identify the honeypot traps in the target SDN servers, and the PHG strategy can be a good solution to this attack strategy, thereby further protecting the security of the IIoT devices.

References

[1] C. Tian, A. Munir, A. X. Liu, J. Yang, and Y. Zhao, "OpenFunction: An extensible data plane abstraction protocol for platform-independent software-defined middleboxes," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1488–1501, Jun. 2018.

[2] M. Du, K. Wang, Y. Chen, X. Wang, and Y. Sun, "Big data privacy preserving in multi-access edge computing for heterogeneous Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 62–67, Aug. 2018.

[3] Q. D. La, T. Q. S. Quek, J. Lee, S. Jin, and H. Zhu, "Deceptive attack and defense game in honeypot-enabled networks for the Internet of Things," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1025–1035, Dec. 2016.

[4] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Y. Yau, and J. Wu, "Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 7, pp. 1838–1853, Jul. 2018.

[5] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and OpenFlow: From concept to implementation," *IEEE Commun. Surveys Tut.*, vol. 16, no. 4, pp. 2181–2206, May 2014.

[6] K. Wang, M. Du, Y. Sun, A. Vinel, and Y. Zhang, "Attack detection and distributed forensics in machine-to-machine networks," *IEEE Netw.*, vol. 30, no. 6, pp. 49–55, Dec. 2016.

[7] M. F. Thompson, "Effects of a honeypot on the cyber grand challenge final event," *IEEE Secur. Privacy*, vol. 16, no. 2, pp. 37–41, Mar. 2018.

[8] K. Wang, M. Du, S. Maharjan, and Y. Sun, "Strategic honeypot game model for distributed denial of service attacks in the smart grid," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2474–2482, Sep. 2017.
[9] X. S. Sun, A. Agarwal, and T. S. Eugene, "Controlling race conditions in OpenFlow to accelerate application verification and packet forwarding," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 2, pp. 263–277, Jun. 2015.
[10] N. Sinclair, D. Harle, I. A. Glover, J. Irvine, C. Robert, and Atkinson, "An advanced SOM algorithm applied to handover management within LTE," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1883–1894, Jun. 2013.
[11] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using OpenFlow: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 493–512, Aug. 2014.
[12] K. Wang, Y. Wang, D. Zeng, and S. Guo, "An SDN-based architecture for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 25–31, Feb. 2017.
[13] L. Zhu, X. Tang, M. Shen, X. Du, and M. Guizani, "Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 628–643, Mar. 2018.
[14] Z. Zhan, M. Xu, and S. Xu, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 11, pp. 1775–1789, Nov. 2013.
[15] X. Jiang, D. Xua, and Y. M. Wang, "Collapsar: AVM-based honeyfarm and reverse honeyfarm architecture for network attack capture and detention," *J. Parallel Distrib. Comput.*, vol. 4, no. 10, pp. 1165–1180, 2006.
[16] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker, "DDoS defense by offense," *ACM Trans. Comput. Syst.*, vol. 28, no. 1, pp. 61–80, 2010.
[17] P. Wang, L. Wu, R. Cunningham, and C. Zou, "Honeypot detection in advanced botnet attacks," *Int. J. Inf. Comput. Secur.*, vol. 4, no. 1, pp. 30–51, 2010.
[18] O. Hayatle, A. Youssef, and H. Otrok, "Dempster-shafer evidence combining for anti-honeypot technologies," *Inf. Sec. J.: A Global Perspective*, vol. 21, no. 6, pp. 306–316, 2012.
[19] B. Rashidi, C. Fung, and E. Bertino, "A collaborative DDoS defence framework using network function virtualization," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 10, pp. 2483–2497, Oct. 2017.
[20] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, "On a mathematical model for low-rate shrew DDoS," *IEEE Trans. Inf. Forensics Secur.*, vol. 9, no. 7, pp. 1069–1083, Jul. 2014.
[21] K. Wang, M. Du, D. Yang, C. Zhu, J. Shen, and Y. Zhang, "Game-theory-based active defense for intrusion detection in cyber-physical embedded systems," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 1, 2016, Art. no. 18.
[22] Z. Anwar and A. W. Malik, "Can a DDoS attack meltdown my data center? A simulation study and defense strategies," *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1175–1178, Jul. 2014.
[23] Q. Yan and F. R. Yu, "Distributed denial of service attacks in software-defined networking with cloud computing," *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 52–59, Apr. 2015.
[24] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, Oct. 2016.

Miao Du received the M.S. degree in information network from the Nanjing University of Posts and Telecommunications, Jiangsu, China, in 2018. He is currently working toward the Ph.D. degree in computer science and technology with Hohai University, Jiangsu, China.

His current research interests include wireless sensor network, social networks, security, game theory, smart grid communications, and cyber-physical systems.

Kun Wang (M'13–SM'17) received the Ph.D. degree from the Nanjing University of Posts and Telecommunications, Jiangsu, China in 2009 and the second Ph.D. degree from the University of Aizu, Japan in 2018, both in computer science.

He was a Postdoctoral Fellow at the University of California, Los Angeles (UCLA), Los Angeles, CA, USA from 2013 to 2015, and a Research Fellow at Hong Kong Polytechnic University, Hong Kong, from 2017 to 2018. He is currently a Research Fellow in UCLA. His current research interests mainly include the area of big data, wireless communications and networking, energy Internet, and information security technologies.

Dr. Kun Wang is the recipient of IEEE GLOBECOM 2016 Best Paper Award, IEEE Technical Committee on Green Communications and Computing Best Magazine Paper Award 2018, and IEEE Systems Journal Best Paper Award 2019. He serves as an Associate Editor for IEEE ACCESS, Editor for *Journal of Network and Computer Applications*, and Guest Editors for IEEE NETWORK, IEEE ACCESS, *Future Generation Computer Systems*, *Peer-to-Peer Networking and Applications*, IEICE TRANSACTIONS ON COMMUNICATIONS, *Journal of Internet Technology*, and *Future Internet*.