IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 1, 2021, accepted April 27, 2021, date of publication May 3, 2021, date of current version May 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077075

Optimal Strategy for Cyberspace Mimic Defense Based on Game Theory

ZEQUAN CHEN[1], GANG CUI[2], LIN ZHANG[2], XIN YANG[1], (Student Member, IEEE), HUI LI[1], YAN ZHAO[3], CHENGTAO MA[4], AND TAO SUN[5]

[1]Shenzhen Graduate School, Peking University, Shenzhen 518055, China
[2]State Key Laboratory of Nuclear Power Safety Monitoring Technology and Equipment, Shenzhen 518055, China
[3]Shenzhen SmartCity Communication Company Ltd., Shenzhen 518055, China
[4]Guangdong Yue Gang Water Supply Company Ltd., Shenzhen 518055, China
[5]Network Information Center, University Town of Shenzhen, Shenzhen 518055, China

Corresponding author: Gang Cui (cuigang@cgnpc.com.cn) and Hui Li (lih64@pkusz.edu.cn)

This work was supported in part by the Research and Development Key Program under Grant 2019B010137001, in part by the National Keystone Research and Development Program of China under Grant 2017YFB0803204, in part by the PCL Future Regional Network Facilities for Large-scale Experiments and Applications under Grant LZC0019, in part by the Natural Science Foundation of China (NSFC) under Grant 61671001, in part by the Shenzhen Research Programs under Grant JCYJ20170306092030521, and in part by the Shenzhen Municipal Development and Reform Commission (Disciplinary development program for data science and intelligent computing).

ABSTRACT Traditional defensive techniques are usually static and passive, and appear weak to confront highly adaptive and stealthy attacks. As a novel security theory, Cyberspace Mimic Defense (CMD) creates asymmetric uncertainty that favors the defender. CMD constructs multiple executors which are diverse functional equivalent variants for the protected target and arbitral mechanism. In this way, CMD senses the results of current running executors and changes the attack surface. Although CMD enhances the security of systems, there are still some critical gaps with respect to design a defensive strategy under costs and security. In this paper, we propose a dual model to dynamically select the number of executors being reconfigured according to the states of the executors. First, we establish a Markov anti-attack model to compare the effects of CMD under different types of attack. Then, we use a dynamic game of incomplete information to determine the optimal strategy, which achieves the balance of the number of reconfiguration and security. Finally, experimental results show that our dual model reduces defensive costs while guarantees security.

INDEX TERMS Cyberspace mimic defense, incomplete information, dynamic game, Markov.

I. INTRODUCTION

Nowadays, advanced cyber-attacks have become a major threat to network security. For example, advanced persistent threats (APTs) are a kind of covert and continuous attack against specific targets to avoid being found and obtain long-term advantages. Traditional defense technology, which only defends against known types of attacks, is unable to deal with APTs. The defender uses prior knowledge to defend the known attack method, but the attacker develops the new attack methods to invade the target host. This kind of difference between attacker and defender leads to the status inequality between defenders and attackers. There are three reasons why defenders are at a disadvantage in confrontation: (a) It is difficult to prove the security of network architectures and avoid software vulnerabilities. (b) The attacker only needs to discover and exploit a vulnerability in the target host to launch an attack, but the defense must fully defend the network system to avoid the attack. (c) The static and determinacy of the network architecture allow that attacker has more time and opportunity to launch an attack. These properties make the defender in a weak position in long-term network confrontation.

The three reasons cause that the attacker knows more about the defender than the defender knows about the attacker. To reverse this imbalance, Moving Target Defense (MTD) [1] and Cyberspace Mimic Defense (CMD) [2] have been proposed, both of which have the characteristics of diversity and dynamics. They increase the difficulty of attacks [3] and restrict the exposure time of vulnerability to improve the anti-attack ability of the system.

CMD identifies the enemy and friend information without relying on the prior knowledge of the attacker or the characteristic information of the attack behavior. As the typical

The associate editor coordinating the review of this manuscript and approving it for publication was Zhitao Guan.

architecture of CMD, DHR (Dynamic Heterogeneous Redundancy structure) consists of multiple heterogeneous executors and arbiter. Executors are the same functions but composed of different components. The arbiter determines the output of DHR after receiving the result of each executor.

When the attacker controls all the executors successfully, the system cannot judge whether the executor is in an abnormal state through the arbiter. Therefore, the system replaces and reconfigures the executors periodically, which makes the attack's state unsustainable. Under limited network resources, it is important to choose the optimal defensive strategy to achieve a balance between security and the number of being reconfigured executors. The antagonism of the offensive and defensive parties in cyberspace is similar to the characteristics of game theory including goal opposition, strategic interdependence, and non-cooperative relationship [4]. To select the optimal defense strategy, we propose a dual model including the Markov model and CMD strategy selection model (CMD-SSM).

We make the following contributions in this paper:

1) Based on the characteristics of the DHR architecture, we establish Markov models to describe the process of the attacker and defender. The simulation experiment verifies the effectiveness of the Markov models.
2) To make the model more realistic, we design the benefit function considering defensive costs, attack costs, and defensive failure cost.
3) To determine the number of executors to be reconfigured, we establish the game model, which is combined with the stability probability of the Markov chain and the benefit of attack and defense.

The rest of this paper is organized as follows. In Section II, we describe the basic principle of CMD. Section III introduces the framework and notations. Section IV introduces the actions of attackers and defenders and establishes Markov models. Section V constructs the model of CMD-SSM. Section VI illustrates the effectiveness of the Markov models and the security of CMD-SSM. Section VII discusses related work. Section VIII provides some concluding remarks.

II. BACKGROUND KNOWLEDGE

A. CMD FUNDAMENTAL PRINCIPLE

The main idea of CMD is to change the combination of executors, which increases attack time and complexity. Dynamic characteristics prevent attackers from further attack. The arbiter monitors the output of each executor to ensure the security of the system as much as possible.

Attackers usually need a certain amount of time to collect target host information, determine the scope of the attack. The research [5] shows that 95% of the enemy's time is spent preparing for an attack, while only 5% of the time is spent executing the attack. Changing the combination of the executor is expected to introduce uncertainty for the attacker and make the reconnaissance effort more costs.

Meanwhile, the research [6] analyzed the vulnerabilities of 11 different OSes over a period of 18 years, which found this number to be low for several combinations of OSes. The analysis shows that selecting appropriate OSes with one can reduce substantially common vulnerabilities from occurring in the replicas of the system. Therefore, the characteristics, which including dynamic, redundant, and heterogeneity make the defender take the initiative in the network confrontation.

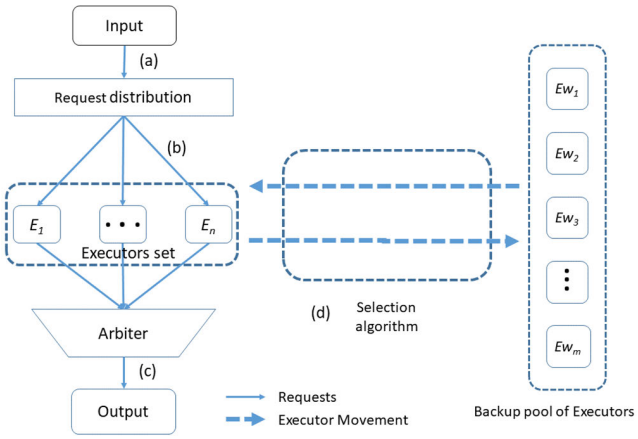

FIGURE 1. The workflow of DHR and executor movement: (a) input requests, (b) distribute information (c) arbiter (d) switch executor(s) and being reconfigure executor(s).

B. DHR ARCHITECTURE

DHR consists of a request distribution, an executor set, an arbiter, and a backup pool of executors, as shown in Fig. 1. The working process of DHR can be described as IPO, that is, ''Input-Processing-Output''.

After receiving the request from the user or attacker, the request distribution copies the request into n copies and distributes it to n heterogeneous executors. The n executors deal with the request independently and send the results to the arbiter. The arbiter receives the output of each executor and determines the return information based on the selection algorithm. If the attack effected only work once, the system tolerates the exception. Once the persistent exception of a single executor exceeds the tolerant threshold of the system, it needs to be offline and reconfigured.

The majority decision is a common decision-making algorithm. The result of output is ''relatively correct'' in which the algorithm returns most of the same output. If the output is not the same, it is regarded as an abnormal state. The system cannot identify whether the output of all executors is invaded or not. Therefore, the system will select executor(s) from the backup pool and replaced it(s) with the executor of the online service. So that the attack successfully escapes but unable to maintain stability.

We assume that the true output of DHR is known. We note symbols (Number of *true*, Number of *false*) to represent the output result. The number of executors n in the executors set is 3. From the perspective of the defender, there have two results of output: (1) If all of the outputs are consistent, the outputs of the three executors are correct $(3, 0)$ or the results of all three outputs are wrong $(0, 3)$; (2) If the two

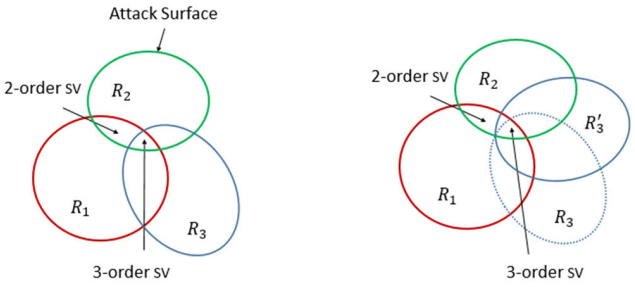

FIGURE 2. The symbiotic vulnerability of attack surface. (a) SV of the three executors. (b) After replacing one of the executors, the change of AS.

results are the same and one exception, there will be two situations: two results are correct and one is wrong $(2, 1)$, or two mistakes with a correct one $(1, 2)$.

Definition 1: The set of DHR output type is denoted as $\Phi = \{\varphi_0, \varphi_1, \cdots, \varphi_k, \cdots, \varphi_{\lfloor n/2 \rfloor}\}$, where $\varphi_k = \{(k, n-k), (n-k, k)\}$ means there are k outputs that are inconsistent with most outputs.

C. ATTACK SURFACE OF DHR

The effectiveness of MTD hopping is described by the attack surface [7]. When the executor is switched, the attack surface of CMD will also change. Similarly, attack surface theory also describes the executor switched of DHR.

Definition 2: Attack Surface (AS) [8] is the set of executors that are protected by defenders to prevent an attack. We denote the set of resources belonging to the executor's attack surface as AS_{R_i}.

Definition 3: Symbiotic Vulnerability (SV) is the same vulnerability between executors. In actual situations, it is difficult to achieve complete heterogeneity between the executors. We assume that the executors have a relationship:

$$AS_{R_1} \cap AS_{R_2} \cap AS_{R_3} \neq \emptyset$$

As shown in Fig. 2(a), the SV between three executors is called the 3-order SV. And so on, the vulnerability between k executors is called the k-order SV.

Definition 4: SV shifting means that the online service executor is replaced with a backup executor. During time t_1 to t_2, the set of SVs also changes, as shown in Fig. 2(b).

$$\bigcap_{i=1} AS_{R_i}(t_1) \neq \bigcap_{i=1} AS_{R_i}(t_2)$$

There are two ways to attack the DHR: single-mode attack and common-mode attack. (1) A single-mode attack exploits the non-SV. We suppose that each single-mode attack will successfully control one executor; (2) A common-mode attack is to exploit SV to attack multiple executors at the same time. Generally, the attacker can successfully control multiple executors that have the same SV with one attack.

TABLE 1. Summary of variable names and descriptions.

Variable	Description
S_n	The DHR has n failed executors
σ_k	Average time to be attacked k-order SV
β	Average time to random disturbance
$\rho_{i,j}$	Transition probability from state i to state j
m_i	Attack type
$\tilde{p}(m_i\|\varphi_k)$	The posterior possibility of the type of attack's type m_i under the condition that the output φ_k are observed
p	The set of prior probability for the attack type of the defender
$\pi(m_i, S_k)$	Stability probability of attacking type m_i in state S_k
rd	Random disturbance selects the number of reconfigurable executors
c	Average time to reconfigure
T_t	Simulation time

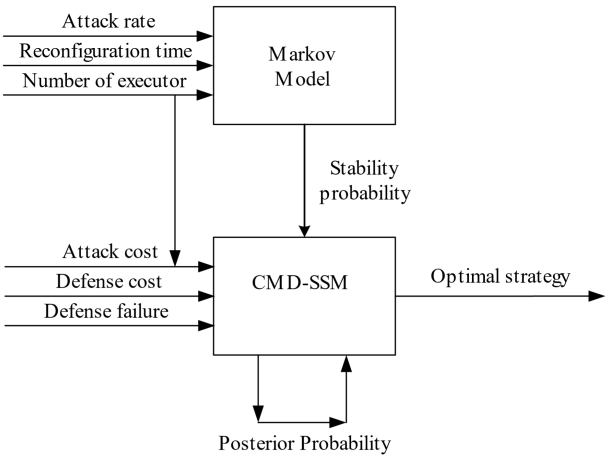

FIGURE 3. Analytic model framework.

III. MODEL FRAMEWORK

For the convenience of explanation, we summarize notations and variables in Table 1.

As shown in Figure 3, the framework for analytic models shows a Markov model at the top and a CMD-SSM at the bottom. Firstly, models are needed to determine the impact of random disturbance period and attack period on system security and the probability of attackers' success. We use Continuous Time Markov Chains (CTMC) to compute the probability distribution of the number of hacked executors. Markov chain is used to describe the change of system state in the process of network confrontation. The Markov models take as inputs the launched attack time, the reconfiguration time, and the number of executors, and produces as outputs the stability probability of the number of invaded executors. The stability probability, along with the attack cost, the defensive cost, and the defensive failure cost are inputs to CMD-SSM, which produces the optimal strategy.

IV. MARKOV PROCESS OF CMD

This section introduces Markov chains to describe the state transition of an executor under single-mode attacks,

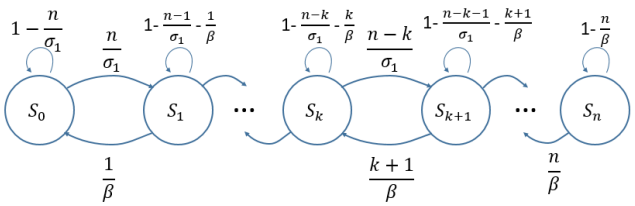

FIGURE 4. Markov model of single-mode attack.

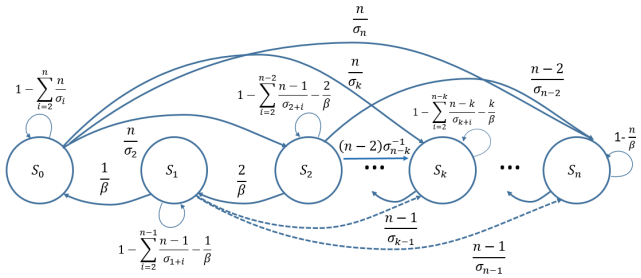

FIGURE 5. Markov model of common-mode attack.

common-mode attacks, and hybrid attack models. The assumption of there is enough redundant executors and no waiting in the switch executor.

A. DESIGN PRINCIPLE

We supposed the environment consists of n executors, and the average rate of attack that can cause effective damage is σ_j which indicates that the attacker exploits successfully j-order SV to control the executors. Each type of attack is independent. The random disturbance periodically selects the executors to reconfigure. The random disturbance period is β, and *rd* executor(s) are randomly selected to be reconfigured. We first introduce the actions of the attacker and the defender in the model.

The following qualitatively describe the actions:

1) Defending action: Periodically replace the service executor with the backup pool of executors.
2) Attacking action: An attacker achieves the purpose of controlling the executor by exploiting the vulnerability, get a shell, and a Trojan horse.
3) Null action: For attackers, the null attack is invalid and the state does not transfer. For defenders, null action means the reconfigured executor is normal.

$S = \{S_0, S_1, \cdots, S_k, \cdots, S_n\}$ represents states that the number of executors is successfully invaded where S_k represents k executors are controlled by the attacker. S_0 is the initial state of the system, in which the attacker does not control any executors. As time progresses, an attacker successfully invades the executor with transfer probability $\rho_{i,j}$, which is in state S_1 at this time, and the subsequent states indicate the number of executors invaded by the attacker. When the attacker controls all executors, the state is S_n. In this case, CMD is completely controlled by the attacker. From the perspective of the defender, the defensive strategy is adopted to select the number of executors for reconfiguration. As a defender, it is difficult to know which executor is in an abnormal state. In this case, the executor is randomly selected for reconfiguration. If the selected executor is normal, it is a null action for the defender. When the state moves toward S_0, it means that the number of controlled executors is decreased until the state returns to 0 and the attacker loses control of any of the executors.

B. SINGLE-MODE ATTACK MODEL

Since the defender plays a memoryless game, it does not matter whether the attacker attacks each executor in a known fixed order. This adaptive confrontation strategy is described by the Markov chains in Fig. 4. Markov chains indicate the birth-death process. The abnormal executor is controlled by the attacker unless it is reconfigured. An attacker who adopts a single-mode attack the system with an attack rate of σ_1, and the system reconfigures the executor at a time interval β. The two events of attack rate and random disturbance period are independent. The transfer probability $\rho_{i,j}$ can be expressed as

$$\rho_{k,k+1} = \frac{n-k}{\sigma_1},\ for\ 0 \le k \le n-1 \tag{1}$$

$$\rho_{k+1,k} = \frac{k+1}{\beta},\ for\ 0 \le k \le n-1 \tag{2}$$

$$\rho_{k,k} = 1 - \rho_{k,k+1} - \rho_{k,k-1},\ for\ 0 \le k \le n-1 \tag{3}$$

$$\rho_{0,0} = 1 - \rho_{0,1}\ and\ \rho_{n,n} = 1 - \rho_{n,n-1} \tag{4}$$

C. COMMON-MODE ATTACK MODEL

In Fig. 5, a common-mode attack enables an attacker to exploit SV to invade multiple executors at one time. Assuming the system has a 2-order SV, the attacker launches an attack against this vulnerability. Then the attacker's request is copied to n copies by the request distribution and sent to each executor. The attacker simultaneously invades the executor with the same vulnerability. We suppose that the attack rate of i-order SV is σ_i. And $\sigma_i\ (i = 2, \ldots, n)$ are independent events. Also, the defender will adopt random disturbance with period β.

$$\rho_{k,k+j} = \frac{n-k}{\sigma_j},\ for\ 0 \le k \le n-1, 2 \le j \le n-j \tag{5}$$

$$\rho_{k+1,k} = \frac{k+1}{\beta},\ for\ 0 \le k \le n-1 \tag{6}$$

$$\rho_{k,k} = 1 - \rho_{k,k-1} - \sum\nolimits_{i=2}^{n-k} \rho_{k,k+i},\ for\ 0 \le k \le n-1 \tag{7}$$

$$\rho_{0,0} = 1 - \sum\nolimits_{i=2}^{n} \frac{n}{\sigma_i}\ and\ \rho_{n,n} = 1 - \rho_{n,n-1} \tag{8}$$

D. HYBRID ATTACK MODEL

Usually, the defender cannot distinguish the attack types, so the hybrid attack model combines single-mode attack and common-mode attack. And the defender can adopt different defense methods, which is to selected *rd* executor(s) for

reconfiguration at the same time.

$$\rho_{k,k+j} = \frac{n-k}{\sigma_j},\ for\ 0 \le k \le n-1, 1 \le j \le n-j \quad (9)$$

$$\rho_{k+j,k} = \frac{C_k^j}{C_n^j}\frac{k+1}{\beta},$$

$$for\ 0 \le k \le n-1, \begin{cases} if\ k - rd \ge 0, & 1 \le j \le rd \\ if\ k - rd < 0, & 1 \le j < k \end{cases} \quad (10)$$

$$\rho_{k,k} = 1 - \sum_{i=1}^{n-k} \rho_{k+i,k} - \sum_{i=1}^{n-k} \rho_{k,k+i},$$
$$for\ 0 \le k \le n-1 \quad (11)$$

$$\rho_{0,0} = 1 - \sum_{i=1}^{n} \frac{n}{\sigma_i}\ and\ \rho_{n,n} = 1 - \rho_{n,n-1} \quad (12)$$

V. CMD STRATEGY SELECTION MODEL

In face of highly adaptive and stealthy attacks, the system is not easy to find abnormalities. When an attacker controls all executors, the system cannot determine whether the attacker completely controls the system. Therefore, a dynamic game based on incomplete information is proposed. According to the probability that the executor may be invaded and the type of attacker, the reconfigured executors are reduced by the optimal defensive strategy.

A. GAME OF INCOMPLETE INFORMATION

In the network confrontation, the defender usually deploys and sets the defensive strategy in advance. Both sides of the attack and defense do not tell each other their key information, and the attacker can obtain the target information through network attacks. Defenders change AS to reduce the time of vulnerability exposure. Offensive and defensive confrontation have the characteristics of incomplete information, both sides do not want their strategy exposed to the other side.

The attack and defense want to be dominant in the confrontation which is like a game. The game is dynamic, which means that the attack and defense actions are carried out in sequence. The attacker adopts the corresponding attack strategy based on a priori knowledge of the defensive type. And then the defender observes that the attacker's attack strategy adopts targeted to take the defensive strategy. The following three executors are used as examples to introduce the game process.

According to Harsanyi transformation [9], virtual participant N_v gives attack type space M_a and a priori probability p. The defender adjusts the defensive strategy by observing the attack behavior and constantly correcting the belief in the attack type. The game tree describes the process selected by the attacker and defender, as shown in Fig. 6. virtual participant N_v randomly selects an attack type m_i from the attack type space, which contains single-mode attack m_1 and common-mode attack m_2. After the attacker send the single, the arbiter has two kinds of output $\Phi = \{\varphi_0, \varphi_1\}$.

After observing Φ, the defender first applies Bayesian law to obtain the posterior probability from the prior probability and selects a strategy from the defensive strategy space D.

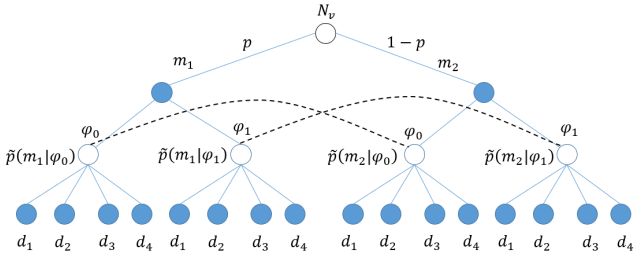

FIGURE 6. Attack and defense game tree.

The defensive strategy concerns the number of executors which should be selected to reconfigure and replaced by the news from the backup pool of executors.

The defenders' decision-making methods are connected by dotted lines. Because the attacker's type is unknown, these nodes that the defender cannot distinguish form an information set. After the defender observes the output Φ, the attacker type is inferred according to the posterior probability $\tilde{p}(m_1|\varphi_0)$ or $\tilde{p}(m_1|\varphi_1)$. Therefore, the attacker has two kinds of attack type, m_1 and m_2, each of which contains a single node corresponding to its type. The two kinds of outputs come from different types of attacks, that is, $\{(\varphi_0|m_1), (\varphi_0|m_2)\}$ and $\{(\varphi_1|m_1), (\varphi_1|m_2)\}$.

After both sides act, players who depend on the sequence of actions obtain different benefits. When computing the benefits of both parties, many factors must be considered. Attackers cost (C_{AC}) is the time to attack the type of vulnerability spent.

The defender periodically selects the executor from the backup pool to replace the online executor. The cost of the defender (C_{DC}) comes from the number of executors that choose to reconfigure. The more executors that are reconfigured, the greater the defensive cost. If the selected executor is normal, the attacker continues to control the abnormal executor leading to the defensive failure cost (C_{DF}).

B. THE CONSTRUCTION OF CMD-SSM

The goal of the attacker is to control the executor to maximize its benefits, while the goal of the defender is to protect key resources at the least cost.

Since the attacker detects the information of the executor to observe the behavior of the defender who sends a signal to the DHR. Because the model is a multi-stage game, the defender will infer the type of attacker based on historical data.

This paper uses game based on incomplete information to describe the non-cooperative, incomplete information, multi-stage dynamic process of network confrontation. In a continuous-time period $T_t\ (t \in Z^+)$, the strategy of attack and defense, which is repeated all the time, is to interact with each other in the game. We assume that the attacker is the sender of the signal and the defender is the receiver of the signal. The defender analyzes the output to infer the output type and modifies the prior probability for the defense strategy optimally.

Definition 5: CMD strategy selection model (CMD-SSM) consists of eight-tuple $\left(N, M, \Phi, D, P, \tilde{P}, R, U\right)$:

1) $N = \{N_a, N_d\}$ is the player set. There are only two players in CMD-SSM, which is an attacker N_a and defender N_d.
2) $M = \{M_a, M_d\}$ represents the type sets of all players' in the attacker and defender space. $M_a = \{m_1, m_2, \cdots, m_n\}$ means the type sets of the attacker. And the only type of defender $M_d = \{m_d\}$.
3) $\Phi = \left\{\varphi_0, \varphi_1, \cdots, \varphi_m\right\}$ represents the output's types set.
4) $D = \{d_1, d_2, \cdots, d_k\}$ represents the strategy set of the defender.
5) $P = \left\{p_1, p_2, \cdots, p_n\right\}$ represents the set of the prior probability of various attack types, according to their initial judgment on the type of attack. It satisfies $\sum_{i=1}^{n} p_i = 1, (p_i > 0)$.
6) $\tilde{P} = \left\{\begin{array}{c} \tilde{p}\left(m_i|\varphi_m\left(T_1\right), h_a\left(T_1\right)\right), \cdots, \\ \tilde{p}\left(m_i|\varphi_m\left(T_t\right), h_a\left(T_t\right)\right) \end{array}\right\}$ represents the historical set of posterior probability that inferences on attack strategies by defenders. $\tilde{p}\left(m_i|\varphi_m\left(T_t\right), h_a\left(T_t\right)\right)$ is defined in definition 6.
7) $R = \{R_a, R_d\}$ represents the revenue set of attacker and defender. It is decided by all players in the game because both the offensive and defensive sides need to consider the cost and benefit when choosing strategy.
8) $U = \{U_a, U_d\}$ represents the reward set of attacker and defender.

Definition 6: $\tilde{p}\left(m_i|\varphi_m\left(T_t\right), h_a\left(T_t\right)\right)$ represents the posterior probability that an attacker of type m_i launches an attack at T_t time-based on the historical strategy set $h_a\left(T_t\right)$. The defender can infer the type of attacker in the T_{t-1} stage by Bayesian law.

$$\tilde{p}\left(m_i|\varphi_m\left(T_t\right), h_a\left(T_t\right)\right) = \frac{p\left(m_i|h_a\left(T_t\right)\right) p\left(\varphi_m\left(T_t\right)|m_i, h_a\left(T_t\right)\right)}{\sum_{m_i \in M_a} p\left(m_i|h_a\left(T_t\right)\right) p\left(\varphi_m\left(T_t\right)|m_i, h_a\left(T_t\right)\right)}$$

where, $h_a\left(T_t\right)$ is the historical strategy of the attacker before T_t. $p\left(m_i|h_a\left(T_t\right)\right)$ is the probability of the attacker type m_i in the history strategy and $p\left(\varphi_m\left(T_t\right)|m_i, h_a\left(T_t\right)\right)$ is the probability that the attacker chooses a strategy $\varphi_m\left(T_t\right)$ base on the historical strategy $h_a\left(T_t\right)$.

Definition 7: The attack cost $C_{AC}\left(\varphi_m, m_i\right)$ indicates the attacker finds the vulnerability and exploits it from the executors.

$$C_{AC}\left(\varphi_m, m_i\right) = \sum\nolimits_{s \in \varphi_m} \pi\left(m_i, S_k\right) \sigma_k \tag{13}$$

where, $\pi\left(m_i, S_k\right)$ is stability probability of attacking type m_i in state S_k, and σ_k is the type of k-order SV.

Definition 8: The defensive cost is $C_{DC}\left(d_h, m_i\right)$ means the cost of the defender's defensive strategy.

$$C_{DC}\left(d_h, m_i\right) = a * c \tag{14}$$

where a is the selected number of executors to be reconfigured, and c is the average time to reconfigure.

Definition 9: Defensive failure cost $C_{DF}\left(d_h, m_i\right)$ means that executors are randomly selected from the serving executor for reconfigured, which may lead to the fact that the attacked executors are not selected. The defensive failure cost is:

$$C_{DF}\left(d_h, m_i\right) = (n-k) \sum\nolimits_{m_i \in M_a} \pi\left(m_i, S_k\right) \beta \tag{15}$$

C. REVENUE CALCULATION

According to the above definition, the attack and defensive reward can be defined.

$$R_a\left(m_i, \varphi_m, d_h\right) = C_{DC}\left(d_h, m_i\right) - C_{AC}\left(\varphi_m, m_i\right) \tag{16}$$

$$R_d\left(m_i, \varphi_m, d_h\right) = C_{AC}\left(\varphi_m, m_i\right) - C_{DC}\left(d_h, m_i\right) - C_{DF}\left(d_h, m_i\right) \tag{17}$$

The revenue to the attacker is

$$U_a\left(\Pi_a, \Pi_d\right) = \sum\nolimits_{m \in M} \sum\nolimits_{\varphi \in \Phi} \sum\nolimits_{d \in D} R_a\left(m, \varphi, d\right) p_a \tag{18}$$

The revenue to the defender is

$$U_d\left(\Pi_a, \Pi_d\right) = \sum\nolimits_{m \in M} \sum\nolimits_{\varphi \in \Phi} \sum\nolimits_{d \in D} R_d\left(m, \varphi, d\right) \times \gamma\left(d\right) \tilde{p}\left(m|\varphi\left(T_t\right), h_a\left(T_t\right)\right) \tag{19}$$

where, $\gamma\left(d\right)$ is the influencing factor of the defender's revenue.

If there is a set of $\Pi_d\left(\gamma\left(d_1\right), \gamma\left(d_2\right), \cdots, \gamma\left(d_k\right)\right)$, no matter what strategy the attacker selects, the attacker's profit no less than a fixed profit. This is called perfect Bayesian equilibrium, and its mathematical meaning is as described follows.

For both sides of the game, there are attack and defense strategies in each stage of the game $\left(\Pi_a^*, \Pi_d^*\right)$. For all Π_d, there are $U_a\left(\Pi_a^*, \Pi_d^*\right) \geq U_d\left(\Pi_a^*, \Pi_d\right)$. Similarly, for all Π_a, there are $U_d\left(\Pi_a^*, \Pi_d^*\right) \geq U_a\left(\Pi_a, \Pi_d^*\right)$. So, the mixed strategy $\left(\Pi_a^*, \Pi_d^*\right)$ is a perfect Bayesian equilibrium, which can ensure that when both sides adopt a Nash equilibrium strategy, it can maximize the confrontation revenue. For a round of the game, the stability probability and the value of variables are taken as inputs shown in Table 2. If the strategies of both players are limited, there is a perfect Bayesian equilibrium [10].

CMD-SSM perfect Bayesian equilibrium solution:

1) The defender establishes a posterior probabilistic inference on each information set $\tilde{p}\left(m_i|\varphi_m\left(T_t\right), h_a\left(T_t\right)\right)$.
2) The attacker deduces the optimal output strategy φ^*.
3) The defender deduces the optimal defense strategy d^*.
4) Perfect Bayesian equilibrium solution (φ^*, d^*).

VI. EXPERIMENTAL RESULTS AND ANALYSIS

Before the launched attack, the attacker scan system vulnerabilities, IP address, service port, the operating system type, and version. Assumed that the attacker has collected enough detailed information, such as the system architecture and existing vulnerabilities. In an actual environment, collecting information is a complicated and time-consuming task.

TABLE 2. Values of variables used in the numerical results.

Variable	Description
T_t	The total simulation is 10^5 time steps
n	The number of online service executors is 3
σ_1	Average 120-time steps to be attacked *1*-order SV
σ_2	Average 240-time steps to be attacked *2*-order SV
σ_3	Average 480-time steps to be attacked *3*-order SV
β	Average 15-time steps to random disturbance
c	Average 10-time steps to being reconfigured

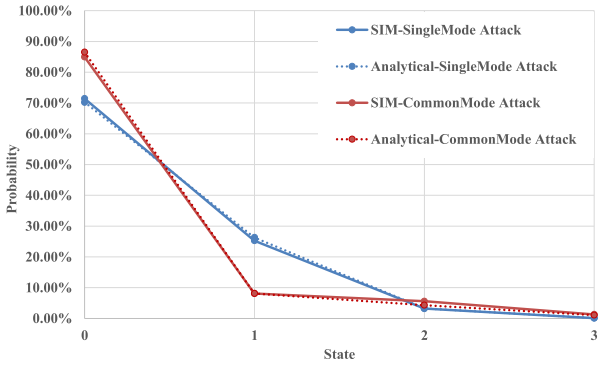

FIGURE 7. The probability of the number of failed executors.

Hence, the actual attack success rate is lower than the model's success rate.

The following section contains two experiments: (1) In part A discusses the effectiveness of the Markov models and compares their threats; (2) In part B, based on the stability probability calculated by part A and the revenue, we can establish the CMD-SSM model. We compare the security of optimal strategy selection and random disturbance under different *rd*.

A. COMPARE SINGLE-MODE ATTACK WITH COMMON-MODE ATTACK

We use Gauss-Seidel [11] to solve the stability probability of Markov chains and SimPy [12] to simulate experiments. SimPy, which supports the competing resource access of multiple processes, is a process discrete event simulation framework based on Python, and automatically processes the event queue when resources are busy to meet the simulation conditions of the experiment.

The experimental simulations compare the state changed between the single-mode attack and the common-mode attack. Assuming there is enough redundant executor of the back pool of executors.

In the experiment, a single-mode attack and a common-mode obey the exponential distribution. And the input parameters are shown in Table 2. During the simulation process, the executor switching is instantaneous, and the result is shown in Fig. 7.

The effectiveness of the model is verified by simulation experiments and the stable probability of the Markov chain. We found that the probability distribution generated by the simulation is close to the stability probability of the analytical model, as shown in Fig. 7. For the same parameters used in a single-mode attack, the stability probability maximum error between the analytical calculation and the simulation result does not exceed 1.2%. Also, the maximum error in a common-mode attack does not exceed 1.6%.

To show the state change of DHR under single-mode attack and common-mode attack, we select 1440-time steps in the experiment as shown in Fig. 8. The initial state of the system is S_0, which means that all executors are in a normal state. When the system reached S_2, the output of the system is controlled by the attacker, which means the system is in an unsafe state. The process of an attacker's game starts from S_0 and reaches S_2 after transitions of state. The results show that the common-mode attack is more destructive than the single-mode attack. In the single-mode attack, there does not reach the state S_3. In the common-mode attack, there have 83 states S_3, and the single-mode attack in S_2 is much less than the common-mode attack. From a security point of view, the greater number of states in S_0, the more secure the system. In terms of system architecture, the single-mode attack can be regarded as the only way to attack completely heterogeneous DHR. In other words, there is no SV in DHR, and the security of the system is also improved.

B. ANALYSIS OF THE PERFORMANCE AND SECURITY OF CMD-SSM

The number of executors in reconfiguration guides how many executors we need to develop. In this experiment, CMD-SSM is compared with a fixed number of reconfigurations, which verifies that CMD-SSM reconfigures fewer executors. The values of the input parameters used in the theoretical and simulations are given in Table 2.

We verify the effectiveness of the hybrid attack model. The experiment selects *rd* executor(s) to reconfigure with the hybrid attack model. The theoretical and simulation value of the model is shown in Table 3. The maximum error between the theoretical and simulation is less than 4.6%.

When the attacker successfully controls the two executors, the attacker can cheat the result of the arbiter in a short time. When the attacker controls all executors, the system cannot judge whether an exception through the output of the executor. Therefore, the system in the state S_2 and S_3 is insecurity. Fig. 9 shows the proportion of the duration of an insecure state in the simulation time of different *rd*.

When $rd = 1$, the system is in an unsafe state about 15% of the simulation time. After reconfiguring the $rd = 3$ executors, the system ensures that all executors are normal. Therefore, it is not a feasible way to improve security by selecting more executors for reconfiguration. The ratio of insecure time of $rd = 2$, $rd = 3$ and CMD-SSM is close. Hence, the security of CMD-SSM can be guaranteed.

There are two types of attacks in CMD-SSM, single-mode attack and common-mode attack. According to the optimal strategy of the CMD-SSM, the defensive strategy can randomly select 0 to 3 executor(s) to replace.

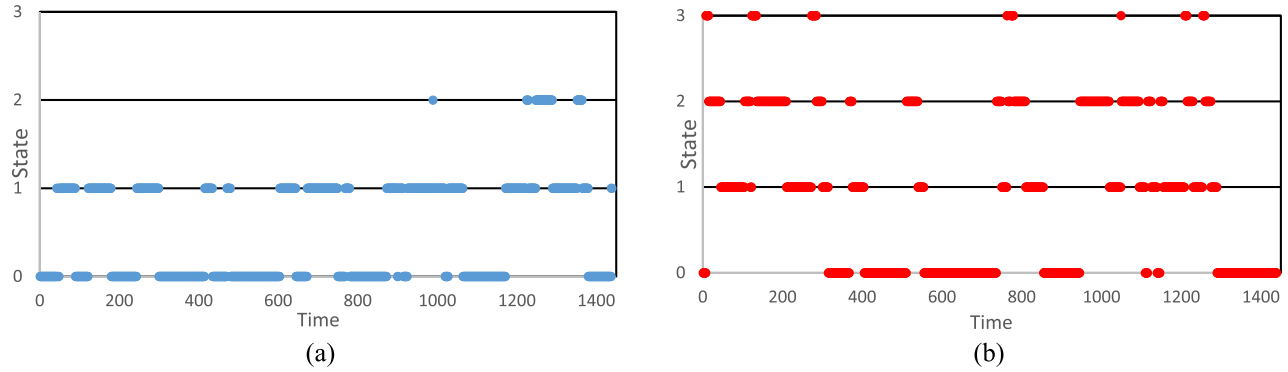

FIGURE 8. The state distribution transfer process. (a) Single-mode attack. (b) Common-mode attack.

TABLE 3. Comparison of simulation and analytical under different *rd*.

Defensive type / State	$rd = 1$		$rd = 2$		$rd = 3$	
	Simulation (%)	Analysis (%)	Simulation (%)	Analysis (%)	Simulation (%)	Analysis (%)
S_0	77.976	76.691	75.476	80.102	90.311	86.583
S_1	13.360	16.777	19.028	15.349	5.149	9.114
S_2	6.769	5.168	3.099	3.259	2.920	1.220
S_3	1.892	1.364	2.394	1.290	1.617	3.083

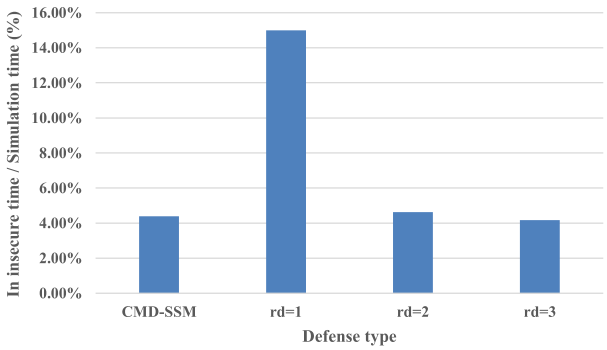

FIGURE 9. The ratio of insecurity time to the simulation time under different *rd*.

In Fig. 10, we select 750-time steps to show the number of executors being reconfigured and DHR status. CMD-SSM outperforms $rd = 2$ and $rd = 3$ in performance. The maximum number of CMD-SSM in reconfiguration is 4 while $rd = 2$ and $rd = 3$ have 6 and 9 respectively. Therefore, we know that the minimum number of executors required for different random disturbance methods. The times of executor switching using CMD-SSM are significantly less than that of random disturbance mode.

VII. RELATED WORK

Active defense has become a hot research field because it can effectively prevent attacks and increase the difficulty of the attacker to collect system information. The cyberspace security defense mechanisms include honeynets [13], moving target defense [1], obfuscation [14], perturbations [15]. [16] and cyberspace mimic defense [2].

Some researchers have proposed different metrics to quantify the effectiveness of the model. Carroll *et al.* [17] proposed models that quantify the probability of attacker success in terms of network size, addresses scanned, vulnerable, and the frequency of shuffling. Rahman *et al.* [18] combine the degree of the system vulnerability and perform an attack based on resource level to quantify the probability that attacks are successfully performed. Cho *et al.* [19] use Stochastic Petri Nets to create a model that describes an integrated defense system. They used the probability of a successful attack to quantify the meantime to security failure. Jafarian *et al.* [20] quantify the effectiveness of MTD to introduce three novel metrics including deterrence, deception, and detectability. Connell *et al.* [21] proposed a quantitative analytic model which evaluates the availability and performance of resource. Meanwhile, the mothed can minimize the probability of success and ensure the performance and stability of maximum reconfiguration rate. Maleki *et al.* [22] proposed a Markov decision process for the MTD game model which shows the results on how the probability of the attacker defeating the MTD strategy and the time/cost spent with the attacker. Chowdhary *et al.* [23] formulate a zero-sum Markov game and use the Common Vulnerability Scoring System(CVSS) to come up with meaningful utility values for this game.

Because of the high compatibility between offensive and defensive confrontation and game theory, the research of game theory based on active defense has been greatly developed. Prakash *et al.* [24] used empirical game-theoretic techniques in a generic cyber-defense scenario, and the result shows that the defender tends to actively move when the detection capabilities are hampered. Kiekintveld *et al.* [25]

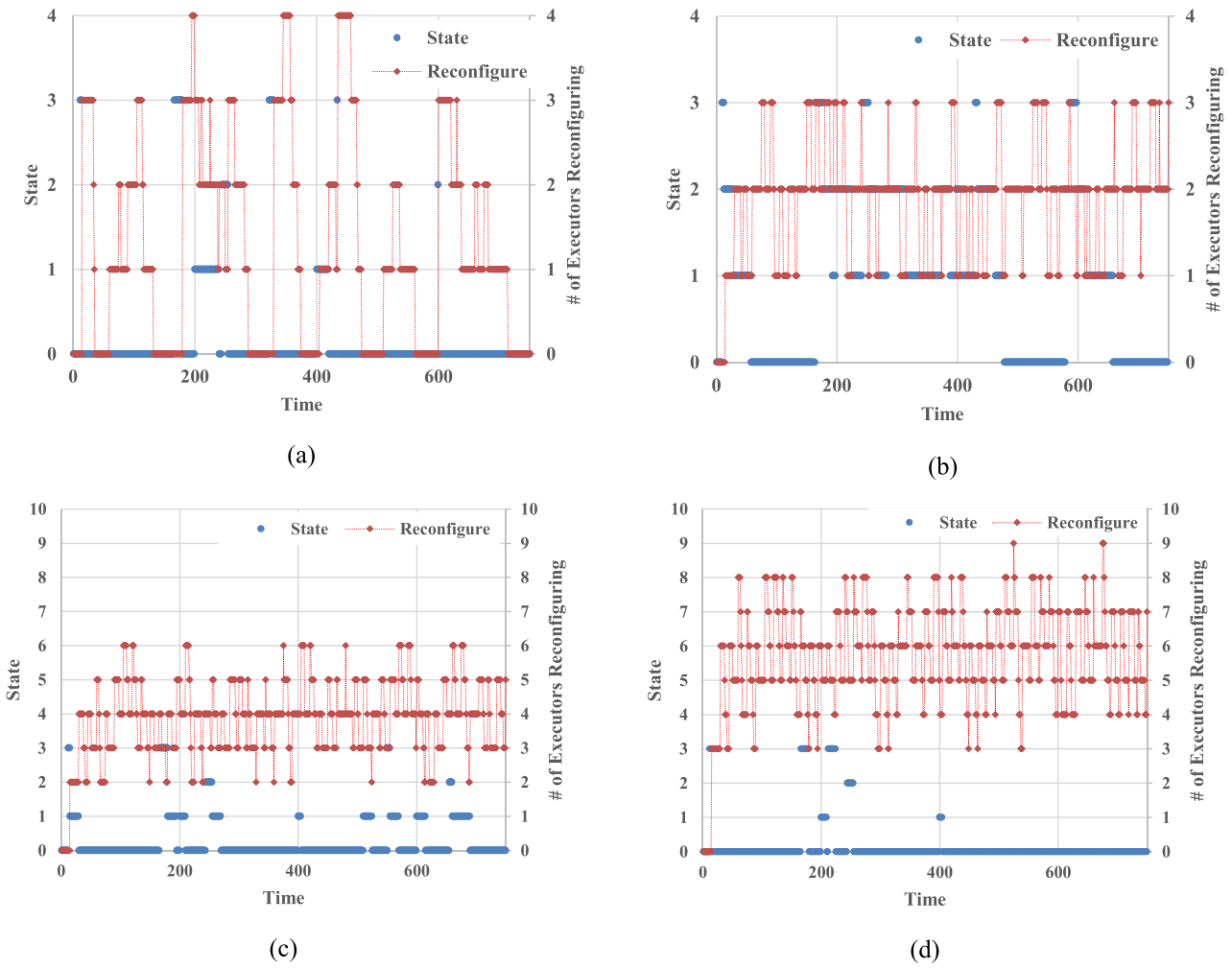

FIGURE 10. The number of executors being reconfigured and the state of DHR. (a) CMD-SSM, (b) $rd = 1$, (c) $rd = 2$, (d) $rd = 3$.

discuss three game-theoretic models that address strategies for deploying honeypots. However, most of these secure game frameworks assume complete information. The limitations of single-state or single-stage hypotheses reduce the value and practicality of research results. Yang Liu *et al.* [26] combines the privacy of participants, the randomness of task arrival, and the cost of the platform.

The incomplete information game is a natural framework for modeling the uncertainty and misinformation brought about by network confrontation. Zhu *et al.* [27] develop a static Bayesian game for spear phishing, and a finite zero-sum game for the final stage of physical-layer infrastructure protection. Huang *et al.* [28] proposed the system behaviors of the critical infrastructures under malicious attacks and the protection strategies by a zero-sum game. However, most of these secure game frameworks assume complete information. Lei *et al.* [29] proposed an incomplete information Markov game-theoretic approach to strategy generation. Huang *et al.* [30] proposed a long-term interaction between an attacker and a defender based on the multi-stage game of incomplete information. La *et al.* [31] discussed the defense of attacks in a honeypot-enable network that used a Bayesian game of incomplete information. Yang Liu *et al.* [32] used Q-learning to the interactions between platforms and participants as a multi-leader multi-follower Stackelberg game and derive the Stackelberg equilibrium (SE) of the game.

VIII. CONCLUSION

CMD is a technology to change the network that is easy to attack and difficult to defend. First, we establish a DHR anti-attack Markov chains. Under theoretical analysis and simulation, it is proved that the effectiveness of Markov models and the maximum error between theoretical and simulation is not more than 1.6%. The simulation shows that common-mode attacks are more harmful than single-mode attacks. Since the attacker and defender do not know each other's information, non-cooperative relationship and dynamic games, which is similar to the feature of game theory. We propose a dynamic game model based on incomplete information for the optimal defensive strategy. We use the game theory to describe multi-stage network confrontation. The simulation results show that CMD-SSM ensures safety and reduces

reconfigured executors. In terms of security, the insecure time with CMD-SSM is in an insecure state is less than the case of random disturbance $rd = 1$, which is close to the time of $rd = 2$ and $rd = 3$. In terms of reconfigured executors, the number of executors being reconfigured is less than $rd = 2$ and $rd = 3$.

REFERENCES

[1] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, vol. 54. New York, NY, USA: Springer, 2011.

[2] H. Hu, J. Wu, Z. Wang, and G. Cheng, "Mimic defense: A designed-in cybersecurity defense framework," *IET Inf. Secur.*, vol. 12, no. 3, pp. 226–237, May 2018.

[3] D. Evans, "Effectiveness of moving target defenses," in *Moving Target Defense*. New York, NY, USA: Springer, 2011, pp. 29–48.

[4] X. He, H. Dai, and P. Ning, "Improving learning and adaptation in security games by exploiting information asymmetry," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2015, pp. 1787–1795.

[5] D. Kewley, R. Fink, J. Lowry, and M. Dean, "Dynamic approaches to thwart adversary intelligence gathering," in *Proc. DARPA Inf. Survivability Conf. Exposit. II. DISCEX*, 2001, pp. 176–185.

[6] M. Garcia, A. Bessani, I. Gashi, N. Neves, and R. Obelheiro, "Analysis of operating system diversity for intrusion tolerance," *Softw., Pract. Exper.*, vol. 44, no. 6, pp. 735–770, Jun. 2014.

[7] R. Zhuang, S. A. DeLoach, and X. Ou, "Towards a theory of moving target defense," in *Proc. 1st ACM Workshop Moving Target Defense (MTD)*, 2014, pp. 31–40.

[8] P. K. Manadhata and J. M. Wing, "An attack surface metric," *IEEE Trans. Softw. Eng.*, vol. 37, no. 3, pp. 371–386, May 2011.

[9] J. C. Harsanyi, "Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility," *J. Political Economy*, vol. 63, no. 4, pp. 309–321, Aug. 1955.

[10] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. New York, NY, USA: Springer-Verlag, 1996.

[11] W. J. Stewart, *Computations With Markov Chains: Proceedings of the 2nd International Workshop on the Numerical Solution of Markov Chains*. New York, NY, USA: Springer, 2012.

[12] N. Matloff, "Introduction to discrete-event simulation and the simpy language," Dept. Comput. Sci., Univ. California Davis, Davis, CA, USA, Aug. 2008, pp. 1–33, vol. 2, no. 2009.

[13] L. Huang and Q. Zhu, "Adaptive honeypot engagement through reinforcement learning of semi-Markov decision processes," in *Proc. Int. Conf. Decis. Game Theory Secur.* Cham, Switzerland: Springer, 2019, pp. 196–216.

[14] J. Pawlick and Q. Zhu, "A Stackelberg game perspective on the conflict between machine learning and data obfuscation," in *Proc. IEEE Int. Workshop Inf. Forensics Secur. (WIFS)*, Dec. 2016, pp. 1–6.

[15] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 172–187, Jan. 2017.

[16] T. Zhang and Q. Zhu, "Distributed privacy-preserving collaborative intrusion detection systems for VANETs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 148–161, Mar. 2018.

[17] T. E. Carroll, M. Crouse, E. W. Fulp, and K. S. Berenhaut, "Analysis of network address shuffling as a moving target defense," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 701–706.

[18] M. A. Rahman, E. Al-Shaer, and R. B. Bobba, "Moving target defense for hardening the security of the power system state estimation," in *Proc. 1st ACM Workshop Moving Target Defense*, 2014, pp. 59–68.

[19] J.-H. Cho and N. Ben-Asher, "Cyber defense in breadth: Modeling and analysis of integrated defense systems," *J. Defense Model. Simul., Appl., Methodol., Technol.*, vol. 15, no. 2, pp. 147–160, Apr. 2018.

[20] J. H. H. Jafarian, E. Al-Shaer, and Q. Duan, "Spatio-temporal address mutation for proactive cyber agility against sophisticated attackers," in *Proc. 1st ACM Workshop Moving Target Defense (MTD)*, 2014, pp. 69–78.

[21] W. Connell, D. A. Menascé, and M. Albanese, "Performance modeling of moving target defenses," in *Proc. Workshop Moving Target Defense*, Oct. 2017, pp. 53–63.

[22] H. Maleki, S. Valizadeh, W. Koch, A. Bestavros, and M. van Dijk, "Markov modeling of moving target defense games," in *Proc. ACM Workshop Moving Target Defense*, Oct. 2016, pp. 81–92.

[23] A. Chowdhary, S. Sengupta, D. Huang, and S. Kambhampati, "Markov game modeling of moving target defense for strategic detection of threats in cloud networks," 2018, *arXiv:1812.09660*. [Online]. Available: http://arxiv.org/abs/1812.09660

[24] A. Prakash and M. P. Wellman, "Empirical game-theoretic analysis for moving target defense," in *Proc. 2nd ACM Workshop Moving Target Defense*, Oct. 2015, pp. 57–65.

[25] C. Kiekintveld, V. Lisý, and R. Píbil, "Game-theoretic foundations for the strategic use of honeypots in network security," in *Cyber Warfare*. Cham, Switzerland: Springer, 2015, pp. 81–101.

[26] Y. Liu, T. Feng, M. Peng, J. Guan, and Y. Wang, "DREAM: Online control mechanisms for data aggregation error minimization in privacy-preserving crowdsensing," *IEEE Trans. Dependable Secure Comput.*, early access, Jul. 24, 2020, doi: 10.1109/TDSC.2020.3011679.

[27] Q. Zhu and S. Rass, "On multi-phase and multi-stage game-theoretic modeling of advanced persistent threats," *IEEE Access*, vol. 6, pp. 13958–13971, 2018.

[28] L. Huang, J. Chen, and Q. Zhu, "A large-scale Markov game approach to dynamic protection of interdependent infrastructure networks," in *Proc. Int. Conf. Decis. Game Theory Secur.* Cham, Switzerland: Springer, 2017, pp. 357–376.

[29] C. Lei, H.-Q. Zhang, L.-M. Wan, L. Liu, and D.-H. Ma, "Incomplete information Markov game theoretic approach to strategy generation for moving target defense," *Comput. Commun.*, vol. 116, pp. 184–199, Jan. 2018.

[30] L. Huang and Q. Zhu, "A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems," *Comput. Secur.*, vol. 89, Feb. 2020, Art. no. 101660.

[31] 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.

[32] Y. Liu, H. Wang, M. Peng, J. Guan, and Y. Wang, "An incentive mechanism for privacy-preserving crowdsensing via deep reinforcement learning," *IEEE Internet Things J.*, early access, Dec. 24, 2020, doi: 10.1109/JIOT.2020.3047105.

ZEQUAN CHEN received the B.Eng. degree from Southeast University, in 2018. He is currently pursuing the master's degree. His research interests include distributed systems and cyber security.

GANG CUI graduated in electrical engineering and automation from the Hebei University of Technology. He served at China Nuclear Power Engineering Company Ltd, as a Senior Electrical Engineer and the Electrical Secondary Chief Designer. His research interests include network security of power industrial control systems, reliability supervision management, and data submission of power monitoring systems. MIN: Co-Governing Multi-Identifier Network Architecture and Its Prototype on Operator's Network was applied to nuclear power industrial control network security.

LIN ZHANG graduated in electrical engineering and automation from Shanghai Electric Power University. He was an Electrical Professional Engineer and the Electrical Chief Design Engineer at China Nuclear Power Engineering Company Ltd. His research interests include cyber security defense in depth of electrical power monitoring system for nuclear power station, prevention and control strategy, and network security supervisory. MIN: Co-Governing Multi-Identifier Network Architecture and Its Prototype on Operator's Network was applied to nuclear power industrial control network security.

XIN YANG (Student Member, IEEE) received the B.Eng. degree from the Department of Computer Science and Engineering, South China University of Technology, in 2016. She is currently pursuing the Ph.D. degree with the School of Information Science, Peking University. Her research interests include cyber security, future network architecture, and distributed storage systems.

HUI LI received the B.Eng. and M.S. degrees from the School of Information Engineering, Tsinghua University, Beijing, China, in 1986 and 1989, respectively, and the Ph.D. degree from the Department of Information Engineering, The Chinese University of Hong Kong, in 2000. He was the Director of Shenzhen Key Laboratory of Information Theory and Future Internet Architecture and PKU Laboratory of China Environment for Network Innovations (CENI), National Major Research Infrastructure. He is currently a Full Professor with the Shenzhen Graduate School, Peking University. His research interests include network architecture, cyberspace security, distributed storage, and blockchain. He proposed the first co-governing future networking MIN based on blockchain technology and implemented its prototype on operator's network in the world, and this project MIN: Co-Governing Multi-Identifier Network Architecture and Its Prototype on Operator's Network was received the award of World Leading Internet Scientific and Technological Achievements by the 6th World Internet Conference on 2019, Wuzhen, China.

YAN ZHAO received the Bachelor of Communication Engineering degree from Sun Yat-sen University. He served with Network Department of Shenzhen SmartCity Communication Company Ltd. He has worked at China Unicom and Huawei Technologies Company Ltd. His main research fields and expertise involve smart city operation, information security, V2X, and information operation and maintenance.

CHENGTAO MA received the master's degree in software engineering from the University of Science and Technology of China, in 2007. He is currently the Director of the Information Center, Guangdong Yuegang Water Supply Company Ltd. His research interests include machine learning, edge computing, data mining techniques and applications, blockchain, and cyber security.

TAO SUN received the Master of Engineering degree in communication and information system from the University Town of Shenzhen. He was a Senior Engineer and the Director of the Network Information Center, University Town of Shenzhen. His research interests include computer network architecture and information systems. He has an intimate knowledge of the principles and related technologies of data communication and computer network systems. He has a comprehensive grasp of various ICT and its application solutions. He has a deep understanding of the information work in education industry and has a rich experience and unique insights in daily operation and maintenance management of campus networks. He has the skill in the implement of large-scale weak current system and network engineering projects, with practical experience and ability in designing, deploying, and managing large and medium-sized networks. He has familiar in the principles and deployment scheme of mainstream information network systems. He had a certain understanding in the field of national defense and building intelligent technology, planned, and implemented a few building intelligence and security project.