IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 18, 2019, accepted July 27, 2019, date of publication July 30, 2019, date of current version August 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2932020

On Dynamic Recovery of Cloud Storage System Under Advanced Persistent Threats

PENGDENG LI AND XIAOFAN YANG, (Member, IEEE)
School of Big Data & Software Engineering, Chongqing University, Chongqing 400044, China
Corresponding authors: Pengdeng Li (pengdengli1992@gmail.com) and Xiaofan Yang (xfyang1964@gmail.com)

This work is supported by the National Natural Science Foundation of China under Grant 61572006.

ABSTRACT Advanced persistent threat (APT) for data theft poses a severe threat to cloud storage systems (CSSs). An APT actor may steal valuable data from the target CSS even in a strategic fashion. To protect a CSS from APT, the cloud defender has to dynamically allocate the limited security resources to recover the compromised storage servers, aiming at mitigating his total loss. This paper addresses this dynamic cloud storage recovery (DCSR) problem by employing differential game theory. First, by introducing an expected state evolution model capturing the CSS's expected state evolution process under a combination of attack strategy and recovery strategy, we measure the APT attacker's net benefit and the cloud defender's total loss. On this basis and in the worst-case situation where the cloud defender assumes that the APT attacker has full knowledge of his expected loss, we reduce the DCSR problem to a differential game-theoretic problem (the DCSR* problem) to characterize the strategic interactions between the two parties. Second, we derive a necessary condition for Nash equilibrium of the DCSR* problem and thereby introduce the concept of competitive strategy profile. Next, we study the structural properties of the competitive strategy profile, followed by some numerical examples. Then, we conduct extensive comparative experiments to exhibit that the competitive strategy profile is superior to a large number of randomly generated strategy profiles in the sense of Nash equilibrium solution concept. Finally, we briefly analyze the practicability (scalability and feasibility) of this paper. Our findings will be helpful to enhance the APT defense capabilities of the cloud defender.

INDEX TERMS Advanced persistent threat, cloud storage recovery, state evolution model, differential game, Nash equilibrium, necessity system, competitive strategy profile.

I. INTRODUCTION

More and more organizations are moving to cloud. In particular, with the financial justifications and increasing functionality, cloud storage systems (CSSs), which typically encompass a collection of storage servers to provide long-term storage services over the Internet [1], are being embraced by organizations. According to [2], more than 72% of global organizations will migrate to cloud by 2022, and the global cloud storage market is projected to be worth USD 101.59 billion by 2023, cementing the rise of cloud storage.

Despite the huge benefit, a CSS is often confronted with a variety of cyber threats. Among them, advanced persistent threat (APT) for data theft is one of the most serious threats to the CSS. Specifically, the following three phases of an APT campaign may be involved.

- Reconnaissance. Gather information about the CSS and the cloud defender, including the system vulnerabilities and work mode of the cloud defender.
- Infiltration. Combine the information obtained in the reconnaissance phase with social engineering attacks, infiltrate the CSS and then establish footholds.
- Data exfiltration. Encrypt and exfiltrate valuable data from the target CSS stealthily and continuously through an established communication channel.

In general, most traditional cyber attacks like computer viruses are single-run and automatic, with the intent of breaking down systems. As a result, the success rate of these cyber attacks is lower and the chance of being detected is higher. In contrast, APT for data theft is time-continuous and highly motivated, and usually performed by well-funded attackers, aiming at stealing valuable data from the target CSS in a covert fashion over a long period of time without being caught [3]. Through extended reconnaissance and by

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

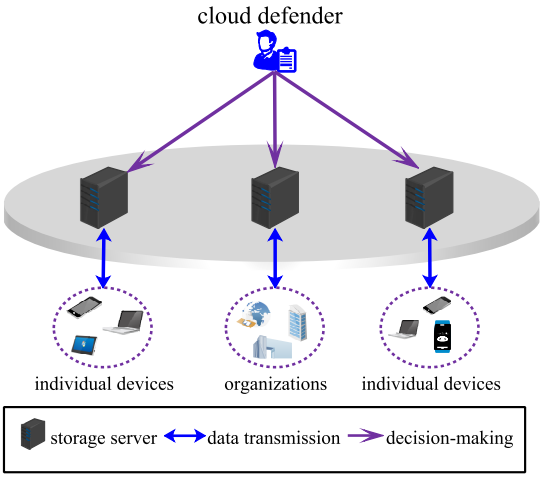

FIGURE 1. Diagram of a typical cloud storage system, where there are multiple storage servers to provide storage services to individuals or organizations, and the cloud defender as the owner is responsible for the decision-making (e.g., recovery) of the whole system.

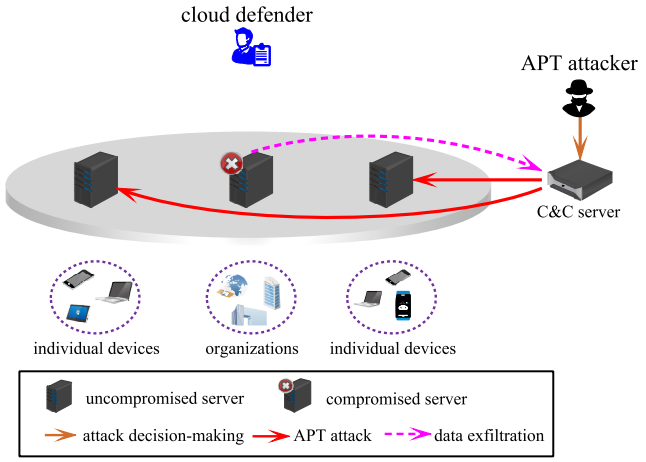

FIGURE 2. Diagram of an APT for data theft on the cloud storage system shown in Fig. 1, where the APT attacker makes the attack decision to compromise the storage servers, and then exfiltrate valuable data from the compromised servers to his C&C server through an established communication channel.

using social engineering attacks, an APT attacker can always infiltrate the CSS, leading to severe data leakage. Due to the fast rate at which the APTs are evolving, it is almost impossible to perfectly protect a CSS from APTs only by employing traditional defense mechanisms such as Intrusion Detection System (IDS) and firewall.

A. MOTIVATION

Consider a CSS consisting of multiple storage servers. The cloud defender as the owner of the CSS is responsible for the decision-making of the whole system. Every day a substantial amount of data will be uploaded to or downloaded from the CSS by individual devices or organizations [4]. See Fig. 1 for the diagram of such a CSS. In this setting, an APT attacker can apply social engineering attacks to the cloud defender to compromise the storage servers and establish footholds. Once having footholds in the target CSS, the APT attacker will be able to encrypt and exfiltrate the valuable data to his remote command-and-control (C&C) server through an established communication channel which applies mainstream protocols such as HTTP, HTTPS, FTP, P2P, and others. See Fig. 2 for the diagram of the APT for data theft on the CSS shown in Fig. 1.

According to [5], when an APT campaign on the CSS is identified and all the compromised storage servers are confirmed by the cloud defender, the next step is to recover all the confirmed compromised storage servers. This recovery work may involve some of the following activities:

- Collect and analyze the system logs, including service logs and traffic logs.
- Search for and kill the suspicious processes in the compromised storage servers.
- Analyze the samples of the attack scripts or toolkits and then develop new patches accordingly.
- Migrate data and then reinstall the systems of the compromised storage servers.

On the one hand, the above recovery work is resource-intensive, i.e., security resources including money and manpower need to be put in place to accomplish this recovery work. On the other hand, the security resources of the cloud defender are usually limited. As a result, the cloud defender has to effectively manage his security resources so that he can make responses in a timely manner when facing an APT.

Resource management, which is the efficient and effective management of an organization's resources and has been widely applied to different research topics such as defense of terrorist attacks [6], trust data sharing in edge computing [7], green services of content-centric IoT [8] and efficient job scheduling and energy-aware resource management in data center networks [9], [10], provides the cloud defender the inspiration to the development of an effective security resources allocation strategy that mitigates his total loss. As the APT campaign is time-continuous, the cloud defender also needs to take continuous actions. In this context, we define a *dynamic recovery (DR) strategy* as a vector-valued function consisting of all the recovery rates of all storage servers at all times in the course of the APT campaign. The DR strategy is controllable by the cloud defender, and the total loss is dependent on the strategy. Therefore, the cloud defender has to deal with the following problem:

Dynamic cloud storage recovery (DCSR) problem: Suppose an APT campaign on a cloud storage system has been identified. Develop an effective DR strategy so that the cloud defender's total loss, which includes the direct loss caused by the leakage of valuable data and the recovery cost for recovering the compromised storage servers, is minimized.

Dealing with this problem will contribute to the enhancement of the APT defense capabilities of the cloud defender.

B. APPROACH

In dealing with the DCSR problem, we need to measure the cloud defender's total loss. This quantity consists of two

parts: the direct loss resulted from the leakage of valuable data and the recovery cost for recovering the compromised storage servers. In our work, the direct loss can be measured by the cloud defender's expected loss, while the recovery cost can be estimated by the expected recovery cost. As both the two measures are closely related to all the CSS's expected states in the course of the APT campaign, we need to establish the CSS's expected state evolution model. In this work, we introduce a differential dynamical system to capture the CSS's expected state evolution process. Thereby, we evaluate the cloud defender's total loss. On this basis, the main task of the cloud defender is to seek a feasible DR strategy so that the total loss is minimized. As the total loss simultaneously relies on the DR strategy and the dynamic attack (DA) strategy which consists of all the attack rates of all storage servers at all times and is typically unknown to the cloud defender, a game-theoretic model is especially suitable to capturing the interactions between the cloud defender and the APT attacker.

In the worst-case situation where the cloud defender assumes that the APT attacker has full knowledge of the his expected loss, seeking a Nash equilibrium is the main task of the cloud defender. As the strategies of both parties are dependent only on time, the equilibrium is an open-loop Nash equilibrium. For convenience, in this paper, unless otherwise specified, we will interchangeably use the two terms *open-loop Nash equilibrium* and *Nash equilibrium*. Given a Nash equilibrium including a DA strategy and a DR strategy. According to the nature of Nash equilibrium, both the cloud defender and the APT attacker cannot achieve a better goal by unilaterally deviating from their respective strategies in the equilibrium. Hence, at least from the worst-case perspective, it is appropriate for the cloud defender to adopt the DR strategy in the equilibrium. Therefore, the DCSR problem is reduced to a game-theoretic problem we refer to as the DCSR* problem, in which the ultimate goal of the cloud defender is to find a Nash equilibrium [11]. A DCSR* game of the DCSR* problem is a differential game, because it relies on a differential dynamical system capturing the CSS's expected state evolution process, and the strategies of both parties are time-varying.

C. CONTRIBUTIONS

The main contributions of this paper are sketched as follows.

- By introducing an expected state evolution model capturing the CSS's expected state evolution process under a combination of DA strategy and DR strategy, we measure the APT attacker's net benefit and the cloud defender's total loss. On this basis and in the worst-case situation where the cloud defender assumes that the APT attacker has full knowledge of his expected loss, we reduce the DCSR problem to a differential game-theoretic problem of seeking a Nash equilibrium, which we refer to as the DCSR* problem.
- According to differential game theory, we derive a necessary condition for Nash equilibrium of the DCSR* problem and thereby introduce the concept of competitive strategy profile. Then we inspect the structural properties of the competitive strategy profile, followed by some numerical examples. Through extensive comparative experiments, we find that the competitive strategy profile outperforms a large number of randomly generated strategy profiles in the sense of Nash equilibrium solution concept. Therefore, we recommend to the cloud defender the DR strategy in the competitive strategy profile. In addition, we briefly analyze the practicability of this work. The results either deepen our understanding of the APT on the CSS or help the cloud defender to enhance the APT defense capabilities.

The rest of this work is organized in the following fashion. We review the related works and illuminate their relationship with our work in Section II. Then we model the DCSR problem as the DCSR* problem in Section III. In Section IV, we derive the necessity system, introduce the concept of competitive strategy profile, examine the structural properties of the competitive strategy profile, and give some numerical examples. The performance of the competitive strategy profile is evaluated in Section V. In Section VI, we briefly analyze the practicability of this work. Finally, Section VII draws conclusions of this work.

II. RELATED WORK

In recent years, the advanced persistent threats (APTs) have posed a severe threat to modern society [5], [12]. By using multiple advanced tools, an APT attacker can often evade conventional defense measures of a system to slowly and covertly exfiltrate valuable data over a long period of time without being noticed. To defend against APT, numerous efforts have been made in cyber security community. For example, many APT detection techniques have been proposed [13]–[16], and different APT defense models have been suggested [17]–[24].

Generally speaking, the APT defense can be classified into two types: *proactive APT defense* and *reactive APT defense*. For the former, the defender uses different proactive defense techniques (e.g., moving target defense (MTD) [25]–[27]) to prevent APT. For the later, the defender takes actions to recover the target system only when an APT campaign has been identified.

A. CLOUD STORAGE DEFENSE AGAINST APT

With significant financial benefits, more and more modern organizations have moved to the cloud, which has changed the APT landscape. Instead of targeting one organization, the cloud could allow the APT to target a cloud storage system (CSS) and then gain access to different organizations' valuable data [5]. Therefore, it is crucial for a cloud defender to guarantee the security of the CSS in the presence of APT. Toward this direction, many defense mechanisms in the cloud have been reported in literature [28]–[30].

Game theory is the dominant formalism for studying the strategic interaction between rational decision-makers [31],

and has been applied to deal with different cybersecurity problems [32]–[34]. In [35], the interaction between an APT attacker and a system defender was modeled as a stealthy takeover game (the FlipIt game). In this work, both the APT attacker and the system defender pursue their objectives without having knowledge of the current state of the target system. Recently, the game framework proposed in [35] has been extensively applied to the defense against APT, particularly in the protection of cloud storage systems. Reference [36] studied the security of the cloud-based system by employing the FlipIt game. Reference [37] applied the FlipIt game to investigate the cloud storage defense against APT. In this work, the prospect theory (PT) was employed to characterize the subjectivity of the APT attacker and the cloud defender, and the target system was assumed to consist of a single storage server. In practice, most CSSs usually consist of multiple storage servers. Thus, [38] extended the game model in [37] to adapt to a more actual situation in which the CSS consists of multiple storage servers. In addition, the game model in [37] was also modified in [39] by employing the cumulative prospect theory (CPT), which is more readily extended to cases with much more outcomes than prospect theory. All the previous works build on an assumption that both the defense and attack resources are unlimited. In real world, the security resources (CPU, money, manpower, etc.) are typically limited. In this case, [40] investigated the cloud storage defense against APT with limited resources over multiple storage servers in the target CSS, by modeling the interaction between the cloud defender and the APT attacker as a Colonel Blotto game (CBG). However, in this work, the data size of each storage server was assumed to be identical and unchanged over time. In practice, the storage servers usually have different amount of data, and the data size also changes over time. Therefore, in [41], the CSS was extended to a dynamic one whose data size changes over time. In [42], evolution game was used to model the long-term behavior of APTs on a CSS. Evolution game was also employed to study the APT defense of fog computing in [43]. Reference [44] addressed the security of cloud-enabled Internet of Controlled Things (IoCT) through the contract theory. Reference [45] studied the interaction among three parties: the administrator of cloud, an attacker, and a device, with the investigation of the nature of a Gestalt Nash equilibrium.

All the above works fall into the proactive APT defense, that is the defender uses different proactive defense techniques to prevent APT.

B. OUR WORK

Different from the previously mentioned works, the present paper aims at solving the dynamic cloud storage recovery (DCSR) problem in the framework of the reactive APT defense. Nonetheless, our work is highly inspired by all of the previous works. In dealing with the DCSR problem, the ultimate goal of the cloud defender is to seek a time-continuous DR strategy to minimize his total loss. However, the total loss simultaneously relies on the DR strategy and the DA strategy, making it a much more complex problem. In the worst-case situation where the cloud defender assumes that the APT attacker has full knowledge of his expected loss, the DCSR problem is reduced to a game-theoretic problem of searching a Nash equilibrium, which we refer to as the DCSR* problem. A DCSR* game of the DCSR* problem is a differential game, because it relies on a differential dynamical system capturing the CSS's expected state evolution process, and the strategies of both parties are time-varying.

Dealing with the DCSR problem is of great importance to the cloud storage defense against APT. To our knowledge, this is the first time the DCSR problem is addressed in this fashion.

III. THE DCSR PROBLEM AND ITS MODELING

This section models the DCSR problem according to the following procedure. First, we introduce some basic terms and notations. Second, we formulate a DA strategy and a DR strategy. Thirdly, we establish the CSS's expected state evolution model under a combination of DA strategy and DR strategy. Next, we measure the APT attacker's net benefit and the cloud defender's total loss. Finally, we reduce the DCSR problem to a differential game-theoretic problem.

A. TERMS AND NOTATIONS

Table 1 summarizes all the notations used in the paper.

Consider a cloud storage system (CSS) shown in Fig. 1. Suppose an APT campaign on the CSS has been detected at time $t = 0$, and the cloud defender is going to continuously defense by recovering the compromised storage servers in the finite time horizon $[0, T]$. Meanwhile, the APT attacker constantly launches attacks to compromise the target CSS during the time horizon.

Suppose the CSS consists of N storage servers. In practice, the amount of data stored in the CSS is time-varying. Hence, we assume that at any time $t \in [0, T]$, storage server i stores data of size $D_i(t)$. Then $\mathbf{D}(t) = (D_1(t), \cdots, D_N(t))$ denotes the CSS's *data size vector* at time t, and $B(t) = \sum_{i=1}^{N} D_i(t)$ denotes the total size of data stored in the CSS at time t. For simplicity, in this work we assume that $D_i(t) \in [0, 1]$, where 0 means there is no any data stored in the storage server, while 1 means the data size reaches the maximum capacity of the storage server. Then the feasible set of $\mathbf{D}(t)$ is

$$\mathbb{D} = \left\{ \mathbf{D}(t) : \mathbf{D}(t) \in \prod_{i=1}^{N} [0, 1], t \in [0, T] \right\}. \tag{1}$$

In this paper, we will generate the data size vector by performing an algorithm which is referred to as the DSV algorithm shown in Algorithm 1, where DSV stands for data size vector.

We assume that at any time $t \in [0, T]$, each and every storage server of the CSS is in one of two possible states: *uncompromised* and *compromised*. An uncompromised storage server is under control of the cloud defender and the data stored in it is safe, whereas a compromised storage server is

TABLE 1. Summary of symbols and notations.

Notation	Definition
N	Number of storage servers in the CSS
$D_i(t)$	Data size of storage server i at time t
$\mathbf{D}(t)$	Data size vector of CSS at time t
$B(t)$	Total size of stored data at time t
$\mathbb{D}$	Feasible set of data size vector
$S_i(t)$	Storage server i's state at time t
$\mathbf{S}(t)$	CSS's state at time t
$C_i(t)$	Storage server i's expected state at time t
$\mathbf{E}(t)$	CSS's expected state at time t
$\alpha_i(t)/\gamma_i(t)$	Attack / Recovery rate on storage server i at time t
$\mathbf{x}/\mathbf{y}$	DA / DR strategy
$\underline{\alpha_i}/\overline{\alpha_i}$	Lower / Upper bound of attack rate on storage server i
$\underline{\alpha}/\overline{\alpha}$	Attack lower / upper bound vector
$\underline{\gamma_i}/\overline{\gamma_i}$	Lower / Upper bound of recovery rate on storage server i
$\underline{\gamma}/\overline{\gamma}$	Recovery lower / upper bound vector
$\mathbb{X}/\mathbb{Y}$	Admissible set of DA / DR strategy
w	Loss coefficient
ϕ_i/ψ_i	Attack / Recovery cost function of storage server i
ϕ/ψ	Attack / Recovery cost function vector

Algorithm 1 DSV

Input: positive real number N, T.
Output: a data size vector $\mathbf{D}$.

```
for i = 1 to N do
    for t = 0 to T do
        choose a random value η ∈ [0, 1];
        D_i(t) ← η;
    end for
end for
return D.
```

under control of the APT attacker and the data stored in it can be stolen by the attacker.

Let $S_i(t) = 0$ and 1 denote that storage server i is uncompromised and compromised at time t, respectively. Then the vector

$$\mathbf{S}(t) = (S_1(t), \cdots, S_N(t)) \tag{2}$$

represents the CSS's state at time t.

Let $C_i(t)$ denote the probability of storage server i being compromised at time t, i.e., $C_i(t) = \Pr\{S_i(t) = 1\}$. Then we get that the probability of storage server i being uncompromised at time t is $1 - C_i(t)$. Thus, the vector

$$\mathbf{E}(t) = (C_1(t), \cdots, C_N(t)) \tag{3}$$

represents the CSS's expected state at time t.

In practice, the attacker can determine whether the attack is detected and stopped by the cloud defender by observing the size of the stolen data. This implies that the APT attacker is aware of the CSS's state at any time. For the cloud defender, the initial expected state of the CSS can be estimated relatively accurately by employing some proven APT detection techniques. Hence, in what follows we assume $\mathbf{E}(0)$ is known to the cloud defender.

B. THE DA STRATEGY AND DR STRATEGY

Let $\alpha_i(t)$ denote the rate at which the APT attack makes $S_i(t)$ to go up at time t. We refer to $\alpha_i(t)$ as the *attack rate* on storage server i at time t. Then the following N-dimensional vector-valued function

$$\mathbf{x}(t) = (\alpha_1(t), \cdots, \alpha_N(t)), \quad t \in [0, T], \tag{4}$$

is referred to as a *dynamic attack (DA) strategy* and is controllable by the APT attacker.

Let $\gamma_i(t)$ denote the rate at which the cloud defender's recovery makes $S_i(t)$ to go down at time t. We refer to $\gamma_i(t)$ as the *recovery rate* on storage server i at time t. Then the following N-dimensional vector-valued function

$$\mathbf{y}(t) = (\gamma_1(t), \cdots, \gamma_N(t)), \quad t \in [0, T], \tag{5}$$

is referred to as a *dynamic recovery (DR) strategy* and is controllable by the cloud defender.

According to stochastic process theory, the diagram of state transition of the storage server i is as shown in Fig. 3. Now we impose a few restrictions on the DA strategy and DR strategy as follows.

First, we assume that $\mathbf{x}$ and $\mathbf{y}$ are both piecewise continuous. Obviously, the piecewise continuous DA strategy and DR strategy are easy to be implemented. Let $\Gamma^N[0, T]$ denote the set of all the piecewise continuous N-dimensional

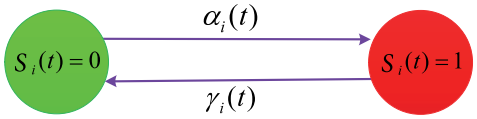

FIGURE 3. The diagram of state transition of the storage server *i* under the combination of the attack rate and recovery rate.

vector-valued functions defined on the interval $[0, T]$. Then, $\mathbf{x}, \mathbf{y} \in \Gamma^N[0, T]$.

Second, let $\underline{\alpha_i} > 0$ and $\overline{\alpha_i} < \infty$ denote the lower bound and upper bound of attack rate on the storage server i at any time, respectively. Let $\underline{\gamma_i} > 0$ and $\overline{\gamma_i} < \infty$ denote the lower bound and upper bound of recovery rate on the storage server i at any time, respectively. Then, for $t \in [0, T]$, $1 \le i \le N$, we have $\underline{\alpha_i} \le \alpha_i(t) \le \overline{\alpha_i}$ and $\underline{\gamma_i} \le \gamma_i(t) \le \overline{\gamma_i}$. Hence, the admissible set of the DA strategy is

$$\mathbb{X} = \left\{ \mathbf{x} \in \Gamma^N[0, T] \mid \mathbf{x}(t) \in \prod_{i=1}^{N} \left[\underline{\alpha_i}, \overline{\alpha_i}\right], t \in [0, T] \right\}, \quad (6)$$

and the admissible set of the DR strategy is

$$\mathbb{Y} = \left\{ \mathbf{y} \in \Gamma^N[0, T] \mid \mathbf{y}(t) \in \prod_{i=1}^{N} \left[\underline{\gamma_i}, \overline{\gamma_i}\right], t \in [0, T] \right\}. \quad (7)$$

We refer to $\underline{\alpha} = \left(\underline{\alpha_1}, \cdots, \underline{\alpha_N}\right)$ as the *attack lower bound vector*, $\overline{\alpha} = (\overline{\alpha_1}, \cdots, \overline{\alpha_N})$ as the *attack upper bound vector*, $\underline{\gamma} = \left(\underline{\gamma_1}, \cdots, \underline{\gamma_N}\right)$ as the *recovery lower bound vector*, and $\overline{\gamma} = (\overline{\gamma_1}, \cdots, \overline{\gamma_N})$ as the *recovery upper bound vector*.

C. THE CSS'S EXPECTED STATE EVOLUTION MODEL

To model the DCSR problem, we need to measure the cloud defender's direct loss, which is closely related to the CSS's expected state evolution process.

Let $\mathbb{E}[\cdot]$ denote the mathematical expectation of a random variable. By definitions of $\alpha_i(t)$ and $\gamma_i(t)$, we get that the rate at which the state of storage server i goes up at time t is $\alpha_i(t)S_i(t) - \gamma_i(t)S_i(t)$. Hence, the average rate at which the expected state of storage server i goes up at time t is $\mathbb{E}[\alpha_i(t)S_i(t) - \gamma_i(t)S_i(t)] = \alpha_i(t)[1 - C_i(t)] - \gamma_i(t)C_i(t)$. Therefore, the CSS's expected state evolves over time according to the following differential dynamical system:

$$\frac{dC_i(t)}{dt} = \alpha_i(t)[1 - C_i(t)] - \gamma_i(t)C_i(t), \quad t \in [0, T], 1 \le i \le N. \quad (8)$$

D. THE APT ATTACKER'S NET BENEFIT AND THE CLOUD DEFENDER'S TOTAL LOSS

In order to model the DCSR problem, we need to measure the APT attacker's net benefit and the cloud defender's total loss. Given a strategy profile $(\mathbf{x}, \mathbf{y})$, the cloud defender's total loss consists of the cloud defender's expected loss incurred by data leakage and the expected cost coming from the implementation of the DR strategy $\mathbf{y}$, while the attacker's net benefit takes the difference between the cloud defender's expected loss and the expected cost coming from the implementation of the DA strategy $\mathbf{x}$. Now we are going to formally measure these quantities.

To measure the cloud defender's expected loss, we introduce the first assumption as follows.

(A_1) The average amount of losses per unit time resulted from the leakage of per unit data volume is $w > 0$. We refer to w as the *loss coefficient*.

According to this assumption, in the infinitesimal time horizon $[t, t + dt)$, the cloud defender's average loss owing to the storage server i is $wD_i(t)dt$ or 0 according as $S_i(t) = 1$ or 0. As a result, the cloud defender's expected loss in the infinitesimal time horizon $[t, t + dt)$ is $wD_i(t)dt \times \Pr\{S_i(t) = 1\} + 0 \times \Pr\{S_i(t) = 0\} = wD_i(t)C_i(t)dt$. Therefore, the cloud defender's expected loss during the time horizon $[0, T]$ is

$$EL(\mathbf{x}, \mathbf{y}) = w \int_0^T \sum_{i=1}^{N} D_i(t)C_i(t)dt. \quad (9)$$

To evaluate the expected cost coming from the implementation of the DA strategy $\mathbf{x}$, we introduce the second assumption as follows.

(A_2) The cost per unit time for attacking the uncompromised storage server i at the rate of α is $\phi_i(\alpha)$, where ϕ_i is referred to as the *attack cost function* of storage server i and is strictly increasing and $\phi_i(0) = 0$. We refer to $\phi = (\phi_1, \cdots, \phi_N)$ as the *attack cost function vector*.

Similarly, according to this assumption, the expected cost coming from the implementation of the DA strategy $\mathbf{x}$ during the time horizon $[0, T]$ is

$$EC_A(\mathbf{x}, \mathbf{y}) = \int_0^T \sum_{i=1}^{N} \phi_i(\alpha_i(t))(1 - C_i(t))dt. \quad (10)$$

To quantify the expected cost coming from the implementation of the DR strategy $\mathbf{y}$, we introduce the third assumption as follows.

(A_3) The cost per unit time for recovering the compromised storage server i at the rate of γ is $\psi_i(\gamma)$, where ψ_i is referred to as the *recovery cost function* of storage server i and is strictly increasing and $\psi_i(0) = 0$. We refer to $\psi = (\psi_1, \cdots, \psi_N)$ as the *recovery cost function vector*.

Similarly, according to this assumption, the expected cost coming from the implementation of the DR strategy $\mathbf{y}$ during the time horizon $[0, T]$ is

$$EC_D(\mathbf{x}, \mathbf{y}) = \int_0^T \sum_{i=1}^{N} \psi_i(\gamma_i(t))C_i(t)dt. \quad (11)$$

Hence, the APT attacker's net benefit is measured by

$$\begin{aligned} U_A(\mathbf{x}, \mathbf{y}) &= EL(\mathbf{x}, \mathbf{y}) - EC_A(\mathbf{x}, \mathbf{y}) \\ &= \int_0^T \sum_{i=1}^{N} \{wD_i(t)C_i(t) - \phi_i(\alpha_i(t))[1 - C_i(t)]\} dt, \end{aligned} \quad (12)$$

and the cloud defender's total loss is measured by

$$U_D(\mathbf{x}, \mathbf{y}) = EL(\mathbf{x}, \mathbf{y}) + EC_D(\mathbf{x}, \mathbf{y}) = \int_0^T \sum_{i=1}^{N} [wD_i(t) + \psi_i(\gamma_i(t))] C_i(t) dt. \quad (13)$$

E. THE GAME MODEL FOR THE DCSR PROBLEM

According to the previous discussions, the DCSR problem is reduced to a game-theoretic problem, which we refer to as the *DCSR* problem*, in which the goal is to seek a DR strategy $\mathbf{y} \in \mathbb{Y}$ so that the $U_D(\mathbf{x}, \mathbf{y})$ is mitigated, provided the DA strategy $\mathbf{x} \in \mathbb{X}$ is unknown to the cloud defender. Specifically, the DCSR* problem consists of all DCSR* games defined as follows.

Definition 1: A DCSR game includes three components*:

C$_1$. *The set of players, P = {APT attacker, Cloud defender}.*

C$_2$. *The admissible set of the DA strategy, $\mathbb{X}$, and the admissible set of the DR strategy, $\mathbb{Y}$.*

C$_3$. *The APT attacker's net benefit, $U_A(\mathbf{x}, \mathbf{y})$, and the cloud defender's total loss, $U_D(\mathbf{x}, \mathbf{y})$, where $(\mathbf{x}, \mathbf{y}) \in \mathbb{X} \times \mathbb{Y}$.*

Every DCSR* game can be represented by an 11-tuple

$$G = \left(N, \mathbf{D}, \underline{\alpha}, \overline{\alpha}, \underline{\gamma}, \overline{\gamma}, \phi, \psi, \mathbf{E}_0, w, T\right). \quad (14)$$

Now let us formally give the definition of the open-loop Nash equilibrium of a DCSR* game.

Definition 2: Given a DCSR game. The strategy profile $(\mathbf{x}^*, \mathbf{y}^*) \in \mathbb{X} \times \mathbb{Y}$ is an* open-loop Nash equilibrium *for the game if*

$$U_A(\mathbf{x}^*, \mathbf{y}^*) \geq U_A(\mathbf{x}, \mathbf{y}^*), \quad \forall \mathbf{x} \in \mathbb{X}, \quad (15)$$

and

$$U_D(\mathbf{x}^*, \mathbf{y}^*) \leq U_D(\mathbf{x}^*, \mathbf{y}), \quad \forall \mathbf{y} \in \mathbb{Y}. \quad (16)$$

For convenience, in this paper, we interchangeably use the two terms *open-loop Nash equilibrium* and *Nash equilibrium*. Given the Nash equilibrium of a DCSR* game, $(\mathbf{x}^*, \mathbf{y}^*)$. According to the nature of Nash equilibrium, both the cloud defender and the APT attacker cannot achieve a better goal by unilaterally deviating from their respective strategies in the equilibrium. Hence, at least from the worst-case perspective, it is appropriate for the cloud defender to adopt the DR strategy $\mathbf{y}^*$ in the equilibrium.

Although many efforts have been taken in some particular classes of differential games such as trilinear games [46] and state-redundant games [47], due to the complexity of the DCSR* problem, we fail to show the existence and uniqueness of the open-loop Nash equilibrium for a DCSR* game. As a result, a DCSR* game may possibly admit no or more than one Nash equilibrium. Nonetheless, in the following section, we will try our best to deal with the DCSR* problem.

IV. DEALING WITH THE DCSR* PROBLEM

In the previous section, we modeled the DCSR problem as the DCSR* problem. In this section, we devote ourselves to dealing with the DCSR* problem. First, we derive a necessary condition for Nash equilibrium of the DCSR* problem and thereby introduce the concept of competitive strategy profile. Second, we investigate the structural properties of the competitive strategy profile. Finally, we present some numerical examples of the competitive strategy profile.

A. A NECESSARY CONDITION

First, we present a necessary condition for the Nash equilibrium. According to differential game [11], the Hamiltonian for the APT attacker is

$$H_A(\mathbf{E}, \mathbf{x}, \mathbf{y}, \lambda) = \sum_{i=1}^{N} [wD_iC_i - \phi_i(\alpha_i)(1 - C_i)] + \sum_{i=1}^{N} \lambda_i [\alpha_i(1 - C_i) - \gamma_i C_i], \quad (17)$$

and the Hamiltonian for the cloud defender is

$$H_D(\mathbf{E}, \mathbf{x}, \mathbf{y}, \mu) = \sum_{i=1}^{N} [wD_i + \psi_i(\gamma_i)] C_i + \sum_{i=1}^{N} \mu_i [\alpha_i(1 - C_i) - \gamma_i C_i], \quad (18)$$

where $\lambda = (\lambda_1, \cdots, \lambda_N)^T$ and $\mu = (\mu_1, \cdots, \mu_N)^T$ are their respective adjoint functions.

The following theorem presents the necessary condition for the Nash equilibrium of a DCSR* game.

Theorem 1: Suppose $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium of a DCSR game, and $\mathbf{E}$ is the solution to the expected state evolution model (8). Then, there exist λ and μ with $\lambda(T) = \mu(T) = \mathbf{0}$ such that*

$$\begin{cases} \dfrac{d\lambda_i(t)}{dt} = -wD_i(t) - \phi_i(\alpha_i(t)) + \lambda_i(t)[\alpha_i(t) + \gamma_i(t)], \\ \dfrac{d\mu_i(t)}{dt} = -wD_i(t) - \psi_i(\gamma_i(t)) + \mu_i(t)[\alpha_i(t) + \gamma_i(t)], \\ t \in [0, T], 1 \leq i \leq N. \end{cases} \quad (19)$$

Moreover,

$$\begin{cases} \alpha_i(t) \in \arg\max_{\alpha \in [\underline{\alpha_i}, \overline{\alpha_i}]} [1 - C_i(t)][\lambda_i(t)\alpha - \phi_i(\alpha)], \\ \gamma_i(t) \in \arg\max_{\gamma \in [\underline{\gamma_i}, \overline{\gamma_i}]} C_i(t)[\psi_i(\gamma) - \mu_i(t)\gamma], \\ 1 \leq i \leq N, t \in [0, T], \end{cases} \quad (20)$$

Proof: According to the Pontryagin Maximum/Minimum Principle [48], there exist λ and μ such that

$$\frac{d\lambda_i(t)}{dt} = -\frac{\partial H_A(\mathbf{E}(t), \mathbf{x}(t), \mathbf{y}(t), \lambda(t))}{\partial C_i}, \quad t \in [0, T], 1 \leq i \leq N, \quad (21)$$

and

$$\frac{d\mu_i(t)}{dt} = -\frac{\partial H_D(\mathbf{E}(t), \mathbf{x}(t), \mathbf{y}(t), \mu(t))}{\partial C_i}, \quad t \in [0, T], 1 \leq i \leq N. \quad (22)$$

Thus, we can get Eqs. (19) by direct calculations, and $\lambda(T) = \mu(T) = \mathbf{0}$ holds. Besides, for $t \in [0, T]$, according to the optimality conditions

$$\mathbf{x}(t) \in \arg\max_{\hat{\mathbf{x}} \in \mathbb{X}} H_A(\mathbf{E}(t), \hat{\mathbf{x}}(t), \mathbf{y}(t), \lambda(t)), \tag{23}$$

and

$$\mathbf{y}(t) \in \arg\min_{\hat{\mathbf{y}} \in \mathbb{Y}} H_D(\mathbf{E}(t), \mathbf{x}(t), \hat{\mathbf{y}}(t), \mu(t)), \tag{24}$$

we can get Eq. (20). ■

In what follows, we refer to the system of Eqs. (8), (19), and (20) with $\mathbf{E}(0) = \mathbf{E}_0$ and $\lambda(T) = \mu(T) = \mathbf{0}$ as the *necessity system* for the DCSR* problem, and each strategy profile in solutions to the necessity system as a *competitive strategy profile* of the DCSR* problem.

We should note that the necessity system is only a necessary condition for Nash equilibrium; a Nash equilibrium must be a competitive strategy profile, but the converse may not hold. Nonetheless, finding a competitive strategy profile and evaluating its performance offer a feasible approach to dealing with the DCSR* problem. Additionally, it is seen from Eq. (20) that a DCSR* game may admit more than one competitive strategy profile. For clarity, whenever solving a necessity system, we always break each tie appearing in Eq. (20) by letting the attack rate or the recovery rate takes on the lower bound.

When it comes to numerical calculation of the necessity system, we describe an algorithm shown in Algorithm 2, which is based on the Forward-Backward Sweep Method [49]. We refer to the algorithm as CSP algorithm, where CSP stands for competitive strategy profile. In this algorithm, $\|\mathbf{h}\| = \max_{0 \le t \le T} \sum_{i=1}^{N} |h_i(t)|$, and we set $\epsilon = 10^{-6}$, $K = 10^3$ for all of the experiments.

Algorithm 2 CSP

Input:
DCSR* game $G = \left(N, \mathbf{D}, \underline{\alpha}, \overline{\alpha}, \underline{\gamma}, \overline{\gamma}, \phi, \psi, \mathbf{E}(0), w, T\right)$, convergence error ϵ, maximum number of iterations K.
Output: a strategy profile $(\mathbf{x}, \mathbf{y})$.

// initialize
for $i = 1$ to N **do**
  for $t = 0$ to T **do**
    $\alpha_i(t) \leftarrow \overline{\alpha_i}$, $\gamma_i(t) \leftarrow \overline{\gamma_i}$;
  end for
end for
// iteration
$k \leftarrow 0$
repeat
  $k \leftarrow k + 1$
  for $t = 0$ to T **do**
    compute $\mathbf{E}^k(t)$ based on Eq. (8);
  end for
  for $t = T$ to 0 **do**
    compute $\lambda^k(t)$ and $\mu^k(t)$ based on Eq. (19) with $\lambda(T) = \mu(T) = \mathbf{0}$;
    compute $\mathbf{x}^k(t)$ and $\mathbf{y}^k(t)$ based on Eq. (20);
  end for
until $\|\mathbf{x}^k - \mathbf{x}^{k-1}\| + \|\mathbf{y}^k - \mathbf{y}^{k-1}\| < \epsilon$ or $k \ge K$
return $\left(\mathbf{x}^k, \mathbf{y}^k\right)$.

B. THE STRUCTURAL PROPERTIES OF THE COMPETITIVE STRATEGY PROFILE

Now, let us study the structural properties of the competitive strategy profile. Let $(\mathbf{x}, \mathbf{y})$, $\mathbf{E}$, and λ and μ be the competitive strategy profile, the solution to the associated expected state evolution model, and the associated adjoints, respectively.

First, let us examine the structural properties of $\mathbf{x}$. Let

$$\theta_i = \frac{\phi_i(\overline{\alpha_i}) - \phi_i(\underline{\alpha_i})}{\overline{\alpha_i} - \underline{\alpha_i}}, \quad 1 \le i \le N. \tag{25}$$

The first and second results in this subsection are given below.

Theorem 2: Suppose ϕ_i is concave. For $t \in [0, T]$, we have

$$\alpha_i(t) = \begin{cases} \underline{\alpha_i} & \text{if } C_i(t) = 1 \text{ or } \lambda_i(t) \le \theta_i, \\ \overline{\alpha_i} & \text{if } C_i(t) < 1 \text{ and } \lambda_i(t) > \theta_i. \end{cases} \tag{26}$$

Proof: If $C_i(t) = 1$, it follows from Eq. (20) and the definition of the competitive strategy profile that $\alpha_i(t) = \underline{\alpha_i}$. If $C_i(t) < 1$, then define the function g_i as follows.

$$g_i(\alpha; t) = \lambda_i(t)\alpha - \phi_i(\alpha), \quad \alpha \in [\underline{\alpha_i}, \overline{\alpha_i}], \tag{27}$$

As ϕ_i is concave, g_i is convex. So, by comparing $g_i(\underline{\alpha_i}; t)$ and $g_i(\overline{\alpha_i}; t)$, we deduce Eq. (26). ■

Theorem 3: Suppose ϕ_i is strictly convex and differentiable. For $t \in [0, T]$, we have

$$\alpha_i(t) = \begin{cases} \underline{\alpha_i} & \text{if } C_i(t) = 1 \text{ or } \lambda_i(t) < \phi_i'(\underline{\alpha_i}), \\ \overline{\alpha_i} & \text{if } C_i(t) < 1 \text{ and } \lambda_i(t) > \phi_i'(\overline{\alpha_i}), \\ \left[\phi_i'\right]^{-1}(\lambda_i(t)) & \text{otherwise}. \end{cases} \tag{28}$$

Proof: If $C_i(t) = 1$, it follows from Eq. (20) and the definition of the competitive strategy profile that $\alpha_i(t) = \underline{\alpha_i}$. If $C_i(t) < 1$, then as ϕ_i is strictly convex, it follows that the function g_i defined by Eq. (27) is strictly concave. Hence, there are three cases.

(a) g_i is strictly decreasing, which implies $\lambda_i(t) < \phi_i'(\underline{\alpha_i})$. Then $\alpha_i(t) = \underline{\alpha_i}$.
(b) g_i is strictly increasing, which implies $\lambda_i(t) > \phi_i'(\overline{\alpha_i})$. Then $\alpha_i(t) = \overline{\alpha_i}$.
(c) g_i is first increasing then decreasing. Then $\frac{dg_i(\alpha_i(t);t)}{d\alpha} = 0$, which implies $\alpha_i(t) = \left[\phi_i'\right]^{-1}(\lambda_i(t))$.

■

Next, let us inspect the structural properties of $\mathbf{y}$. Let

$$\eta_i = \frac{\psi(\overline{\gamma_i}) - \psi(\underline{\gamma_i})}{\overline{\gamma_i} - \underline{\gamma_i}}. \tag{29}$$

The third and fourth results in this subsection are given below.

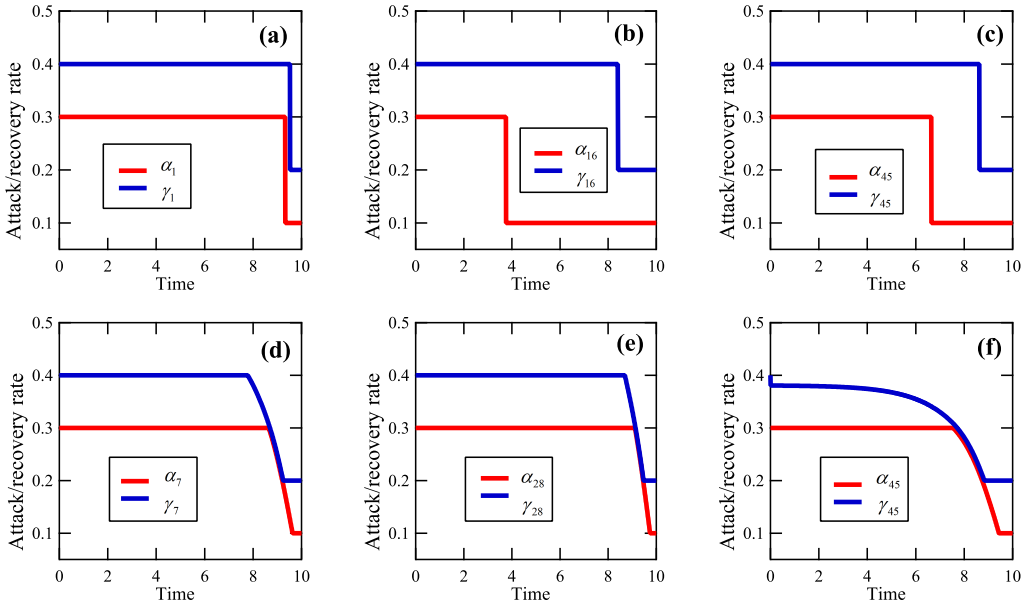

FIGURE 4. Three attack rate functions and three recovery rate functions in each of the two competitive strategy profiles given in Example 1, where (a)–(c) are for the DCSR* game G^1, (d)–(f) are for the DCSR* game G^2.

Theorem 4: Suppose ψ_i is concave. For $t \in [0, T]$, we have

$$\gamma_i(t) = \begin{cases} \underline{\gamma_i} & \text{if } C_i(t) = 0 \text{ or } \mu_i(t) \le \eta_i, \\ \overline{\gamma_i} & \text{if } C_i(t) > 0 \text{ and } \mu_i(t) > \eta_i. \end{cases} \tag{30}$$

Theorem 5: Suppose ψ_i is strictly convex and differentiable. For $t \in [0, T]$, we have

$$\gamma_i(t) = \begin{cases} \underline{\gamma_i} & \text{if } C_i(t) = 0 \text{ or } \mu_i(t) < \psi_i'(\underline{\gamma_i}), \\ \overline{\gamma_i} & \text{if } C_i(t) > 0 \text{ and } \mu_i(t) > \psi_i'(\overline{\gamma_i}), \\ \left[\psi_i'\right]^{-1}(\mu_i(t)) & \text{otherwise}. \end{cases} \tag{31}$$

The proofs of the two theorems are similar to those of Theorems 2 and 3 and hence are omitted.

Theorems 2–5 partly illuminate the structural properties of the competitive strategy profile.

C. EXPERIMENT SETTINGS

All the experiments in this paper are conducted on a PC with Intel Xeon E3-1231 CPU and 8GB RAM. Furthermore, we need to obtain the competitive strategy profiles of a set of DCSR* games by solving the corresponding necessity systems. For this purpose, we will generate these DCSR* games by setting the parameters in detail as follows.

- Set N. Let $N^1 = 50$, $N^2 = 100$. In all the experiments, $N \in \{N^1, N^2\}$.
- Set $\mathbf{D}$. Let $\mathbf{D}^*$ be the data size vector generated by invoking the DSV algorithm. In all the experiments, $\mathbf{D} = \mathbf{D}^*$.
- Set $\underline{\alpha}$, $\overline{\alpha}$, $\underline{\gamma}$, and $\overline{\gamma}$. For $m \in \{1, 2, \cdots, 5\}$, let $\alpha^m = (0.m, \cdots, 0.m)$ and $\gamma^m = (0.m, \cdots, 0.m)$ be the vectors with N components, where $N \in \{N^1, N^2\}$. In all the experiments, $(\underline{\alpha}, \overline{\alpha}) \in \left\{\left(\alpha^1, \alpha^3\right), \left(\alpha^2, \alpha^4\right)\right\}$, $\left(\underline{\gamma}, \overline{\gamma}\right) \in \left\{\left(\gamma^2, \gamma^4\right), \left(\gamma^3, \gamma^5\right)\right\}$.
- Set ϕ and ψ. Let $\phi^{\frac{1}{2}}(\alpha) = \left(\sqrt{\alpha}, \cdots, \sqrt{\alpha}\right)$, $\phi^2(\alpha) = \left(\alpha^2, \cdots, \alpha^2\right)$, $\psi^{\frac{1}{2}}(\gamma) = \left(\sqrt{\gamma}, \cdots, \sqrt{\gamma}\right)$, and $\psi^2(\gamma) = \left(\gamma^2, \cdots, \gamma^2\right)$ be the vectors with N components, where $N \in \{N^1, N^2\}$. In all the experiments, $\phi \in \left\{\phi^{\frac{1}{2}}, \phi^2\right\}$, $\psi \in \left\{\psi^{\frac{1}{2}}, \psi^2\right\}$.
- Set $\mathbf{E}_0$. Let $\mathbf{E}^* = (0.2, \cdots, 0.2)$ with N 0.2 components, where $N \in \{N^1, N^2\}$. In all the experiments, $\mathbf{E}_0 = \mathbf{E}^*$.
- Set w. For $m \in \{4, 5\}$, let $w^m = m$. In all the experiments, $w \in \{w^4, w^5\}$.
- Set T. Let $T^* = 10$. In all the experiments, $T = T^*$.

D. EXAMPLES OF THE COMPETITIVE STRATEGY PROFILE

In this subsection, we solve some DCSR* games to obtain their competitive strategy profiles.

Example 1: (a) Consider the DCSR game*

$$G^1 = \left(N^1, \mathbf{D}^*, \alpha^1, \alpha^3, \gamma^2, \gamma^4, \phi^{\frac{1}{2}}, \psi^{\frac{1}{2}}, \mathbf{E}^*, w^4, T^*\right).$$

By solving the necessity system, we get the competitive strategy profile. Fig. 4(a)–(c) plots three attack rate functions and three recovery rate functions in the strategy profile.

(b) Consider the DCSR game*

$$G^2 = \left(N^1, \mathbf{D}^*, \alpha^1, \alpha^3, \gamma^2, \gamma^4, \phi^2, \psi^2, \mathbf{E}^*, w^4, T^*\right).$$

By solving the necessity system, we get the competitive strategy profile. Fig. 4(d)–(f) plots three attack rate functions and three recovery rate functions in the strategy profile.

Example 2: (a) Consider the DCSR game*

$$G^3 = \left(N^2, \mathbf{D}^*, \alpha^2, \alpha^4, \gamma^3, \gamma^5, \phi^{\frac{1}{2}}, \psi^{\frac{1}{2}}, \mathbf{E}^*, w^5, T^*\right).$$

By solving the necessity system, we get the competitive strategy profile. Fig. 5(a)–(c) plots three attack rate functions and

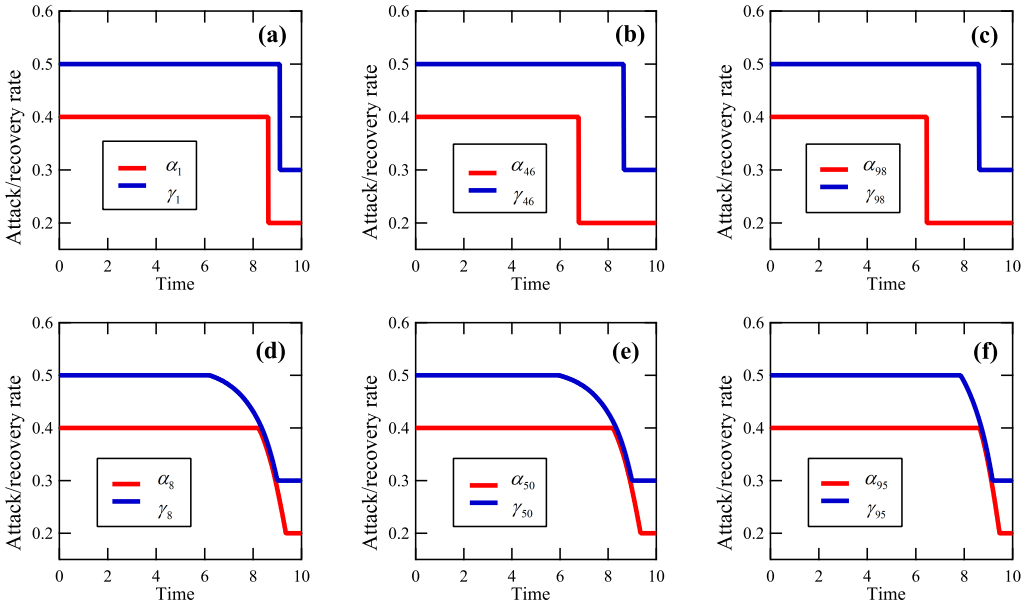

FIGURE 5. Three attack rate functions and three recovery rate functions in each of the two competitive strategy profiles given in Example 2, where (a)–(c) are for the DCSR* game G^3, (d)–(f) are for the DCSR* game G^4.

three recovery rate functions in the strategy profile.

(b) Consider the DCSR game*

$$G^4 = \left(N^2, \mathbf{D}^*, \alpha^2, \alpha^4, \gamma^3, \gamma^5, \phi^2, \psi^2, \mathbf{E}^*, w^5, T^*\right).$$

By solving the necessity system, we get the competitive strategy profile. Fig. 5(d)–(f) plots three attack rate functions and three recovery rate functions in the strategy profile.

From the above examples we conclude that for any DCSR* game, each rate function in the competitive strategy profile first stays at a high value (mostly, the upper bound), and then falls sharply or gradually to the lower bound, and finally stays at the lower bound.

On the one hand, the above results are completely in accordance with theorems 2–5. Specifically, when the cost function of a rate is concave, then the rate will only take on its lower bound or upper bound (see Fig. 4(a)–(c) and Fig. 5(a)–(c)). In contrast, when the cost function of a rate is strictly convex and differentiable, then the rate may take on some values in between its lower bound and its upper bound (see Fig. 4(d)–(f) and Fig. 5(d)–(f)). On the other hand, in the course of the APT campaign, the APT attacker first selects high attack rates to gain as much benefit as possible from data theft at the beginning. However, as time goes by, the cost will overtake the benefit. Hence, the APT attacker reduces the cost to achieve a better tradeoff. Similarly, the cloud defender needs to make responses in a timely manner by selecting high recovery rates at the beginning of the APT campaign, and then reduce the cost to achieve a better tradeoff as well.

V. PERFORMANCE OF THE COMPETITIVE STRATEGY PROFILE

In the previous section, we derived a necessity system for seeking the competitive strategy profile of a DCSR* game.

Algorithm 3 RAS

Input:
DCSR* game $G = \left(N, \mathbf{D}, \underline{\alpha}, \overline{\alpha}, \underline{\gamma}, \overline{\gamma}, \phi, \psi, \mathbf{E}(0), w, T\right)$.
Output: a DA strategy $\mathbf{x}$.

```
for i = 1 to N do
    for t = 0 to T do
        choose a random value δ ∈ [α_i, ᾱ_i];
        α_i(t) ← δ;
    end for
end for
return x.
```

In this section, we are going to evaluate the performance of the competitive strategy profile through computer experiments. Let $(\mathbf{x}^*, \mathbf{y}^*)$ be the competitive strategy profile of a DCSR* game. For our purpose, we generate a set of 100 feasible DA strategies by executing algorithm 3 (the RAS algorithm, where RAS stands for *random attack strategy*) 100 times, denoted $\mathbb{X}_{\text{rand}} = \{\mathbf{x}_1, \cdots, \mathbf{x}_{100}\}$, and a set of 100 feasible DR strategies by executing algorithm 4 (the RRS algorithm, where RRS stands for *random recovery strategy*) 100 times, denoted $\mathbb{Y}_{\text{rand}} = \{\mathbf{y}_1, \cdots, \mathbf{y}_{100}\}$.

Experiment 1: Consider the two DCSR games given in Example 1. Fig. 6 shows the comparative results. It can be seen from these results that for either of the two games, we have*

(1) $U_A(\mathbf{x}^*, \mathbf{y}^*) > U_A(\mathbf{x}, \mathbf{y}^*), \quad \mathbf{x} \in \mathbb{X}_{rand}$,
(2) $U_D(\mathbf{x}^*, \mathbf{y}^*) < U_D(\mathbf{x}^*, \mathbf{y}), \quad \mathbf{y} \in \mathbb{Y}_{rand}$,
(3) $U_D(\mathbf{x}^*, \mathbf{y}^*) > U_D(\mathbf{x}, \mathbf{y}^*), \quad \mathbf{x} \in \mathbb{X}_{rand}$.

Experiment 2: Consider the two DCSR games given in Example 2. Fig. 7 shows the comparative results. Again, it can be seen from these results that for either of the two games,*

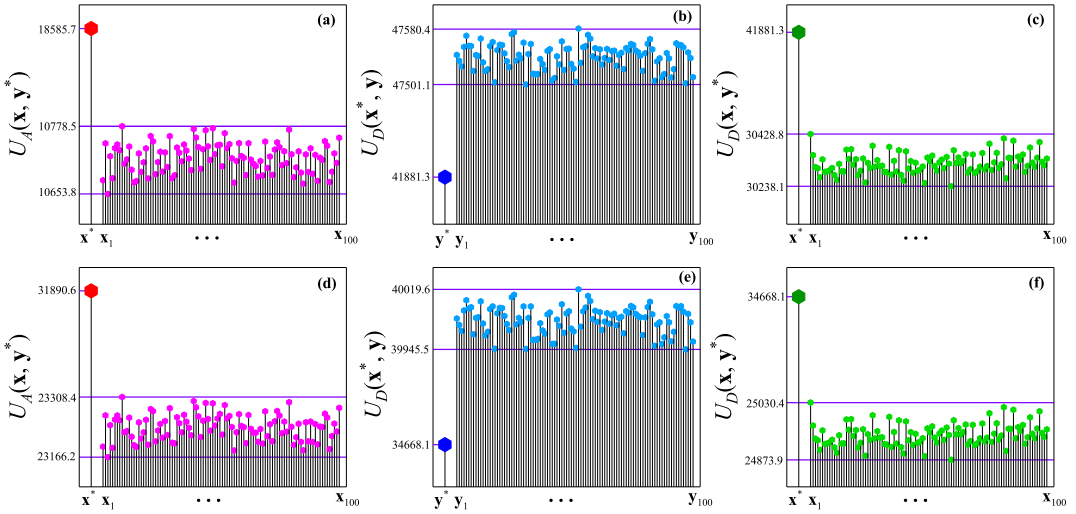

FIGURE 6. The performances of the competitive strategy profiles in Experiment 1, where (a)-(c) are for the DCSR* game G^1, (d)-(f) are for the DCSR* game G^2.

Algorithm 4 RRS

Input:
DCSR* game $G = \left(N, \mathbf{D}, \underline{\alpha}, \overline{\alpha}, \underline{\gamma}, \overline{\gamma}, \phi, \psi, \mathbf{E}(0), w, T\right)$.
Output: a DR strategy $\mathbf{y}$.

```
for i = 1 to N do
    for t = 0 to T do
        choose a random value δ ∈ [γ_i, γ̄_i];
        γ_i(t) ← δ;
    end for
end for
return y.
```

we have

(1) $U_A(\mathbf{x}^*, \mathbf{y}^*) > U_A(\mathbf{x}, \mathbf{y}^*), \quad \mathbf{x} \in \mathbb{X}_{rand}$,
(2) $U_D(\mathbf{x}^*, \mathbf{y}^*) < U_D(\mathbf{x}^*, \mathbf{y}), \quad \mathbf{y} \in \mathbb{Y}_{rand}$,
(3) $U_D(\mathbf{x}^*, \mathbf{y}^*) > U_D(\mathbf{x}, \mathbf{y}^*), \quad \mathbf{x} \in \mathbb{X}_{rand}$.

Now let us take a few minutes to discuss the above experiment results in detail as follows.

First, Experiment 1(1)-(2) and Experiment 2(1)-(2) are in accordance with the definition of Nash equilibrium (Definition 2), which means that for a given DCSR* game, the DA strategy and DR strategy in the competitive strategy profile are respectively optimal for the APT attacker and the cloud defender under the Nash equilibrium solution concept. Therefore, from the worst-case perspective, we recommend to the cloud defender the DR strategy in the competitive strategy profile.

Second, in this paper, we solve the DCSR problem in the worst-case situation where the cloud defender assumes that the APT attacker has full knowledge of his expected loss. However, in practice, due to the lack of information or bounded rationality, the APT attacker may well be unaware of the exact form of the cloud defender's expected loss. In this context, the attacker may not be able to find the DA strategy in the competitive strategy profile. As a result, the attacker may subjectively choose a DA strategy. From Experiment 1(3) and Experiment 2(3) we can conclude that the APT attacker may well choose a DA strategy in favor of the cloud defender (i.e., a lower total loss) when the attacker does not have full knowledge of the cloud defender's expected loss, which is an advantage for the cloud defender to realize the DR strategy in the competitive strategy profile.

VI. PRACTICABILITY OF OUR WORK

In previous sections, we presented a full framework for dealing with the DCSR problem. In this section, we briefly analyze the practicability of our work.

A. SCALABILITY

From a holistic perspective, game theory provides a suitable framework for modeling a variety of cybersecurity problems. In a cybersecurity game, the main task of the defender is to seek an equilibrium (Nash, Stackelberg, etc.) of the game. Though classical works provide rich mathematical foundations and equilibrium concepts, searching the admissible set of strategies for an equilibrium will become extremely difficult with the increasing scale and complexity of the target system. Computational game theory aims at addressing such algorithmic issues, and different algorithms were proposed to tackle complex cybersecurity game problems.

Despite the advancement of application of computational game theory to cybersecurity domain, the curse of dimensionality is still a stumbling block for dealing with many cybersecurity games. In truth, the main difficulty comes from the process of finding the equilibrium of a cybersecurity game. In most cases, the algorithms (learning, uphill/downhill, etc.) executed by the defender start from some initial points in the admissible set of strategies and then search the set for

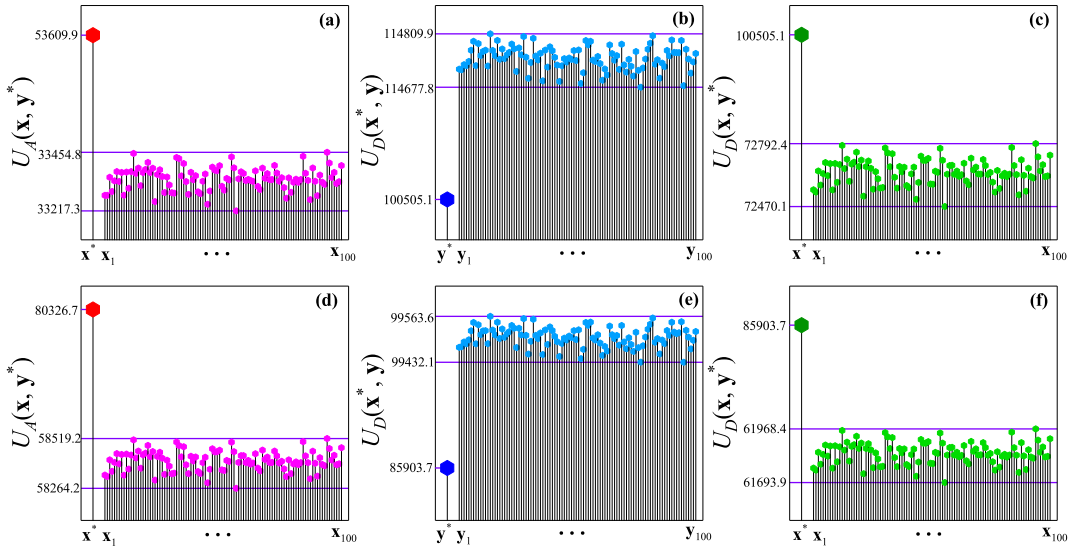

FIGURE 7. The performances of the competitive strategy profiles in Experiment 2, where (a)-(c) are for the DCSR* game G^3, (d)-(f) are for the DCSR* game G^4.

an equilibrium according to some basic rules. Due to the lack of knowledge of the form of the equilibrium, the time cost of some of these algorithms may become prohibitive quite quickly with the increasing scale of the target system. Consequently, the scalability of some of these algorithms is questionable.

In our work, the necessity system provides guidance for the defender to search the admissible set of strategies for the competitive strategy profile. In fact, the necessity system shows the specific form that the competitive strategy profile must satisfy. As a result, the time cost is much lower than that of the above mentioned algorithms. Therefore, our work delivers superior scalability, making it a good solution to the DCSR problem.

B. FEASIBILITY

In this subsection, we discuss the potential of applying a DCSR* game model to solve the real-world DCSR problem.

In practice, it is important to determine the model parameters. The parameters related to the APT attacker can be estimated by the cloud defender. Specifically, by conducting repeated APT attack-defense manoeuvres, the cloud defender can generate a mass of data. On this basis, the attack lower and upper bound vectors can be estimated by statistical analysis of the generated data, and the attack cost function vector can be approximated by fitting the generated data. The other parameters are controllable by the cloud defender. That is, the recovery lower and upper bound vectors are determined by the security requirement and budget, the recovery cost function vector is configured manually, the initial CSS's expected state can be estimated by some proven APT detection tools, and the loss coefficient is dependent on the value of the data stored in the CSS which is typically known to the cloud defender.

To solve a real-world DCSR problem, the cloud defender needs to take the following two steps.

- *Step 1:* At time $t = 0$, estimate/configure model parameters, numerically solve the necessity system to obtain the competitive strategy profile $(\mathbf{x}^*, \mathbf{y}^*)$.
- *Step 2:* During the time horizon $(0, T]$, allocate the security resources to recover the CSS in accordance with the DR strategy $\mathbf{y}^*$ obtained in Step 1.

In practice, the cloud defender can repeat the above procedure so that he can defend against APT in a long time span, though our work is restricted to the time horizon $[0, T]$. In addition, this procedure can be executed in a relatively flexible fashion. Specifically, on the one hand, a larger T means a lower frequency of re-estimation of the model parameters, which reduces the overheads for re-estimation works. Nevertheless, the behavior of the game model may deviate from the real scenario, resulting in insufficient DR strategies. On the other hand, a smaller T means the model parameters will be re-estimated more frequently, which will inevitably increase the overheads, though it helps to guarantee the accuracy of the behavior of the game model. Consequently, by dynamically adjusting T, the cloud defender may achieve a better balance in the course of dealing with a real-world DCSR problem.

VII. CONCLUSION

This paper has addressed the problem of finding an effective dynamic recovery (DR) strategy to mitigate the cloud defender's total loss under an identified APT campaign, which we refer to as the dynamic cloud storage recovery (DCSR) problem. Based on an expected state evolution model, the APT attacker's net benefit and the cloud defender's total loss have been measured. From the worst-case point of view, the DCSR problem has been reduced to a differential game-theoretic problem (the DCSR* problem), in which the ultimate goal of the cloud defender is to seek a Nash equilibrium strategy profile. A necessity system has been derived and the concept of competitive strategy

profile has been introduced. The structural properties of the competitive strategy profile have been examined. Extensive comparative experiments have shown that the competitive strategy profile outperforms a large number of randomly generated strategy profiles. Finally, the practicability (scalability and feasibility) of this work has been analyzed briefly.

There are some problems to be solved. In practice, the APT attacker may well have only partial information about the cloud defender's expected loss. Hence, the DCSR problem can be addressed by using Bayesian game approach [50], [51]. In addition, this work is based on expected utility theory (EUT), in which the players choose their strategies to optimize their expected utilities. In the situation in which the players with bounded rationality may suffer the deviations of decisions from EUT-based results, the prospect theory (PT) can be used to deal with the DCSR problem [52]. The challenge of an APT campaign is escalated when it is facilitated by the insiders, which often have privileged access to the CSS and could trade valuable data to the APT attacker for financial benefits [53], [54]. In this context, the interaction among the cloud defender, the APT attacker, and the insiders should be judiciously studied to find an effective recovery strategy. Moreover, with the development of mobility, it is worthwhile to protect mobile cloud computing from the APTs [55], [56].

ACKNOWLEDGMENTS

The authors are grateful to the five anonymous reviewers and the editor for their valuable comments and suggestions that have improved the quality of the paper greatly.

REFERENCES

[1] D. C. Marinescu, *Cloud Computing: Theory and Practice*, 2nd ed. Cambridge, MA, USA: Morgan Kaufmann, 2018.

[2] *Cloud Storage Market Size, Share—Segmented by Solution (Cloud Storage Gateway, Primary Storage, Backup Storage, Data Archiving), Service (Managed Services, Professional Services), Deployment (Private Cloud, Public Cloud, Hybrid Cloud), End User, and Region-Growth, Trends, and Forecast (2019–2024)*, document, Mordor Intell., Hyderabad, India, 2019.

[3] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, "Cloud-trust—A security assessment model for infrastructure as a service (IaaS) clouds," *IEEE Trans. Cloud Comput.*, vol. 5, no. 3, pp. 523–536, Jul./Sep. 2017.

[4] S. Mu, K. Chen, P. Gao, F. Ye, Y. Wu, and W. Zheng, "μLibCloud: Providing high available and uniform accessing to multiple cloud storages," in *Proc. ACM/IEEE 13th Int. Conf. Grid Comput.*, Sep. 2012, pp. 201–208.

[5] E. Cole, *Advanced Persistent Threat: Understanding the Danger and How to Protect Your Organization*. Waltham, MA, USA: Syngress, 2013.

[6] Y. Zhang, B. An, L. Tran-Thanh, Z. Wang, J. Gan, and N. R. Jennings, "Optimal escape interdiction on transportation networks," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Aug. 2017, pp. 3936–3944.

[7] C. Xu, K. Wang, P. Li, S. Guo, J. Luo, B. Ye, and M. Guo, "Making big data open in edges: A resource-efficient blockchain-based approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 4, pp. 870–882, Apr. 2019.

[8] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric IoT," *IEEE Trans. Emerg. Topic Comput.*, to be published. doi: 10.1109/TETC.2018.2805718.

[9] K. Wang, Q. Zhou, S. Guo, and J. Luo, "Cluster frameworks for efficient scheduling and resource allocation in data center networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3560–3580, 4th Quart., 2018.

[10] C. Xu, K. Wang, and M. Guo, "Intelligent resource management in blockchain-based cloud datacenters," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 50–59, Nov./Dec. 2017.

[11] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. New York, NY, USA: SIAM, 1999.

[12] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1851–1877, 2nd Quart., 2019.

[13] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Comput. Secur.*, vol. 48, pp. 35–57, Feb. 2015.

[14] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, and F. Aparicio-Navarro, "Detection of advanced persistent threat using machine-learning correlation analysis," *Futur. Gener. Comput. Syst.*, vol. 89, pp. 349–359, Dec. 2018.

[15] L. Xiao, D. Xu, N. B. Mandayam, and H. V. Poor, "Attacker-centric view of a detection game against advanced persistent threats," *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2512–2523, Nov. 2018.

[16] B. L. J. Chuan, M. M. Singh, and A. R. M. Shariff, "APTGuard: Advanced persistent threat (APT) detections and predictions using Android smartphone," in *Computational Science and Technology*. Singapore: Springer, 2018, pp. 545–555.

[17] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, "Security evaluation of the cyber networks under advanced persistent threats," *IEEE Access*, vol. 5, pp. 20111–20123, 2017.

[18] P. Li, X. Yang, Q. Xiong, J. Wen, and Y. Y. Tang, "Defending against the advanced persistent threat: An optimal control approach," *Secur. Commun. Netw.*, vol. 2018, pp. 1–14, Feb. 2018, Art. no. 2975376.

[19] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, "A risk management approach to defending against the advanced persistent threat," *IEEE Trans. Dependable Secur. Comput.*, to be published. doi: 10.1109/TDSC.2018.2858786.

[20] L.-X. Yang, P. Li, Y. Zhang, X. Yang, Y. Xiang, and W. Zhou, "Effective repair strategy against advanced persistent threat: A differential game approach," *IEEE Trans. Inf. Forensic Security*, vol. 14, no. 7, pp. 1713–1728, Jul. 2018.

[21] R. Zheng, W. Lu, and S. Xu, "Preventive and reactive cyber defense dynamics is globally stable," *IEEE Trans. Netw. Sci. Eng.*, vol. 5, no. 2, pp. 156–170, Apr./Jun. 2018.

[22] M. Xu, K. M. Schweitzer, R. M. Bateman, and S. Xu, "Modeling and predicting cyber hacking breaches," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 11, pp. 2856–2871, Nov. 2018.

[23] P. Du, Z. Sun, H. Chen, J.-H. Cho, and S. Xu, "Statistical estimation of malware detection metrics in the absence of ground truth," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 12, pp. 2965–2980, Dec. 2018.

[24] Z. Lin, W. Lu, and S. Xu, "Unified preventive and reactive cyber defense dynamics is still globally convergent," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1098–1111, Jun. 2019.

[25] J. H. Jafarian, E. AI-Shaer, and Q. Duan, "Adversary-aware IP address randomization for proactive agility against sophisticated attackers," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 738–746.

[26] R. Zhuang, A. Bardas, S. DeLoach, and X. Ou, "A theory of cyber attacks: A step towards analyzing MTD systems," in *Proc. 2nd ACM Workshop Moving Target Defense*, Oct. 2015, pp. 11–20.

[27] S. Hosseinzadeh, S. Rauti, S. Laurén, and J.-M. Mäkelä, "Diversification and obfuscation techniques for software security: A systematic literature review," *Inf. Softw. Technol.*, vol. 104, pp. 72–93, Dec. 2018.

[28] J. V. Chandra, N. Challa, and S. K. Pasupuleti, "Advanced persistent threat defense system using self-destructive mechanism for cloud security," in *Proc. IEEE Int. Conf. Eng. Technol. (ICETECH)*, Mar. 2016, pp. 7–11.

[29] H. Zhuang, R. Rahman, P. Hui, and K. Aberer, "Optimizing information leakage in multicloud storage services," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2018.2808275.

[30] M. H. Ameri, M. Delavar, J. Mohajeri, and M. Salmasizadeh, "A key-policy attribute-based temporary keyword search scheme for secure cloud storage," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2018.2825983.

[31] M. J. Osborne, *An Introduction to Game Theory*. New York, NY, USA: Oxford Univ. Press, 2003.

[32] Q. Zhu, C. Fung, R. Boutaba, and T. Basar, "GUIDEX: A game-theoretic incentive-based mechanism for intrusion detection networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 11, pp. 2220–2230, Dec. 2012.

[33] Y. Wang, F. R. Yu, H. Tang, and M. Huang, "A mean field game theoretic approach for security enhancements in mobile ad hoc networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1616–1627, Mar. 2014.
[34] R. Shokri, G. Theodorakopoulos, and C. Troncoso, "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Trans. Privacy Secur.*, vol. 19, no. 4, Art. no. 11, 2017.
[35] M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "FlipIt: The game of 'stealthy takeover,'" *J. Cryptol.*, vol. 26, no. 4, pp. 655–713, 2013.
[36] J. Pawlick, S. Farhang, and Q. Zhu, "Flip the cloud: Cyber-physical signaling games in the presence of advanced persistent threats," in *Decision and Game Theory for Security*. Cham, Switzerland: Springer, 2015, pp. 289–308.
[37] D. Xu, Y. Li, L. Xiao, N. B. Mandayam, and H. V. Poor, "Prospect theoretic study of cloud storage defense against advanced persistent threats," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[38] L. Xiao, D. Xu, C. Xie, N. B. Mandayam, and H. V. Poor, "Cloud storage defense against advanced persistent threats: A prospect theoretic study," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 534–544, Mar. 2017.
[39] D. Xu, L. Xiao, N. B. Mandayam, and H. V. Poor, "Cumulative prospect theoretic study of a cloud storage defense game against advanced persistent threats," in *Proc. IEEE Int. Conf. Comput. Commun. BigSecurity Workshop (INFOCOM)*, May 2017, pp. 541–546.
[40] M. Min, L. Xiao, C. Xie, M. Hajimirsadeghi, and N. B. Mandayam, "Defense against advanced persistent threats: A colonel blotto game approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[41] M. Min, L. Xiao, C. Xie, M. Hajimirsadeghi, and N. B. Mandayam, "Defense against advanced persistent threats in dynamic cloud storage: A colonel blotto game approach," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4250–4261, Dec. 2018.
[42] A. A. A. Abass, L. Xiao, N. B. Mandayam, and Z. Gajic, "Evolutionary game theoretic analysis of advanced persistent threats against cloud storage," *IEEE Access*, vol. 5, pp. 8482–8491, 2017.
[43] S. Feng, Z. Xiong, D. Niyato, and P. Wang, "Dynamic resource management to defend against advanced persistent threats in fog computing: A game theoretic approach," *IEEE Trans. Cloud Comput.*, to be published. doi: 10.1109/TCC.2019.2896632.
[44] J. Chen and Q. Zhu, "Security as a service for cloud-enabled Internet of controlled things under advanced persistent threats: A contract design approach," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2736–2750, Nov. 2017.
[45] J. Pawlick and Q. Zhu, "Strategic trust in cloud-enabled cyber-physical systems with an application to glucose control," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 2906–2919, Dec. 2017.
[46] S. Jörgensen, G. Martín-Herrán, and G. Zaccour, "The Leitmann–Schmitendorf advertising differential game," *Appl. Math. Comput.*, vol. 217, no. 3, pp. 1110–1116, 2010.
[47] R. Cellini and L. Lambertini, "Weak and strong time consistency in a differential oligopoly game with capital accumulation," *J. Optim. Theory Appl.*, vol. 138, no. 1, pp. 17–26, 2008.
[48] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton, NJ, USA: Princeton Univ. Press, 2012.
[49] K. Atkinson, W. Han, and D. Stewart, *Numerical Solution of Ordinary Differential Equation*. Hoboken, NJ, USA: Wiley, 2009.
[50] L. Huang and Q. Zhu, "Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 2, pp. 52–56, 2019.
[51] L. Huang and Q. Zhu, "Dynamic Bayesian games for adversarial and defensive cyber deception," in *Autonomous Cyber Deception*. Cham, Switzerland: Springer, 2019, pp. 75–97.
[52] A. Sanjab, W. Saad, and T. Başar, "Prospect theory for enhanced cyber-physical security of drone delivery systems: A network interdiction game," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.
[53] L. Liu, O. de Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and preventing cyber insider threats: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1397–1417, 2nd Quart., 2018.
[54] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici, and M. Ochoa, "Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Comput. Surv.*, vol. 52, no. 2, 2019, Art. no. 30.
[55] Y. Liu, Y. Zhang, J. Ling, and Z. Liu, "Secure and fine-grained access control on e-healthcare records in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 1020–1026, Jan. 2018.
[56] A. J. Ferrer, J. M. Marqués, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Comput. Surv.*, vol. 51, no. 6, 2019, Art. no. 111.

PENGDENG LI received the B.Sc. degree from the School of Software Engineering, Chongqing University, in 2015, where he is currently pursuing the Ph.D. degree. He has published about 14 academic papers in peer-reviewed international journals. His research interests include epidemic dynamics and cybersecurity.

XIAOFAN YANG received the B.Sc. degree from the Department of Mathematics, Sichuan University, in 1985, the M.Sc. degree from the Department of Applied Mathematics, Chongqing University, in 1988, and the Ph.D. degree from the Department of Computer Science, Chongqing University, in 1994. He visited the University of Reading, from 1998 to 1999. He is currently a Professor of computer science with Chongqing University. He has published more than 160 academic papers in peer-reviewed international journals, and more than 20 students have received Ph.D. degree under his supervision. His research interests include fault-tolerant computing, epidemic dynamics, and cybersecurity.