Computation Offloading and Security with Q-Learning

Songyang Ge[1,2], Beiling Lu[3], Jie Gong[3], and Xiang Chen[1,2(✉)]

[1] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
chenxiang@mail.sysu.edu.cn
[2] Key Lab of EDA, Research Institute of Tsinghua University in Shenzhen (RITS), Shenzhen 518075, China
[3] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Abstract. With the rapid development of the technology and wireless communication, the user cannot support the computation-intensive applications, owing to the restricted computation resources, energy supply, limited memory space and communication resources. The emerging computation mode, called mobile edge computing (MEC), provides a solution that the user can unload parts of tasks to edge servers. This communication process should be finished in the wireless network. However, computation offloading in the wireless network can encounter many kinds of attacks. Specifically, edge servers located in the edge of network are vulnerable to these security threats, such as spoofing, jamming and eavesdropping. Moreover, the computation offloading has much time latency and energy consumption. Then, how to minimize this consumption is the another problem to be solved. To improve the security and minimize the consumption, we formulate a system containing a primary user (PU), a second user (SU), an attacker and several edge servers. They communicate with each other by multiple input multiple output (MIMO) technology. In this system, the SU chooses an MEC server from the set of not being occupied by PU, determines an offloading rate and a transmission power, then the attacker selects the action of attack. The aim of this system is to optimize the utility of SU. To solve this problem, a Q-learning based optimal offloading strategy is proposed in dynamic environments. Simulation results show that our proposed scheme can improve the capacity of SU and efficiently decrease the attack rate of the attacker.

Keywords: Computation offloading · System security · Q-learning

The work is supported in part of Science, Technology and Innovation Commission of Shenzhen Municipality (No. JCYJ20170816151823313), NSFC (No. U1734209, No. 61501527), States Key Project of Research and Development Plan (No. 2017YFE0121300-6), The 54th Research Institute of China Electronics Technology Group Corporation (No. B0105) and Guangdong Provincial Special Fund For Modern Agriculture Industry Technology Innovation Teams (No. 2019KJ122).

B. Li et al. (Eds.): IoTaaS 2019, LNICST 316, pp. 71–81, 2020.
https://doi.org/10.1007/978-3-030-44751-9_7

1 Introduction

With the rapid development of mobile internet and interest of thing (IoT), many kinds of new services are constantly emerging, which makes the explosive growth of mobile communication outflow possible. The mobile terminals and smart phones, are gradually replaced by personal computer which is the main tool in the daily work, study, entertainment and social association. Meanwhile, a large number of IoT terminal devices, such as smart watches, cameras and a variety of sensors, are comprehensively applied in plenty of industries, including traffic, smart home, education, health care and agriculture.

In order to meet the demand above, different types of solutions are born at the right moment, such as Cloud computing, fog computing and mobile edge computing. Cloud computing characterizes its centralization of computing and storing data and network management by making use of cloud center [1]. Though it is convenient for humans that terminal devices access cloud computing center directly, it brings heavy network burdens, long computation delay and higher requirements for bandwidth. MEC gives a good scheme to solve the serious problems above by putting servers into the edge of network. Thus, MEC is one of the most popular schemes and regarded as a vital promoter of evolution for cellular base station. Besides, MEC can be applied in many scenarios [2], comprising of dynamic connect optimization, computational offloading in IoT, mobile big data analytic and smart transportation.

With computation Offloading, user terminals in the mode of MEC can unload tasks to the edge MEC servers, such as base stations, access points and laptops, to decrease the delay of computation, prolong the life of battery and save the computing resources [3]. There are two ways of offloading, including binary offloading and partial offloading. In this paper, the partial offloading is considered. The task can be divided into two parts in partial offloading, in which one part is for computing locally and the other is for offloading to edge servers [4,5].

In addition, owing to the fast process of development of technology, it is a challenge for MEC server to face complex wireless network. Firstly, terminal users do not know the action of other users. For instance, compared with secondary user (SU), the primary user (PU) has a priority of using spectrum is referred to [6]. Secondly, There are more and more attackers and types of smart attacks. The interaction between a smart attacker and an end-user by using prospect theory is formulated in [7]. Because MEC servers are located in the edge of network, they are closer to attackers. Besides of the advanced persistent threats to cloud storage researched in [8,9], mobile edge computing can meet more classes of attacks. In order to provide secure offloading to MEC servers, the solution to different kinds of attacks by applying reinforcement learning methods are summarized in [10].

In this paper, we propose a computation offloading game against smart attacks under the condition of existence of one primary user and a second user. Besides, we propose a Q-learning [11] based scheme for SU by choosing proper MEC server, an offloading rate and a transmission power to optimize the utility.

The main contributions of this work is summarized as follows:

(1) We investigate the computation offloading of SU with multi-antennas in the wireless network. For simplicity, we set two states of MEC server, occupied by PU and not. SU chooses the proper MEC server from the idle set. Next, SU terminal allocates the accurate amount of task to server and ensures the transmission power.
(2) A Q-learning based optimal computation offloading strategy is developed to improve the utility of SU, after observing the time varying channel information. The simulation results show that our proposed scheme can improve the utility and decrease the attack rate.

The organization of the rest is as follows. We review the related work in Sect. 2 and formulate a computation offloading game against smart attacks in Sect. 3. Moreover, a Q-learning algorithm based computation offloading with unknown channel model is proposed in Sect. 4. Then, we provide the simulation results in Sect. 5 and make conclusions in Sect. 6.

2 Related Work

Multiple input and multiple output (MIMO) is one of popular research directions nowadays. We assume that one SU, several MEC servers and an attacker are all with multiple antennas. The interaction between the receiver with multi-antennas and one spoofing node is formulated as a zero-sum physical-layer authentication game in [12]. But it only investigates one type of attack. To reduce the speed of attack and improve the secrecy capacity, one noncooperative MIMO transmission against smart attacks game, including eavesdropping, jamming, and spoofing, is proposed in [13].

In the scene of computation offloading, some researchers focus on the attack defense. By jointly optimizing the energy transmit beamforming at access point, the frequency of central process unit and offloading rate, a solution to minimize the energy consumption and time delay of a single user is derived in [14]. For the computation offloading model, a reinforcement learning based offloading frame is formulated in [15], after observing the battery level, the previous radio bandwidth and the amount of energy harvested. In this paper, We combine the MIMO scenario and computation model to simulate more complex communication environment. Furthermore, in the dynamic MEC network with varying channel state, SU cannot optimize the offloading policy against various types of smart attacks quickly and accurately. We use one of reinforcement learning algorithms, Q-learning, to derive the optimal offloading strategy.

3 Computation Offloading Game Against Attacks

3.1 System Model

We consider a mobile edge offloading system with MIMO transmissions as shown in Fig. 1, consisting a PU, a SU with N_u antennas, M MEC servers

with N_m antennas and an attacker with N_a antennas. Because of the limited computation ability and battery level, SU cannot compute total task. SU chooses one specific MEC server to offload tasks, which is not occupied by PU. Besides, when MEC servers receive the signals from SU or PU, it might be attacked by attacker in the way of 4 types, including keeping silent, spoofing, jamming and eavesdropping.

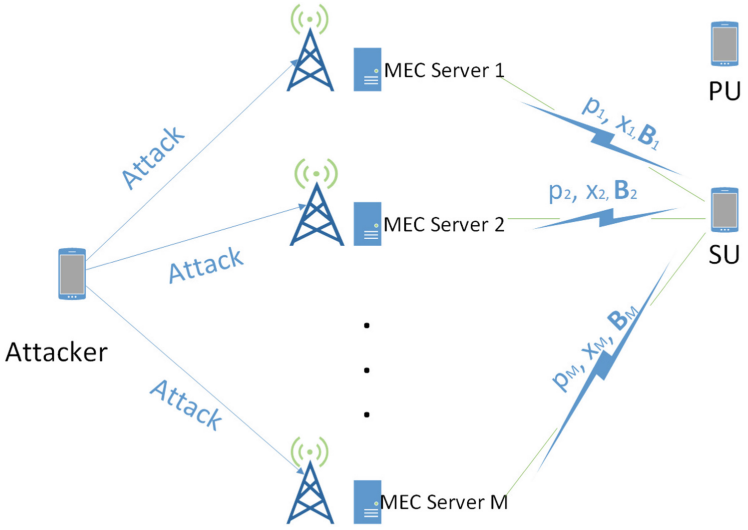

Fig. 1. Sytem model.

3.2 Offloading Model

With the development of wireless communication, the task to be computed by user becomes lager and larger. Generally, mobile device cannot meet this demand due to limited computing resources. The primary user can occupy MEC server. Therefore, MEC server has two states, being occupied or not. SU can choose an MEC sever from the idle server set to offload partial tasks, which can lighten the load greatly. Assuming that the time structure of computation offloading process is slotted, we denote the time slot index by k, with $k \in \mathcal{K} = \{0, 1, \ldots\}$. The size of computing task generated by SU at time slot k is denoted by $L^{(k)}$ (in bits). The index of the MEC server selected is i, which satisfies $1 \leq i \leq M$. The proportion of the offloading task, called offloading rate, is denoted by $x_i^{(k)}$, with $0 \leq x_i^{(k)} \leq 1$. More specifically, if $x_i^{(k)} = 0$, the whole task would be computed by SU; if $x_i^{(k)} = 1$, then the task would be handled by MEC_i totally; if $0 < x_i^{(k)} < 1$, the task $x_i^{(k)} L^{(k)}$ would be computed by MEC_i, the left $(1 - x_i^{(k)}) L^{(k)}$ is to be

operate by SU locally. Simply, we quantize the offloading rate into $N_x + 1$ levels, i.e., $x_i^{(k)} \in \{l/N_x\}_{0 \le l \le N_x}$.

For local-computing model, we first ensure that the task computed locally is $(1-x_i^{(k)})L^{(k)}$. The CPU cycles required for computation are denoted by ϕ. The CPU frequency of mobile device of SU, i.e., the computing speed of operator, is represented by f_j, with $f_j \le f_{max}$. We denote the computing energy efficiency of operator chip as k_j. Then, the energy consumption of local computation e_0 [3] is represented as

$$e_0 = \sum_{j=1}^{(1-x_i^{(k)})L^{(k)}\phi} k_j f_j^2. \tag{1}$$

The computing time can be written as

$$t_0 = \sum_{j=1}^{(1-x_i^{(k)})L^{(k)}\phi} \frac{1}{f_j}. \tag{2}$$

For computation-offloading model, we should be clear about the task to be offloaded is $x_i^{(k)}L^{(k)}$. Moreover, SU chooses the appropriate power $p_i^{(k)}$ to transmit signals to MEC_i, where the transmission power has bounds, i.e. $0 \le p_i^{(k)} \le p_{max}$. The bandwidth between SU and MEC is B_i. We only consider the time and energy overhead in the process of transmission. The energy consumption can be denoted as

$$e_1 = \frac{p_i^{(k)} x_i^{(k)} L^{(k)}}{B_i C_g}. \tag{3}$$

The total time of offloading computation t_1 is

$$t_1 = \frac{x_i^{(k)} L^{(k)}}{B_i C_g}. \tag{4}$$

In the analysis above, we neglect the time required in the back transmission of computation results from MEC server to SU. It is due to the reality that the amount of result data of computation task is much smaller than the size of input data. Thus, we only take the offloading time into account rather than the time delay in the back transmission.

3.3 Attack Defense Model

In this computation offloading system with MIMO transmission, SU sends M-dimensional signal vector with power $p_i^{(k)}$. From the transmitting antennas at SU to the receiving antennas at MEC_i, the channel gains can be described as channel matrix $\mathbf{H}_{um}$. In the same way, The channel matrix between SU and attacker (or between MEC server and attacker) is $\mathbf{H}_{ua}$ (or $\mathbf{H}_{ma}$). We assume that the distribution of each channel matrix follows independently and identically

distributed (i.i.d) complex Gaussian distribution, i.e., $\mathbf{H}_n \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2\mathbf{I})$, with $n = um, ua, ma$.

There is an attacker generating attack to SU in the way of 4 types, including keeping silent, spoofing, jamming and eavesdropping. The attack mode can be denoted as g, with $g \in \mathcal{A}_a = \{0, 1, 2, 3\}$. The secrecy capacity C_g is formulated as [13].

For ease of reference, we list the key notation of our system model in Table 1.

Table 1. Summary of symbols and notation

Notation	Definition
N_u	Number of antennas at SU
N_m	Number of antennas at MEC server
N_a	Number of antennas at attacker
M	Number of MEC server
$L^{(k)}$	Total task of SU in time slot k
$x_i^{(k)}$	Offloading rate on MEC_i in time slot k
$p_i^{(k)}$	Transmission power to MEC_i in time slot k
f_j	CPU frequency in j-th cycle
ϕ	CPU cycles required for local computation
B_i	Bandwidth between SU and MEC_i
$t_{0/1}$	Computing time of SU or MEC
$e_{0/1}$	Energy consumption of SU or MEC
$\mathbf{H}_{um/ma/ua}$	Channel matrix
C	Secrecy Capacity
$\omega_{0/1/2/3}$	Attack cost
U_a	Utility of SU
α	Learning rate
δ	Discount factor

3.4 Game Model

We formulate the relationship between the attacker and SU as a computation offloading game against smart attacks in the MIMO wireless environment. In this process, SU is firstly constrained by PU. As a result of the limited spectrum resource, PU has priority over utilizing MEC server to unload computation task. Next, SU should select the idle server to take offloading with some transmission power. Furthermore, attacker chooses attack model to launch different kinds of smart attacks. In each time slot, attacker takes action to decrease the cost and improve the utility. Meanwhile, SU tries its best to maximize utility. In a word, we provide the game with two players maximizing their own utilities.

In the computation offloading game above, SU chooses MEC_i from idle server set, suitable transmission power and offloading rate under smart attacks. This process of choice surely makes some cost and energy consumption. We divide the total cost to 3 parts. We assume that the results of three parts have no units. We regard each part as a factor which influences the utility in the different degree and direction. The first part is about the amount of offloading computation task. Due to its beneficial property, we make it a positive number. The second section is energy consumption consisting of local computation and transmission energy consumption. In addition, we add a coefficient to the front of the sum of energy consumption to represent the impact, denoted as ρ. In order to meet the time delay constraint, SU is supposed to reduce its utility. Thus, the third portion is the overhead of time behind the corresponding coefficient ν. Obviously, the second and third parts are overhead of SU, and decrease the utility of mobile user. Therefore, we add a negative sign in the front of the value. Thus, the utility of SU in the game, denoted by U_u, relies on offloading rate, energy consumption, and delay constraint. We write the utility of SU as

$$U_u = x_i^{(k)} L^{(k)} - \rho(e_0 + e_1) - \nu(t_0 + t_1). \tag{5}$$

At the same time, attacker selects one attack mode g from $\{0, 1, 2, 3\}$, corresponding to keeping silent, spoofing, jamming and eavesdropping respectively. But attacker cannot launch blind attack for the cost. We classify the cost into 4 types according to different attacks, represented as $\{\omega_0, \omega_1, \omega_2, \omega_3\}$. Thus the utility of attacker is defined as

$$U_a = (-C_g - \omega_g),\ g = 0, 1, 2, 3. \tag{6}$$

4 Offloading Strategy of SU in the MIMO System Based on Q-Learning

In a dynamic computation offloading computing game, it is hard for SU to estimate the current environment state, including dynamic channel condition, the action of PU and attacks of diverse types. We model the system as a Markov Decision Process, which has a finite state set and is continuous. Moreover, we represent the process as $<\mathcal{S}, \mathcal{A}_u, \mathcal{P}(s, a, s'), \mathcal{R}(s, a, s')>$, where $\mathcal{S}$ is the state set of SU and $\mathcal{A}_u$ denotes action set. $\mathcal{P}(s, a, s')$ indicates the transition probability, i.e., the agent in the current state s by choosing action a would arrive the next state s'. Moreover, $\mathcal{R}(s, a, s')$ represents the direct reward in the time of choosing action a in the current state s.

The process of seeking optimal strategy is summarized in Algorithm 1. Based on the communication environment, SU chooses one specific action $a^{(k)}$, comprising of MEC_i, $x_i^{(k)}$ and $p_i^{(k)}$, from the action set $\mathcal{A}_u$ in time slot k. Assume that SU regards the attack type of last time slot as its system state, represented as $s^{(k)} = g^{(k-1)}$. We define the direct reward as the utility discussed in the last section.

Since the transition probability is unknown in the system, we apply a kind of reinforcement learning method, called Q learning algorithm. With the reinforcement learning methods emerging as the time requires, SU can make the best of one of these methods, Q-learning algorithm, to achieve optimal mobile edge offloading strategies and get the most gain. Moreover, Q-learning algorithm can obtain the optimal strategy via trial-and-error under the condition of not assuming any probability model. The contents above are the main elements of process of Q-learning algorithm. Besides, they are also basic components of decision making process.

We write the Q function of SU as $Q(s^{(k)},\ a^{(k)})$, which is the state-action value function of SU. The value function defined as $V(s^{(k)})$ is the highest value of the current state in time slot k, called state value function. Then we have the iteration equation denoted as

$$Q(s^{(k)},\ a^{(k)}) \longleftarrow (1-\alpha)Q(s^{(k)},\ a^{(k)}) + \alpha(U_a(s^{(k)},\ a^{(k)}) + \delta V(s^{(k+1)})), \quad (7)$$

$$V(s^{(k)}) = \max_{a \in \mathcal{A}_u} Q(s^{(k)},\ a), \quad (8)$$

where α is the learning rate of this algorithm, $\delta \in [0, 1]$ is the discount factor about future reward. By the iteration in the learning, SU can find the optimal policy. In the current state, agent can choose the best action, observe next state and direct reward value. Lastly, Q function updates according to (7) and value function renews by (8). It can be shown that given sufficient number of iterations Q learning can converge to optimal result and max long term reward.

Because of the advantage of the policy, it is favorable for SU to choose the best action based on the system state and improve convergence performance. Consequently, SU applies the ε-greedy policy in the process of learning. That is, SU selects action with highest probability from the optimal Q function, gains the maximum of direct reward and selects the other action randomly. Thus, the learning process can balance exploration and exploitation. The probability equation above can be given by

$$\Pr(a_i^{(k)} = \tilde{a}) \begin{cases} 1-\varepsilon, \ \text{if} \ \ \tilde{a} = \arg\max\limits_{a \in \mathcal{A}_u} Q(s^{(k+1)},\ a) \\ \frac{\varepsilon}{N_x}, \ \text{otherwise.} \end{cases} \quad (9)$$

During the process of learning, SU learns the system state and unloads partial task $x_i^{(k)} L^{(k)}$ to MEC_i with transmission $p_i^{(k)}$ to increase the long-term reward. The mobile edge offloading scheme against smart attacks with Q-learning is summarized in Algorithm 1.

Algorithm 1. Q-learning based Computation Offloading Scheme

Initialize $g^{(0)} = 0$, $Q(s, a) = 0$, $V(s) = 0$, $\forall s,\ a$.
for each episode **do**
  for $n = 1, 2, 3, \cdots$ **do**
    Update the state $s^{(k)} = g^{(k-1)}$;
    Choose $a^{(k)}$ with $\varepsilon-$ greedy policy;
    Observe the attack type $g^{(k)}$ and U_u ;
    Update the Q function and value function,
    $Q(s^{(k)},\ a^{(k)}) \longleftarrow (1-\alpha)Q(s^{(k)},\ a^{(k)}) + \alpha(U_a(s^{(k)},\ a^{(k)}) + \delta V(s^{(k+1)}))$,
    $V(s^{(k)}) = \max_{a \in \mathcal{A}_u} Q(s^{(k)},\ a)$.
  end for
end for

5 Simulation Results

We evaluate the performance of the computation offloading computing scheme via simulations with $L = 100$, $P = 6 : 10$, $M = 3$, $N_u = 5$, $N_m = N_a = 2$. As shown in Fig. 2, the average utility of SU increases with the growth of time slot. It raises by 173.8% over 500 time slots. Between 0-th and 500-th time slot, the utility has a rapid development and converge at 384-th time slot gradually. The reason why the curve of mobile edge offloading computing in MIMO systems changes is SU can choose different transmission powers and offloading rates according to diverse MEC server.

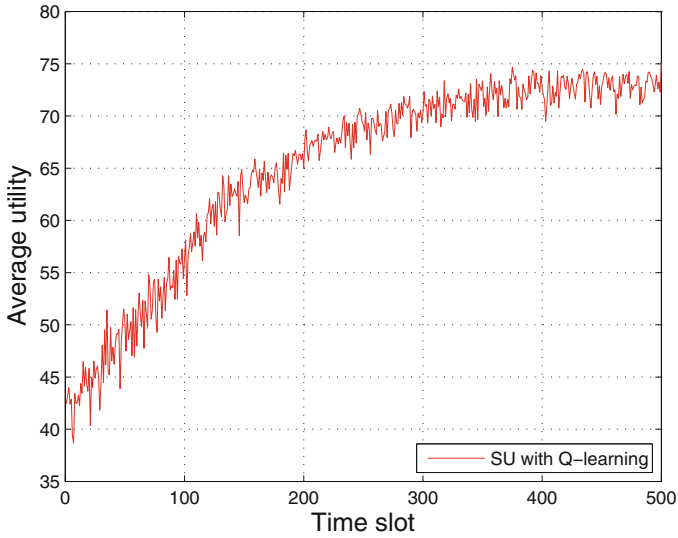

Fig. 2. Performance of computation offloading computing utility of SU for $5 \times 2 \times 2$ MIMO system with $L = 100$, $P = 6 : 10$, $M = 3$.

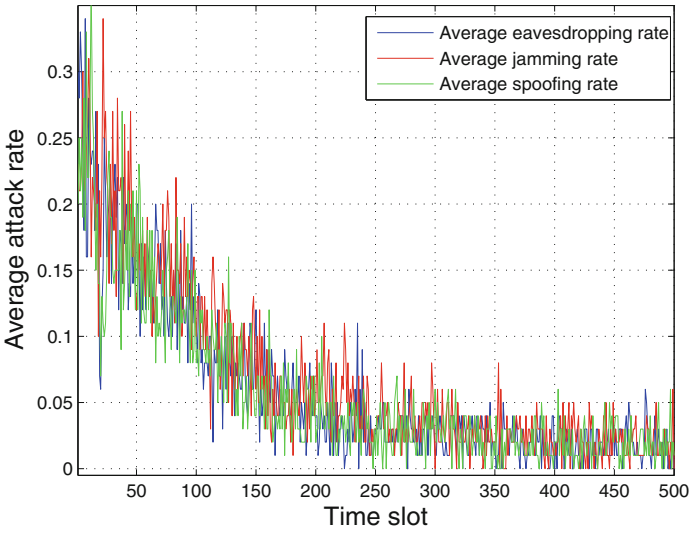

Fig. 3. The attack rate of three attacks in computation offloading scheme with $L = 100$, $P = 6 : 10$, $M = 3$.

As shown in Fig. 3, the proposed computation offloading scheme in the MIMO systems reduces rapidly the attack rate of different types of smart attack, including spoofing, jamming and eavesdropping. For instance, the scheme decreases the attack frequency of spoofing from 0.3 to 0.01 after 200 time slots. Besides, it is the first one to be close to zero. Similarly, the attack rate of eavesdropping is the second one to converge to zero. Small amplitude of fluctuation close to zero was shown in the attack rate of jamming.

6 Conclusions

In this paper, we have formulated a computation offloading game against attacks in MIMO systems, in which SU chooses the MEC server from the set of not being occupied by PU, offloading rate and transmission power and the attacker selects the action of attack. The attack types includes spoofing, jamming and eavesdropping. Besides, the object is to optimize the utility and performance of SU. Then, a Q-learning algorithm based optimal offloading strategy is proposed, under the condition of dynamic environment with unknown channel information. Simulation results show that our proposed scheme can improve the capacity of SU and efficiently decrease the attack rate of spoofing, jamming and eavesdropping.

References

1. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-art and research challenges. J. Internet Serv. Appl. **1**(1), 7–18 (2010)

2. Ahmed, A., Ahmed, E.: A survey on mobile edge computing. In: 2016 10th International Conference on Intelligent System & Control Intelligent Systems and Control (ISCO), Coimbatore, India, pp. 1–2 (2016)
3. Mao, Y., You, C., Zhang, J., et al.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017)
4. Li, Y., Li, Q., Liu, J., et al.: Mobile cloud offloading for malware detections with learning. In: IEEE International Conference on Computer Communications (INFOCOM), BigSecurity, Hongkong (2015)
5. Wan, X., Sheng, G., Li, Y., et al.: Reinforcement learning based mobile offloading for cloud-based malware detection. In: IEEE Global Communications Conference (GLOBECOM), Singapore (2017)
6. Duan, L., Gao, L., Huang, J.: Cooperative spectrum sharing: a contract-based approach. IEEE Trans. Mob. Comput. **13**(1), 174–187 (2014)
7. Xie, C., Xiao, L.: User-centric view of smart attacks in wireless networks. In: IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB), invited talk, Nanjing, China (2016)
8. Xiao, L., Xu, D., Xie, C., et al.: Cloud storage defense against advanced persistent threats: a prospect theoretic study. IEEE J. Sel. Areas Commun. **35**(3), 534–544 (2017)
9. Abass, A., Xiao, L., Mandayam, N.B., et al.: Evolutionary game theoretic analysis of advanced persistent threats against cloud storage. IEEE Access **5**, 8482–8491 (2017)
10. Xiao, L., Wan, X., Dai, C., et al.: Security in mobile edge caching with reinforcement learning. IEEE Wirel. Commun. Mag. **25**(3), 116–122 (2018)
11. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
12. Xiao, L., Chen, T., Han, G., et al.: Game theoretic study on channel-based authentication in MIMO systems. IEEE Trans. Veh. Technol. **66**(8), 7474–7484 (2017)
13. Li, Y., Xiao, L., Dai, H., et al.: Game theoretic study of protecting MIMO transmission against smart attacks. In: IEEE International Conference on Communications (ICC), Paris (2017)
14. Wang, F., Xu, J., Wang, X., et al.: Joint offloading and computing optimization in wireless powered mobile-edge computing. IEEE Trans. Wirel. Commun. **17**(3), 1784–1797 (2018)
15. Min, M., Xu, D., Xiao, L., et al.: Learning-based computing offloading for IoT devices with energy harvesting. IEEE Trans. Veh. Technol. **68**(2), 1930–1941 (2019)