Adaptive Honeypot Defense Deployment: A Stackelberg Game Approach with Decentralized DRL for AMI Protection

Abdullatif Albaseer, Moqbel Hamood, Mohamed Abdallah
Division of Information and Computing Technology, College of Science and Engineering,
Hamad Bin Khalifa University, Doha, Qatar
{aalbaseer, moha19838, moabdallah}@hbku.edu.qa

***Abstract*—Honeypot defenses are pivotal for safeguarding the Industrial Internet of Things (IIoT), notably the Advanced Metering Infrastructure (AMI), against cyber threats. The success of AMI defense relies on the strategic deployment of small-scale power suppliers (SPSs) and their interaction with traditional power retailers (TPR). Existing methods require exhaustive information exchange, which is not feasible. Prior studies also neglected the competitive aspect among the SPSs in task allocation. Our paper introduces a Stackelberg game model to address TPR-SPS interactions and SPS competition comprehensively. Our proposed approach stands out by eliminating the need for prior deployment and data sharing. It leverages the multiagent deep deterministic policy gradient (MADDPG) algorithm, centralized training, and distributed execution, effectively adapting to changing environments. Without relying on historical data, each SPS actively learns from its surroundings. Our simulations validate the efficiency of this novel approach.**

***Index Terms*—Smart Grids, AMI, DRL, Honeypots Deployment, Stackelberg Game, MADDPG**

I. Introduction

The smart grid (SG) power system is advancing to accommodate the requirements of the industrial Internet of Things (IIoT), such as smart homes [1]. Households, by installing wind turbines, have evolved into small-scale power suppliers (SPSs), reducing the load on traditional power retailers (TPRs) by injecting energy into the grids [2]. This has broadened the reach of advanced metering infrastructure (AMI), an SG cornerstone, across both SPSs and TPRs.

The intricate and interconnected architecture of SG and the integration of online devices render it susceptible to many cyber threats. Such vulnerabilities may emanate from legacy software or systems [3], [4]. The inherent complexities of SGs pose challenges to promptly detecting and mitigating these potential threats. Given these vulnerabilities, the entire power system's integrity becomes precariously balanced. Consequently, there is a compelling necessity to employ proactive defensive measures, such as deploying honeypots. Honeypots strategically entice potential attackers, serving as decoy systems and facilitating the assessment of their methodologies and tactics while safeguarding pivotal assets [5].

Within the domain of SG, the strategic placement of honeypots within the SPS infrastructure is imperative for achieving thorough defense coverage. Consequently, TPRs must offer proper incentives to SPSs to facilitate honeypot deployment, thereby enabling a collaborative paradigm to enhance grid security [6]. The reward design proposed for the SPSs should be proportional to their respective contributions, considering the caliber and magnitude of the data shared. However, given the prevalent information asymmetry, TPRs grapple with delineating appropriate rewards, especially in light of potential data misrepresentation by SPSs [7].

The literature contains various techniques to encourage end-users for honeypot deployment [6], [8]. For instance, [9], [10] utilized contract-theory-based methods for data relay incentives, while [2] focused on a contract-based incentive for direct energy trading in SGs. A recent approach by Tian et al. [11] considered information asymmetry for honeypot incentives. However, these methodologies overlook the specific nature of SGs, especially the complexities tied to SG services and protocols. An individual SPS cannot host honeypots for every service (i.e., communication protocols within SGs) due to resource constraints. Specifically, the heavy dependence on extensive data sharing, overlooking SPS inter-competition, and inadequate attention to historical data and edge device constraints in security protocols are evident. Recognizing this, TPRs ought to pinpoint specific services for honeypot deployment, subsequently choosing qualified SPSs based on their traffic volume.

Motivated by these observations, our work introduces a novel solution addressing these challenges, accounting for TPR-SPS interactions, resource constraints, and the heterogeneity in SG communication protocols. Our approach, formulated as a multistage Stackelberg game, emphasizes a strategic collaboration between TPR and SPSs, relying on a multiagent deep deterministic policy gradient (MADDPG) methodology. Key contributions include:

- A Stackelberg game-based approach encapsulating the dynamics between TPR and SPSs, factoring in resource constraints and communication diversities.
- An adaptive strategy that minimizes dependence on prior information, aligning with edge device capabilities.

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

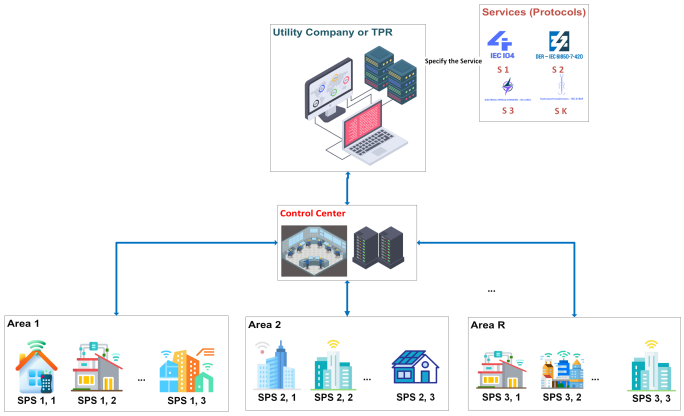


Fig. 1: Micro-grid Architecture for the considered system model

- Employing MADDPG, a responsive deep reinforcement learning (DRL) strategy adjusted to environmental shifts and uncertainties.
- A self-learning aspect, negates the need for prior training data, with each SPS acting as an independent learning entity.
- Validated efficacy of our methodology through simulations, emphasizing its practical viability.

The forthcoming sections provide an in-depth exploration of our proposed system model, problem formulation, and solutions, followed by performance evaluation and conclusion.

II. System Model and Problem Formulation

As shown in Fig. 1, an AMI consists of a utility company or TPR and multiple K SPSs where $\mathcal{K} = \{1, 2, \ldots, K\}$ are distributed in different areas or regions $\mathcal{R} = \{1, 2, ..., R\}$. Initially, we specify the TPR and SPS models and show the interaction procedure. The SPSs deploy the honeypots to capture network traffic cyberattacks, wrap the defense data, and then send it back to TPR via the network. Then, the SPSs receive either a monetary payment, a reduction in their bill, or an upgrade to their defense model. It is worth mentioning that the data shared by SPS is an information asymmetry, as the SPSs can realize their valid defense data (VDD), whereas the TPR cannot. The VDD (unknown attack interaction log) is stored in the database along with existing detection methods and typical defense logs. This information is continuously used to enhance the security of the AMI system by implementing new defense mechanisms.

Due to the limitation in the resources equipped with the edge devices in SG networks (i.e., smart meter and IEDs) and the several services used for communication between different entities in SG (i.e., GOOSE, R-GOOSE, Distributed Energy Resources – IEC 61850-7-420, Electrical vehicle charging – IEC 61851, Power electronics for electrical transmission and distribution systems – IEC 61954, etc.), a single SPS can not deploy a honeypot for all services. Therefore, the utility company or TPR should identify the specific services they aim to improve their defense models and then choose the most suitable SPS to run a single service or several services. Specifically, the TPR specify the services $\mathcal{N} = \{1, 2, \ldots, N\}$. For each service, a predetermined number of SPSs is selected. Each SPS $k \in \mathcal{K}$ has limited resources to run the honeypot and report the collected logs. We assume that the TPR can ask the SPSs to run the honeypot for a specific time to minimize the TPR's cost. We can summarize the interactions between the SPSs and TPR as follows:

- As shown in Fig. 1, the TPR or utility company determines the service (i.e., protocol) that needs to collect information about it, the time, and the allocated budget.
- The control center then announces to all SPSs in different regions.
- Each SPS determines its price and time to run the honeypot and upload the collected logs.
- The control center selects the best candidates to perform the deployment task, collects the logs and sends them back to the TPR.

A. Stackelberg game-based model

Game theory has been investigated as a pertinent tool for investigating decentralized decision-making among strategic players in various optimization problems [12]. One of the essential tools is the Stackelberg game, a new tool for simulating interactions between leaders and followers. It is widely used for optimization problems, such as computation offloading, resource trading, and energy transfer, to maximize or minimize utility between leaders and followers. In particular, players in the Stackelberg game model are leaders and followers. The leaders devise a strategy, and the followers act in accordance with it. Leaders (i.e., the TPR) and followers (i.e., the SPSs) seek to maximize their own rewards and utility in the game. The Stackelberg game can be modeled as a multi-stage game in which various decision-makers make decisions at different stages simultaneously. In our proposed approach, the TPR finds the optimal strategy to maximize its utility. Then, the SPSs account for observing the TPR's strategy and maximizing their utilities. Considering the competition amongst all SPSs, the optimum reaction of every SPS is determined using a MADDPG algorithm. Specifically, each SPS interacts with its environment, trying to learn a policy that maximizes its long-term profits with no prior knowledge about the actions of others. In our scenario, we can define the Stackelberg game as follows: First, the TPR and the SPSs are **player**s in which the TPR is the leader, and the SPS are the followers. Second, for the adopted **strategy**, the SPS's strategy is to specify the cost based on the honeypot deployment costs, while the TPR's strategy aims to select the best SPSs based on its budget. It is worth noting that if the cost is high, the TPR may decide to upgrade the model through security retailers. Third, for the

utility, the functions for SPSs, TPR, and AMI are explained in the following subsections.

1) SPS's Utility

For each service n, the time purchased from the SPS to run the honeypot is denoted by τ_k^n. Then, the utility of each SPS can be given as follows:

$$\varphi_k\left(p_k, \boldsymbol{\tau}_k\right) = p_k \sum_{n\in\mathcal{N}} \tau_k^n - c_k^n,\ \forall k \in \mathcal{K}, \tag{1}$$

where p_k is the price determined by SPS k, $\boldsymbol{\tau}_k$ is the total time spent running the honeypot for all services where $\boldsymbol{\tau}_k = \{\tau_k^n\}_{n\in\mathcal{N}}$, and c_k^n is the cost of running the honeypot for service n. As seen in the following sections, each SPS should ensure a positive utility.

2) TPR's Utility

Accordingly, we can calculate the utility of the TPR for each service as follows:

$$\Phi_n\left(\boldsymbol{\tau}_n, \boldsymbol{c}\right) = \theta_n\left(\sum_{k\in\mathcal{K}} \vartheta_k^n \tau_k^n\right) - \sum_{k\in\mathcal{K}} p_k \tau_k^n,\ \forall n \in \mathcal{N} \tag{2}$$

where $\boldsymbol{\tau}_n = \{\tau_k^n\}_{k\in\mathcal{K}}$, $\boldsymbol{c} = \{p_k\}_{k\in K}$, ϑ_k^n denotes the quality of data uploaded by SPS k for service n and θ_n is the utility function, showing the quality of collected data for each service. It is worth mentioning that θ_n is a concave function showing an increasing return rate as quality-weighted collecting time increases.

3) AMI welfare

Finally, as the TPR and SPSs compose the AMI, its welfare [13] is the summation utilities of both, which can be given by:

$$\begin{aligned}\Phi\left(\boldsymbol{\tau}, \boldsymbol{c}\right) &= \sum_{k\in\mathcal{K}} \varphi_k\left(p_k, \boldsymbol{\tau}_k\right) + \sum_{n\in\mathcal{N}} \Phi_n\left(\boldsymbol{\tau}_n, \boldsymbol{c}\right) \\ &= \sum_{n\in\mathcal{N}} \theta_n\left(\sum_{k\in\mathcal{K}} \vartheta_k^n \tau_k^n\right)\end{aligned} \tag{3}$$

B. Problem Formulation

In this paper, we formulate our main problem as a set of optimization challenges, encompassing the AMI welfare optimization, the TPR optimization, and the SPS optimization problems. First, the AMI aims to maximize the total social welfare payoff (i.e., total utility for the TPRs and SPSs). Given the cost $\boldsymbol{c}$, the problem can be posted as:

$$\mathbf{P1}: \max_{\boldsymbol{\tau}} \Phi\left(\boldsymbol{\tau}, \boldsymbol{c}\right) \tag{4}$$

subject to.

$$C1.1: \sum_{n\in\mathcal{N}} \tau_k^n \le t_k,\ \forall k \in \mathcal{K} \tag{5}$$

$$C1.2: \sum_{k\in\mathcal{K}} p_k \tau_k^n \le b_n,\ \forall n \in \mathcal{N} \tag{6}$$

$$C1.3: \sum_{k\in\mathcal{K}} \tau_k^n > 0 \tag{7}$$

$$C1.4: t_k > 0,\ \forall k \in \mathcal{K} \tag{8}$$

$$C1.5: \tau_k^n > 0,\ \forall k \in \mathcal{K} \tag{9}$$

$$C1.6: b_n > 0,\ \forall n \in \mathcal{N} \tag{10}$$

where t_k is the time in which the SPS k is available to perform the deployment tasks, and b_n is the budget adopted for each service n. $C1$ ensures that the time dedicated for all services by SPS k does not exceed its time budget, while in $C2$, the amount of purchased time for all SPSs for any given service has to be aligned with the allocated budget. The constraint, $C3$, ensures that at least one service is assigned per SPS. The constraints from $C4$ to $C6$ are non-negative constraints for t_k, τ_k, and b_n, respectively. It is worth noting that **P1** can be reformulated to maximize the utility of each TPR as follows:

$$\mathbf{P2}: \max_{\boldsymbol{\tau_n}} \Phi_n\left(\boldsymbol{\tau}_n, \boldsymbol{c}\right) \tag{11}$$

subject to.

$$C2.1: \tau_k^n \le t_n,\ \forall k \in \mathcal{K} \tag{12}$$

$$C2.2: \sum_{k\in\mathcal{K}} p_k \tau_k^n \le b_n \tag{13}$$

$$C1.3 - C1.5 \quad \text{in} \quad \mathbf{P1} \tag{14}$$

Now, we present the optimization problem on the SPS side in which each SPS selects the best policy considering the TPR's strategy. Mathematically speaking, each SPS aims to maximize its profit by choosing the best price as follows:

$$\mathbf{P3}: \max_{p_k} \varphi_k\left(p_k, \boldsymbol{\tau}_k\right) \tag{15}$$

s.t.

$$C3.1: p_k \ge 0. \tag{16}$$

$$C3.2: \sum_{n\in\mathcal{N}} c_k^n \le \Omega_k \tag{17}$$

where the $\mathcal{N}$ is a set of services that the SPS k is willing to run, Ω_k is the resource capabilities.

As we can see, **P2** and **P3** can be solved as optimization problems if all prior information is available. However, it is difficult to determine the optimal prices beforehand, and the quality of traffic data collected may vary due to the constantly changing network environment. Additionally, ensuring fair rewards for all participating SPSs is crucial. These challenges make it essential to use DRL to tackle these challenges effectively.

III. DRL-BASED FRAMEWORK

Utilizing DRL, the framework aids the SPSs in optimizing utility based on feedback from their environment. Meanwhile, the AMI welfare relies on implicit DRL feedback. In this system, agents (TPRs and SPSs) make decisions depending on the observed experiences of other entities. At every time interval t, agents make decisions to adapt to environmental shifts. The overarching objective is to bolster both TPR and

SPS utility by refining their decision-making strategies over time.

For each player $k \in K$, observations $O_k(t)$ form part of state S_t at each time slot t. The subsequent actions the players take give rise to rewards from the environment. The goal is to develop a policy maximizing long-term rewards dependent on state spaces and actions. We define the state space, action space, and reward as:

- **State Space:** Denoted as $S = \{S_1, S_2, \ldots, S_K\}$, with each S_K being the local state of a given SPS k.

$$o_k^t = \{p_k^{t-1}, \boldsymbol{\tau}_k^{t-1}, \ldots, p_k^{t-T}, \boldsymbol{\tau}_k^{t-T}\}. \tag{18}$$

- **Action Space:** At state s_t, the appropriate action a_t is chosen.

$$a_k^t = \mu_k \left(o_k^t | \boldsymbol{\theta}_k^{\mu}\right). \tag{19}$$

- **Reward Function:** For each agent, it is defined as:

$$r_k^t = \log\left(1 + p_k^t \sum_{n \in \mathcal{N}} \tau_{kn}^t - c_{kn}^t \sum_{n \in \mathcal{N}} \tau_{kn}^t\right). \tag{20}$$

A. MADDPG Technique

Given the complex environment, the MADDPG approach, emphasizing centralized training with decentralized execution, is implemented. This method modifies the actor-critic mechanism employing the Deep Q-Network (DQN) paradigm.

1) Actor Network

The actor-network is designed to maximize expected rewards. Here, the actor-network policies are parameterized by $\omega^{\mu} = \{\omega_1^{\mu}, \omega_2^{\mu}, \ldots, \omega_K^{\mu}\}$.

2) Critic Network

The critic network evaluates agent actions in relation to the expected future rewards, incorporating all states and actions.

$$\mathcal{L}_k = \frac{1}{T} \sum_{t}^{T} (y_k^t - Q_k(\boldsymbol{o}^t, \boldsymbol{a}^t | \boldsymbol{\omega}^{Q_k}))^2, \tag{21}$$

3) Experience Replay Buffer

Each agent possesses its local experience replay buffer, storing tuples $\{s_k, a_k, r_k, s_k'\}$. Policy training uses mini-batch sampling from this buffer, updating both the actor and critic networks.

IV. Performance Evaluation

A. Simulation Setup

We consider an AMI with 1 TPR, 9, 10, 15, and 20 SPSs disseminated in different regions, one control center to coordinate between the TPR and SPS, and 2, 3, 4, and 5 targeted services. The data quality of each SPS is randomly distributed. Each SPS aims to maximize its profit, so the reward discount factor is set to 0. We use one input layer, 5 hidden layers, and one output layer for the actor and critic networks. For the activation function, we use Relu for all hidden layers for both actor and critic, and only the Tanh is used for the output layer of the actor-network.

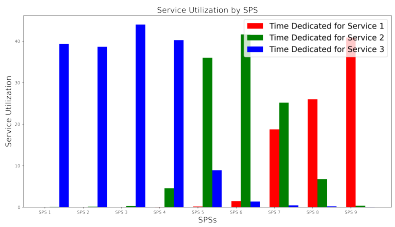


Fig. 2: Time Allocated for each service amongst all SPSs

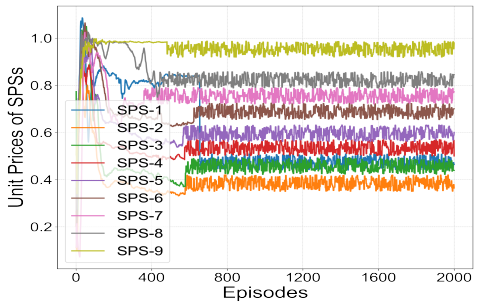
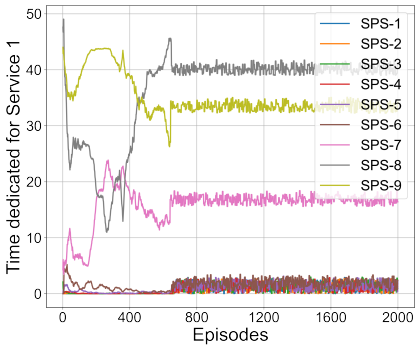


Fig. 3: The Prices Given by SPSs

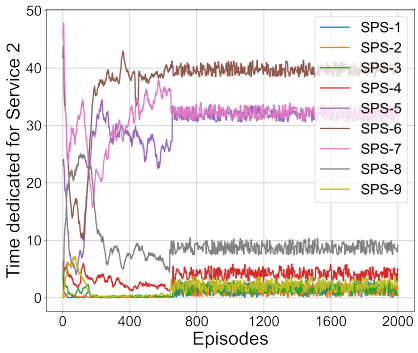
B. Numerical Results

As shown in Fig. 2, the services are assigned to all SPSs based on their prices. Some SPSs are selected for only one service. For example, SPS-1, SPS-2, and SPS-3 are assigned only for service 3. On the other hand, SPS-4 to SPS-8 are selected for two services. Fig. 3 also shows the prices each SPS charged to join the services. The SPSs set for multiple services charge higher prices, while the SPSs selected for only one service charge lower prices. The allocation of services to SPSs is influenced by the prices that the SPSs charge to join. SPSs that charge higher prices are more likely to be selected for multiple services, while SPSs that charge lower prices are more likely to be selected for only one task. This suggests that pricing can effectively allocate tasks to devices in a decentralized deployment system. It is worth mentioning that the quality of the reported data plays an essential role for each SPS selected to run the honeypot for a specific service where we set the SPSs to report less quality for service 1, and the system learns the optimal policy, not to select those devices.

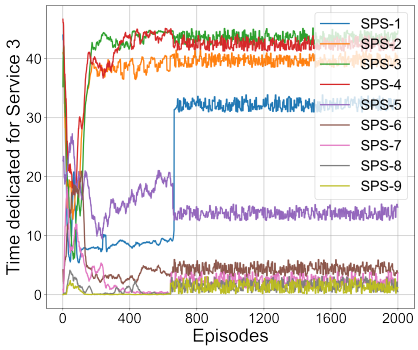
Figs. 4a–4c show the time allocated for each SPS deploying honeypots of different protocols (i.e., services) and sharing the defense data, which mainly depend on the local data quality (i.e., weights) and the prices. The significance here refers to the value or importance of the SPS in the SG network. Therefore, an SPS with a higher weight collects more valuable data and provides better insights into attack patterns. The higher the importance of the SPS, the more influential the honeypot is at catching attacks' patterns. However, the budget allocated to each service also plays a significant role in the quality of the data collected and the profits gained. We also note that



(a) Time allocated for Service 1



(b) Time allocated for Service 2



(c) Time allocated for Service 3

Fig. 4: The time allocated for each service, showing the participation of each SPS and the dedicated time.

the higher budget allows more SPSs to participate, resulting in better data quality. As a result, the appropriate weights for each device and allocating sufficient budgets to each service are critical to achieving optimal outputs when deploying honeypots across different SPSs, running other protocols, and sharing the related defense data. This enables TPR to obtain valuable insights into the attack patterns, detect new threats, and prevent potential cyber-attacks.

V. Conclusion

This paper introduced a Stackelberg game to model TPR and SPSs interactions in SG networks, addressing device limitations and communication protocol diversity. Our proposed approach is robust, needing no prior knowledge of SPSs' deployment or sharing decisions. Using MADDPG with centralized training but distributed execution, our method captures the dynamic environment efficiently. Crucially, each SPS learns in real-time without needing historical data. Simulations confirm the efficacy of our approach in guiding SPSs on honeypot deployment and log-sharing based on their resources and traffic.

Acknowledgement

This publication was made possible by NPRP Cluster project (NPRP-C) Twelve (12th) Cycle grant # NPRP12C-33905-SP-67 from the Qatar National Research Fund (a member of Qatar Foundation). The findings herein reflect the work, and are solely the responsibility of the authors.

References

[1] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, "Smart grid metering networks: A survey on security, privacy and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.

[2] T. Morstyn, A. Teytelboym, and M. D. McCulloch, "Bilateral contract networks for peer-to-peer energy trading," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2026–2035, 2018.

[3] W. Chen, D. Ding, H. Dong, and G. Wei, "Distributed resilient filtering for power systems subject to denial-of-service attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1688–1697, 2019.

[4] D. Du, X. Li, W. Li, R. Chen, M. Fei, and L. Wu, "Admm-based distributed state estimation of smart grid under data deception and denial of service attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 8, pp. 1698–1711, 2019.

[5] P. H. Mirzaee, M. Shojafar, H. Cruickshank, and R. Tafazolli, "Smart grid security and privacy: From conventional to machine learning issues (threats and countermeasures)," *IEEE Access*, 2022.

[6] K. Wang, M. Du, S. Maharjan, and Y. Sun, "Strategic honeypot game model for distributed denial of service attacks in the smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2474–2482, 2017.

[7] R. Zhang and Q. Zhu, "`FlipIn` : A game-theoretic cyber insurance framework for incentive-compatible cyber risk management of internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2026–2041, 2020.

[8] W. Tian, X.-P. Ji, W. Liu, J. Zhai, G. Liu, Y. Dai, and S. Huang, "Honeypot game-theoretical model for defending against apt attacks with limited resources in cyber-physical systems," *Etri Journal*, vol. 41, no. 5, pp. 585–598, 2019.

[9] Y. Zhang, L. Song, W. Saad, Z. Dawy, and Z. Han, "Contract-based incentive mechanisms for device-to-device communications in cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2144–2155, 2015.

[10] Y. Zhang, "Contract theory framework for wireless networking," Ph.D. dissertation, 2016.

[11] W. Tian, M. Du, X. Ji, G. Liu, Y. Dai, and Z. Han, "Contract-based incentive mechanisms for honeypot defense in advanced metering infrastructure," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4259–4268, 2021.

[12] Y. Zhan, C. H. Liu, Y. Zhao, J. Zhang, and J. Tang, "Free market of multi-leader multi-follower mobile crowdsensing: An incentive mechanism design by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 10, pp. 2316–2329, 2020.

[13] Z. Chen, T. Ni, H. Zhong, S. Zhang, and J. Cui, "Differentially private double spectrum auction with approximate social welfare maximization," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 11, pp. 2805–2818, 2019.