Faster Learning and Adaptation in Security Games by Exploiting Information Asymmetry

Xiaofan He, *Student Member, IEEE*, Huaiyu Dai, *Senior Member, IEEE*, and Peng Ning, *Senior Member, IEEE*

***Abstract*—With the advancement of modern technologies, the security battle between a legitimate system (LS) and an adversary is becoming increasingly sophisticated, involving complex interactions in unknown dynamic environments. Stochastic game (SG), together with multi-agent reinforcement learning (MARL), offers a systematic framework for the study of information warfare in current and emerging cyber-physical systems. In practical security games, each player usually has only incomplete information about the opponent, which induces information asymmetry. This paper exploits information asymmetry from a new angle, considering how to exploit information unknown to the opponent to the player's advantage. Two new MARL algorithms, termed minimax post-decision state (minimax-PDS) and Win-or-Learn Fast post-decision state (WoLF-PDS), are proposed, which enable the LS to learn and adapt faster in dynamic environments by exploiting its information advantage. The proposed algorithms are provably convergent and rational, respectively. Also, numerical results are presented to show their effectiveness through three important applications.**

***Index Terms*—Cloud computing, cognitive radio, energy harvesting, jamming, reinforcement learning, security, stochastic game.**

I. Introduction

While bringing unprecedented convenience and productivity to our life, information systems have also exhibited vast vulnerabilities for the adversaries to explore, leaving its security an ever present concern. In addition, with the advancement of information processing and electronic technologies, the security battle in modern cyber-physical systems is becoming increasingly sophisticated, in that both the defender and the attacker can and should take a tactical analysis of the opponent and determine the responding strategies accordingly; this in turn leads to the wide applications of game theory in network security studies recently [1], [2].

However, classical game theory and direct application of it usually assume a static environment, while the arms race between the defender and attacker in practice often takes place in (unknown) dynamic environments. Variations in wireless environments and available communication and computational resources are possible sources of such dynamics. Take the emerging technologies, energy harvesting (EH) [3], cognitive radio (CR) [4], and cloud computing [5], as examples: besides the frequently changing channel conditions, the battery level of an EH communication system (EHCS) will change over time depending on the random energy harvesting and expending processes; the spectrum availability of a secondary user (SU) in a CR system may vary due to dynamic primary user (PU) and SU activities; the amount of computational resource offered by a cloud server to a specific client may also be influenced by more urgent demands from other clients. This suggests that, in practical applications, legitimate systems (LS) have to not only meticulously deal with the intelligent adversary but also carefully accommodate their strategies in accordance to these environmental dynamics on-the-fly, so as to ensure normal system operation. With this consideration, a *dynamic game* formulation of the competitions between the LS and the attacker is a more appealing choice as compared to the static ones assumed by classical game theory, and is expected to facilitate the design and analysis of the strategic interactions between the attacker and defender in current and emerging cyber-physical systems.

Stochastic game (SG) [6], an extension of the Markov decision process (MDP) to the multi-player setting, is a good fit for the problem we consider. In a general SG, both the LS and the adversary can employ multi-agent reinforcement learning (MARL) algorithms [6] to gradually learn good (ideally optimal) strategies with respect to long-term goals through trial-and-error interactions with both the opponent and the unknown dynamic environment. Although SG together with MARL can address the environmental dynamics in security games in a systematic manner, there is a notable mismatch between their underlying modeling and practice: Conventional MARL algorithms usually assume that both the LS and the adversary are equally knowledgeable about the ongoing competitions, while in practical security games, each player usually has only incomplete information about the opponent, which induces *information asymmetry*. A Bayesian approach has been taken in literature (e.g., [7]) to deal with the unknown information at the opponent, where it is usually assumed that environmental dynamics are known. Nonetheless, another aspect of information asymmetry, how to exploit local information unknown to the opponent to the player's advantage, seems unexplored in literature. For example, an EHCS preserves information about its energy harvesting process; a SU is often aware of its data arrival statistics and transmission schedules; and a cloud client keeps its own statistics about the cloud resource availability. In practice such

Manuscript received January 23, 2015; revised September 1, 2015; accepted March 14, 2016. Date of publication March 31, 2016; date of current version May 20, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Eduard Jorswieck. This work was supported in part by the National Science Foundation under Grants CNS-1016260, ECCS-1307949, and EARS-1444009. Part of this work was presented at the IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, April 2015.

X. He and H. Dai are with the Department of Electrical and Computer Engineering, North Carolina State University, NC 27695 USA (e-mail: xhe6@ncsu.edu; Huaiyu_Dai@ncsu.edu).

P. Ning is with the Department of Computer Science, North Carolina State University, NC 27695-7911 USA (e-mail: pning@ncsu.edu).

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

Digital Object Identifier 10.1109/TSP.2016.2548987

information is unlikely to be known accurately by the adversary. This observation motivates the new algorithms developed in this work.

In this work, two new MARL algorithms, termed minimax post-decision state (minimax-PDS) and Win-or-Learn Fast post-decision state (WoLF-PDS), are proposed, which enable the defense system to *learn and adapt faster* in unknown dynamic environments by exploiting its information advantage. The contributions of this work are highlighted as follows:

1) To the best of our knowledge, the proposed algorithms are the first to explore such information asymmetry for faster learning in SG settings.
2) The proposed learning algorithms are general and admit various applications. Anti-jamming problems in EHCS and CR systems are demonstrated as concrete examples. In addition, as compared to the conference version of this work [8], the security game in the emerging paradigm of security as a cloud service is considered as another application of the proposed algorithms.[1] To our knowledge, we are among the first to consider the security game between a cloud-based defender and an intelligent attacker in such a paradigm.
3) The proposed minimax-PDS and WoLF-PDS are provably convergent and rational, respectively.

The remainder of this paper is organized as follows. Section II first presents an abstract security game model under the SG framework and then introduces the basics of conventional MARL algorithms. The proposed algorithms and their applications in EHCS, CR and cloud-based security systems are presented in Sections III and IV, respectively. Simulation results are presented in Section V, and related works are discussed in Section VI. Section VII concludes the paper.

II. Problem Formulation and Background

A. Problem Formulation

In the context of SG [6], an abstract security game between an LS and an attacker (opponent) can be characterized by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, R^O \rangle$, where $\mathcal{S}$ stands for the state space (e.g., the channel states); $\mathcal{A}$ denotes the action space (e.g., the transmit powers and channels to select) of the LS, while $\mathcal{O}$ denotes that of the attacker (which is the opponent from the LS's perspective); $R : \mathcal{S} \times \mathcal{A} \times \mathcal{O} \to \mathbb{R}$ represents the (random) reward function (e.g., the throughput) of the LS, which (randomly) maps the current state $s \in \mathcal{S}$ and the joint actions of the LS and attacker $(a, o) \in \mathcal{A} \times \mathcal{O}$ into a real number $R(s, a, o) \in \mathbb{R}$, while R^O denotes that of the attacker; and $T : \mathcal{S} \times \mathcal{A} \times \mathcal{O} \to p(\mathcal{S})$ is the state transition function that maps the state-action tuple $(s, a, o) \in \mathcal{S} \times \mathcal{A} \times \mathcal{O}$ into a distribution $p(s'|s, a, o)$ of the future state $s' \in \mathcal{S}$. As shown in Fig. 1, the LS adopts MARL to learn a policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that specifies the probability $\pi(s, a)$ of taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$, such that a long-term performance objective is maximized in the dynamic environment. Usually the long-term objective is given in the form of an expected cumulative discounted reward $\mathbb{E}\{\sum_{n=0}^{\infty} \beta^n R(s_n, a_n, o_n)\}$ with discounting factor $\beta \in [0, 1)$, and similar descriptions also apply to the attacker. At each time step n, both the LS and the attacker observe the current state $s_n \in \mathcal{S}$ and take actions $a_n \in \mathcal{A}$ and $o_n \in \mathcal{O}$, respectively, according to their own learned policies π_n and π_n^O.[2] Then the LS receives an immediate reward r_n which is a realization of the random reward function $R(s_n, a_n, o_n)$ corresponding to the state-action pair (s_n, a_n, o_n) (and $r_n = R(s_n, a_n, o_n)$ when the reward function R is deterministic). To capture the attacker's intention of degrading the LS's reward and without loss of generality, the common zero-sum assumption is adopted throughout this work, which assumes that the attacker's reward is the opposite of the LS's reward (i.e., $R^O = -R$ and $r_n^O = -r_n$). Meanwhile, the environment transits to a new state $s_{n+1} \in \mathcal{S}$ according to the (often *unknown*) state transition function T.

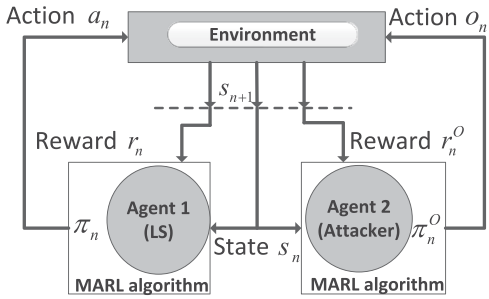


Fig. 1. Block diagram of a stochastic security game.

B. Conventional MARL

Minimax-Q [9] is a well-known MARL algorithm for policy learning in SG. In this algorithm, the optimal quality function $Q_*(s, a, o)$ of a state-action pair (s, a, o) for the LS is defined as the total expected discounted reward attained by taking action a, given current state s and opponent action o, and then following the optimal policy from then on. It satisfies the following Bellman optimality equation[3,4]

$$Q_*^{(m)}(s, a, o) \triangleq \mathbb{E}_{R,S'}\left[R(s, a, o) + \beta V_*^{(m)}(S')\right], \quad (1)$$

where the value function of a state s for the LS is defined based on the minimax principle as

$$V_*^{(m)}(s) \triangleq \max_{\pi(s)} \min_{o} \sum_{a} Q_*^{(m)}(s, a, o)\, \pi(s, a). \quad (2)$$

[1] Besides the application in cloud-based security systems, performance evaluation of two interesting scenarios, minimax-PDS LS vs. WoLF attacker and WoLF-PDS LS vs. minimax-Q attacker, is added (cf. Figs. 8 and 9) along with corresponding discussions. More technical details are provided as well, including Appendix A and C and some enhancement in the proof of Proposition 1 in Appendix B.

[2] Perfect monitoring, i.e., both players can perfectly observe each other's action, is assumed throughout this work.

[3] In this work, superscripts (m), (w), (mp) and (wp) will be used to distinguish similar quantities in the conventional minimax-Q and WoLF, and the proposed minimax-PDS and WoLF-PDS algorithms, respectively.

[4] Note that, for a given state-action pair (s, a, o), the corresponding reward $R(s, a, o)$ of the LS is a random variable. In (1), the capital letter S' denotes the random variable for the future state, and $\mathbb{E}_{R,S'}$ denotes the expectation against the probability measure generated by the random variables $R(s, a, o)$ and S'. Similar notations will be used throughout this work.

Note that $\pi(s) \triangleq [\pi(s,a)]_{a\in\mathcal{A}}$ is used in (2) to denote the probability distribution over the action set $\mathcal{A}$ at state s corresponding to a policy π, and similar notations will be used in the rest part of this work. Based on the quality function, the minimax optimal policy for the LS at each state s can be found by

$$\pi_*^{(m)}(s) = \arg\max_{\pi(s)} \min_{o} \sum_{a} Q_*^{(m)}(s,a,o)\,\pi(s,a). \quad (3)$$

The corresponding quantities of the attacker are defined similarly by replacing R and π with R^O and π^O, respectively, and switching a and o in the above expressions.

The minimax-Q algorithm enables the LS and attacker to learn the optimal quality functions and policies gradually. In particular, after the n-th round of interaction, the LS updates its quality function with a learning rate $\alpha_n \in (0,1)$ by

$$Q_{n+1}^{(m)}(s,a,o) = \begin{cases} (1-\alpha_n)Q_n^{(m)}(s,a,o) + \alpha_n\left[r_n + \beta V_n^{(m)}(s_{n+1})\right], & \text{for } (s,a,o) = (s_n,a_n,o_n), \\ Q_n^{(m)}(s,a,o), & \text{otherwise.} \end{cases} \quad (4)$$

The corresponding updated value function $V_{n+1}^{(m)}$ and policy $\pi_{n+1}^{(m)}$ can be computed by replacing $Q_*^{(m)}$ with $Q_{n+1}^{(m)}$ in the right hand side (RHS) of (2) and (3), respectively. The learning procedure of the attacker is similar. This minimax-Q iteration ensures that $Q_n^{(m)}$ ($\pi_n^{(m)}$) converges to the $Q_*^{(m)}$ ($\pi_*^{(m)}$) of the SG; but the minimax principle in (3) is conservative and may misguide the agent to blindly learn the worst case policy and cause performance loss. Intuitively, this may be best illustrated using the example given in [10]: Consider an opponent in a rock-paper-scissors game playing almost exclusively Rock, but playing Paper and Scissors with some small probability. Minimax-Q will converge to an equilibrium solution which randomizes among each of its three actions equally likely, but this is not a best response (playing only Paper in this situation is the only best response). Formally, this characteristic is known as irrationality, and the corresponding definition is given below [6], [10].

Definition 1: A MARL algorithm is said to be *rational* if it converges to a best response (i.e., an optimal policy that maximizes the expected reward) when the opponent plays a stationary policy (i.e., a policy that does not change over time).

To overcome the irrationality issue, a rational MARL algorithm, called Win-or-Learn Fast (WoLF), was developed in [10], with the penalty of loosing convergence assurance. As indicated by its name, the WoLF algorithm updates the policy using a slow (fast) learning parameter δ_{win} (δ_{lose}) when winning (losing). In particular, it is assumed that the agent is winning (losing) when its current policy provides larger (smaller) expected reward than an empirical average policy (defined later in Section III.B). In addition, the optimal quality and value functions for the LS in the WoLF algorithm are defined as

$$Q_*^{(w)}(s,a) \triangleq \mathbb{E}_{O,R,S'}\left[R(s,a,O) + \beta \cdot V_*^{(w)}(S')\right], \quad (5)$$

$$V_*^{(w)}(s) \triangleq \max_a Q_*^{(w)}(s,a). \quad (6)$$

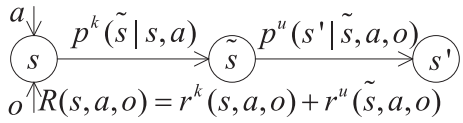


Fig. 2. Multi-agent PDS-learning.

The corresponding quantities for the attacker can be obtained similarly by switching the roles of the two.

It can be observed from (5) (with expectation over O) that, the WoLF algorithm includes the average effect of the opponent's current action into the reward, instead of the presumed worst-case effect from the opponent as in minimax-Q. The corresponding quality function can be updated through learning as

$$Q_{n+1}^{(w)}(s,a) = \begin{cases} (1-\alpha_n)Q_n^{(w)}(s,a) + \alpha_n[r_n + \beta V_n^{(w)}(s_{n+1})], & \text{for } (s,a) = (s_n,a_n), \\ Q_n^{(w)}(s,a), & \text{otherwise,} \end{cases} \quad (7)$$

while the updated value function $V_{n+1}^{(w)}$ can be obtained by replacing $Q_*^{(w)}$ with $Q_{n+1}^{(w)}$ in the RHS of (6).

C. Limitation of Conventional MARL

As will be exemplified in Section IV, the LS often holds certain extra information (e.g., statistics of its own communication and computational resources and schedules) as compared to the attacker in practical attack/defense games. However, none of the existing MARL algorithms can exploit such information advantage for performance improvement, even though it is highly desirable.

III. Multi-Agent PDS Learning

In this section, two multi-agent post-decision state (PDS) learning algorithms, termed minimax-PDS and WoLF-PDS, are developed to enable an agent to learn and adapt faster in unknown dynamic environments when extra partial information is available. The analysis is given in the context of a general SG, and specific applications will be discussed in Section IV.

In the proposed multi-agent PDS-learning, it is assumed that after taking action a at state s given the opponent's action o, the PDS-learning agent (i.e., the agent having extra information) will first receive a known reward $r^k(s,a,o)$ and the state transits to an intermediate state $\tilde{s}$, termed post-decision state, with a known probability $p^k(\tilde{s}|s,a,o)$, which is unknown to the opponent; then the state further transits to a future state s' with an unknown probability $p^u(s'|\tilde{s},a,o)$, and a reward $r^u(\tilde{s},a,o)$ that depends on the random PDS $\tilde{s}$ is received.[5] Particularly, it is assumed that the future state s' is independent of the current state s given the PDS $\tilde{s}$ and that the reward can be decomposed into the sum of known and unknown rewards at the post-decision state and the next state, respectively. This process is illustrated in Fig. 2. Mathematically, the transition from s to s' admits

$$p(s'|s,a,o) = \sum_{\tilde{s}} p^u(s'|\tilde{s},a,o)\,p^k(\tilde{s}|s,a,o), \quad (8)$$

[5]Note that, in this work, it is assumed that the randomness in $R(s,a,o)$ is solely due to the PDS $\tilde{s}$ and both r^k and r^u are deterministic functions, which hence are denoted by lower case letters.

and it can be verified that the expected reward of the state-action pair (s,a,o) is given by

$$\mathbb{E}_{\tilde{S}}[R(s,a,o)] = r^k(s,a,o) + \sum_{\tilde{s}} p^k(\tilde{s}|s,a,o) r^u(\tilde{s},a,o). \tag{9}$$

A. Minimax-PDS

The basic idea of the proposed minimax-PDS is that, instead of directly updating a single quality function at a time, an agent with the extra information on $p^k(\tilde{s}|s,a,o)$ and $r^k(s,a,o)$ can first learn the PDS quality function $\tilde{Q}^{(\mathrm{mp})}$ defined below and then simultaneously update multiple quality functions.

In particular, the optimal PDS quality function $\tilde{Q}^{(\mathrm{mp})}_*$ for the post-decision state-action pair $(\tilde{s},a,o)$ is defined as

$$\tilde{Q}^{(\mathrm{mp})}_*(\tilde{s},a,o) \triangleq r^u(\tilde{s},a,o) + \beta \sum_{s'} p^u(s'|\tilde{s},a,o) V^{(\mathrm{mp})}_*(s'), \tag{10}$$

and the optimal quality and value functions $Q^{(\mathrm{mp})}_*$ and $V^{(\mathrm{mp})}_*$ in the proposed minimax-PDS are identical to $Q^{(m)}_*$ and $V^{(m)}_*$ defined in (1) and (2), respectively. Using the extra information $p^k(\tilde{s}|s,a,o)$ and $r^k(s,a,o)$, it can be shown (cf. Appendix A) that $Q^{(\mathrm{mp})}_*$ can be further expanded, for all state-action pairs (s,a,o), as

$$Q^{(\mathrm{mp})}_*(s,a,o) = r^k(s,a,o) + \sum_{\tilde{s}} p^k(\tilde{s}|s,a,o) \tilde{Q}^{(\mathrm{mp})}_*(\tilde{s},a,o). \tag{11}$$

In the proposed minimax-PDS algorithm, after observing the sample $\left(s_n, a_n, o_n, r^k(s_n,a_n,o_n), \tilde{s}_n, r^u(\tilde{s}_n,a_n,o_n), s_{n+1}\right)$, the PDS-learning agent first updates the PDS quality function $\tilde{Q}^{(\mathrm{mp})}$ by

$$\begin{aligned}\tilde{Q}^{(\mathrm{mp})}_{n+1}(\tilde{s}_n,a_n,o_n) &= (1-\alpha_n)\tilde{Q}^{(\mathrm{mp})}_n(\tilde{s}_n,a_n,o_n) \\ &\quad + \alpha_n[r^u(\tilde{s}_n,a_n,o_n) + \beta \cdot V^{(\mathrm{mp})}_n(s_{n+1})],\end{aligned} \tag{12}$$

and $\tilde{Q}^{(\mathrm{mp})}_{n+1}(\tilde{s},a,o) = \tilde{Q}^{(\mathrm{mp})}_n(\tilde{s},a,o)$ for $(\tilde{s},a,o) \neq (\tilde{s}_n,a_n,o_n)$.

After obtaining $\tilde{Q}^{(\mathrm{mp})}_{n+1}$, $Q^{(\mathrm{mp})}_{n+1}$ can be updated using the RHS of (11) by replacing $\tilde{Q}^{(\mathrm{mp})}_*$ with $\tilde{Q}^{(\mathrm{mp})}_{n+1}$. Note that in the conventional minimax-Q, at each interaction, only a single entry (s_n,a_n,o_n) of the quality function is updated using (4); while in contrast, all entries are updated in (11),[6] and hence the learning speed is substantially accelerated. The corresponding $V^{(\mathrm{mp})}_{n+1}$ and $\pi^{(\mathrm{mp})}_{n+1}$ can be updated by replacing $Q^{(m)}_*$ with $Q^{(\mathrm{mp})}_{n+1}$ in the RHS of (2) and (3), respectively, for *all* states s.

The convergence property of the proposed minimax-PDS is given by the following proposition.

Proposition 1: Using minimax-PDS, $Q^{(\mathrm{mp})}_n$ converges to the minimax optimal quality function $Q^{(\mathrm{mp})}_*$ with probability 1 when the learning rate sequence α_n satisfies the conditions $\alpha_n \in [0,1)$, $\sum_{n=0}^{\infty}\alpha_n = \infty$ and $\sum_{n=0}^{\infty}\alpha_n^2 < \infty$.[7]

Proof: Please see Appendix B. ■

However, as will be shown in Section V, the minimax-PDS inherits the unfavorable irrationality property from the minimax-Q. To address this, a rational multi-agent PDS learning algorithm, termed WoLF-PDS, is developed in the next subsection.

B. Wolf-PDS

WoLF-PDS is developed by incorporating the PDS-learning principle into the WoLF algorithm. As compared to minimax-PDS, the following extra assumption is required in WoLF-PDS:

$$p^k(\tilde{s}|s,a,o) = p^k(\tilde{s}|s,a), \tag{13}$$

which essentially requires that the transition from the current state s to the PDS $\tilde{s}$ is independent of the opponent's action o.[8] The optimal quality and value functions in the proposed WoLF-PDS $Q^{(\mathrm{wp})}_*$ and $V^{(\mathrm{wp})}_*$ are defined identically to $Q^{(w)}_*$ and $V^{(w)}_*$, respectively, as in (5) and (6). The PDS quality function $\tilde{Q}^{(\mathrm{wp})}_*(\tilde{s},a)$ in the WoLF-PDS algorithm is defined as

$$\tilde{Q}^{(\mathrm{wp})}_*(\tilde{s},a) \triangleq \mathbb{E}_O\left[r^u(\tilde{s},a,O) + \beta \sum_{s'} p^u(s'|\tilde{s},a,O) V^{(\mathrm{wp})}_*(s')\right]. \tag{14}$$

Using the extra information and assumption (13), it can be shown (cf. Appendix C) that $Q^{(\mathrm{wp})}_*(s,a)$ can be further expanded as

$$Q^{(\mathrm{wp})}_*(s,a) = \mathbb{E}_O[r^k(s,a,O)] + \sum_{\tilde{s}} p^k(\tilde{s}|s,a) \tilde{Q}^{(\mathrm{wp})}_*(\tilde{s},a). \tag{15}$$

In the proposed WoLF-PDS learning, after observing the sample $(s_n,a_n,o_n,r^k(s_n,a_n,o_n),\tilde{s}_n,r^u(\tilde{s}_n,a_n,o_n),s_{n+1})$, the agent first updates the PDS quality function $\tilde{Q}^{(\mathrm{wp})}$ using

$$\begin{aligned}\tilde{Q}^{(\mathrm{wp})}_{n+1}(\tilde{s}_n,a_n) = \ & (1-\alpha_n)\tilde{Q}^{(\mathrm{wp})}_n(\tilde{s}_n,a_n) + \alpha_n[r^u(\tilde{s}_n,a_n,o_n) \\ & + \beta \cdot V^{(\mathrm{wp})}_n(s_{n+1})],\end{aligned} \tag{16}$$

and $\tilde{Q}^{(\mathrm{wp})}_{n+1}(\tilde{s},a) = \tilde{Q}^{(\mathrm{wp})}_n(\tilde{s},a)$ for $(\tilde{s},a) \neq (\tilde{s}_n,a_n)$. In addition, it updates an empirical reward function $\overline{r}^k$ using

$$\overline{r}^k_{n+1}(s_n,a) = (1-\alpha_n)\overline{r}^k_n(s_n,a) + \alpha_n \cdot r^k(s_n,a,o_n), \tag{17}$$

for all actions a, and $\overline{r}^k_{n+1}(s,a) = \overline{r}^k_n(s,a)$ for $s \neq s_n$, so as to keep track of the empirical average performance. Then, the corresponding quality function $Q^{(\mathrm{wp})}_{n+1}$ is updated by replacing $\mathbb{E}_O[r^k(s,a,O)]$ and $\tilde{Q}^{(\mathrm{wp})}_*(\tilde{s},a)$ in the RHS of (15) with $\overline{r}^k_{n+1}(s,a)$ and $\tilde{Q}^{(\mathrm{wp})}_{n+1}(\tilde{s},a)$, respectively, for all state-action pairs (s,a).

After obtaining the quality functions, the rest steps of WoLF-PDS are similar to the original WoLF algorithm. In particular, the state occurrence count c, the empirical average policy $\overline{\pi}$ and the policy $\pi^{(\mathrm{wp})}$ (with initial values given in Algorithm 2) are

[6] Note that, when the state space is non-irreducible, only the reachable states will be updated by (11).

[7] Note that these requirements on α_n appear in most of the RL algorithms (e.g., MDP, minimax-Q and WoLF) and are not specific to the proposed algorithms.

[8] This assumption is valid in many practical applications as exemplified in Sections IV.A and IV.C.

Algorithm 1: Minimax-PDS Algorithm.

Initialization: $n = 1$, $Q^{(\mathrm{mp})} = \mathbf{0}$, $V^{(\mathrm{mp})} = \mathbf{0}$ and $\pi^{(\mathrm{mp})}$ uniform.

Taking action a_n at current state s_n

1) uniformly at random with probability p_{explor};
2) otherwise, with probability $\pi_n^{(\mathrm{mp})}(s_n, a_n)$.

Learning: after receiving a reward $r^k(s_n, a_n, o_n)$ and observing the state transition from s_n to $\tilde{s}_n$ and then to s_{n+1}

3) Update $\tilde{Q}^{(\mathrm{mp})}$ using (12);
4) Update $Q^{(\mathrm{mp})}$ using (11) (with $\tilde{Q}_*^{(\mathrm{mp})}$ replaced by the updated $\tilde{Q}^{(\mathrm{mp})}$);
5) Update $V^{(\mathrm{mp})}$ and $\pi^{(\mathrm{mp})}$ using (2) and (3) (with $Q_*^{(m)}$ replaced by the updated $Q^{(\mathrm{mp})}$), respectively.

Repeat.

Algorithm 2: WoLF-PDS Algorithm.

Initialization: $n = 1$, $Q^{(\mathrm{wp})} = \mathbf{0}$, $c = \mathbf{0}$, $\overline{r} = \mathbf{0}$, and $\pi^{(\mathrm{wp})}$ and $\overline{\pi}$ uniform.

Taking action a_n at current state s_n

1) uniformly at random with probability p_{explor};
2) otherwise, with probability $\pi_n^{(\mathrm{wp})}(s_n, a_n)$.

Learning: after receiving a reward $r^k(s_n, a_n, o_n)$ and observing the state transition from s_n to $\tilde{s}_n$ and then to s_{n+1}

1) Update $\tilde{Q}^{(\mathrm{wp})}$ using (16);
2) Update $\overline{r}^k$ using (17);
3) Update $Q^{(\mathrm{wp})}$ using (15) (with $\mathbb{E}_O\left[r^k(s, a, O)\right]$ and $\tilde{Q}_*^{(\mathrm{wp})}$ replaced by the updated $\overline{r}^k$ and $\tilde{Q}^{(\mathrm{wp})}$, respectively);
4) Update c using (18);
5) Update $\overline{\pi}$ and $\pi^{(\mathrm{wp})}$ using (19) and (20), respectively.

Repeat.

updated, respectively, as

$$c_{n+1}(s_n) = c_n(s_n) + 1, \tag{18}$$

$$\overline{\pi}_{n+1}(s_n, a) = \overline{\pi}_n(s_n, a) + \frac{\pi_n^{(\mathrm{wp})}(s_n, a) - \overline{\pi}_n(s_n, a)}{c_{n+1}(s_n)}, \quad \forall a. \tag{19}$$

$$\pi_{n+1}^{(\mathrm{wp})}(s_n, a) = \pi_n^{(\mathrm{wp})}(s_n, a) + \Delta_{\mathrm{sa}}, \quad \forall a, \tag{20}$$

where $\Delta_{\mathrm{sa}} = -\delta_{\mathrm{sa}}$ if $a \neq \arg\max_{a'} Q^{(\mathrm{wp})}(s, a')$ and $\Delta_{\mathrm{sa}} = \sum_{a \neq a'} \delta_{sa'}$ otherwise, with $\delta_{\mathrm{sa}} = \min\{\pi^{(\mathrm{wp})}(s, a), \frac{\delta}{|\mathcal{A}|-1}\}$ and $|\mathcal{A}|$ denoting the size of the LS's action space $\mathcal{A}$; intuitively, (20) moves the policy towards the highest valued action with speed controlled by the parameter δ. At each round, δ is determined by the WoLF principle. Particularly, $\delta = \delta_{\mathrm{win}}$ if $\sum_{a'} \pi_n^{(\mathrm{wp})}(s_n, a') Q_{n+1}^{(\mathrm{wp})}(s_n, a') > \sum_{a'} \overline{\pi}_n(s_n, a') Q_{n+1}^{(\mathrm{wp})}(s_n, a')$ (the winning condition) and $\delta = \delta_{\mathrm{lose}}$ $(> \delta_{\mathrm{win}})$ otherwise, where both δ_{win} and δ_{lose} vanish over (usually inversely proportionally to) time [10].

Through updating multiple $Q^{(\mathrm{wp})}$'s at a time by taking advantage of extra information, the proposed WoLF-PDS algorithm can substantially expedite the learning speed as compared to the original WoLF algorithm. In addition, it admits the favorable rationality property.

Proposition 2: The WoLF-PDS is rational when the sequence α_n satisfies the conditions in Proposition 1.

Proof: Please see Appendix D. ■

Nonetheless, similar to the conventional WoLF algorithm, the proposed WoLF-PDS has no convergence assurance in general SG.

In practice, an agent has to switch between taking actions uniformly at random (with probability p_{explor}) and following the learned policy (with probability $1 - p_{\mathrm{explor}}$), to ensure sufficient explorations of the underlying SG [6]. The corresponding algorithms of the proposed minimax-PDS and WoLF-PDS are given in Algorithm 1 and Algorithm 2, respectively.

IV. Applications in Practical Security Games

Three applications of the proposed algorithms are presented in this section to provide concrete instances for the general framework presented above. EH and CR technologies are promising for future communication networks as they improve the utilization efficiency of the scarce energy and spectrum resources, respectively. The first two examples illustrate how EHCS and CR systems can employ the proposed algorithms as effective defense against jamming attacks. Cloud based security service [11], [12] is another emerging application in which users can utilize cloud resource to fulfill various resource-demanding security applications. The third example demonstrates how the proposed algorithms can guide a cloud user to strategically utilize the dynamic cloud resource so as to conduct most effective security defense.

A. Anti-Jamming in EHCS

The first example considers the jamming/anti-jamming competition between an EHCS and a jammer (J) in a dynamic environment, where the channel power gain $h_n \in \mathcal{H} \triangleq \{h^{(1)}, \ldots, h^{(m)}\}$ varies over time following an Markov process with transition probability $p(h_{n+1} = h^{(j)} | h_n = h^{(i)}) = p_H(j|i)$ unknown to both the EHCS and the jammer. As illustrated in Fig. 3, the competition proceeds as follows: At each timeslot n, the EHCS chooses a suitable transmit power a_n, which is constrained by its current battery energy level $b_n \in \mathcal{B} \triangleq \{b^{(1)}, \ldots, b^{\max}\}$ with $b^{(1)} = 0$ and $b^{\max}$ the battery capacity. Without loss of generality, it is assumed that δb amount of new energy will be harvested by the EHCS after each transmission (with duration T_s), transiting its battery level from $b_n - a_n T_s$ to $b_{n+1} = \min\{b_n - a_n T_s + \delta b, b^{\max}\}$ with probability $p_{\mathrm{EH}}(b_{n+1} | b_n - a_n T_s)$. In addition, we assume that p_{EH} is known only to the EHCS itself, inducing *information asymmetry*. On the other hand, due to a per unit jamming power cost c_J, the smart jammer needs to select a proper jamming power $o_n \in \mathcal{O}$ for effective jamming. Considering the zero-sum

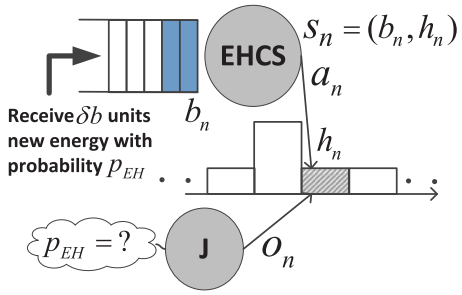


Fig. 3. The jamming/anti-jamming competition between an EHCS and a jammer.

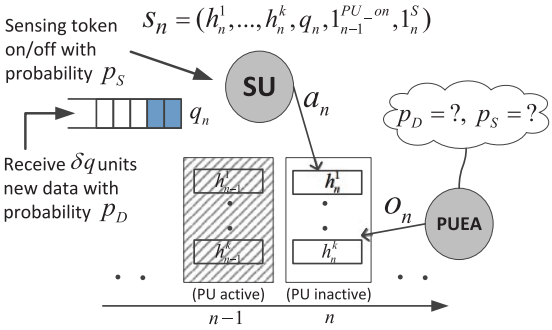


Fig. 4. The jamming/anti-jamming competition between a SU and a PUEA.

assumption, the total reward of the EHCS at each timeslot n is modeled as the sum of the throughput reward and the cost of the jammer, and is given by

$$r_n = \underbrace{h_n \cdot a_n/(o_n + N)}_{\text{signal-to-interference-plus-noise ratio}} + \underbrace{c_J \cdot o_n}_{\text{jamming cost}}, \quad (21)$$

where N is the noise power and a unit jamming channel power gain is assumed for simplicity. The jammer's reward is the opposite.

B. Anti-Jamming in CR Systems

The second example envisages the competition between a SU and a primary user emulation attacker (PUEA), as shown in Fig. 4. In the PUE attack, the adversary emits emulated PU signals to spoof SUs, aiming at obtaining exclusive access of the spectrum holes [13], and may be considered as a smart-type jammer to CR systems. Assume that the on/off state of the PU is Markovian, and the probability $\phi_{1,0}$ $(\phi_{0,1})$ that the PU (e.g., a TV tower) transits from the active (inactive) to inactive (active) state is known to both the SU and the PUEA [14]. The PU spectrum is divided into k sub-channels, and the channel power gain h^i of each sub-channel i independently varies among m different states $\mathcal{H}^i = \{h^{i,(1)}, \ldots, h^{i,(m)}\}$ in a Markovian manner with unknown statistics p_{h^i} to both the SU and the PUEA; also note that all the k sub-channels will be occupied when PU is active. At each timeslot n, a sensing token $\mathbf{1}_n^S \in \{0, 1\}$ indicating whether the SU can sense/transmit or not will be assigned to the SU, which captures, for example, the random access scenario in a CSMA based secondary network; the transition probability $p_S(\mathbf{1}_n^S | \mathbf{1}_{n-1}^S)$ of the sensing token is assumed known only to the SU. If an inactive sensing token $\mathbf{1}_n^S = 0$ is obtained, the SU will keep silent, denoted by $a_n = 0$, otherwise it will choose a target channel $i \in \{0, 1, \ldots, k\}$ for sensing, denoted by $a_n = i$ (including the option of keeping silent). If no PU signal is detected on the sensed channel, the SU will transmit on this channel, given that the current data queue length $q_n > 0$. At the same time, the PUEA will also either choose a target channel i' to emit faked PU signals, denoted by $o_n = i'$, or keep silent (to avoid a penalty c_p that reflects the risk of being detected by the PU when the real and the faked PU signals collide), denoted by $o_n = 0$. For simplicity, we assume that spectrum sensing is perfect but the SU cannot distinguish between the real and the faked PU signals. Also note that PUE attack must be launched at the SU sensing period to be effective. In each timeslot, $\Delta q_n = \mathbf{1}_n^S \cdot \sum_{i=1}^k \mathbf{1}_{\{a_n = i, o_n \neq i\}} \cdot \min\left\{q_n, C\left(h_n^i\right)\right\}$ amount of data can be successfully transmitted by the SU, where $\mathbf{1}_{\{\cdot\}} \in \{0, 1\}$ is the indicator function and $C\left(h_n^i\right) = B_i \log\left(1 + h_n^i \cdot P/N_i\right)$ is the capacity of the ith sub-channel with channel power gain $h_n^i \in \mathcal{H}^i$, bandwidth B_i, transmit power P and noise power N_i. In addition, δq units of new data will arrive at the SU, transiting the data queue state to $q_{n+1} = q_n - \Delta q_n + \delta q$ with probability $p_D\left(q_{n+1} | q_n - \Delta q_n\right)$, assumed known only to the SU, which, together with the knowledge on the sensing token transition probability p_S, induces *information asymmetry*. The expected reward of the SU is modeled as

$$r_n = \underbrace{\phi_{\mathbf{1}_{n-1}^{\mathrm{PU\_on}}, 0} \cdot \Delta q_n}_{\text{expected goodput}} - \underbrace{\mathbf{1}_n^S \cdot \mathbf{1}_{\{a_n > 0\}} \cdot c_s}_{\text{sensing cost}} + \underbrace{c_p \cdot \mathbf{1}_{\{o_n > 0\}} \cdot \phi_{\mathbf{1}_{n-1}^{\mathrm{PU\_on}}, 1}}_{\text{expected attack penalty}}, \quad (22)$$

where c_s is the sensing cost (e.g., due to circuit power consumption) of the SU; $\mathbf{1}_{n-1}^{\mathrm{PU\_on}} \in \{0, 1\}$ is the PU state in the previous timeslot $n-1$ which is assumed known to both the SU and the PUEA. The PUEA's reward is the opposite.

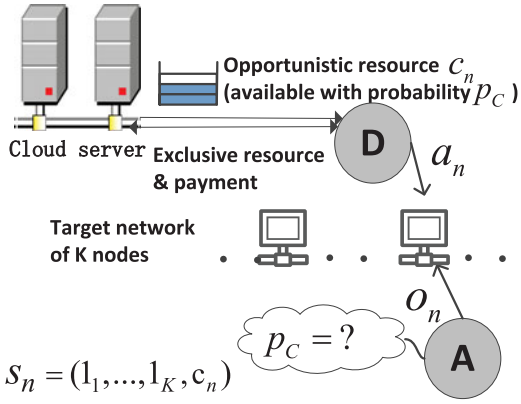


Fig. 5. The cloud-based security game.

C. Cloud-Based Security Game

The third example considers a cloud-based security game as depicted in Fig. 5. At each timeslot n, the defender (D) and

the attacker (A) choose a_n and o_n nodes in the target network to enforce security protection and to inject malware, respectively. To fulfill the security protection to these a_n nodes, the defender needs to request a_n units of computing resource from the cloud. As in [15], it is assumed in this work that the cloud provides both exclusive and opportunistic resources; the exclusive resource has a per unit price φ_a while the opportunistic resource is free and its availability follows a Markov process with transition probabilities p_C, which is assumed known only to the defender and induces *information asymmetry*. Denoting by $c_n \in \mathcal{C} = \{0, \ldots, c_{\max}\}$ the amount of opportunistic resource available at timeslot n, the payment from the defender will be $\varphi_a \cdot \max\{a_n - c_n, 0\}$. While for the attacker, a per node attacking cost φ_o is assumed. In this work, it is assumed that each of the K nodes in the target network can be in either a healthy or an infected state. In addition, it is assumed that when both the defender and the attacker act on the same node, this node will transit from infected (healthy) state to healthy (infected) state with probability p_{10} (p_{01}), which is unknown to both the defender and attacker[9]; when only the defender (attacker) acts on a node, this node will result in a healthy (infected) state; otherwise, the state of the node remains unchanged. Considering the possibility of malware spreading, it is further assumed that, after both the attacker and the defender take actions, a spreading phase will occur, in which, any healthy node may be infected by a infected node with unknown probability $p_{\inf}$.[10] In this security game, the defender aims at maximizing the number of healthy nodes with minimum payment and its instant reward at timeslot n is given by

$$r_n = (K - k_n) - \varphi_a \cdot \max\{a_n - c_n, 0\} + \varphi_o \cdot o_n, \quad (23)$$

where k_n denotes the number of infected nodes at timeslot n. The reward of the attacker is assumed to be the opposite.

D. Applications of the Proposed Algorithms

To apply the proposed algorithms, the EHCS anti-jamming problem is first formulated as a SG with the state defined as $s_n = (b_n, h_n)$ and the corresponding actions and rewards defined as in Section IV.A. In addition, the PDS can be defined as $\tilde{s}_n = (\tilde{b}_n, h_n)$ with $\tilde{b}_n \triangleq \min\{b_n - a_n T_s + \delta b, b^{\max}\} = b_{n+1}$, which is the system state after transmission and arrival of new energy. With this PDS definition, the corresponding known and unknown state transition probabilities are $p^k = p_{\rm EH}$ and $p^u = p_H$, respectively; the known reward r_n^k is given by (21) and $r_n^u = 0$.

For the SU anti-jamming problem, the state is defined as $s_n = (h_n^1, \ldots, h_n^k, q_n, \mathbf{1}_{n-1}^{\rm PU\_on}, \mathbf{1}_n^S)$, and the PDS is defined as $\tilde{s}_n = (h_n^1, \ldots, h_n^k, \tilde{q}_n, \mathbf{1}_{n-1}^{\rm PU\_on}, \widetilde{\mathbf{1}_n^S})$ with $\tilde{q}_n \triangleq q_n - \Delta q_n + \delta q = q_{n+1}$ and $\widetilde{\mathbf{1}_n^S} \triangleq \mathbf{1}_{n+1}^S$, which is the system state after the data transmission and the arrivals of new data and sensing token; accordingly, $p^k = p_D \cdot p_S$ and $p^u = \prod_{i=1}^k p_{h^i}$, and r_n^k is given by (22) and $r_n^u = 0$. Note that, since the transition from s_n to $\tilde{s}_n$ depends on the PUEA's action (which affects the successfulness of a SU transmission and its data queue state), (13) is violated. Therefore, WoLF-PDS cannot be applied and only minimax-PDS will be considered in this SU anti-jamming problem.

For the cloud-based security game, the state is defined as $s_n = (\mathbf{1}_1, \ldots, \mathbf{1}_K, c_n)$, where $\mathbf{1}_i = 0$ if the i-th node is healthy and otherwise $\mathbf{1}_i = 1$, and the corresponding PDS is defined as $\tilde{s}_n = (\mathbf{1}_1, \ldots, \mathbf{1}_K, \tilde{c}_n)$ with $\tilde{c}_n \triangleq c_{n+1}$. With these definitions, $p^k = p_C$, and p^u is determined by the probabilities of node state transition p_{01} and p_{10} and the probability of malware spreading $p_{\inf}$. Accordingly, the known part of the reward r_n^k is given by (23) and $r_n^u = 0$.

In our setting, the jammer (in the EHCS and CR applications) and the attacker (in the cloud-based security game) cannot employ the proposed minimax-PDS or WoLF-PDS due to lack of knowledge about $p_{\rm EH}$, p_D and p_S, and p_C. Although there may be scenario in which the attacker holds extra local information against the defender and can employ the proposed algorithms to assist its learning, this work focuses on applications in which the defender has extra local information. In the face of the same attacker, the performance gain at the defender side through the proposed PDS-learning will remain.[11] Also, to focus on the main theme of this work, we have made some assumptions about the underlying problems above for simplicity; some of them will be addressed in Section V while the others are left to future work.

V. Simulations

In this section, numerical results are presented to justify the effectiveness of the proposed algorithms. Specifically, the performance gain of using the proposed algorithms is measured by

$$\eta(n) \triangleq [\tilde{r}_{\rm PDS}(n) - \tilde{r}(n)]/\bar{\tilde{r}}(n) \times 100\%, \quad (24)$$

where $\tilde{r}_{\rm PDS}(n) \triangleq \frac{1}{n}\sum_{i=1}^n r(s_i, a_i, o_i)$ is the average accumulative reward [16] till timeslot n when the LS adopts PDS-learning; $\tilde{r}(n)$ is defined similarly when the LS adopts the conventional minimax-Q or WoLF algorithm; $\bar{\tilde{r}}(n)$ denotes the average of $\tilde{r}(n)$ over all Monte Carlo runs and serves as a normalization factor here. For learning speed comparison, the relative distance between the learned and the optimal quality functions Q_n and Q_*, defined as

$$\Delta Q_n \triangleq ||\mathbf{vec}(Q_n - Q_*)||_1 / ||\mathbf{vec}(Q_*)||_1 \times 100\%, \quad (25)$$

is the metric of interest, with $\mathbf{vec}(\cdot)$ the vectorization operator and $||\cdot||_1$ the 1-norm.[12] For minimax-Q and minimax-PDS the corresponding Q_* can be found by numerically computing the

[9]It is usually difficult to predict p_{10} and p_{01} beforehand, since they may depend on the effectiveness of the specific defense to the malware.

[10]In practice, $p_{\inf}$ is determined by the properties of both the malware and the target network, and thus, neither the defender nor the attacker can predict $p_{\inf}$ unilaterally.

[11]Although the jammer and the attacker may adopt model-learning approaches [6], in which an agent jointly estimates the unknown model and computes the strategy, it is beyond the scope of this work, since our main objective in this work is to improve the learning performance of model-free algorithms [6] (e.g., minimax-Q and WoLF) by exploiting information asymmetry. Also, model-learning approaches suffer from significant higher computation complexity (due to the costly fixed point evaluations at *every* interaction) as compared to the model-free ones (only using simple updates as (4)).

[12]Note that in a finite-dimensional vector space, convergences in 1-norm and max-norm are equivalent, but, in our view, 1-norm is more natural for relative distance comparison here.

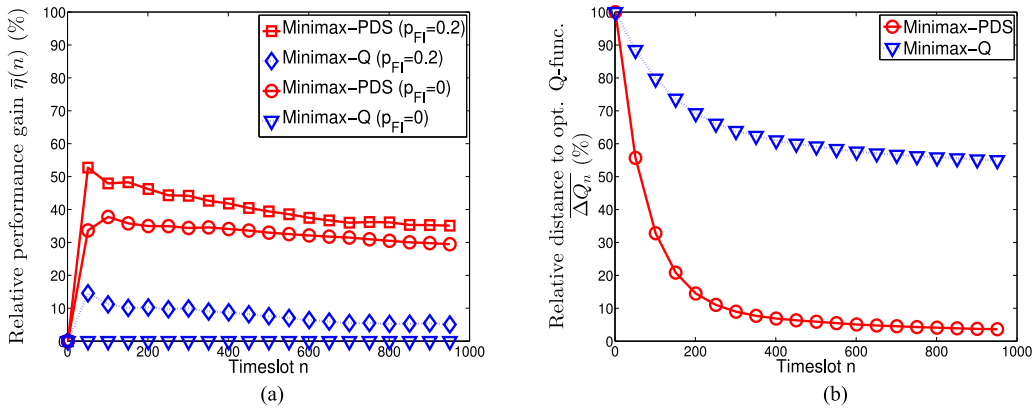


Fig. 6. Minimax-PDS vs. minimax-Q in scenario I. (a) Anti-jamming performance comparison. (b) Convergence performance comparison.

fixed point of (1) and (2); for WoLF and WoLF-PDS, Q_* in the stationary jammer case can be found by computing the fixed point of (5) and (6).[13] The learning rates α_n's in minimax-PDS and WoLF-PDS are set according to [9] and [10], respectively. All the average values are obtained through 100 Monte Carlo runs.

A. EHCS Anti-Jamming

For the EHCS anti-jamming application, the following scenario (scenario I) is considered. Specifically, the EHCS battery level varies among $\mathcal{B} = \{b^{(1)}, b^{(2)}, b^{(3)}\} = \{0, 5, 10\}$ and the corresponding transition probability matrix

$$p_{\mathrm{EH}} = \begin{bmatrix} 0.6 & 0.4 & 0 \\ 0 & 0.9 & 0.1 \\ 0 & 0 & 1 \end{bmatrix};$$

the channel power gain changes between two states $\mathcal{H} = \{h^{(1)}, h^{(2)}\} = \{1, 100\}$ with transition probability matrix

$$p_H = \begin{bmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \end{bmatrix};$$

the jammer chooses the jamming power from the action set $\mathcal{O} = \{0, 5, 10\}$; p_{explor} and the discounting factor β are set to 0.2 and 0.75, respectively; the transmission duration T_s and noise power N are set to 1 and 0.5, respectively; the jammer power cost per unit c_J is set to 1.

1) Minimax-PDS and WoLF-PDS: The performance of the proposed minimax-PDS and the conventional minimax-Q when the EHCS faces a minimax-Q jammer is examined first. Fig. 6(a) shows the average performance gain $\overline{\eta}(n)$ provided by the proposed minimax-PDS (curve Minimax-PDS ($p_{\mathrm{FI}} = 0$)) over the conventional minimax-Q (curve Minimax-Q ($p_{\mathrm{FI}} = 0$)), and Fig. 6(b) compares the learning speeds. It can be seen from Fig. 6(b) that the proposed minimax-PDS enables the EHCS to learn the minimax optimal policy substantially faster as compared to the conventional minimax-Q, which in turn leads to a significant average performance gain for the EHCS. For example, as shown in Fig. 6(b), even till $n = 800$, the average relative distance $\overline{\Delta Q_n}$ between Q_* and the $Q_n^{(m)}$ learned by the conventional minimax-Q algorithm is still above 50%; in contrast, the minimax-PDS favorably learns a $Q_n^{(\mathrm{mp})}$ with average relative distance around 3% to Q_*, and as a consequence, an average performance gain of 31% is exhibited in Fig. 6(a). In addition, it can be observed from Fig. 6(a) that minimax-PDS only benefits EHCS at the learning phase, and the corresponding gain decreases over time. The reason is that for fixed channel statistics p_H, both minimax-Q and minimax-PDS will eventually converge to the optimal policy and hence perform identically from then on. However, in practice, the channel statistics p_H will also vary over time, forcing both the EHCS and the jammer to stay frequently in the learning phase. In such cases, the minimax-PDS is apparently more favorable. The corresponding gain may be coined as the *hiding target defense gain*, as it is achieved by the uncertainty (hiding) of some information at the LS side.

In the above simulation, we give the jammer the privilege of perfectly observing the battery state b_n of the EHCS, which could be true in applications where the EHCS has to report its battery state to a control center and the information is intercepted by the jammer. In practice, the jammer may fail to obtain such information or the EHCS can deliberately send out falsified information with certain probability p_{FI}. In such cases, assuming that the jammer will take a conservative estimate of b_n as $b^{\max}$ whenever it fails to intercept b_n, the corresponding performance curves for $p_{\mathrm{FI}} = 0.2$ are also shown in Fig. 6(a). It can be seen that, a nonzero p_{FI} leads to extra anti-jamming gains for both minimax-Q and minimax-PDS EHCSs, since the jammer's learning process is disrupted by the false information. The advantage obtained through this type of information asymmetry may be termed *falsifying target defense gain*. A more comprehensive examination of the impacts of such falsified information is beyond the scope of this work, and $p_{\mathrm{FI}} = 0$ is assumed in the rest simulations for simplicity.

The performance of the WoLF-PDS EHCS against a WoLF jammer is shown in Fig. 7, and it can be seen that the proposed WoLF-PDS provides a significant anti-jamming performance gain (up to 92% at $n = 50$) to the EHCS as compared to the conventional WoLF. This can be explained by its capability

[13] For WoLF and WoLF-PDS with non-stationary jammers, opponent independent Q_* does not exist in general.

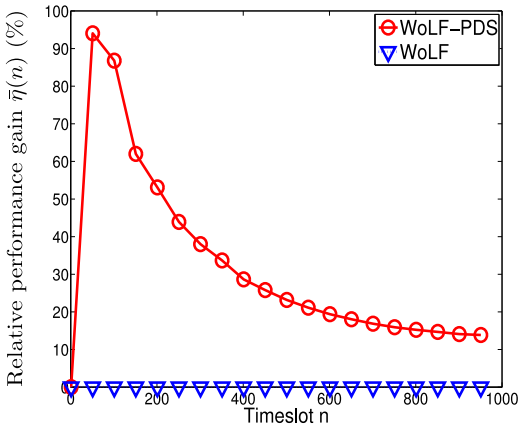


Fig. 7. WoLF-PDS vs. WoLF in scenario I.

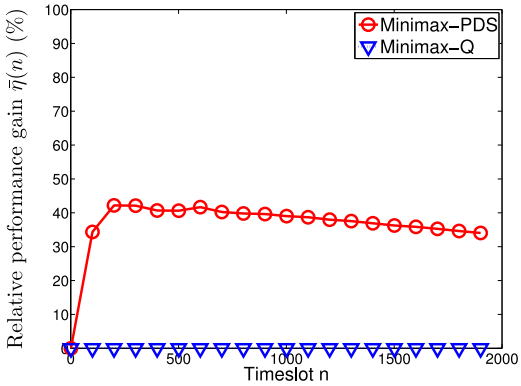


Fig. 8. Minimax-PDS vs. Minimax-Q in scenario I when facing a WoLF jammer.

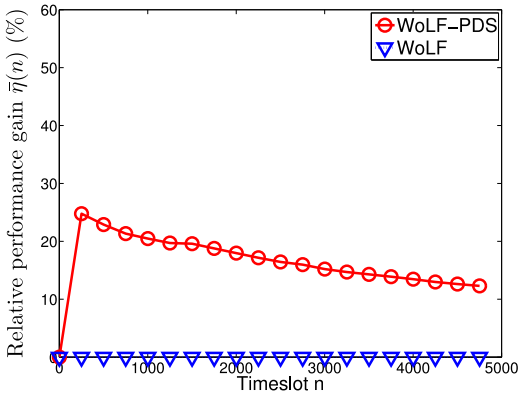


Fig. 9. WoLF-PDS vs. WoLF in scenario I when facing a Minimax-Q jammer.

of faster adaptation to the jammer. Since WoLF-PDS generally does not have convergence assurance against a dynamic opponent, e.g., the WoLF jammer considered here, its faster adaptation capability will be illustrated next, through stationary jammer examples.

Moreover, the proposed minimax-PDS also demonstrates similar advantages when facing a WoLF jammer, and so does the proposed WoLF-PDS when facing a minimax-Q jammer, as shown in Figs. 8 and 9, respectively. These results indicate that

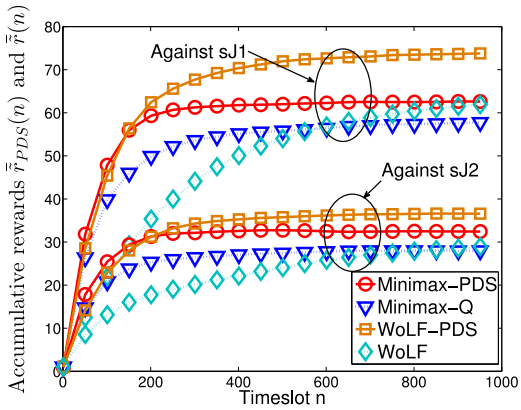


Fig. 10. Accumulative rewards of the proposed and conventional algorithms in scenario II.

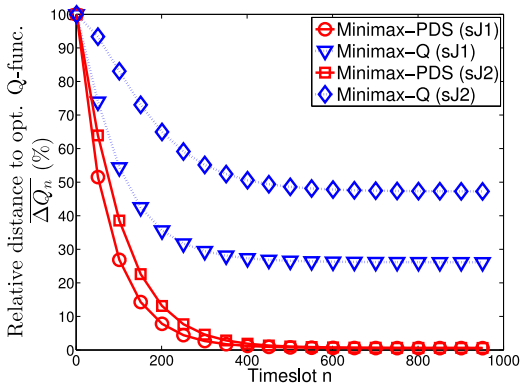


Fig. 11. Convergence of minimax-Q and minimax-PDS in scenario II.

the effectiveness of the proposed algorithms does not depend on the opponent's learning process.

2) Convergence vs. Rationality: Another scenario (scenario II) is deliberately chosen to discuss the convergence and rationality aspects of the proposed algorithms. Particularly, it is assumed that

$$p_{\mathrm{EH}} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathcal{H} = \left\{h^{(1)}, h^{(2)}\right\} = \{0.5, 40\},$$

$$p_H = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}, \qquad \beta = 0.3,$$

and other parameters remain the same as in scenario I. In addition, two stationary jammers, $sJ1$ and $sJ2$, are considered. Specifically, $sJ1$ adopts a stationary policy $\pi_{sJ1}(s) = [1/3, 1/3, 1/3]$ at all states s, i.e., taking a jamming action o from $\mathcal{O}$ with equal probability each time, while $sJ2$ adopts a more aggressive stationary policy $\pi_{sJ2}(s) = [0.1, 0.1, 0.8]$ at all states s.

The reward performance of the conventional and the proposed algorithms are compared in Fig. 10 when the EHCS faces $sJ1$ and $sJ2$, respectively. From Fig. 10, it can be observed that in both cases, the minimax-PDS does not provide as much anti-jamming gain as the WoLF-PDS, which is due to its *irrationality* property inherited from the conventional minimax-Q. In fact, the

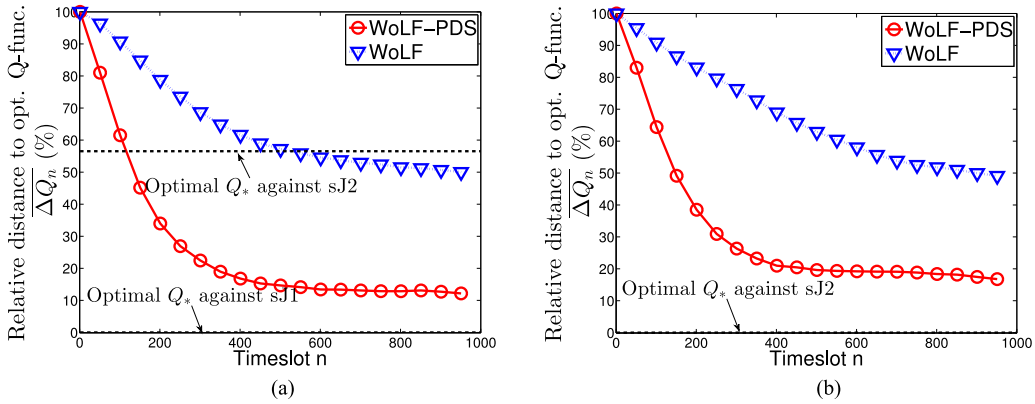


Fig. 12. Convergence of WoLF and WoLF-PDS in scenario II. (a) Against $sJ1$. (b) Against $sJ2$.

minimax principle forces the EHCS to take a conservative policy (best response to the worst opponent strategy) even when the opponents ($sJ1$ and $sJ2$) are actually not playing with the worst case policy. Nonetheless, it is worth noting that the minimax-PDS converges to the same minimax optimal quality function $Q_*^{(\mathrm{mp})}(s)$ in both cases as shown in Fig. 11, as promised by Proposition 1. In addition, the minimax optimality indicates that the jammer cannot further degrade the performance, and the minimax-PDS allows the EHCS to obtain such a stable policy more quickly as compared to minimax-Q.

Unlike the minimax-PDS that always converges to an opponent independent minimax optimal $Q_*^{(\mathrm{mp})}$, it can be seen from the trends in Fig. 12(a) and (b) that, the rational (cf. Definition 1) WoLF-PDS will converge to two different Q_*'s corresponding to the two different best responses against $sJ1$ and $sJ2$, respectively, as suggested by Proposition 2. (In Fig. 12(b), the optimal Q_* against $sJ1$ is out of the border.) In addition, the proposed WoLF-PDS offers substantially faster learning as compared to the conventional WoLF. When facing a dynamic jammer (as considered in scenario I), faster learning capability along with rationality allows more agile adaptation to the opponent's current policy and thus leads to better LS performance.

B. SU Anti-Jamming

For the SU anti-jamming application, it is assumed that the PU transition statistics are $\phi_{1,0} = \phi_{0,1} = 0.5$; the PU spectrum is divided into $k = 2$ sub-channels, and the first (second) channel varies between two power gains $\mathcal{H}^1 = \{h^{1,(1)}, h^{1,(2)}\}$ $\left(\mathcal{H}^2 = \{h^{2,(1)}, h^{2,(2)}\}\right)$ with transition probabilities

$$p_{h^1} = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix} \quad \left(p_{h^2} = \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix}\right)$$

and the corresponding channel capacities are $C\left(h^{1,(1)}\right) = 1$ and $C\left(h^{1,(2)}\right) = 2$ ($C\left(h^{2,(1)}\right) = 0$ and $C\left(h^{2,(2)}\right) = 1$); the data queue length $q_n \in \{0, 1, 2\}$ and corresponding transition matrix is

$$p_D = \begin{bmatrix} 0.1 & 0.8 & 0.1 \\ 0 & 0.2 & 0.8 \\ 0 & 0 & 1 \end{bmatrix};$$

the sensing token transition matrix

$$p_S = \begin{bmatrix} 0.2 & 0.8 \\ 0.5 & 0.5 \end{bmatrix};$$

and $c_s = 0.7$, $c_p = 2$, $p_{\mathrm{explor}} = 0.2$, and $\beta = 0.75$.

The performance of the proposed minimax-PDS and the conventional minimax-Q at a SU when facing a minimax-Q PUEA is compared in Fig. 13, similar to what is presented in Fig. 6 for the EHCS application. It can be seen from Fig. 13(b) that the SU learns the optimal policy much faster by using minimax-PDS, which in turn, results in a substantial performance gain (up to 45%) in rewards in the learning phase, as shown in Fig. 13(a). In addition, similar observation can be made as in the EHCS case when the SU sends out falsified information (with $p_{\mathrm{FI}} = 0.2$).

C. Cloud-Based Security Game

For the cloud-based security game, it is assumed that there are $K = 3$ nodes in the target network; the per node attacking cost is $\varphi_o = 0.5$ and the per unit exclusive cloud resource price is $\varphi_a = 1.5$; the probabilities of the node state transition and malware spreading are set to $p_{01} = p_{10} = 0.3$ and $p_{\mathrm{inf}} = 0.5$, respectively; the available opportunistic cloud resources change among the values $\mathcal{C} = \{0, 1, 2\}$ with transition probabilities

$$p_C = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix};$$

p_{explor} and β are set to 0.75 and 0.2, respectively. Note that, in the cloud-based security game, it is reasonable to assumed that both the node states and the amount of available opportunistic cloud resource are publicly known, and thus we set $p_{\mathrm{FI}} = 0$ in this example.

Similar to the previous two applications, the proposed minimax-PDS and WoLF-PDS again demonstrate significant performance gain over the conventional minimax-Q and WoLF algorithms, as shown in Figs. 14 and 15 respectively.

VI. Related Works

Game theory has been widely adopted in the network security study ([1], [2] and references therein), and both the

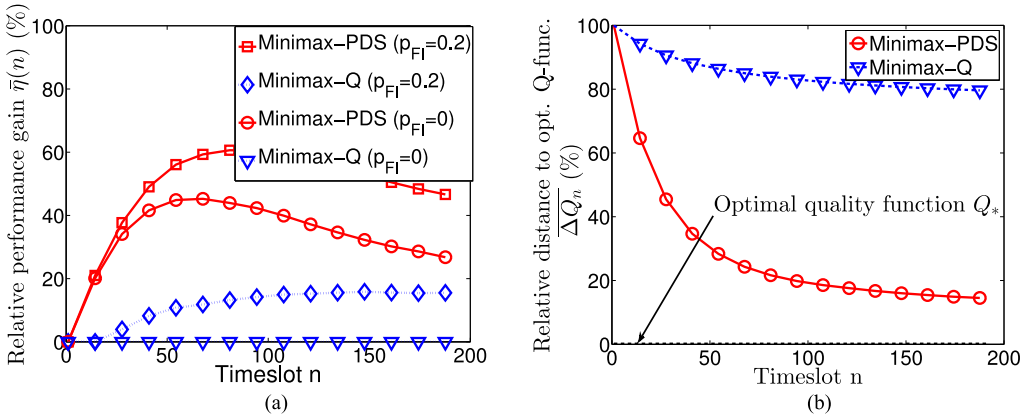


Fig. 13. Minimax-PDS vs. minimax-Q in CR anti-jamming application. (a) Anti-jamming performance comparison. (b) Convergence performance comparison.

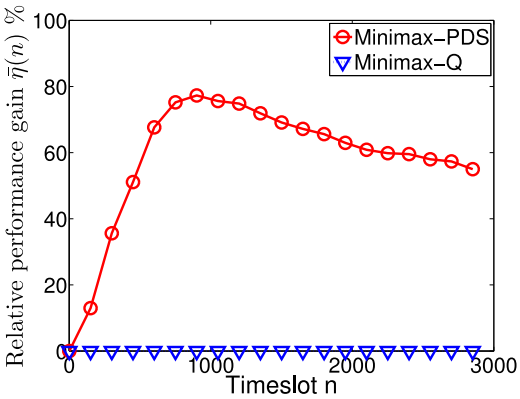


Fig. 14. Minimax-PDS vs. Minimax-Q in cloud-based security game (with a Minimax-Q attacker).

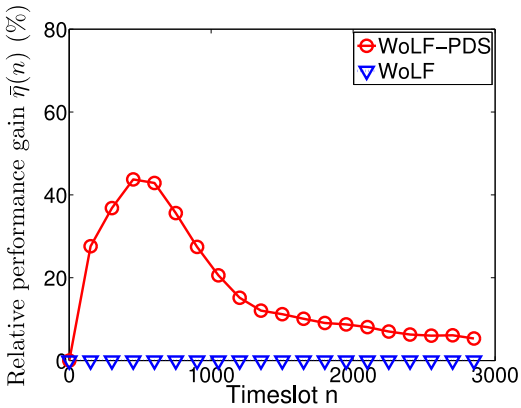


Fig. 15. WoLF-PDS vs. WoLF in cloud-based security game (with a WoLF attacker).

simultaneous-move game model (e.g., [17]) and Stackelberg game model (e.g., [18]) have been considered; but the majority only considers non-stochastic games. Security games with information asymmetry receive research interest recently. For example, in [7] and references therein, it is shown that incomplete information about the opponent will usually lead to performance degradation in jamming/anti-jamming games. In [19]–[21], [22], a Bayesian Stackelberg game framework is employed to guide security patrolling at the airport in the presence of adversary type uncertainty. Our work differs from these prior works in two aspects. First, these works adopt the classical game theory framework where a known environment is often assumed, while in our work, more complicated unknown dynamic environments are considered under the framework of stochastic games. Second, existing works on security games with information asymmetry mainly study the cases that a player (mostly the defender) has incomplete information about the opponent, and Bayesian approaches are usually taken to handle the involved uncertainties. Considering that a player may also hold some information advantage against the opponent, our work takes a new research angle to the existing literature and endeavors to address how such extra local information may be exploited by the player to gain advantage over the opponent. This work also complements our previous works [23], [24] on stochastic game with incomplete information.

Among the existing works adopting SG formulation and MARL, [16] and [25] are the most relevant to our work, where the minimax-Q and the WoLF algorithms were applied, respectively, to find anti-jamming strategies in multi-channel CR systems. In [16] and [25], the LS and the attacker are treated as two equally knowledgeable learning agents. When the LS lacks sufficient knowledge about the opponent, existing works either impose restrictions to the considered opponent models (e.g., [26], [27]) or treat the opponent and the dynamic environment as an integrated entity (e.g., [28]) and then further invoke MDP or multi-armed bandit (MAB) algorithms for strategy learning. Our work is orthogonal to these works and focuses on exploiting local information unknown to the opponent for performance improvement. In addition, our algorithms advance the pioneer works on single agent PDS-learning [29], [30], which expedites system learning speed in benign environments, to the more complicated multi-agent cases so that they can deal with intelligent adversaries. In [31] and [32], a non-zero-sum SG and a cooperative multi-agent system are considered respectively. In these two works, novel decomposition methods are used to transform the considered multi-agent SGs into multiple single-agent decision problems by exploiting the specific problem structures there, and then single-agent PDS-learning algorithms are employed to further improve learning speeds.

Moreover, the research on EHCS security and cloud-based security systems is still in the infancy, as compared to that on CR security [33]. Some pioneer works in EHCS security include the study of the secrecy data rate in [34] and [35], energy harvesting friendly jamming in [36], and the impact of energy harvesting on smart meter privacy in [37]; but there exists little work that can exploit unique features of the EHCS for anti-jamming performance enhancement, and our work contributes in this direction. As to the cloud-based security systems, existing works mainly focus on architectural designs and (proof-of-concept) implementations [11], [12], [38], [39], and our work is the first to consider security game in such systems.

VII. Conclusions and Future Works

Two new MARL algorithms, minimax-PDS and WoLF-PDS, that enable a game player to learn and adapt faster in unknown dynamic environments by exploiting information asymmetry between itself and the opponent, are proposed in this work, which are provably convergent and rational, respectively. Numerical results verify that the proposed algorithms can provide substantial performance improvement as compared to conventional ones. The proposed learning framework is general enough to admit wide applications.

An interesting direction for future work is to extend the current framework beyond the zero-sum setting, and incorporate the PDS-learning principle into other relevant MARL algorithms (such as Nash-Q and FoF-Q [6]), considering further applications in similarly structured problems. It is also interesting to consider stochastic security games where information asymmetry exists at both the defender and the attacker sides, and possible methods of increasing (reducing) information asymmetry against (about) the opponent.

Appendix A
Proof of (11)

Since $Q_*^{(\mathrm{mp})}$ is identical to $Q_*^{(m)}$ (cf. (1)) by definition, it admits the following expansion

$$\begin{aligned} Q_*^{(\mathrm{mp})}(s,a,o) &\triangleq \mathbb{E}_{R,S'}[R(s,a,o)+\beta V_*^{(m)}(S')] \\ &= \mathbb{E}_{\tilde{S}}[R(s,a,o)] + \beta\sum_{s'} p(s'|s,a,o)V_*^{(\mathrm{mp})}(s') \\ &= r^k(s,a,o) + \sum_{\tilde{s}} p^k(\tilde{s}|s,a)\Bigg[r^u(\tilde{s},a,o) \\ &\quad + \beta\sum_{s'} p^u(s'|\tilde{s},a,o)V_*^{(\mathrm{mp})}(s')\Bigg] \\ &= r^k(s,a,o) + \sum_{\tilde{s}} p^k(\tilde{s}|s,a)\tilde{Q}_*^{(\mathrm{mp})}(\tilde{s},a,o), \end{aligned} \tag{26}$$

where the second equality invokes the facts that the randomness of $R(s,a,o)$ is solely due to $\tilde{S}$ and that $V_*^{(\mathrm{mp})}$ is the same as $V_*^{(m)}$ by definition; the third equality follows from (8) and (9); the last one is by the definition of $\tilde{Q}_*^{(\mathrm{mp})}$ in (10).

Appendix B
Proof of Proposition 1

To streamline the proof for Propositions 1 (as well as Proposition 2), several relevant definitions and useful lemmas are introduced first.

Definition 2: The max-norm of an n-dimensional matrix $A=[a_{i_1,\ldots,i_n}]$ is defined as $||A||_\infty = \max_{i_1,\ldots,i_n} |a_{i_1,\ldots,i_n}|$, and unless otherwise noted, $||\cdot||_\infty$ represents the max-norm.

Definition 3: For any two $m\times n$ matrices $A=[a_{i,j}]$ and $B=[b_{i,j}]$, $A\geq B$ if $a_{i,j}\geq b_{i,j}$ for $i=1,\ldots,m$ and $j=1,\ldots,n$.

Lemma 1: (Szepesvari and Littman [40]) Assume a learning rate sequence α_n that satisfies $0\leq\alpha_n<1$, $\sum_{n=0}^{\infty}\alpha_n=\infty$ and $\sum_{n=0}^{\infty}\alpha_n^2<\infty$, and a sequence of (random) mappings $\mathcal{T}_n$ from $\mathbb{Q}$ to $\mathbb{Q}$ (with $\mathbb{Q}$ denoting the set of quality functions) that satisfies:

(C1) $\mathbb{E}[\mathcal{T}_n Q_*]=Q_*$;

(C2) There exist a number $0<\gamma<1$ and a positive sequence λ_n converging to zero with probability 1, such that $||\mathcal{T}_nQ-\mathcal{T}_nQ_*||_\infty\leq\gamma\cdot||Q-Q_*||_\infty+\lambda_n$, for all $Q\in\mathbb{Q}$.

Then the iteration defined by

$$Q_{n+1}(s,a,o)=\begin{cases}(1-\alpha_n)Q_n(s,a,o)\\ \quad+\alpha_n[(\mathcal{T}_nQ_n)(s,a,o)],\\ \text{if } (s,a,o)=(s_n,a_n,o_n),\\ Q_n(s,a,o), \quad \text{otherwise,}\end{cases}$$

converges to Q_* with probability 1.

Lemma 2: For any two quality functions (Q_1 and Q_2) and corresponding value functions (V_1 and V_2),

$$|V_1(s)-V_2(s)|\leq||Q_1(s)-Q_2(s)||_\infty. \tag{27}$$

Particularly, for any quality function Q in minimax-PDS, $Q(s)\triangleq[Q(s,a,o)]_{a\in\mathcal{A},o\in\mathcal{O}}$ is the $|\mathcal{A}|\times|\mathcal{O}|$ Q-matrix for state s, with $|\mathcal{A}|$ and $|\mathcal{O}|$ the cardinalities of action sets of the LS and the attacker, respectively; in WoLF-PDS, $Q(s)\triangleq[Q(s,a)]_{a\in\mathcal{A}}$ reduces to a $|\mathcal{A}|\times 1$ vector.

Proof: The minimax-PDS case is proved first. Denoting $\mathbf{1}_{|\mathcal{A}|}$ and $\mathbf{1}_{|\mathcal{O}|}$ all-one column vectors of length $|\mathcal{A}|$ and $|\mathcal{O}|$, respectively, it can be noticed that for all s,

$$\begin{aligned} Q_2(s)&\geq Q_1(s)-\|Q_1(s)-Q_2(s)\|_\infty\mathbf{1}_{|\mathcal{A}|}\mathbf{1}_{|\mathcal{O}|}^T,\\ Q_2(s)&\leq Q_1(s)+\|Q_2(s)-Q_1(s)\|_\infty\mathbf{1}_{|\mathcal{A}|}\mathbf{1}_{|\mathcal{O}|}^T. \end{aligned} \tag{28}$$

Then applying the fact that $\mathbf{val}(Q'(s))\geq\mathbf{val}(Q''(s))$ for any matrices $Q'(s)\geq Q''(s)$ (where $\mathbf{val}(Q(s))\triangleq\max_{\pi(s)}\min_o\sum_a Q(s,a,o)\pi_a(s)$ in minimax-PDS) on (28), we have

$$\begin{aligned} V_2(s) &= \mathbf{val}(Q_2(s))\\ &\geq \mathbf{val}\left(Q_1(s)-\|Q_1(s)-Q_2(s)\|_\infty\mathbf{1}_{|\mathcal{A}|}\mathbf{1}_{|\mathcal{O}|}^T\right)\\ &= V_1(s)-\|Q_1(s)-Q_2(s)\|_\infty, \end{aligned} \tag{29}$$

and

$$\begin{aligned} V_2(s) =& \ \mathbf{val}(Q_2(s)) \\ \leq& \ \mathbf{val}\left(Q_1(s) + \|Q_2(s) - Q_1(s)\|_\infty \mathbf{1}_{|\mathcal{A}|}\mathbf{1}_{|\mathcal{O}|}^T\right) \\ =& \ V_1(s) + \|Q_2(s) - Q_1(s)\|_\infty, \end{aligned} \tag{30}$$

which implies $|V_1(s) - V_2(s)| \leq ||Q_1(s) - Q_2(s)||_\infty$.

For the WoLF-PDS case, we have

$$\begin{aligned} |V_1(s) - V_2(s)| &= |\max_a Q_1(s,a) - \max_a Q_2(s,a)| \\ &\leq \max_a |Q_1(s,a) - Q_2(s,a)| = \|Q_1(s) - Q_2(s)\|_\infty. \end{aligned} \tag{31}$$

■

Now we are ready to prove Proposition 1.

Proof: For simplicity, the superscript (mp) will be omitted in this proof. To show Q_n converges to Q_* in Proposition 1, it is sufficient to show $\tilde{Q}_n$ converges to $\tilde{Q}_*$, considering the definitions of Q_n and Q_* in minimax-PDS (cf. (11)). To this end, let the mapping sequence $\mathcal{T}_n$ be such that

$$\left(\mathcal{T}_n\tilde{Q}\right)(\tilde{s}_n, a_n, o_n) \triangleq r^u(\tilde{s}_n, a_n, o_n) + \beta \cdot V(s_{n+1}), \tag{32}$$

and for $(\tilde{s}, a, o) \neq (\tilde{s}_n, a_n, o_n)$,

$$\left(\mathcal{T}_n\tilde{Q}\right)(\tilde{s}, a, o) \triangleq \tilde{Q}(\tilde{s}, a, o). \tag{33}$$

First notice that (C1) in Lemma 1 holds, since

$$\begin{aligned} &\mathbb{E}_{S_{n+1}}[(\mathcal{T}_n\tilde{Q}_*)](\tilde{s}_n, a_n, o_n) \\ &= \sum_{s_{n+1}} p^u(s_{n+1}|\tilde{s}_n, a_n, o_n)[r^u(\tilde{s}_n, a_n, o_n) + \beta \cdot V_*(s_{n+1})] \\ &= r^u(\tilde{s}_n, a_n, o_n) + \beta \sum_{s_{n+1}} p^u(s_{n+1}|\tilde{s}_n, a_n, o_n) V_*(s_{n+1}) \\ &= \tilde{Q}_*(\tilde{s}_n, a_n, o_n), \end{aligned} \tag{34}$$

where the last step follows from the definition in (10), and $\mathbb{E}\left[\left(\mathcal{T}_n\tilde{Q}_*\right)\right](\tilde{s}, a, o) = \tilde{Q}_*(\tilde{s}, a, o)$ also holds for $(\tilde{s}, a, o) \neq (\tilde{s}_n, a_n, o_n)$. The contraction property (C2) is shown as follows for any $\tilde{Q}$:

$$\begin{aligned} &||\mathcal{T}_n\tilde{Q} - \mathcal{T}_n\tilde{Q}_*||_\infty \leq \beta \cdot \max_{s'} |V(s') - V_*(s')| \\ &\leq \beta \cdot \max_{s'} ||Q(s') - Q_*(s')||_\infty = \beta \cdot \max_{s'} \max_{a,o} \\ &\times \left|\sum_{\tilde{s}} p^k(\tilde{s}|s', a, o)\left(\tilde{Q}(\tilde{s}, a, o) - \tilde{Q}_*(\tilde{s}, a, o)\right)\right| \\ &\leq \beta \cdot \max_{s'} \max_{a,o} \sum_{\tilde{s}} p^k(\tilde{s}|s', a, o)\left|\tilde{Q}(\tilde{s}, a, o) - \tilde{Q}_*(\tilde{s}, a, o)\right| \\ &\leq \beta \cdot \max_{a,o} \max_{\tilde{s}} \left|\tilde{Q}(\tilde{s}, a, o) - \tilde{Q}_*(\tilde{s}, a, o)\right| \\ &= \beta \cdot ||\tilde{Q} - \tilde{Q}_*||_\infty, \end{aligned} \tag{35}$$

where the first inequality is due to (32); the second inequality is due to Lemma 2; and the third inequality is due to triangle inequality. Hence by Lemma 1 with $\lambda_n = 0$, $\tilde{Q}$ converges to $\tilde{Q}_*$ with probability 1 for learning rate sequence α_n satisfying the conditions there.

APPENDIX C
PROOF OF (15)

As $Q_*^{(\mathrm{wp})}$ is defined identically to $Q_*^{(w)}$ (cf. (5)), it admits the following expansion

$$\begin{aligned} &Q_*^{(\mathrm{wp})}(s,a) \\ &\triangleq \mathbb{E}_{O,R,S'}\left[R(s,a,O) + \beta \cdot V_*^{(w)}(S')\right], \\ &= \mathbb{E}_{O,\tilde{S}}\left[R(s,a,O)\right] + \mathbb{E}_O\left[\beta \sum_{s'} p(s'|s,a,o) \cdot V_*^{(\mathrm{wp})}(s')\right], \\ &= \mathbb{E}_O\left[r^k(s,a,O)\right] + \mathbb{E}_O\left[\sum_{\tilde{s}} p^k(\tilde{s}|s,a,O)\, r^u(s,a,O)\right] \\ &+ \mathbb{E}_O\left[\beta \sum_{s'} \sum_{\tilde{s}} p^u(s'|\tilde{s},a,O)\, p^k(\tilde{s}|s,a,O)\, V_*^{(\mathrm{wp})}(s')\right] \\ &= \mathbb{E}_O\left[r^k(s,a,O)\right] + \sum_{\tilde{s}} p^k(\tilde{s}|s,a)\, \tilde{Q}_*^{(\mathrm{wp})}(\tilde{s},a), \end{aligned} \tag{36}$$

where the third equality is due to assumptions (8) and (9) and the fact that $V_*^{(\mathrm{wp})}$ is the same as $V_*^{(w)}$ by definition; the last one follows from (14) and (13).

APPENDIX D
PROOF OF PROPOSITION 2

For simplicity, the superscript (wp) will be omitted in this proof. First notice that in WoLF [10], the learnt policy π_n will converge to the best response as long as the corresponding Q_n converges to Q_*, when facing a stationary opponent. Further notice that WoLF-PDS and WoLF admit the same relationship between π_n and Q_n. Therefore, to show the rationality of the proposed WoLF-PDS, it is sufficient to show Q_n defined in WoLF-PDS converges to Q_*. For this purpose, we will adopt a similar approach as in the proof of Proposition 1. The convergence of $\tilde{Q}$ will be shown first and then the convergence of Q follows from the definitions of Q_n and Q_* in WoLF-PDS (cf. (15)). Define the mapping sequence $\mathcal{T}_n$ as

$$\left(\mathcal{T}_n\tilde{Q}\right)(\tilde{s}_n, a_n) \triangleq r^u(\tilde{s}_n, a_n, o_n) + \beta \cdot V(s_{n+1}), \tag{37}$$

and for $(\tilde{s}, a) \neq (\tilde{s}_n, a_n)$,

$$\left(\mathcal{T}_n\tilde{Q}\right)(\tilde{s}, a) \triangleq \tilde{Q}(\tilde{s}, a). \tag{38}$$

It follows from (15) that for a stationary opponent, (C1) in Lemma 1 holds since

$$\begin{aligned} \mathbb{E}_{O_n, S_{n+1}}[(\mathcal{T}_n\tilde{Q}_*)(\tilde{s}_n, a_n)] &= \mathbb{E}_{O_n}\Bigg[\sum_{s_{n+1}} p^u(s_{n+1}|\tilde{s}_n, a_n, O_n) \\ &\times \left(r^u(\tilde{s}_n, a_n, O_n) + \beta \cdot V_*(s_{n+1})\right)\Bigg], \\ &= \mathbb{E}_{O_n}\Bigg[r^u(\tilde{s}_n, a_n, O_n) + \beta \sum_{s_{n+1}} p^u(s_{n+1}|\tilde{s}_n, a_n, O_n) \\ &\times V_*(s_{n+1})\Bigg] = \tilde{Q}_*(\tilde{s}_n, a_n), \end{aligned} \tag{39}$$

and $\mathbb{E}[(\mathcal{T}_n\tilde{Q}_*)(\tilde{s},a)] = \tilde{Q}_*(\tilde{s},a)$ also holds for $(\tilde{s},a) \neq (\tilde{s}_n,a_n)$. Further notice that

$$\begin{aligned}
&\|\mathcal{T}_n\tilde{Q} - \mathcal{T}_n\tilde{Q}_*\|_\infty \leq \beta \cdot \max_{s'} |V(s') - V_*(s')| \\
&\leq \beta \cdot \max_{s'} \|Q(s') - Q_*(s')\|_\infty = \beta \cdot \max_{s'} \max_{a,o} \\
&\times \left| \sum_{\tilde{s}} p^k(\tilde{s}|s',a) \left(\tilde{Q}(\tilde{s},a,o) - \tilde{Q}_*(\tilde{s},a,o) \right) \right| + \lambda_n \\
&\leq \beta \cdot \max_{s'} \max_{a,o} \sum_{\tilde{s}} p^k(\tilde{s}|s',a) \left\| \tilde{Q}(\tilde{s},a,o)\tilde{Q}_*(\tilde{s},a,o) \right\| + \lambda_n \\
&\leq \beta \cdot \max_{a,o} \max_{\tilde{s}} \left| \tilde{Q}(\tilde{s},a,o) - \tilde{Q}_*(\tilde{s},a,o) \right| + \lambda_n \\
&= \beta \cdot ||\tilde{Q} - \tilde{Q}_*||_\infty + \lambda_n, \qquad (40)
\end{aligned}$$

where similar arguments as in the proof of Proposition 1 are invoked for the above derivation, and λ_n is defined as

$$\lambda_n \triangleq \max_{s,a} |\overline{r}_n^k(s,a) - \mathbb{E}_O\left[r^k(s,a,O)\right]|. \qquad (41)$$

Hence, if λ_n converges to zero with probability 1, it follows that (C2) in Lemma 1 holds. This implies that, with suitable learning rate, $\tilde{Q}_n$ converges to $\tilde{Q}_*$ with probability 1.

To show that λ_n converges to zero with probability 1, it is sufficient to show $\overline{r}_n^k(s,a) \overset{w.p.1}{\to} \mathbb{E}_O[r^k(s,a,O)]$ for all (s,a). To this end, for each (s,a), consider a mapping sequence $\mathcal{F}_n$ such that, for any real number $r \in \mathbb{R}$,

$$\mathcal{F}_n r \triangleq r^k(s,a,o_n), \qquad (42)$$

where the image $r^k(s,a,o_n)$ of the mapping $\mathcal{F}_n$ is independent of its pre-image r. Further notice that (17) can be written as

$$\overline{r}_{n+1}^k(s,a) = (1-\alpha_n)\overline{r}_n^k(s,a) + \alpha_n \cdot \mathcal{F}_n\overline{r}_n^k(s,a). \qquad (43)$$

Then, it follows from the definition in (42) and the stationarity of the opponent's policy that

$$\mathbb{E}\left[\mathcal{F}_n\mathbb{E}_O\left[r^k(s,a,O)\right]\right] = \mathbb{E}_O\left[r^k(s,a,O)\right], \qquad (44)$$

and that for any $r \in \mathbb{R}$ and $\gamma \in (0,1)$,

$$|\mathcal{F}_n r - \mathcal{F}_n\mathbb{E}_O[r^k(s,a,O)]| = 0 \leq \gamma \cdot |r - \mathbb{E}_O[r^k(s,a,O)]|. \qquad (45)$$

These imply that (C1) and (C2) of Lemma 1 hold for $\mathcal{F}_n$ (by reducing Q and $\mathbb{Q}$ to a $|\mathcal{S}|$-dimensional vector and $\mathbb{R}^{|\mathcal{S}|}$, respectively, with $|\mathcal{S}|$ the cardinality of the state space), and hence $\overline{r}_n^k(s,a) \overset{w.p.1}{\to} \mathbb{E}_O\left[r^k(s,a,O)\right]$.[14] ■

REFERENCES

[1] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A survey of game theory as applied to network security," *Proc. IEEE HICSS*, Honolulu, HI, USA, pp. 1–10, Jan. 2010.

[2] X. Liang and Y. Xiao, "Game theory for network security," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 472–486, Feb. 2013.

[3] R. Prasad, S. Devasenapathy, V. Rao, and J. Vazifehdan, "Reincarnation in the ambiance: Devices and networks with energy harvesting," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 195–213, Feb. 2014.

[14] Note that in the special case of $\alpha_n = \frac{1}{n}$, $\overline{r}_n^k(s,a) \overset{w.p.1}{\to} \mathbb{E}_o\left[r^k(s,a,o)\right]$ readily follows from the well-known weak law of large numbers.

[4] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, no. 13, pp. 2127–2159, 2006.

[5] M. Armbrust *et al.*, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[6] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, 2008.

[7] Y. E. Sagduyu, R. A. Berry, and A. Ephremides, "Jamming games in wireless networks with incomplete information," *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 112–118, 2011.

[8] X. He, H. Dai, and P. Ning, "Improving learning and adaptation in security games by exploiting information asymmetry," *Proc. IEEE INFOCOM*, Hong Kong, China, pp. 1787–1795, Apr. 2015.

[9] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. ACM ICML*, New Brunswick, NJ, pp. 157–163, Jul. 1994.

[10] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artif. Intell.*, vol. 136, no. 2, pp. 215–250, 2002.

[11] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, "Virtualized in-cloud security services for mobile devices," *Proc. ACM Workshop Virtualization Mobile Comput.*, Breckenridge, CO, USA, pp. 31–35, Jun. 2008.

[12] V. Varadharajan and U. Tupakula, "Security as a service model for cloud environment," *IEEE Trans. Netw. Serv. Manag.*, vol. 11, no. 1, pp. 60–75, 2014.

[13] R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 25–37, 2008.

[14] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 589–600, 2007.

[15] T. He, S. Chen, H. Kim, L. Tong, and K.-W. Lee, "Scheduling parallel tasks onto opportunistically available cloud resources," *Proc. IEEE CLOUD*, Honolulu, HI, USA, pp. 180–187, Jun. 2012.

[16] B. Wang, Y. Wu, K. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, 2011.

[17] E. Altman, K. Avrachenkov, and A. Garnaev, "A jamming game in wireless networks with transmission cost," *Network Control and Optimization*, New York, NY, USA: Springer, 2007, pp. 1–12.

[18] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang, "Coping with a smart jammer in wireless networks: A Stackelberg game approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 4038–4047, 2013.

[19] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles International Airport," *Proc. ASM AAMAS*, Estoril, Portugal, pp. 125–132, May 2008.

[20] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus, "Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games," *Proc. ACM AAMAS*, Estoril, Portugal, pp. 895–902, May 2008.

[21] J. Pita, M. Jain, F. Ordónez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Using game theory for Los Angeles airport security," *AI Mag.*, vol. 30, no. 1, pp. 43–57, 2009.

[22] M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*, Cambridge, U.K.: Cambridge Univ. Press, 2011.

[23] X. He, H. Dai, P. Ning, and R. Dutta, "A stochastic multi-channel spectrum access game with incomplete information," presented at IEEE ICC, London, U.K., pp. 4799–4804, Jun. 2015.

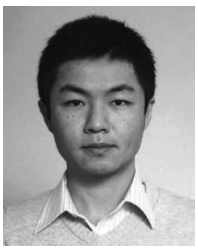
[24] X. He, H. Dai, P. Ning, and R. Dutta, "Dynamic IDS configuration in the presence of intruder type uncertainty," *Proc. IEEE GLOBECOM*, San Diego, CA, USA, pp. 1–6, Dec. 2015.

[25] B. F. Lo and I. F. Akyildiz, "Multiagent jamming-resilient control channel game for cognitive radio ad hoc networks," *Proc. IEEE ICC*, London, U.K., pp. 1821–1826, 2012.

[26] Y. Wu, B. Wang, K. Liu, and T. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 4–15, 2012.

[27] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part I: Known channel statistics," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3566–3577, 2010.

[28] Q. Wang, P. Xu, K. Ren, and X.-Y. Li, "Towards optimal adaptive UFH-based anti-jamming wireless communication," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 1, pp. 16–30, 2012.
[29] N. Salodkar, A. Bhorkar, A. Karandikar, and V. S. Borkar, "An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 732–742, 2008.
[30] N. Mastronarde and M. van der Schaar, "Fast reinforcement learning for energy-efficient wireless communication," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6262–6266, 2011.
[31] F. Fu and U. C. Kozat, "Stochastic game for wireless network virtualization," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 84–97, 2013.
[32] Y. Xiao and M. van der Schaar, "Optimal foresighted multi-user wireless video," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 1, pp. 89–101, 2015.
[33] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 428–445, 2013.
[34] L. Liu, R. Zhang, and K. Chua, "Secrecy wireless information and power transfer with MISO beamforming," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1850–1863, 2014.
[35] D. W. K. Ng, L. Xiang, and R. Schober, "Multi-objective beamforming for secure communication in systems with wireless information and power transfer," *Proc. IEEE PIMRC*, London, U.K., pp. 7–12, Sep. 2013.
[36] A. Mukherjee and J. Huang, "Deploying multi-antenna energy-harvesting cooperative jammers in the MIMO wiretap channel," *Proc. IEEE ASILOMAR*, Pacific Grove, CA, USA, pp. 1886–1890, Nov. 2012.
[37] O. Tan, D. Gunduz, and H. V. Poor, "Increasing smart meter privacy through energy harvesting and storage devices," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1331–1341, Jul. 2013.
[38] W. Yassin, N. I. Udzir, Z. Muda, A. Abdullah, and M. Abdullah, "A cloud-based intrusion detection service framework," *Proc. IEEE CyberSec*, Kuala Lumpur, pp. 213–218, Jun. 2012.
[39] Y. Meng, W. Li, and L. Kwok, "Design of cloud-based parallel exclusive signature matching model in intrusion detection," *Proc. IEEE PHCC/EUC*, Hunan, China, pp. 175–182, Nov. 2013.
[40] C. Szepesvári and M. Littman, "A unified analysis of value-function-based reinforcement-learning algorithms," *Neural Comput.*, vol. 11, no. 8, pp. 2017–2060, 1999.



Xiaofan He (S'13) received the B.S. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2008, the M.A.Sc. degree in electrical and computer engineering from McMaster University, Hamilton, ON, Canada, in 2011, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, in 2015.

His research interests are in the areas of wireless communications and networking, and detection and estimation. His current research focuses on the security issues in wireless communications and networking with a physical layer emphasis.

Huaiyu Dai (M'03–SM'09) received the B.E. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ in 2002.

He was with Bell Labs, Lucent Technologies, Holmdel, NJ, in summer 2000, and with AT&T Labs-Research, Middletown, NJ, in summer 2001. Currently he is a Professor of Electrical and Computer Engineering at NC State University, Raleigh. His research interests are in the general areas of communication systems and networks, advanced signal processing for digital communications, and communication theory and information theory. His current research focuses on networked information processing and crosslayer design in wireless networks, cognitive radio networks, wireless security, and associated information-theoretic and computation-theoretic analysis.

He has served as an editor of IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. Currently he is an Area Editor in charge of wireless communications for IEEE Transactions on Communications. He co-edited two special issues of EURASIP journals on distributed signal processing techniques for wireless sensor networks, and on multiuser information theory and related applications, respectively. He co-chaired the Signal Processing for Communications Symposium of IEEE Globecom 2013, the Communications Theory Symposium of IEEE ICC 2014, and the Wireless Communications Symposium of IEEE Globecom 2014.

Peng Ning (M'01–SM'12) received the B.S. degree in information sciences from the University of Science and Technology of China (USTC), Hefei, China, in 1994, the M.E. degree in communications and electronics systems from USTC, Graduate School in Beijing, Beijing, China, in 1997, and the Ph.D. degree in information technology from George Mason University, Fairfax, VA, in 2001.

He is a Professor of Computer Science at NC State University, where he also serves as the Technical Director for Secure Open Systems Initiative (SOSI). He is a recipient of National Science Foundation (NSF) CAREER Award in 2005. He is currently the Secretary/Treasurer of the ACM Special Interest Group on Security, Auditing, and Control (SIGSAC), and is on the Executive Committee of ACM SIGSAC. He is an editor for Springer Briefs in Computer Science, responsible for Briefs on information security. He has served or is serving on the editorial boards of several international journals, including ACM TRANSACTIONS ON SENSOR NETWORKS, *Journal of Computer Security*, *Ad-Hoc Networks*, *Ad-Hoc & Sensor Networks: An International Journal*, *International Journal of Security and Networks*, and *IET Proceedings Information Security*. He also served as the Program Chair or Co-Chair for ACM SASN 2005, ICICS 2006 and ESORICS 2009, ICDCS-SPCC 2010, and NDSS 2013, the General Chair of ACM CCS 2007 and 2008, and Program Vice Chair for ICDCS 2009 and 2010—Security and Privacy Track. He served on the Steering Committee of ACM CCS from 2007 to 2011, and is a founding Steering Committee member of ACM WiSec and ICDCS SPCC. His research has been supported by NSF, Army Research Office (ARO), the Advanced Research and Development Activity (ARDA), IBM Research, SRI International, and the NCSU/Duke Center for Advanced Computing and Communication (CACC). Peng Ning is a senior member of the ACM and the ACM SIGSAC.