Hide and Seek: A Markov-Based Defense Strategy Against Off-Sensing Attack in Cognitive Radio Networks

Moinul Hossain, *Member, IEEE* and Jiang Xie, *Fellow, IEEE*

***Abstract*—In a cognitive radio-based network (CRN), secondary users (SUs) opportunistically access underutilized spectrum resources and stop utilizing these resources when licensed or primary users (PUs) reappear. However, this way of opportunistic spectrum access is susceptible to novel vulnerabilities. Recently, a new attack, *off-sensing* (OS), has shed light on a vulnerability in the Federal Communications Commission (FCC) policy of CRN, which affects the channel utilization of the victim SU by creating an illusion of a PU's presence. However, prior work on OS-attack considers a deterministic approach that is unrealistic and is futile to fortify against conventional defense techniques. In this paper, we propose a new random approach, the *random-OS* attack, which adapts to realistic scenarios and is difficult to detect using conventional techniques. Then, we model the interaction between the victim SU and attackers as a stochastic zero-sum Markov game and propose a novel safeguard approach based on the Markov decision process to defend the proposed attack, namely *hide and seek*. Finally, we introduce an OS-attack detection strategy, which utilizes the sensing history to detect the presence of attackers without violating any policy or design constraints and without any networking overhead. Mathematical analysis and extensive simulation results exhibit the superior performance of our proposed work and advent a direction in designing safeguard strategies without amending the current FCC policies.**

***Index Terms*—Cognitive radio networks, off-sensing attacks, Markov chain, and Markov decision process.**

I. Introduction

The demand for wireless services continues to increase exponentially. However, the constrained amount of radio resources has been impeding the growth to meet this demand. On the other hand, the Federal Communications Commission (FCC) has concluded that the radio spectrum is not balanced in terms of resource and traffic-load; a significant portion of the radio spectrum remains underutilized, whereas high volume of traffic appears in another portion. Cognitive Radio (CR) has been proposed as an enabling technology to off-set this unbalanced utilization of the spectrum. A CR-enabled device (or secondary user, SU) can opportunistically access an underutilized licensed channel (i.e., white spaces) and utilize it until a licensed user (or primary user, PU) reappears. *Spectrum sensing* helps CR-enabled devices to be aware of and to be sensitive to the changes in its network environment [1]–[3]. It helps CR-enabled devices to detect white spaces and PU's presence without interfering with the primary network.

However, like traditional wireless networks, CR-based networks (CRNs) are prone to conventional network attacks [4] (e.g., jamming, packet drop, and eavesdropping). In addition, new genres of attacks have emerged in CRNs due to its unique way of operation (i.e., opportunistic spectrum access) [5]–[7].

Two most studied attacks specifically in CRNs that try to compromise the spectrum sensing process are primary user emulation (PUE) [8] and spectrum sensing data falsification (SSDF) [9]. Depending on the motive, these attacks help the attacker to either maximize its own channel utilization (i.e., selfish attacker) or to sabotage the network operation of the victim (i.e., malicious attacker). In PUE, an attacker masquerades as a PU during the sensing interval of the victim to trick it into avoiding the channel; a PUE attacker forges the transmission characteristics of a benign PU and tries to compromise the spectrum sensing process of the victim. To avoid PUE attacks and sensing errors, cooperative spectrum sensing approach is proposed [10], [11] as an alternative decision process to collectively estimate the spectrum availability. Nonetheless, this consensus-based approach is also vulnerable to intelligent attacks, such as SSDF. In SSDF, an attacker shares engineered sensing information with its neighbors (i.e., victims) to manipulate the consensus on the channel availability in the cooperative spectrum sensing.

Under both of the above attacks, *sensing interval* is the attack surface in both attacks. In [12], a novel genre of attacks in CRNs, off-sensing (OS), is introduced. In contrast to PUE and SSDF, OS-attack achieves similar goals with a different attack surface: the *off-sensing interval* (i.e., the transmission or reception interval). In OS-attack, an attacker interferes with the victim's transmission only when the victim is not sensing but transmitting (or receiving). The attacker tries to corrupt packets of the victim and to cause transmission failures. As current radio designs do not permit SUs to sense the operating channel during transmission, a victim SU would believe that it is interfering with a reappeared PU, thereby creating an

Manuscript received February 8, 2020; revised May 30, 2020; accepted July 3, 2020. Date of publication July 24, 2020; date of current version December 30, 2020. This work was supported in part by the US National Science Foundation (NSF) under Grants 1718666, 1731675, 1910667, 1910891, and 2025284. Recommended for acceptance by Dr. Yunhuai Liu. *(Corresponding author: Jiang Xie.)*

Moinul Hossain is with the Department of Computer and Information Sciences, Towson University, Towson, MD 21252 USA (e-mail: mhossain@towson.edu).

Jiang Xie is with the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC 28223 USA (e-mail: linda.xie@uncc.edu).

Digital Object Identifier 10.1109/TNSE.2020.3011707

illusion. Since FCC regulation requires that an SU should leave the channel within 2 seconds of PU reappearance [13], it will stop using the channel if the reason of interference remains inconclusive.

Motivations: Prior work on OS-attack considered two attack scenarios: the attacker always stays on a particular channel and attacks anyone who tries to access the channel (i.e., selfish attacker), or the attacker knows the channel hopping sequence of the victim SU and interferes with each transmission attempt of the victim to create a Denial-of-Service (DoS) situation (i.e., DoS attacker). In either case, the attacker plays a deterministic role from a victim's perspective in terms of the operating channel (i.e., the victim can infer the future attack channel). This deterministic hopping sequence of OS-attackers makes it difficult to fortify against traditional defense techniques [5], [7]. Similarly, the assumption that the attacker has the perfect knowledge of the victim's hopping sequence makes it a critical disadvantage for the victim and creates unrealistic scenarios (hopping sequence depends on each SU's surrounding environment, which varies in time and space). Therefore, in realistic conditions, OS-attackers desire a sequence that is random.

Previous work on the defense and detection of the PUE and SSDF attack focused on the sensing interval and the cooperative nature of CRNs, respectively. However, these proposed methods cannot detect the OS-attack due to different attack surfaces and, to the best of our knowledge, the defense of OS-attacks remains unstudied. Hence, the OS-defense requires focused efforts into the off-sensing interval to safeguard SUs.

Challenges on OS-Attackers: SUs can follow any channel-hopping process to rendezvous with each other [14]. Moreover, the rendezvous channel (the channel where two SUs meet) and the transmission channel may differ [15]. Therefore, it is difficult for an attacker to find the operating channel of the victim to perpetrate an OS-attack without any predetermined knowledge. In addition, OS-DoS attack requires successive detection of the victim's operating channel; that is, more challenging.

Challenges on Defense Against OS-Attacks: A straight-forward approach to identify an OS-attacker is to sense the channel when transmitting. However, hardware limitations (e.g., the transmission antenna would overwhelm the sensing antenna), design considerations (e.g., half-duplex radio), and a decrease in channel utilization (e.g., the victim SU could use an extra-sensing time to utilize another white space) restrain this approach. Therefore, the defense and detection process of OS-attack must adhere to these constraints.

Moreover, most previous research on defense considered that attackers are always present and safeguard process(es) are deployed regardless of the presence of attackers. This assumption costs SUs networking, computational, and energy overhead. Therefore, in resource constrained networks, the safeguard process must be aware of the presence of attackers, and it is deployed only when under-attack. Additionally, it must provide the flexibility to trade-off between networking and security performance.

Contributions: In this paper, we study these research challenges and propose solutions to these problems. The novel contributions of this paper are summarized in the following:

1. We propose a random strategy for OS-DoS attackers, where attackers iteratively hop through channels to detect the operating channel of the victim and persistently perpetrate OS-attacks to cause a DoS situation.
2. We propose a Markov decision process (MDP) based safeguard approach, where victims avoid the OS-attack by randomly hopping through different channels and detect the attacker when deemed necessary according to the parameters (i.e., the trade-off between networking and security performance). The defender learns the MDP game through reinforcement learning.
3. We consider that attackers may not always be present, and the safeguard process must be aware of attackers' presence. We propose an attack inference model to detect the presence of attackers without any networking overhead.

Paper Organization: The rest of this paper is organized as follows. In Section II, prior DoS attacks and their defense techniques are reviewed briefly. Then in Section III, the system model that is considered in this paper is explained. We provide an overview of the proposed attack model in Section IV followed by the formulation of the Markov game in Section V to counteract the random-OS attack. Then, we propose the attack inference model to detect the presence of potential OS-attackers in Section VI. Simulation results are shown and discussed in Section VII, followed by the concluding remarks in Section VIII.

II. Related Work

Unlike traditional PUE attacks, the OS-attack does not rely on the transmission characteristics of a PU. Additionally, in contrast to jamming attacks, it does not depend on a strong noise signal either. Instead, it creates enough interference using regular transmissions to corrupt the reception of the victim. Therefore, the OS-attack is neither the PUE nor jamming attack; however, we compare it to both of these attacks because of its close resemblance to these attacks from the perspective of denial-of-service attacks.

The security research community has proposed numerous vulnerabilities and their defenses in CRNs [16]–[22], which laid the groundwork for the future research on dynamic spectrum access. The PUE attack is discussed in [23], where the vulnerability is exploited in a multi-hop channel environment; if a PUE attack is launched and the victim SU has no available channel, the transmission is dropped or delayed. The dropped and the delayed transmissions result in unreliable communication and lower quality of service, respectively [24]. An optimal online learning algorithm is proposed in [25], where it can be utilized by a PUE attacker without any prior knowledge of the PU activity and secondary user channel access strategies. In [26], a cross-layer route manipulation attack is proposed in CR-based wireless mesh networks, where OS-DoS attack is utilized as a front-end attack to manipulate the traffic-flow and to

induce congestion in the network. Though the PUE and the OS-attack have similar attack objectives, the OS-attack exploits a different attack surface, i.e., the off-sensing interval. In addition, unlike the PUE attack, the OS attacker must know the activity of the victim.

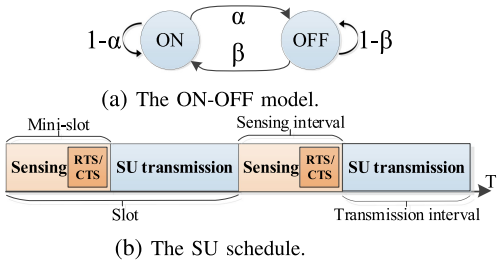
A wide range of jamming strategies have been studied in [27]–[31]. However, we only focus on the papers that are most relevant to our proposed work. In [32], a frequency hopping strategy against a jammer in 802.11 networks is proposed; the proposed hopping strategy optimizes the channel residence time. In [33], a similar hopping strategy is developed using Markov decision process for a cognitive radio network. In [34], a strategy that combines frequency hopping and rate adaption techniques is proposed to defend jamming attacks; the rate adaptation method helped to increase the diversity in defense against a power constrained jammer. A different direction to counteract jamming attacks is introduced in [35], where the latest advances in deep learning and artificial intelligence are leveraged. A sweep jammer strategy is proposed in [36] where jammers sweep through all channels to find the operating channel of any user. However, these proposed attack strategies are either ineffective in realistic scenarios or does not consider the DoS situation. Unlike previous research, we devise a sophisticated attack strategy for OS-DoS attackers to adapt to more realistic conditions and to force the victim in dropping packets.

Regarding the defense strategies, a game theoretical approach is proposed in [37], [38] to counteract PUE attacks by adopting a combination of extra-sensing and surveillance process. In [39], an MDP-based anti-jamming strategy is proposed to counteract jamming attacks in CRNs. A zero-sum Markov game is proposed in [34] and an optimal strategy to defend against the reactive-sweep jammer is devised. Similarly, in [36], an MDP-based strategy is proposed to thwart jamming attacks in multi-channel networks, where radios are equipped with in-band full-duplex capability. However, all these works neither consider an iterative attack model to prevent DoS attacks nor adopt an intelligent attack detection model. In contrast, we consider a more sophisticated attack model where the attacker can identify an individual victim's transmission and perpetrate a DoS attack on the victim, but our proposed model can detect the presence of such attackers.

Moreover, in the proposed defenses of these attacks, researchers have mostly considered that the victim can detect the unauthorized transmissions of attackers in the sensing interval. In contrast, an OS-attacker [12] ingeniously avoids the sensing interval and interferes with the victim during the transmission interval.

III. System Model

We consider two SUs who are trying to communicate between themselves in the presence of OS-attackers. These two SUs could be network entities of either an infrastructure-based network (i.e., one SU is a CR access point that opportunistically accesses the licensed spectrum, and the other is a CR user communicating with other network users through the access point) or an ad-hoc network. They are located within the interference region of OS-attackers, and OS-attackers are authorized and authenticated entities in the network.



(a) The ON-OFF model.

(b) The SU schedule.

Fig. 1. The network model.

A. Network Model

In this subsection, we explain the traffic model of both PUs and SUs, and we illustrate the rendezvous-based channel access mechanism for SUs.

1) PU and SU Model: We consider the presence of M homogeneous channels (and M PUs), each with a fixed bandwidth. Time is divided into equal slots. Transmissions are packet based for both PUs and SUs, and a packet transmission starts at the beginning of a mini-slot and finishes at the end of a mini-slot. The length of a mini-slot is the time to perform a fast-sensing [40] and to exchange a request-to-send/clear-to-send (RTS/CTS) handshake, and a slot is a multiple of mini-slots. Each PU randomly selects a channel to access and alternates between the ON and OFF state, according to an ON-OFF model (Fig. 1(a)). Let α and β denote the transition probabilities from the ON to OFF state and from the OFF to ON state, respectively. We consider a saturated SU traffic scenario , which means that SUs always have a packet in their buffer to transmit. Hence, an SU continuously transmits on a channel until it finds the current channel busy during a sensing interval or experiences a transmission failure (e.g., if an ACK is not received from the other SU). Transmission failures can result from two reasons: collision with a reappeared PU and interference from an OS-attacker. However, SUs are unable to determine the exact reason of transmission failures due to their inability to sense the channel during transmission or reception.

2) SU Access Protocol: Each transmission attempt of an SU must be preceded by a sensing interval. As shown in Fig. 1(b), SUs periodically operate between the sensing and transmission intervals. An SU is allowed to access a channel when it finds the sensing result suitable to transmit (e.g., senses that no PU is present). After sensing the channel available, two SUs exchange RTS/CTS messages to reserve the channel. Each SU is equipped with one half-duplex radio for spectrum sensing, control information exchange, and data transmission. With one radio, an SU can sense the channel only before initiating the transmission (i.e., in the sensing interval). During a sensing interval, if an SU senses that the current channel is busy, it pauses the communication attempt on the current channel, performs a spectrum handoff to a new channel, and resumes the communication attempt on the new channel (if the new channel is sensed available).

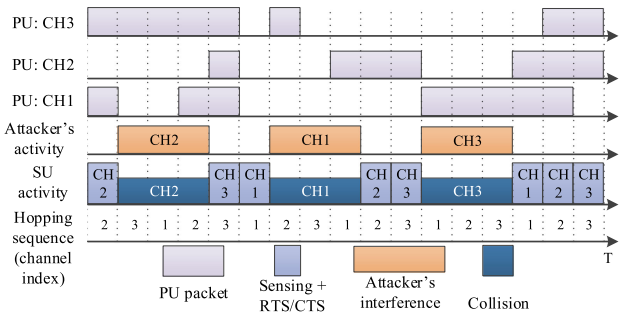


Fig. 2. OS-DoS attack under periodic channel hopping process.

B. Network Coordination Scheme

In this paper, we assume that a common control channel (CCC) is unavailable and two SUs must find a common available channel between them to initiate a data transmission. Rendezvous technique works as the process for two SUs following a channel hopping process to meet and exchange control information on a common available channel. A significant amount of research has been conducted on rendezvous techniques. However, the choice of a specific rendezvous scheme does not impact the performance of our proposed attack and defense mechanisms, as long as attackers have no prior knowledge of the victim's hopping sequence. Thereby, we assume that benign SUs have successfully performed rendezvous with each other, using any existing blind rendezvous scheme, and they share a time-seeded pseudo-random channel hopping sequence for future communications.

C. OS-DoS Attack

The OS-attacker intelligently interferes with a victim's transmission in the transmission interval (by avoiding the sensing interval) and misleads the victim SU into believing that the victim is interfering with a reappeared PU. With current designs, an SU does not sense the channel during transmission. Therefore, it cannot detect the origin of an interference. In addition, according to the FCC regulation, an SU must leave the channel within 2 seconds [13] and perform a spectrum handoff. These two factors facilitate an attacker to confuse the victim with the presence of a reappeared PU and to force the victim to leave the channel. An OS-attacker detects the transmission of a particular victim SU from the RTS/CTS message that precedes each transmission attempt. Fig. 2 provides an illustration of the OS-DoS attack under a periodic channel hopping process.

In Fig. 2, the OS-attacker knows the channel hopping sequence of the victim SU and interferes with each transmission originating from and to the victim (by overhearing RTS/CTS messages). Here, the attacker interferes the whole packet time to make sure that the victim cannot decode the packet and tries to create a DoS situation for the victim SU by causing consecutive successful collisions. However, in reality, it is likely that the attacker does not have any knowledge of the victim's hopping sequence, and it requires shrewder efforts from the attacker to perpetrate successive transmission failures. Next, we propose a novel strategy for an attacker to perpetrate the OS-DoS attack, *without any knowledge of the victim's hopping sequence and operating channel*.

IV. Proposed Random-OS Attack Model

In our proposed OS-DoS attack, the short-term goal is to cause successive transmission failures, and the long-term goal is to reach the maximum limit of transmission attempts to force the victim to drop the current packet. As shown in Fig. 2, if the maximum transmission attempt is 3, then the SU packet would have been dropped. However, the assumption that attackers know the channel hopping sequence of the victim is unrealistic and so is the strategy of an attacker to interfere with each transmission of the victim (due to the deterministic hopping sequence of the attacker); the victim can infer the attacker's activity and detect the attacker with a longer fine-sensing (explained in Section V). Therefore, we propose a new random strategy for OS-DoS attackers, where the attackers have no prior knowledge of the victim's channel hopping sequence, and they randomly hop to different channels in each slot to detect the victim and to perpetrate the OS-DoS attack.

Basic Principles: We assume the presence of m OS-attackers ($m < M$) with the same hardware configuration as benign SUs. We consider that these OS-attackers coordinate among themselves using an out-of-band secure channel (i.e., a secure control channel for attackers only), and they attack non-overlapping channels to increase their chance to detect the operating channel of the victim sooner. Attackers detect a transmission of a particular victim by listening to the RTS/CTS messages. Then, they perform the OS-attack in the transmission interval of the victim by interfering the victim's transmission. As discussed earlier, this attack happens only when the victim is transmitting and not sensing. The interference in the off-sensing interval (i.e., the transmission interval) tricks the victim into believing that it is interfering with a legitimate PU; hence, the victim leaves the channel.

Short-Term Strategy: With the help of coordination, the m attackers visit m different channels during each slot. Here, attackers randomly generate a channel hopping sequence after each successful attack (i.e., transmission failure) and hop through the sequence periodically until they find the operating channel of the victim SU. As the network has M channels, there are $M!$ sequences with equal probability of being selected. This strategy of channel hopping helps attackers to put an upper bound on how long (i.e., the channel residence time) a victim SU can continuously use a channel. The upper bound will be discussed later in this section. Fig. 3(a) shows an illustration of the attack sequence with $M = 10$ and $m = 2$. It shows the hopping sequence of two attackers before a successful OS-attack. Here, the operating channel of the victim SU is channel-3 and, in slot-3, attackers detect the victim and perpetrate the attack on channel-3. Also, the attacker must attack sufficiently long enough to corrupt the packet, otherwise the victim can recover the packet from minor interference. Now, let a_i represents the channel on which attackers have conducted

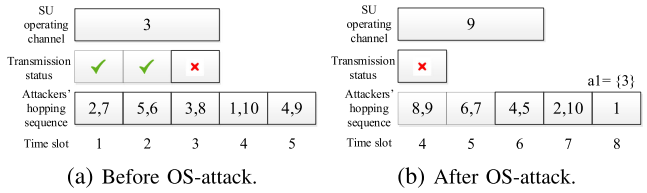


Fig. 3. First phase of the random-OS attack.

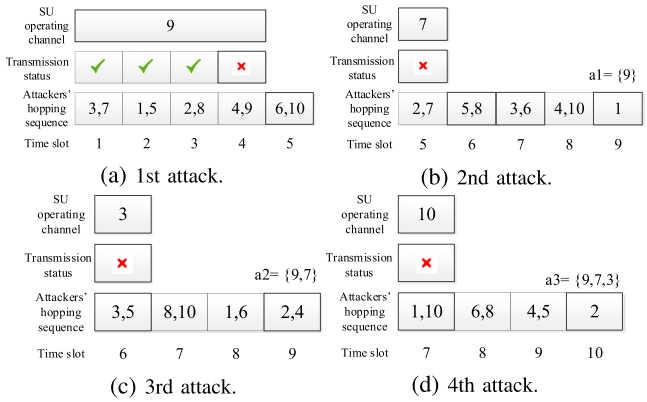


Fig. 4. A scenario of a successful OS-DoS attack with $G = 4$.

the OS-attack and i represents the number of successive attacks (or transmission failures). a_i works as a queue of channels and includes all the attacked channels, from $i, i-1, \cdots, 1$, on which successive transmission failures have happened. However, attackers will exclude these a_i channels from the next random sequence and re-visit them at the end of the hopping sequence. Fig. 3(b) shows an example of the new hopping sequence of attackers to increase their chance of a successive attack by excluding the attacked channel (i.e., channel-3) in the previous slot (i.e., slot-3).

In the OS-attack, a victim cannot determine the exact reason of the transmission failure. Thereby, the victim will randomly hop to a new channel (believing that it has interfered with a reappeared PU), try to stay on that channel as long as plausible, and not hop back to the previously attacked channels (i.e., a_i) until it achieves a successful packet transmission. Hence, it is inefficient for attackers to revisit the previously attacked channels for a particular packet. After each successful perpetration of the attack (or transmission failure), attackers randomize their hopping sequence, excluding a_i. Therefore, after i successive transmission failures, attackers have $M - i$ channels to randomize. Fig. 3(b) illustrates a new hopping sequence of the attackers.

Long-Term Strategy: As the OS-DoS attack considers that the victim must experience G consecutive transmission failures ($G < M$) before discarding the current packet, attackers stay persistent to increase their chance of successful attacks after each successive OS-attack. Hence, they keep excluding channels that were already attacked earlier, for the current packet. Fig. 4 shows an illustration of a scenario, where $G = 4$, and attackers are successful to drop the packet with 4 successive attacks. In the illustration, we can observe that the attackers keep discarding the earlier consecutive attacked channels at each time-slot, i.e., a_i, and eventually after 4 successive attacks, force the victim to drop the packet. It creates a DoS scenario for the victim. However, attackers may not successfully perpetrate the attack in a consecutive manner, and they must take this into consideration in the subsequent time-slot to increase the chance of a successful attack.

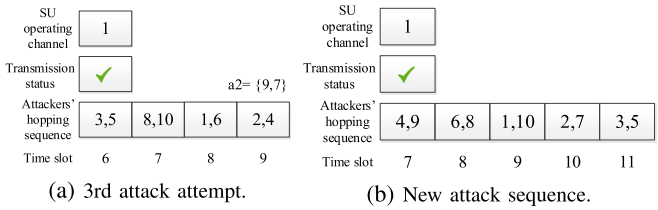


Fig. 5. Re-randomization of the attack sequence.

After the i_{th} successful attack, if attackers are not successful in the subsequent time-slot, they consider that the victim had a successful transmission. Hence, they will re-randomize their hopping sequence (i.e., nullify a_i), excluding the channels they have visited in the current slot (since currently visited channels are free, there is no need to visit them again in this period), and begin a new period (one period $= \lceil M/m \rceil$ slots). Fig. 5 provides an illustration of this scenario. Fig. 5(a) illustrates an alternate scenario if the victim SU had chosen channel-1 (instead of channel-3) in Figs. 4(c), and 5(b) illustrates the new randomized sequence. Thereby, it is inefficient for attackers to visit the attacked channel soon, and hence the attackers exclude these channels.

If attackers cannot detect the operating channel and one period has finished, they will revisit the channels following the same sequence. Given M channels and m OS-attackers, if the victim SU stays on the same channel, the operating channel of the victim will be detected within $\lceil M/m \rceil$ slots. Thereby, the maximum number of consecutive successful transmissions an SU can have in a channel is $K = \lceil M/m \rceil - 1$. This is the upper-bound that was discussed earlier in this section.

Summary: The proposed OS-DoS attack strategy introduces uncertainties in actions of attackers; hence, we name it random-OS attack. Unlike the deterministic approach shown in Fig. 2, the proposed strategy introduces a random hopping sequence for attackers. Due to this randomness, it is not guaranteed that the victim can detect an attacker's interference by a single fine-sensing [41], rather it may require multiple attempts to detect an attacker. Therefore, the victim SU must use the fine-sensing interval (explained in the next section) wisely to maximize the chance of detection.

V. Proposed Safeguard Approach: Hide and Seek

In this section, we propose a solution to the random-OS attack problem by modeling it as an MDP-based game with three actions: *stay*, *hop*, and *extra-sense*. Besides *stay* and *hop*, we propose an action *extra-sense* to increase the diversity of defense (Fig. 6). In *extra-sense*, instead of transmitting

Fig. 6. The extra-sensing interval.

in the transmission interval, an SU tries to detect OS-attackers by fine-sensing the channel which we call the extra-sensing interval. With fine-sensing, an SU can differentiate between the transmission of a PU and an attacker. Now, with these available actions, the MDP deduces an optimal policy, which provides the optimal action to take at each state that maximizes the reward of playing this MDP-based game. One important point to note, the attack strategy is integrated into the stochastic process where the attacker acts as the environment; this strategy reduces the game complexity from a multi-agent problem to a single-agent problem. Therefore, in this section, we model the attack and defense problem as an MDP, and we develop a single agent (i.e., a victim SU) MDP-based defense method to counteract the random-OS attack.

A. Formation of the MDP

We assume that the channel-hopping sequence of the victim SU is unknown to the attacker; however, the attacker can iteratively sweep through the available channels and detect the presence of the victim SU. As we consider the presence of multiple (i.e., m) OS-attackers and coordination among themselves, they will not hop to the same channel together. Instead, they will hop to m different channels to determine the operating channel of the victim SU faster. The SU will decide its action at the end of each time slot, based on the observation of the current state. The SU receives an immediate reward $U(n)$ in the n_{th} time slot,

$$\begin{aligned} U(n) = {} & R.\mathbf{1}(Successful\ transmission) \\ & - L.\mathbf{1}(Transmission\ failure) \\ & - C.\mathbf{1}(Hopping\ cost) - B.\mathbf{1}(Busy\ channel) \\ & - F.\mathbf{1}(Penalty\ for\ policy\ violation) \\ & - Q.\mathbf{1}(Packet\ drop) + E.\mathbf{1}(Attacker\ detection), \end{aligned} \tag{1}$$

where $\mathbf{1}(\cdot)$ is an indicator function of the event in brackets.

As the employed strategy impacts the current state and also the future states, the expected reward of this game is,

$$\overline{U} = \sum_{n=1}^{\infty} \delta^{n-1} U(n), \tag{2}$$

where δ represents the discount factor ($0 < \delta \leq 1$). It measures the significance of the future reward values.

B. Markov Model

This subsection demonstrates the proposed MDP model and defines the state space, action space, rewards, and transition probabilities. We assume that attackers sweep through all channels periodically; hence, the probability of an operating channel being detected depends on the channels that have been visited earlier in the sequence. This consideration helps us to conform the requirement of a Markov process (i.e., a future state of the Markov process depends only on the current state).

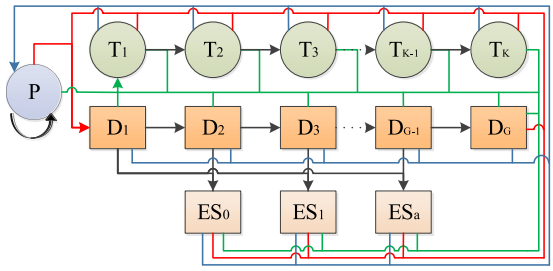


Fig. 7. The proposed MDP.

Markov States: The state denotes the status of an SU at the end of a time-slot. Here, the proposed Markov model (Fig. 7) has six kinds of states:

P: The SU senses that the channel is occupied by a PU.
T_i: The SU hopped onto a new channel and had i consecutive successful transmissions ($1 \leq i \leq K$).
D_j: The SU had j consecutive transmission failures in j different channels ($1 \leq j \leq G$).
ES_0: The SU employed the action $extra\text{-}sense$ and found the channel is free (i.e., no PU or OS attacker).
ES_1: The SU employed the action $extra\text{-}sense$ and found the channel is reoccupied by a PU.
ES_a: The SU employed the action $extra\text{-}sense$ and detected an OS-attacker successfully.

We represent the whole state space as $\mathbb{X} \triangleq \{P, T_1, T_2, \cdots, D_1, D_2, \cdots, ES_0, ES_1, ES_a\}$.

Actions: Here, we have three actions available at each state:

stay (s): The SU remains on the current channel in the next time-slot and initiates a transmission.
hop (h): The SU hops to a new channel in the next time-slot and initiates a transmission.
extra-sense (es): The SU hops to a new channel in the next time-slot and fine-senses the channel for interference.

We represent the whole action space as $\mathbb{A} \triangleq \{s, h, es\}$.

Rewards: When an SU performs a handoff, it is required to perform radio-frequency front-end reconfiguration that consumes insignificant time and we must make it accountable. Though the duration of this reconfiguration process depends on the hardware (e.g., 30ms in USRP [42]), loss in throughput is inevitable. In addition, synchronization between the transmitter and the receiver nodes may engender more loss in throughput. Collectively, we denote the mean cost of a channel handoff by C. Let $U(S, a, S')$ represents the reward when an SU takes an action $a \in \mathbb{A}$ in state $S \in \mathbb{X}$ and enters into state $S' \in \mathbb{X}$. Now using (1), we define rewards:

$$U(S,a,S') =$$

$$\begin{cases} R, & \text{if } \{S,a,S'\} = \{T_i, s, T_{i+1}\}, i = 1,\ldots,K-1 \\ R-C, & \text{if } \{S,a,S'\} = \{\mathbb{X}, h, T_1\} \\ -L, & \text{if } \{S,a,S'\} = \{T_i, s, D_1\}, i = 1,\ldots,K-1 \\ -L-C, & \text{if } \{S,a,S'\} = \{\mathbb{X}, h, D_j\}, j = 1,\ldots,G-1 \\ -Q-C, & \text{if } \{S,a,S'\} = \{D_{G-1}, h, D_G\} \\ -B, & \text{if } \{S,a,S'\} = \{T_i, s, P\}, i = 1,\ldots,K-1 \\ -B-C, & \text{if } \{S,a,S'\} = \{\mathbb{X}, h, P\} \\ -F, & \text{if } \{S,a,S'\} = \{Z, s, \mathbb{X}\}, Z \in \{D, P\} \\ -Q, & \text{if } \{S,a,S'\} = \{\mathbb{X}, es, Z\}, Z \in \{ES_0, ES_1\} \\ E-Q, & \text{if } \{S,a,S'\} = \{\mathbb{X}, es, ES_a\}. \end{cases} \tag{3}$$

Transition Probabilities: As m attackers are going through their attack channel sequence, at state T_i, only $\max(M - im, 0)$ channels have yet to be visited by attackers, and another m channels will be visited in the subsequent slot. Therefore, the probability of an OS-attack (with action $stay$) in absence of a PU on the channel,

$$Pr_{at|s} = \begin{cases} \frac{m}{M-im}, & \text{if } i < K \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

The transition probabilities from state T_i with action $stay$ is,

$$\begin{aligned} Pr(T_{i+1}|T_i, s) &= (1-\beta)^{l+1}(1 - Pr_{at|s}), \\ Pr(D_1|T_i, s) &= (1-\beta)\{1-(1-\beta)^l\} \\ &\quad + (1-\beta)^{l+1} Pr_{at|s}, \\ Pr(P|T_i, s) &= \beta, \end{aligned} \tag{5}$$

where an SU packet is l mini-slots long, and each SU packet is preceded by 1 mini-slot long sensing interval. Note that the action $stay$ is a violation of hard-coded network policy in state P and D_j and subject to penalty (i.e., $-F$).

When there are plenty of channels in the network, the time interval of visiting back to a channel is long; hence, we can approximate the probability of finding the channel busy with action hop as the steady-state probability,

$$Pr(P|S,h) = \frac{\beta}{\alpha+\beta} = \rho, \quad S \in \mathbb{X}. \tag{6}$$

Now, the SU takes action hop and selects a new channel randomly from $M-1$ channels (the SU does not hop to the same channel it found busy in the current slot) from the current state P and hands off to that channel. Provided that the new channel is available, the probability of an OS-attack is,

$$Pr_{at|h,P} = \frac{1}{M} \cdot \frac{m-1}{M-1} + \frac{M-1}{M} \cdot \frac{m}{M-1}. \tag{7}$$

Since, attackers do not know the current state of the victim SU, they will keep hopping through the predetermined sequence consisting of M channels. Now, let us assume that channel ch was sensed busy by the victim SU in the previous slot. Then, the former and latter part of (7) represents the scenario where attackers visit the channel ch and do not visit the channel ch in the current slot, respectively. Now, the transition probabilities from state P with action hop is,

$$\begin{aligned} Pr(T_1|P,h) &= (1-\rho)(1-\beta)^l(1 - Pr_{at|h,P}), \\ Pr(D_1|P,h) &= (1-\rho)\{1-(1-\beta)^l\} \\ &\quad + (1-\rho)(1-\beta)^l Pr_{at|h,P}. \end{aligned} \tag{8}$$

When an SU takes action hop from state T_i, it randomly selects a channel from $M-1$ channels (excluding the current one). The probability that attackers will attack the new channel in the next slot depends on two cases:

- The new channel has already been visited by attackers: The new channel is one of the im channels visited by attackers.
- The new channel has not been visited by attackers: The new channel is among the $M - im - 1$ channels that have not been visited by attackers, and it will not be visited by attackers in the next slot.

Therefore, the probability of OS-attack,

$$Pr_{at|h,T} = 1 - \left(\frac{mi}{M-1} + \frac{M-im-1}{M-1}(1 - Pr_{at|s}) \right). \tag{9}$$

The transition probabilities from state T_i with action hop is,

$$\begin{aligned} Pr(T_1|T_i,h) &= (1-\rho)(1-\beta)^l(1 - Pr_{at|h,T}), \\ Pr(D_1|T_i,h) &= (1-\rho)\{1-(1-\beta)^l\} \\ &\quad + (1-\rho)(1-\beta)^l Pr_{at|h,T}. \end{aligned} \tag{10}$$

When an SU takes action hop from state D_j, it randomly selects a channel from $M-j$ channels. As the SU has already experienced transmission failures j times in j different channels, it does not visit back to these channels until it successfully transmits the current packet. Since attackers also randomize their attack sequence, excluding these j channels, the probability that attackers will attack the new channel in the next slot is uniformly distributed over $M-j$ channels. Therefore, the probability of an OS-attack is,

$$Pr_{at|h,D} = \frac{m}{M-j}. \tag{11}$$

The transition probabilities from state D_j with action hop is,

$$\begin{aligned} Pr(T_1|D_j,h) &= (1-\rho)(1-\beta)^l(1 - P_{at|h,D}), \\ Pr(D_{j+1}|D_j,h) &= (1-\rho)\{1-(1-\beta)^l\} \\ &\quad + (1-\rho)(1-\beta)^l Pr_{at|h,D}. \end{aligned} \tag{12}$$

The transition probabilities from state D_j with action es is,

$$\begin{aligned} Pr(ES_0|D_j,es) &= (1-\rho)(1-\beta)^l(1 - Pr_{at|h,D}), \\ Pr(ES_1|D_j,es) &= (1-\rho)\{1-(1-\beta)^l\}, \\ Pr(ES_a|D_j,es) &= (1-\rho)(1-\beta)^l Pr_{at|h,D}, \\ Pr(P|D_j,es) &= \rho. \end{aligned} \tag{13}$$

Lemma 1: The longer a defender stays on a channel, the higher the chance of avoiding the attack on the next channel.

Proof: The proof of this lemma follows by verifying that $Pr(T_1|T_i, h)$ is an increasing function of i, i.e.,

$$Pr(T_1|T_{i+1}, h) > Pr(T_1|T_i, h). \tag{14}$$

From (4), we can understand that, the more a defender SU stays on a certain channel, the higher the probability of experiencing attack in the next time-slot. However, by combining (4) and (9), we can also find that, the longer a defender stays on a certain channel and transmits successfully, the lower the probability of experiencing attack in the next time-slot when it hops randomly to a new channel. Intuitively, the longer a defender stays on a channel undetected, the more channels attackers have swept—in the current sweeping cycle—unsuccessfully. It provides the defender an extra-room to hop to a random channel from a larger subset of available channels and it increases the probability of experiencing a successful transmission in the next time-slot. ■

Lemma 2: The more successive attacks attackers can perpetrate, the higher the chance of successful attack in the next slot.

Proof: The proof of this lemma follows by verifying that $Pr(T_1|D_j, h)$ is a decreasing function of j, i.e.,

$$Pr(T_1|D_j, h) > Pr(T_1|D_{j+1}, h). \tag{15}$$

Intuitively, the more an SU experiences consecutive transmission failures, the fewer channels it has to hop onto for the current packet transmission. Hence, when it hops, it is more likely to be detected by attackers. Each transmission failure comes with a significant cost to the victim SU. However, as the chance of experiencing an OS-attack increases, so does the chance of detection by the victim SU if action es is employed. This means that SUs should balance their strategy between the encounter of the OS-attack and the detection of an attacker on the new channel, when they hop. ■

C. Optimal Defense Strategy

An MDP consists of four components: a finite set of states, a finite set of actions, transition probabilities, and immediate rewards. We have modeled the defense problem as an MDP. Now, we can find the optimal defense strategy by solving it.

For an MDP, a *policy* is defined as the action to take in each state, i.e., $\pi : S_n \rightarrow a_n$. In other words, a policy maps each state $S \in \mathbb{X}$ to an action $a \in \mathbb{A}$ and is represented by $\pi(S)$. Among all possible policies, the optimal policy returns the maximum expected total discounted payoffs. The value of a state S is defined as the highest expected payoff, starting from the state S and represented as,

$$V^*(s) = \max_{\pi} E\left[\sum_{n=1}^{\infty} \delta^{n-1} U(n) \Big| S = s\right]. \tag{16}$$

Here, the optimal policy $\pi^*(S)$ returns the maximum expected payoff. One important point is that, after making a move from the current state, the remaining part of an optimal policy should still be optimal. Therefore, the first move should maximize the immediate payoff and the future expected payoff, which are conditioned on the current action. This is called Bellman equation [43],

$$\begin{aligned} Q(S, a) &= \sum_{S'} Pr(S'|S, a)(U(S, a, S') + \delta V^*(S')), \\ V^*(S) &= \max_{Q} Q(S, a), \\ \pi^*(S) &= \text{argmax}\, Q(S, a). \end{aligned} \tag{17}$$

Now, we can use the value iteration method to derive the optimal defense strategy and show that the solution has a structure mentioned in Proposition 1.

Proposition 1: The optimal policy can be represented by two critical states $k^* \in \{1, 2, \cdots, K\}$ and $g^* \in \{1, 2, \cdots, G\}$, i.e.,

$$\pi^*(T_i) = \begin{cases} s, & \text{if } T_i < T_{k^*} \\ h, & \text{otherwise} \end{cases}, \pi^*(D_j) = \begin{cases} h, & \text{if } D_j < D_{g^*} \\ es, & \text{otherwise.} \end{cases} \tag{18}$$

Proof: From (4) and (5), the probability of a successful transmission with action *stay* (i.e., $Pr(T_{i+1}|T_i, s)$) decreases over i. Therefore, from the definition of $Q(S, a)$ in (17), $Q(T_i, s) - Q(T_{i-1}, s) < 0$. Now, (9) indicates that the probability of a successful transmission with action *hop* (i.e., $Pr(T_1|T_i, h)$) increases over i. Therefore, $Q(T_i, h) - Q(T_{i-1}, h) > 0$. Now, the optimal action at state T_i is *stay* if $Q(T_i, s) \geq Q(T_i, h)$, or *hop* if $Q(T_i, h) \geq Q(T_i, s)$. Since $Q(T_i, s)$ is decreasing and $Q(T_i, h)$ is increasing, there exists a k^*, where $Q(T_{k^*-1}, s) \geq Q(T_{k^*-1}, h)$ and $Q(T_{k^*}, h) > Q(T_{k^*}, s)$, and $k^* \in \{1, 2, \ldots, K\}$. This concludes the first part of the proof.

Similarly, from (11)-(15), we can show that $Q(D_j, h) < Q(D_{j-1}, h)$ and $Q(D_j, es) > Q(D_{j-1}, es)$. Therefore, there exists a g^*, where $Q(D_{g^*-1}, h) \geq Q(D_{g^*-1}, es)$ and $Q(D_{g^*}, es) > Q(D_{g^*}, h)$, and $g^* \in \{1, 2, \cdots, G\}$. This concludes the second part of the proof. ■

Please note that since the defender hops to another channel when it reaches the state k^*, it refrain itself from entering the states larger than k^*. Therefore, in a scenario where $k^* < K$, the Markov chain becomes irreducible.

Corollary 1: The threshold k^* is decreasing in L, and increasing in both C and M.

Proof: We begin the proof by shedding light on the fact that for any $T_{i'} > T_i$ (where $T_{i'} \in \{2, 3, \cdots, K\}$ and $T_i \in \{1, 2, \cdots, K-1\}$), $Q(T_{i'}, s) - Q(T_i, s)$ is increasing in L and decreasing in K, where K is an increasing function of M. In addition, $Q(T_i, h)$ is decreasing in C, thus verifies that k^* is increasing in C. This concludes the proof. ■

Corollary 2: The threshold g^* is decreasing in L, E, and C, and increasing in M.

Proof: The proof follows by noting that for any $D_{j'} > D_j$ (where $D_{j'} \in \{2, 3, \cdots, G\}$ and $D_j \in \{1, 2, \cdots, G-1\}$), $Q(D_{j'}, h) - Q(D_j, h)$ is increasing in L and C, and

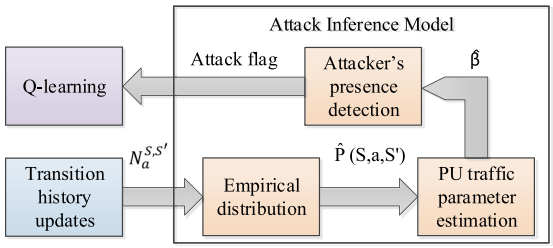


Fig. 8. The Q-learning and attack inference model.

decreasing in M. Moreover, $Q(D_j, es)$ is increasing in E, thus verifies that g^* is decreasing in E. This concludes the proof. ■

Summary: An SU's strategy to use an underutilized channel *as long as plausible* and the *iterative process* of random-OS attacks facilitate the design of the attack and defense problem as an MDP. The proposed defense can be summarized in two aspects: (1) an SU keeps utilizing an underutilized channel for k^* time-slots and then hops to another channel, and (2) after g^* successive transmission failures, an SU takes the action $extra\text{-}sense$. In this paper, we consider that the strategy of attackers remains unchanged, and the strategy of attackers can be learned over time. Nevertheless, an attack and defense problem is comparable to an arms race: the attacker and defender will change their strategies to outsmart each other. Moreover, most existing research works consider that attackers are ubiquitous, i.e., attackers are always present. This consideration demands wireless devices to take defensive actions all the time, even if these actions come at the cost of networking performance. We propose a detection technique (explained in the next section) to infer the presence of attackers and to deploy defensive strategies accordingly.

VI. Proposed Attack Inference Model

In this section, we propose an attack inference model to detect the presence of attackers. The proposed model has two features: 1) it utilizes the in-hand sensing history of the victim; hence, no networking overhead occurs to estimate PU parameters, and 2) it does not violate any policy and hardware constraints; hence, no policy change and extra hardware required. Depending on the parameters of the model, it helps the safeguard process to detect the presence of attackers.

The optimal defense strategy in each state depends on the transition probabilities, which requires the exact knowledge of network parameters (i.e., α, β, m). In reality, it is impossible for a victim to know the exact network parameters to devise the MDP, especially, when it can change over time (e.g., attackers' presence is uncertain, the number of attackers may change, and PU activities may change). Therefore, an SU must learn the MDP over time. A *model-based* learning technique requires the Markov process to exhibit constant parameters over time, and it has a limitation in scalability; hence, a *model-free* learning is best suitable for this scenario. We employ the Q-learning technique that works as a model-free off-policy method, learns the game without the need of transition probabilities, and fits well with sudden changes in MDP parameters. Fig. 8 shows the framework of the proposed attack inference model and Q-learning.

A. Q-Learning

The Q-learning tries to approximate the unknown transition probability by the empirical distribution of states that have been experienced over time. It iteratively calculates and updates the Q-value based on the state-action tuple (S, a, S').

$$\begin{aligned} Q_n(S,a) =& Q_{n-1}(S,a) + \gamma[\{R(S,a,S') + \delta V_n(S')\} \\ & - Q_{n-1}(S,a)], \\ V_n(S) =& \max_Q \; Q_n(S,a), \end{aligned} \tag{19}$$

where γ is the learning rate and δ is the discount factor.

In Q-learning, there is no fixed policy while learning the MDP and agents take random actions (with probability ϵ) to discover the MDP. However, the randomness decreases over time (i.e., $\epsilon \to 0$) and defenders are more likely to take actions with highest Q-values. After Q-values converge, the learning process ends. The optimal policy after the learning period is,

$$\pi^*(S) = \text{argmax}\, Q_n(S,a), \quad a \in \mathbb{A}, S \in \mathbb{X}. \tag{20}$$

In quest of learning the optimal policy, the defender makes mistakes and takes random decisions to explore the MDP. Hence, Q-learning engenders a cost in performance, and it is represented by *regret* that quantifies the difference between the expected rewards (while learning) and the optimal rewards. Therefore, the more the defender learns, the fewer mistakes it makes (i.e., regret is a decreasing function of time).

Hence, to minimize the learning cost, the attack inference model re-initializes the learning process (i.e., reinitialize ϵ) when the model detects the presence of OS-DoS attackers.

B. Attacker's Presence Detection

In this approach, benign SUs initiate their operation with three policies: 1) stay on the current channel until a transmission failure (i.e., $\pi(T) = s$) occurs, 2) hop to another channel after a transmission failure (i.e., $\pi(D) = h$), and 3) hop to another channel after sensing the channel busy in the sensing interval (i.e., $\pi(P) = h$). Without detecting the presence of OS-attackers, Q-learning does not employ the action es.

With recorded historic states and actions, SUs are able to compute the occurrences of transitions given any action. For example, the notation $N_a^{S,S'}$ represents the total number of transitions from state S to S', taking action a.

We define $T_p \triangleq \max\{T : N_s^{T_i,T_{i+1}} = 0\}$ (e.g., under-attack, $T_p = K$). From (5), we can understand that the absence of attack (i.e., $Pr_{at|s} = 0$) will result in an empirical probability $\widehat{Pr}(D_1|T_i, s) = \frac{N_s^{T_i,D_1}}{N_s^{T_i,D_1} + N_s^{T_i,P} + N_s^{T_i,T_{i+1}}}$ that is close to the probability of transmission failure by PUs only,

$$Pr(D_1|T_i, s, Pr_{at|s} = 0) = (1 - \widehat{\beta})\{1 - (1 - \widehat{\beta})^l\}, \tag{21}$$

where $\widehat{\beta}$ represents the PU traffic parameter from empirical observations, which will be explained later in this section.

Now, with the presence of attackers (i.e., $Pr_{at|s} > 0$), $\widehat{Pr}(D_1|T_i, s) > Pr(D_1|T_i, s)$. We represent this by,

$$X_i^n = \frac{\widehat{Pr}_n(D_1|T_i, s) - Pr_n(D_1|T_i, s; Pr_{at|s} = 0)}{Pr_n(D_1|T_i, s; Pr_{at|s} = 0)}, \quad (22)$$

where $\widehat{Pr}_n$ and Pr_n represent empirical probabilities after n time-slots (i.e., $\widehat{Pr}_n$ and Pr_n are running parameters).

SUs track these values of X_i over time. From (4) and (5), we can observe that $Pr_{at|s}$ increases with the residence time of SUs on a channel. Therefore, to deduce the presence of attackers, X_i values should conform to the requirement below,

$$X_1^n < X_2^n < \cdots < X_{p-1}^n < X_p^n. \quad (23)$$

This inequality characterizes the primary condition to detect the random-OS attack. It differentiates the random-OS attack from the naive attack where m attackers randomly choose m channels in each slot with equal probabilities (i.e., m/M), and it does not consider which channels have been detected in the past. Therefore, X_i^n will not meet the requirement in (23), instead, the values of X_i^n will lie within a close approximation,

$$X_1^n = X_2^n = \cdots = X_{p-1}^n = X_p^n \approx c, \quad (24)$$

where c is a constant.

Since each channel has an equal probability of encountering attack in the naive approach, hopping strategy cannot reduce the risk of attacks. Moreover, the hopping cost makes it a futile effort to avoid the attack by hopping from one channel to another. Hence, SUs stay on the same channel until they sense the PU reappearance or experience a transmission failure.

Next, we consider a safety margin τ to finally trigger the presence of attackers in the network. Besides a safety margin, τ also works as a trade-off parameter between performance and security. We compare the value of X_1^n to τ to decide the presence of attackers. Since the state T_1 is visited more frequently than other T states, we make an educated choice of comparing the safety margin with X_1^n. Therefore, the second requirement is,

$$X_1^n > \tau. \quad (25)$$

We can further control it by starting a counter when (23) and (25) are met, then triggering the attack flag once these requirements are consistently met for a certain time.

C. PU Traffic Parameter Estimation

We define $\mathbb{S} \triangleq \{T_1, T_2, \cdots, T_p - 1\}$ and $\mathbb{H} \triangleq \{P, D, T_p\}$. Now, given the state transition history $N_a^{S,S'}$ over time, we can deduce the empirical value of the PU traffic parameter,

$$\widehat{\beta} = \frac{\sum_{T\in\mathbb{S}} N_s^{T,P}}{\sum_{T\in\mathbb{S}} \left(N_s^{T,P} + N_s^{T,D} + N_s^{T,T+1} \right)}, \quad (26)$$

$$\widehat{\rho} = \frac{\sum_{S\in\mathbb{H}} N_h^{S,P}}{\sum_{S\in\mathbb{H}} \left(N_h^{S,P} + N_h^{S,D} + N_h^{S,T_1} \right)}. \quad (27)$$

The empirical value of $\widehat{\beta}$ remains unaffected by the presence of attackers. It depends on the results from the sensing interval, and OS-attackers remain inactive during this interval. Therefore, (26) provides a close estimation of the actual PU parameter to decide the presence of attackers in the network.

D. Empirical Distribution

For transmission states (i.e., T_i), the estimated probability from sample transitions are,

$$\widehat{Pr}(S'|T_i, s) = \frac{N_s^{T_i,S'}}{\sum_{S'} N_s^{T_i,S'}}, \quad (28)$$

where $S' \in \{P, D_1, T_i + 1\}$ and $T_i \in \{T_1, T_2, \cdots, T_p - 1\}$.

For transmission failure states (i.e., D_i), the estimated probability from sample transitions are,

$$\widehat{Pr}(S'|D_j, h) = \frac{N_h^{D_j,S'}}{\sum_{S'} N_h^{D_j,S'}}, \quad (29)$$

where $S' \in \{P, D_j + 1, T_1\}$ and $D_j \in \{D_1, D_2, \cdots, D_G - 1\}$.

And, for the busy state (i.e., P), the estimated probability is,

$$\widehat{Pr}(S'|P, h) = \frac{N_h^{P,S'}}{\sum_{S'} N_h^{P,S'}}, \quad (30)$$

where $S' \in \{P, T_1, D_1\}$.

Summary: Unlike previous research, we consider the absence and the presence of attackers. It helps us to avoid unnecessary defensive measures (e.g., action es), when attackers are absent. When attackers initiate an OS-DoS attack, the proposed attack inference model detects the attack using empirical observations from its sensing results and re-initializes the Q-learning process (i.e., re-initialization of ϵ) to minimize the regret (i.e., learning cost) and to take appropriate action (i.e., action es).

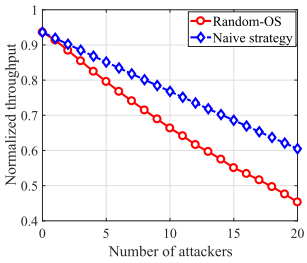
VII. Performance Evaluation

In this section, we present simulation results to evaluate the performance of our proposed research. Here, we consider that the victim SU detects an attacker, but does not oust it from the network; the appropriate attack response (e.g., network isolation, bandwidth limitation, and network elimination) is an open research issue. Unless otherwise stated, the simulation parameters are:

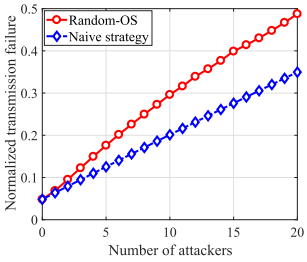
The presented simulation results are the average of 100 independent trials.

A. Random-OS Attack

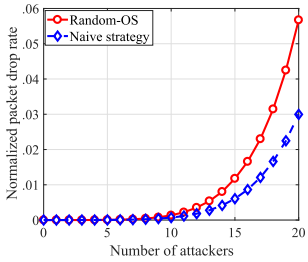
In this work, we consider that attackers do not have any predetermined knowledge of the victim's channel hopping



(a) Normalized throughput.



(b) Normalized transmission failure.



(c) Normalized packet drop rate.

Fig. 9. Performance of the random-OS attack.

sequence and operating channels. Therefore, we discard the comparison with conventional OS-DoS attack where the victim experiences null throughput regardless of the number of attackers (i.e., unrealistic scenario). Fig. 9 demonstrates the performance of the random-OS strategy in contrast to the naive approach, where attackers do not consider the knowledge of which channels have been visited in the past, instead randomly select channels at each time-slot.

In Fig. 9(a), the normalized throughput is shown, where victims experience less throughput in the random-OS attack due to the iterative process and the re-randomization technique of random-OS. Likewise, victims of the random-OS attack suffer more transmission failures (Fig. 9(b)) and higher rate of packet drop (Fig. 9(c)). However, transmission failures are not enough to cause significant packet drop or DoS attack unless attackers can perpetrate it consecutively. This reflects in Fig. 9 (c) where the packet drop rate follows a different trend than the rate of transmission failure; the packet drop rate starts to increase exponentially after $m = 10$. Therefore, in this scenario, more than 10 attackers are required to cause significant damage to the victim.

B. Critical States

We demonstrate the critical states k^* and g^* of the optimal policy (Fig. 10) derived from the value iteration of the MDP, with the change in the number of attackers (m), the cost of transmission failure (L), the reward of attacker detection (E), the cost of channel hopping (C), and the number of operating channels (M).

Effect of m: In Fig. 10(a)-(h), both k^* and g^* decrease with the increase in the number of attackers. As m increases, attackers can visit more channels in each time-slot; hence, K starts to decrease, and SUs have less channels to hop on after each transmission failure. Therefore, the channel residence time decreases and SUs have to hop more frequently to avoid the attack.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Communication gain, R	5
Cost of transmission failure, L	5
Hopping cost, C	1
Cost of busy channel, B	1
Penalty for policy violation, F	50
Maximum transmission attempt, G	7
Cost of packet drop, Q	$G \cdot L$
Reward for detecting an attacker, E	20
SU packet length l	5
Discount factor, δ	0.95
Learning rate, γ	$1/\sqrt{\text{number of time-slots}}$
PU parameters	$\beta = 0.01$, $\rho = 0.1$
Number of channels, M	60

Effect of L: In Fig. 10(a) and 10(e), as the cost of transmission failure L increases, SUs tend to hop more to avoid imminent transmission failures, thus k^* decreases. However, g^* demonstrates relatively less sensitivity towards changes in L due to the significantly high cost of Q. In transmission failure states, choosing action es over h means that the defender has to compromise its packet transmission regardless of the outcome of the action es; hence, the defender is reluctant to take action es.

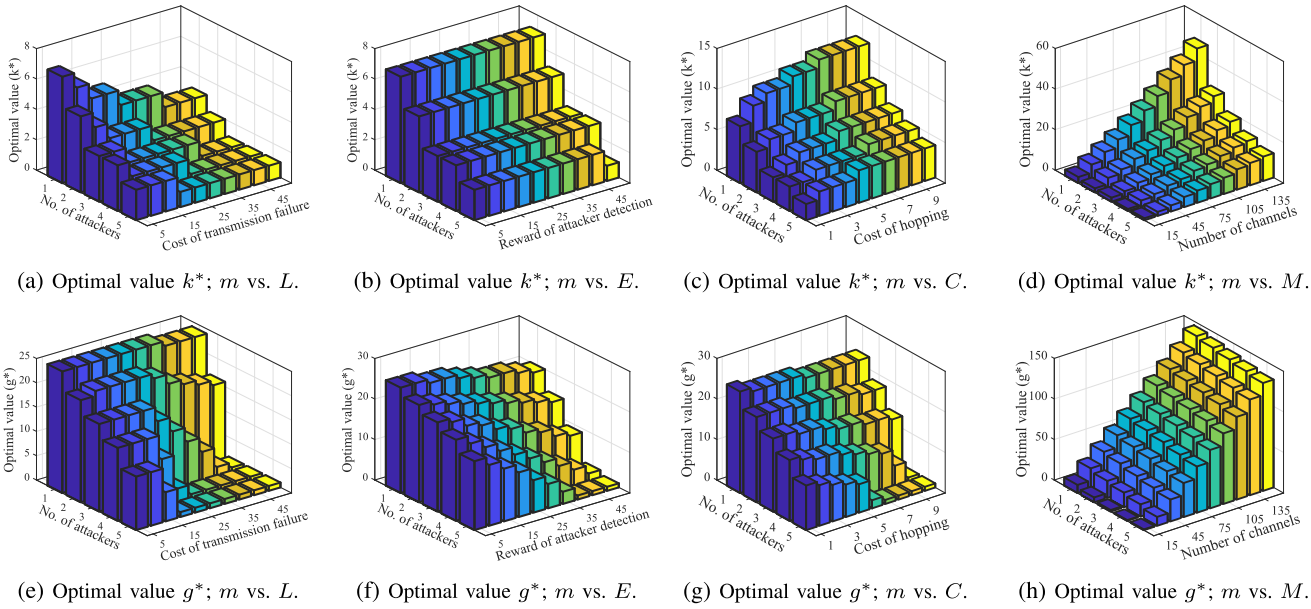
Effect of E: In Fig. 10(b), k^* remains almost insensitive to the change in the reward of attacker detection E. Because E largely dictates the action es only, $stay$ and hop from transmission states remain out of its influence. For the similar reason, in Fig. 10(f), g^* illustrates linear sensitivity to the change in E. Therefore, as the reward for detecting an attacker increases, SUs become more motivated to take the action es instead of hop, to detect attackers. The parameter E works as a trade-off parameter between the networking performance and the security performance. Lower and higher values of E mean that victims have more tendency toward avoiding and victims have more tendency toward detecting OS-attackers, respectively.

Effect of C: As discussed in Section V, channel hopping engenders insignificant cost in terms of channel throughput; we quantify this cost by C. In Fig. 10(c), we can observe that k^* increases with C. As C increases, defenders become reluctant to take action hop and stays in a channel longer. Therefore, the cost of hopping significantly impacts the proposed defense strategy because defenders become limited in their capability to utilize the channel diversity a multi-channel network has to offer. However, unlike k^*, g^*—though exhibits very low sensitivity—decreases with C (Fig. 10(g)).

Effect of M: As the number of channels M increases, the maximum channel residence time K increases. Therefore, attackers have more channels to sweep through and defenders have more time to stay on a channel. In Fig. 10(d), we can observe that k^* increases linearly with the increase of M. Similarly, as M increases, defenders experience more incentive to hop through different channels than to detect attackers. As a result, g^* increases with M.

C. Hide and Seek

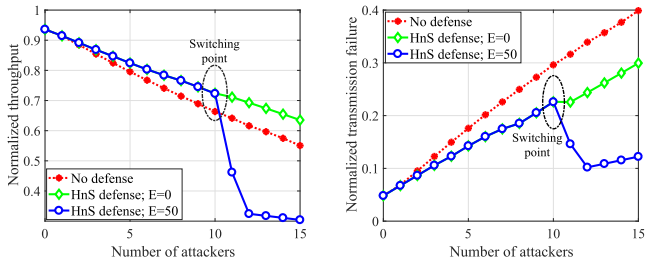
Fig. 11(a) compares the performance of our proposed hide and seek strategy with three scenarios: no defense, hide and



(a) Optimal value k^*; m vs. L. (b) Optimal value k^*; m vs. E. (c) Optimal value k^*; m vs. C. (d) Optimal value k^*; m vs. M.

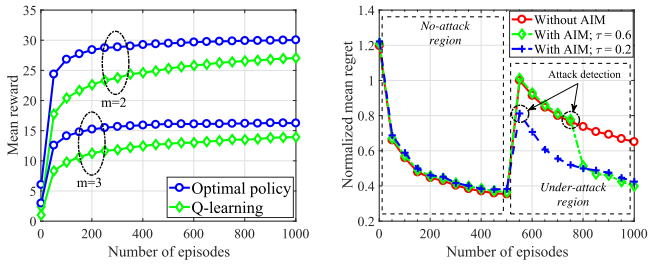
(e) Optimal value g^*; m vs. L. (f) Optimal value g^*; m vs. E. (g) Optimal value g^*; m vs. C. (h) Optimal value g^*; m vs. M.

Fig. 10. The sensitivity of optimal values to the changes in L, E, C, and M.



(a) Normalized throughput. (b) Normalized transmission failure.

Fig. 11. Performance of Hide and Seek.



(a) Q-learning. (b) Attack inference model.

Fig. 12. Performance of Q-learning and attack inference model.

seek with no reward ($E = 0$), and hide and seek with a high reward ($E = 50$). It illustrates that both $E = 0$ and $E = 50$ follow the same line until the number of attackers surpasses $m = 10$ (when $E = 50$); the throughput drops below the no defense line afterwards. We call this moment the *switching point* after which the victim prefers to detect attackers (using the action es) rather than avoiding them (using the action hop); hence, the throughput drops. As E increases, the victim becomes more motivated to detect attackers and the switching point moves to the left. As discussed earlier, E works as a tuning parameter between the networking and security performance. Likewise, in Fig. 11(b), we can observe that the transmission failure decreases after the switching point. However, after $m = 12$, it starts to increase again due to the increasing number of attackers.

D. Q-Learning and Attack Inference Model

We evaluate the performance of Q-learning (Fig. 12 (a)) by showing the difference in mean reward after each episode between an SU that knows the optimal values and an SU that learns the MDP over time via Q-learning. Here, we can observe that in both cases (i.e., $m = 2$ and $m = 3$), the reward converges to the optimal reward. However, with $m = 3$, the agent converges more quickly due to the fewer amount of states.

In Fig. 12(b), the performance of our proposed attack inference model is shown with different values of the threshold τ. We change the scenario from $m = 0$, $M = 10$ to $m = 2$, $M = 10$ at $epsiode = 501$. As the MDP progresses, an SU takes fewer random actions (i.e., ϵ decreases); hence, it takes more time to track the changes without the assistance from the attack inference model. The proposed model assists the Q-learning to detect changes in the MDP and re-initializes the parameter ϵ to minimize the regret based on the threshold τ. With $\tau = 0.2$ and $\tau = 0.6$, the attack inference model detects the presence of the attacker on $epsiode = 549$ and $epsiode = 753$, respectively. Hence, a lower value of τ assists the SU to track the changes sooner and yields in less regret.

VIII. Conclusion

In this paper, we proposed a new strategy, random-OS, to perpetrate OS-DoS attacks without any predetermined knowledge of the victim's channel hopping sequence. Afterwards, we proposed an MDP-based safeguard approach, hide and

seek, to avoid and detect the proposed attack. We showed that by hopping to random channels, an SU can avoid the OS-DoS attack, and when it becomes necessary (based on rewards) to detect interference, it employs an extra-sensing interval to detect the attack. Here, the victim SU learns the optimal policy using Q-learning. Lastly, we proposed an attack inference model to detect the presence of attackers and to reinitialize the learning process to incur less regret.

Numerical investigations and simulation results showed that the random-OS outperforms the naive approach and the hide and seek improves the network throughput without ousting the attackers. To the best of our knowledge, this is the first work that introduced a new avenue in designing defensive measures of OS-attack without changing the FCC policy.

REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 40–48, Apr. 2008.

[2] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[3] IEEE 802.22 Working Group *et al.*, IEEE Standard for Wireless Regional Area Networks Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands, IEEE Standard, 2011.

[4] D. Hlavacek and J. M. Chang, "A layered approach to cognitive radio network security: A survey," *Elsevier Comput. Netw.*, vol. 75, pp. 414–436, 2014.

[5] R. K. Sharma and D. B. Rawat, "Advances on security threats and countermeasures for cognitive radio networks: A survey," *IEEE Commun. Surv. Tut.*, vol. 17, no. 2, pp. 1023–1043, Apr.–Jun. 2015.

[6] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "SpecGuard: Spectrum misuse detection in dynamic spectrum access systems," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2925–2938, Dec. 2018.

[7] A. G. Fragkiadakis, E. Z. Tragos, and I. G. Askoxylakis, "A survey on security threats and detection techniques in cognitive radio networks," *IEEE Commun. Surv. Tut.*, vol. 15, no. 1, pp. 428–445, Oct.–Dec. 2013.

[8] R. Chen and J.-M. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *Proc. IEEE Workshop Netw. Technologies Softw. Defined Radio Netw.*, 2006, pp. 110–119.

[9] Y. Song and J. Xie, "Finding out the liars: Fighting against false channel information exchange attacks in cognitive radio ad hoc networks," in *Proc. IEEE Global Commun. Conf.*, 2012, pp. 2095–2100.

[10] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio networks," in *Proc. IEEE Int. Symp. New Frontiers Dyn. Spectrum Access Netw.*, 2005, pp. 137–143.

[11] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Phys. Commun.*, vol. 4, no. 1, pp. 40–62, 2011.

[12] M. Hossain and J. Xie, "Impact of off-sensing attacks in cognitive radio networks," in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.

[13] T. Bansal, B. Chen, and P. Sinha, "FastProbe: Malicious user detection in cognitive radio networks through active transmissions," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 2517–2525.

[14] X. Liu and J. Xie, "A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 2085–2093.

[15] X. Liu and J. Xie, "A 2D heterogeneous rendezvous protocol for multi-wideband cognitive radio networks," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[16] G. Baldini *et al.*, "Security aspects of policy controlled cognitive radio," in *Proc. Int. Conf. New Technologies, Mobility Secur.*, 2012, pp. 1–5.

[17] J. L. Burbank, "Security in cognitive radio networks: The required evolution in approaches to wireless network security," in *Proc. 3rd Int. Conf. Cogn. Radio Oriented Wireless Netw. Commun.*, 2008, pp. 1–7.

[18] M. Hossain and J. Xie, "Covert spectrum handoff: An attack in spectrum handoff processes in cognitive radio networks," in *Proc. IEEE Global Commun. Conf.*, 2018, pp. 1–6.

[19] M. Hossain and J. Xie, "Detection of hidden terminal emulation attacks in cognitive radio-enabled IoT networks," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.

[20] M. Hossain and J. Xie, "Hide and seek: A defense against off-sensing attack in cognitive radio networks," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 613–621.

[21] M. Hossain and J. Xie, "Hidden terminal emulation: An attack in dense IoT networks in the shared spectrum operation," in *IEEE Proc. IEEE Global Commun. Conf.*, 2019, pp. 1–6.

[22] M. Hossain and J. Xie, "Third eye: Context-aware detection for hidden terminal emulation attacks in cognitive radio-enabled IoT networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 214–228, Mar. 2020.

[23] Z. Jin, S. Anand, and K. Subbalakshmi, "Performance analysis of dynamic spectrum access networks under primary user emulation attacks," in *Proc. IEEE Global Commun. Conf.*, 2010, pp. 1–5.

[24] Z. Jin, S. Anand, and K. P. Subbalakshmi, "Impact of primary user emulation attacks on dynamic spectrum access networks," *IEEE Trans. Commun.*, vol. 60, no. 9, pp. 2635–2643, Sep. 2012.

[25] M. Dabaghchian, A. Alipour-Fanid, K. Zeng, Q. Wang, and P. Auer, "Online learning with randomized feedback graphs for optimal PUE attacks in cognitive radio networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2268–2281, Oct. 2018.

[26] M. Hossain and J. Xie, "Off-sensing and route manipulation attack: A cross-layer attack in cognitive radio based wireless mesh networks," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 1376–1384.

[27] 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, Aug. 2011.

[28] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Commun. Surv. Tut.*, vol. 13, no. 2, pp. 245–257, Apr.–Jun. 2010.

[29] L. Xiao, T. Chen, J. Liu, and H. Dai, "Anti-jamming transmission Stackelberg game with observation errors," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 949–952, Jun. 2015.

[30] L. Xiao, J. Liu, Q. Li, N. B. Mandayam, and H. V. Poor, "User-centric view of jamming games in cognitive radio networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 12, pp. 2578–2590, Dec. 2015.

[31] G.-Y. Chang, S.-Y. Wang, and Y.-X. Liu, "A jamming-resistant channel hopping scheme for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6712–6725, Oct. 2017.

[32] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *Proc. IEEE Conf. Comput. Commun.*, 2007, pp. 2526–2530.

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

[34] M. K. Hanawal, M. Abdel-Rahman, and M. Krunz, "Game theoretic anti-jamming dynamic frequency hopping and rate adaptation in wireless systems," in *Proc. 12th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw.*, 2014, pp. 247–254.

[35] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 1, pp. 2–14, Mar. 2019.

[36] M. K. Hanawal, D. N. Nguyen, and M. Krunz, "Jamming attack on in-band full-duplex communications: Detection and countermeasures," in *Proc. IEEE Conf. Comput. Commun.*, 2016, pp. 1–9.

[37] N. Nguyen-Thanh, P. Ciblat, A. T. Pham, and V. Nguyen, "Surveillance strategies against primary user emulation attack in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 9, pp. 4981–4993, Sep. 2015.

[38] D. Ta, N. Nguyen-Thanh, P. Maillé, and V. Nguyen, "Strategic surveillance against primary user emulation attacks in cognitive radio networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 582–596, Sep. 2018.

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

[40] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with MAC-layer sensing in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 5, pp. 533–545, May. 2008.

[41] H. Kim and K. G. Shin, "In-band spectrum sensing in cognitive radio networks: Energy detection or feature detection?" in *Proc. ACM MobiCom*, 2008, pp. 14–25.

[42] R. Bell, "Maximum supported hopping rate measurements using the universal software radio peripheral software defined radio," in *Proc. GNU Radio Conf.*, vol. 1, no. 1, Sep. 2016.

[43] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

Moinul Hossain (Member, IEEE) received the B.S. degree in electronics and communication engineering from Khulna University of Engineering and Technology, Khulna, Bangladesh, in 2011, and the Ph.D. degree in electrical engineering from the University of North Carolina at Charlotte (UNC-Charlotte), Charlotte, NC, USA, in 2020. In August 2020, he joined the Department of Computer and Information Sciences, Towson University, Towson, MD, USA, as an Assistant Professor. His research interests include wireless network security, wireless networking, Internet of Things, and spectrum coexistence.

Jiang Xie (Fellow, IEEE) received the B.E. degree from Tsinghua University, Beijing, China, in 1997, the M.Phil. degree from The Hong Kong University of Science and Technology, Hong Kong, in 1999, and the M.S. and Ph.D. degrees from Georgia Institute of Technology, Atlanta, GA, USA, in 2002 and 2004, respectively, all in electrical and computer engineering. In August 2004, she joined the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte (UNC-Charlotte), Charlotte, NC, USA, as an Assistant Professor, where she is currently a Full Professor. Her current research interests include resource and mobility management in wireless networks, mobile computing, Internet of Things, and cloud/edge computing. She is on the Editorial Boards for the IEEE/ACM TRANSACTIONS ON NETWORKING and *Journal of Network and Computer Applications* (Elsevier). She received the US National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award in 2010, a Best Paper Award from IEEE Global Communications Conference (Globecom 2017), a Best Paper Award from IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2010), and a Graduate Teaching Excellence Award from the College of Engineering at UNC-Charlotte in 2007. She is a senior member of the ACM.