Online Learning with Randomized Feedback Graphs for Optimal PUE Attacks in Cognitive Radio Networks

Monireh Dabaghchian, *Student Member, IEEE,* Amir Alipour-Fanid, *Student Member, IEEE,* Kai Zeng, *Member, IEEE,* Qingsi Wang, *Member, IEEE,* Peter Auer, *Member, ACM*

***Abstract*—In a cognitive radio network, a secondary user learns the spectrum environment and dynamically accesses the channel where the primary user is inactive. At the same time, a primary user emulation (PUE) attacker can send falsified primary user signals and prevent the secondary user from utilizing the available channel. The best attacking strategies that an attacker can apply have not been well studied. In this paper, for the first time, we study optimal PUE attack strategies by formulating an online learning problem where the attacker needs to dynamically decide the attacking channel in each time slot based on its attacking experience. The challenge in our problem is that since the PUE attack happens in the spectrum sensing phase, the attacker cannot observe the reward on the attacked channel. To address this challenge, we utilize the attacker's observation capability. We propose online learning-based attacking strategies based on the attacker's observation capabilities. Through our analysis, we show that with no observation within the attacking slot, the attacker loses on the regret order, and with the observation of at least one channel, there is a significant improvement on the attacking performance. Observation of multiple channels does not give additional benefit to the attacker (only a constant scaling) though it gives insight on the number of observations required to achieve the minimum constant factor. Our proposed algorithms are optimal in the sense that their regret upper bounds match their corresponding regret lower-bounds. We show consistency between simulation and analytical results under various system parameters.**

***Index Terms* — Online learning, multi-armed-bandits, feedback graphs, cognitive radio networks, PUE attacks.**

I. Introduction

NOWADAYS the demand to wireless bandwidth is growing rapidly due to the increasing growth in various mobile and IoT applications, which raises the spectrum shortage problem. To address this problem, Federal Communications Commission (FCC) has authorized opening spectrum bands (e.g., 3550-3700 MHz and TV white space) owned by licensed primary users (PU) to unlicensed secondary users (SU) when the primary users are inactive [1, 2]. Cognitive radio (CR) is a key technology that enables secondary users to learn the spectrum environment and dynamically access the best available channel. Meanwhile, an attacker can send signals emulating the primary users to manipulate the spectrum environment, preventing a secondary user from utilizing the available channel. This attack is called primary user emulation (PUE) attack [3, 4, 5, 6, 7].

Existing works on PUE attacks mainly focus on PUE attack detection [8, 9] and defending strategies [5, 6]. However, there is a lack of study on the optimal PUE attacking strategies. Better understanding of the optimal attacking strategies will enable us to quantify the severeness or impact of a PUE attacker on the secondary user's throughput. It will also shed light on the design of defending strategies.

In practice, an attacker may not have any prior knowledge of the primary user activity characteristics or the secondary user dynamic spectrum access strategies. Therefore, it needs to learn the environment and attack at the same time. In this paper, for the first time, we study the optimal PUE attacking strategies without any assumption on the prior knowledge of the primary user activity or secondary user accessing strategies. We formulate this problem as an online learning problem. We use the words *play*, *action taking*, and *attack* interchangeably throughout the paper.

Different from all the existing works on online learning based PUE attack defending strategies [5, 6], in our problem, an attacker cannot observe the reward on the attacked channel. Considering a time-slotted system, the PUE attack usually happens in the channel sensing period, in which a secondary user attempting to access a channel conducts spectrum sensing to decide the presence of a primary user. If a secondary user senses the attacked channel, it will believe the primary user is active so that it will not transmit the data in order to avoid interfering with the primary user. In this case, the PUE attack is effective since it disrupts the secondary user's communication and affects its knowledge of the spectrum availability. In the other case, if there is no secondary user attempting to access the attacked channel, the attacker makes no impact on the secondary user, so the attack is ineffective. However, the attacker cannot differentiate between the two cases when it launches a PUE attack on a channel because no secondary user will be observed on the attacked channel no matter a secondary user has ever attempted to access the channel or not.

The key for the attacker to launch effective PUE attacks is to learn which channel or channels a secondary user

Manuscript received ; revised ; accepted ; approved by IEEE/ACM Transactions on Networking Editor C. W. Tan. This work was partially supported by the National Scinece Foundation under grant No. CNS-1502584 and CNS-1464487. This paper is an extended version of the conference paper presented at the IEEE CNS, Philadelphia, USA, 2016.

Monireh Dabaghchian, Amir Alipour-Fanid, and Kai Zeng are with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, 22030 USA (e-mail: {mdabaghc, aalipour, kzeng2}@gmu.edu).

Qingsi Wang is with Google Inc., Kirkland, WA 98033 USA (email: qingsi@umich.edu).

Peter Auer is with the Department of Mathematics and Information Technology, Montanuniversitaet Leoben, Austria (email: auer@unileoben.ac.at).

is most likely to access. To do so, the attacker needs to make observations on the channels. As a result, the attacker's performance is dependent on its observation capability. We define the observation capability of the attacker as the number of the channels it can observe within the same time slot after launching the attack. We propose two attacking schemes based on the attacker's observation capability. In the first scheme called *Attack-OR-Observe* (AORO), an attacker if attacks, cannot make any observation in the same time slot due to the short slot duration in highly dynamic systems [10] or when the channel switching time is long. We call this attacker an attacker with no observation capability within the same time slot (since no observation is possible if it attacks). To learn the most rewarding channel though, the attacker needs to dedicate some time slots for observation only without launching any attacks. On the other hand, an attacker could be able to attack a channel in the sensing phase and observe other channels in the data transmission phase within the same time slot if it can switch between channels fast enough. We call this attacking scheme, *Attack-But-Observe-Another* (ABOA).

For the AORO case, we propose an online learning algorithm called POLA *-Play or Observe Learning Algorithm-* to dynamically decide between attack and observation and then to choose a channel for the decided action, in a given time slot. Through theoretical analysis, we prove that POLA achieves a regret in the order of $\tilde{O}(\sqrt[3]{T^2})$ where T is the number of slots the CR network operates. Its higher slope regret is due to the fact that it cannot attack (gain reward) and observe (learn) simultaneously in a given time slot. We show the optimality of our algorithm by matching its regret upper bound with its lower bound.

For the ABOA case, we propose PROLA *-Play and Random Observe Learning Algorithm-* as another online learning algorithm to be applied by the PUE attacker. PROLA dynamically chooses the attacking and observing channels in each time slot. The PROLA learning algorithm's observation policy fits a specific group of graphs called time-varying partially observable feedback graphs [11]. It is derived in [11] that these feedback graphs lead to a regret in the order of $\tilde{O}(\sqrt[3]{T^2})$. However, our algorithm, PROLA, is based on a new theoretical framework and we prove its regret is in the order of $\tilde{O}(\sqrt{T})$. We then prove its regret lower bound is in the order of $\tilde{\Omega}(\sqrt{T})$ which matches its upper bound and shows our algorithm's optimality.

Both algorithms proposed address the attacker's observation capabilities and can be applied as optimal PUE attacking strategies without any prior knowledge of the primary user activity and secondary user access strategies.

We further generalize PROLA to multi-channel observations where an attacker can observe multiple channels within the same time slot. By analyzing its regret, we come to the conclusion that increasing the observation capability from one to multiple channels does not give additional benefit to the attacker in terms of regret order. It only improves the constant factor of the regret.

Our main contributions are summarized as follows:

- We formulate the PUE attack as an online learning problem without any assumption on the prior knowledge of either primary user activity characteristics or secondary user dynamic channel access strategies.
- We propose two attacking schemes, AORO and ABOA, that model the behavior of a PUE attacker. For the AORO case, a PUE attacker, in a given time slot, dynamically decides either to attack or observe; then chooses a channel for the decision it made. While in the ABOA case, the PUE attacker dynamically chooses one channel to attack and chooses at least one other channel to observe within the same time slot.
- We propose an online learning algorithm POLA for the AORO case. POLA achieves an optimal learning regret in the order of $\tilde{\Theta}(\sqrt[3]{T^2})$. We show its optimality by matching its regret lower and upper bounds.
- For the ABOA case, we propose an online learning algorithm, PROLA, to dynamically decide the attacking and observing channels. We prove that PROLA achieves an optimal regret order of $\tilde{\Theta}(\sqrt{T})$ by deriving its regret upper bound and lower bound. This shows that:
  1) with an observation capability of at least one within the attacking slot, there is a significant improvement on the performance of the attacker.
  2) Theoretical contribution: For PROLA, despite observing the actions partially, it achieves an optimal regret order of $\tilde{\Theta}(\sqrt{T})$ which is better than a known bound of $\tilde{\Theta}(\sqrt[3]{T^2})$. We accomplish it by proposing randomized time-variable feedback graphs.
- The algorithm and the analysis of the PROLA are further generalized to multi-channel observations.
- We conduct simulations to evaluate the performance of the proposed algorithms under various system parameters.

Through theoretical analysis and simulations under various system parameters, we have the following findings:

- With no observation at all within the attacking slot, the attacker loses on the regret order. While with the observation of at least one channel, there is a significant improvement on the attacking performance.
- Observation of multiple channels in the PROLA algorithm does not give additional benefit to the attacker in terms of regret order. The regret is proportional to $\sqrt{1/m}$, where m is the number of channels observed by the attacker. Based on this relation, the regret is a monotonically decreasing and a convex function of the number of observing channels. As more observing channels are added, the reduction in the regret becomes marginal. Therefore, a relatively small number of observing channels is sufficient to approach a small constant factor.
- The attacker's regret is proportional to $\sqrt{K \ln K}$, where K is the total number of channels.

II. Related Work

A. PUE Attacks in Cognitive Radio Networks

Existing work on PUE attacks mainly focus on PUE attack detection [8, 9] and defending strategies [5, 6].

There are few work discussing attacking strategies under dynamic spectrum access scenarios. In [5], the attacker applies partially observable Markov decision process (POMDP)

framework to find the attacking channel in each time slot. It is assumed that the attacker can observe the reward on the attacking channel. That is, the attacker knows if a secondary user is ever trying to access a channel or not. In [6], it is assumed that the attacker is always aware of the best channel to attack. However, there is no methodology proposed or proved on how the best attacking channel can be decided.

The optimal PUE attack strategy without any prior knowledge of the primary user activity and secondary user access strategies is not well understood. In this paper, we fill this gap by formulating this problem as an online learning problem. Our problem is also unique in that the attacker cannot observe the reward on the attacking channel due to the nature of PUE attack.

B. Multi-armed Bandit Problems

There is a rich literature about online learning algorithms. The most related ones to our work are multi-armed bandit (MAB) problems [12, 13, 14, 15, 16]. The MAB problems have many applications in cognitive radio networks with learning capabilities [6, 17, 18, 19]. In such problems, an agent plays a machine repeatedly and obtains a reward when it takes a certain action at each time. Any time when choosing an action the agent faces a dilemma of whether to take the best rewarding action known so far or to try other actions to find even better ones. Trying to learn and optimize his actions, the agent needs to trade off between exploration and exploitation. On one hand the agent needs to explore all the actions often enough to learn which is the most rewarding one and on the other hand he needs to exploit the believed best rewarding action to minimize his overall regret.

For most existing MAB frameworks as explained, the agent needs to observe the reward on the taken action. Therefore, these frameworks cannot be directly applied to our problem where a PUE attacker cannot observe the reward on the attacking channel. Most recently, Alon et al. generalize the cases of MAB problems into feedback graphs [11]. In this framework, they study the online learning with side observations. They show in their work that if an agent takes an action without observing its reward, but observes the reward of all the other actions, it can achieve an optimal regret in the order of $\tilde{O}(\sqrt{T})$. However, the agent can only achieve a regret of $\tilde{O}(\sqrt[3]{T^2})$ if it cannot observe the rewards on all the other actions simultaneously. In other words, even one missing edge in the feedback graph leads to the regret of $\tilde{O}(\sqrt[3]{T^2})$.

In this paper, we propose two novel online learning policies. The first one, POLA, for the case when only either observation or attack is possible within a time slot, is shown to achieve a regret in the order of $\tilde{O}(\sqrt[3]{T^2})$. We then advance this theoretical study by proposing a strategy, PROLA, which is suitable for an attacker (learning agent) with higher observation capabilities within the acting time step. We prove that PROLA achieves an optimal regret in the order of $\tilde{O}(\sqrt{T})$ without observing the rewards on all the channels other than the attacking one simultaneously. In PROLA, the attacker uniformly randomly selects at least one channel other than the attacking one to observe in each time slot. Our framework is called randomized time-variable feedback graph.

C. Jamming Attacks

There are several works formulating jamming attacks and anti-jamming strategies as online learning problems [18, 20, 21, 22]. In jamming attacks, an attacker can usually observe the reward on the attacking channel where an ongoing communication between legitimate users can be detected. Also it is possible for the defenders to learn whether they are defending against a jammer or a PUE attacker by observing the reward on the accessed channel. PUE attacks are different in that the attacker attacks the channel sensing phase and prevents a secondary user from utilizing an available channel. As a result, a PUE attacker cannot observe the instantaneous reward on the attacked channel. That is, it cannot decide if an attack is effective or not.

D. Adaptive Opponents

Wang et al. study the outcome/performance of two adaptive opponents when each of the agents applies a no regret learning-based access strategy [18] . One agent tries to catch the other on a channel and the other one tries to evade it. Both learning algorithms applied by the opponents are no regret algorithms and are designed for oblivious environments. In other words, each of the learning algorithms is designed for benign environments and non of them assumes an adaptive/learning-based opponent. It is shown in this work, that both opponents applying a no-regret learning-based algorithm, reach an equilibrium which is in fact a Nash equilibrium in that case. In other words, despite the fact that the learning-based algorithms applied are originally proposed for oblivious environments, the two agents applying these algorithms act reasonably well to achieve an equilibrium. Motivated by this work, in our simulations in Section V, we have considered a learning-based secondary user. More specifically, even though we proposed learning-based algorithms for oblivious environments, in the simulations, we evaluate its performance in an adaptive environment. The simulation results show the rationality of the attacker that even against an adaptive opponent (learning-based cognitive radio), it performs well and achieves a regret below the derived upper bound.

III. System Model and Problem Formulation

We consider a cognitive radio network consisting of several primary users, multiple secondary users, and one attacker. There are K ($K > 1$) channels in the network. We assume the system is operated in a time-slotted fashion.

A. Primary User

In each time slot, each primary user is either active (on) or inactive (off). We assume the on-off sequence of PUs on the channels is unknown to the attacker a priori. In other words, the PU activity can follow any distribution or can even be arbitrary.

B. Secondary User

The secondary users may apply any dynamic spectrum access policy [23, 24, 25]. In each time slot, each SU conducts

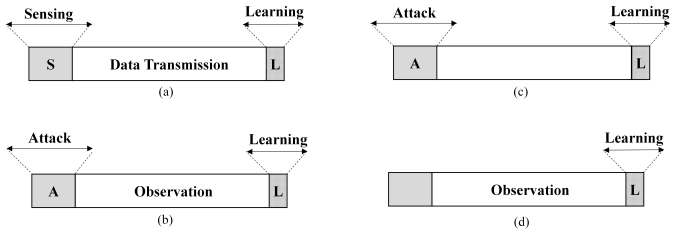


Figure 1: Time slot structure of a) an SU and b,c,d) a PUE attacker.

Table I: Main Notation

T	total number of time slots
K	total number of channels
I_t	index of the channel to be attacked at time t
J_t	index of the channel to be observed at time t
R	total regret of the attacker in Algorithm 1
γ	exploration rate used in algorithm 2
η	learning rate used in Algorithms 1 and 2
$p_t\,(i)$	attack distribution on channels at time t in Algorithms 1 and 2
$q_t\,(i)$	observation distribution on channels at time t in Algorithm 1
$\omega_t\,(i)$	weight assigned to channel i at time t in Algorithms 1 and 2
δ_t	observation probability at time t in Algorithm 1

spectrum sensing, data transmission, and learning in three consecutive phases as shown in Fig. 1(a).

At the beginning of each time slot, each secondary user senses a channel it attempts to access. If it finds the channel idle (i.e., the primary user is inactive), then accesses this channel; otherwise, it remains silent till the end of the current slot in order to avoid interference to the primary user. At the end of the slot, it applies a learning algorithm to decide which channel it will attempt to access in the next time slot based on its past channel access experiences.

We assume the secondary users cannot differentiate the attacking signal from the genuine primary user signal. That is, in a time slot, when the attacker launches a PUE attack on the channel a secondary user attempts to access, the secondary user will not transmit any data on that channel.

C. Attacker

We assume a smart PUE attacker with learning capability. We do not consider attack on the PUs. The attacker may have different observation capabilities. Observation capability is defined as the number of channels the attacker can observe after the attack period within the same time-slot. Observation capabilities of the attacker are impacted by the overall number of it's antennas, time slot duration, switching time, etc.

Within a time-slot, an attacker with at least one observation capability conducts three different operations. First in the attacking phase, the attacker launches the PUE attack by sending signals emulating the primary user's signals [5] to attack the SUs. Note that, in this phase, the attacker has no idea if its attack is effective or not. That is, it does not know if a secondary user is ever trying to access the attacking channel or not. In the observation phase, the attacker is supposed to observe the communication on at least one other channel. The attacker can even observe the attacked channel in the observation phase, however, in that case it will sense nothing since it has attacked the channel by emulating the PU signal and scared away any potential SU attempting to access the channel. So from learning point of view, observing the attacked channel does not provide any useful information on the SUs' activity. Therefore, it only gets useful information when observing channels other than the attacked one. Observing a different channel, the attacker may detect a primary user signal, a secondary user signal, or nothing on the observing channel. The attacker applies its past observations in the learning phase to optimize its future attacks. At the end of the learning period, it decides which channels it attempts to attack and observe in the next slot. Figure 1(b) shows the time slot structure for an attacker with at least one observation capability.

If the attacker has no observation capability within the attacking time slot, at the end of each time slot it still applies a learning strategy based on which, it decides whether to dedicate the next time slot for attack or observation, then chooses a channel for either of them. Figure 1(c,d) show the time slot structure for an attacker with no observation capability in the attacking time slot. As shown in these two figures, the attacker conducts either attack or observation at each time slot.

D. Problem Formulation

Since the attacker needs to learn and attack at the same time and it has no prior knowledge of the primary user activity or secondary user access strategies, we formulate this problem as an online learning problem.

We consider T as the total number of time slots the network operates. We define $x_t(j)$ as the attacker's reward on channel j at time slot t ($1 \leq j \leq K$, $1 \leq t \leq T$). Without loss of generality, we normalize $x_t(j) \in [0, 1]$. More specifically:

$$x_t(j) = \begin{cases} 1, & \textit{SU is on channel } j \textit{ at time } t \\ 0, & o.w. \end{cases}. \tag{1}$$

Suppose the attacker applies a learning policy φ to select the attacking and observing channels. The aggregated expected reward of attacks by time slot T is equal to

$$G_{\varphi}(T) = \mathbf{E}_{\varphi}\left[\sum_{t=1}^{T} x_t(I_t)\right], \tag{2}$$

where I_t indicates the channel chosen at time t to be attacked. The attacker's goal is to maximize the expected value of the aggregated attacking reward, thus to minimize the throughput of the secondary user,

$$\text{maximize} \quad G_{\varphi}(T). \tag{3}$$

For a learning algorithm, regret is commonly used to measure its performance. The regret of the attacker can be defined as follows

$$Regret = G_{max} - G_{\varphi}\,(T)\,, \tag{4}$$

where

$$G_{max} = \max_{j} \sum_{t=1}^{T} x_t\,(j)\,. \tag{5}$$

The regret measures the gap between the accumulated reward achieved applying a learning algorithm and the maximum accumulated reward the attacker can obtain when it keeps attacking the single best channel. Single best channel is the channel with highest accumulated reward up to time T [14]. Then the problem can be transformed to minimize the regret

$$\text{minimize} \quad G_{max} - G_{\varphi}(T). \tag{6}$$

Table I summarizes the main notation used in this paper.

IV. Online Learning-Based Attacking Strategy

In this section we propose two novel online learning algorithms for the attacker. These algorithms do not require the attacker to observe the reward on the attacked channel. Moreover, our algorithms can be applied in any other application with the same requirements.

The first algorithm proposed, POLA, is suitable for an attacker with no observation capability in the attacking time slot . For this attacker, either attack or observation is feasible within each time slot. Based on this learning strategy, the attacker decides at each time slot whether to attack or observe and chooses a channel for either of them, dynamically. The assumption here is that, any time the attacker decides to make observation, it observes only one channel since time slot duration has been considered short or switching costs are high compared to the time slot duration.

The second algorithm, PROLA, is proposed for an attacker with at least one observation capability. Based on this learning policy, at each time, the attacker chooses channels dynamically for both attack and observation. In the following, we assume the attacker's observation capability is one. We generalize it to multiple channel observation capability in Section IV-C.

Both algorithms are considered as no-regret online learning algorithms in the sense that the incremental regret between two consecutive time slots diminishes to zero as time goes to infinity. The first one, POLA, achieves an optimal regret in the order of $\tilde{O}(\sqrt[3]{T^2})$. This higher slope arises from the fact that at each time slot, the attacker gains only some reward by attacking a channel without updating its learning or it learns by making observation without being able to gain any reward. PROLA, is also optimal with the regret proved in the order of $\tilde{O}(\sqrt{T})$. Comparing these algorithms and the generalization of the latter together shows that, changing from no observation capability in the same time slot to one observation capability results in a significant improvement in the regret order; this is in comparison to changing from one observation capability to multiple observation capability which does not give any benefits in terms of regret order. However, multiple channel observation capability, provides insight to find the appropriate number of observations required to achieve the minimum constant factor in the regret upper bound.

A. Attacking Strategy 1: POLA Learning Algorithm

POLA is an online learning algorithm for the learning of an agent with no observation capability within the attacking time slot. In other words, the agent cannot play and observe simultaneously. If modeled properly as a feedback graph as is shown in Appendix A, this problem can be solved based on the EXP3.G algorithm presented in [11]. However, our proposed learning algorithm, POLA, leads to a smaller regret constant compared to the one in [11] and it is much easier to understand. Based on the POLA attacking strategy, the attacker at each time slot decides between attacking and observation, dynamically. If it decides to attack, it applies an exponential weighting distribution based on the previous observations it has made. It chooses a channel uniformly at random if it decides to make an observation. Figure 2 represents the decision making structure of POLA. The proposed algorithm is presented in Algorithm 1.

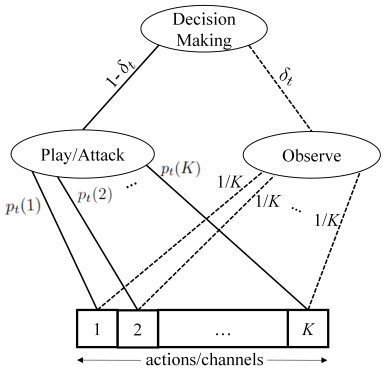


Figure 2: Decision Making process based on the POLA strategy: First the agent decides between play and observe, then for either of them chooses a channel dynamically.

Since the attacker is not able to attack and observe within the same time slot simultaneously, it needs to trade off between attacking and observation cycles. From one side the attacker needs to make observations to learn the most rewarding channel to attack. From the other side, it needs to attack often enough to minimize his regret. So, δ_t which represents the trade off between attack and observation needs to be chosen carefully.

In step 2 of Algorithm 1, $\hat{x}_t(j)$ represents an unbiased estimate of the reward $x_t(j)$. In order to derive $\hat{x}_t(j)$, we divide the $x_t(j)$ by the probability this channel is chosen to be observed which is equal to δ_t/K. The term δ_t/K indicates that in order for each channel to be chosen to be observed, first the algorithm needs to decide to make observation with probability δ_t and then to choose that channel for observation with probability $1/K$.

For any agent that acts based on the Algorithm 1, the following theorem holds.

Theorem 1. *For any $K \geq 2$ and for any $\eta \leq \sqrt[3]{\frac{\ln K}{K^2 T}}$, the upper bound on the expected regret of Algorithm 1*

$$G_{max} - \mathbf{E}[G_{POLA}] \leq (e-2)\eta K \left(\frac{3}{4} \sqrt[3]{\frac{(T+1)^4}{K \ln K}} + \frac{K \ln K}{4} \right) + \frac{\ln K}{\eta} + \frac{3}{2} \sqrt[3]{K T^2 \ln K} \tag{7}$$

holds for any assignment of rewards for any $T > 0$.

Algorithm 1: POLA, Play or Observe Learning Algorithm

Parameter: $\eta \in \left(0, \sqrt[3]{\frac{\ln K}{K^2 T}}\right]$.
Initialization: $\omega_1(i) = 1, \quad i = 1, ..., K$.
For each $t = 1, 2, ..., T$
1. Set $\delta_t = \min\left\{1, \sqrt[3]{\frac{K \ln K}{t}}\right\}$.

Observe with probability δ_t and go to step 2.
Attack with probability $1 - \delta_t$ and go to step 3.

2. Set
$$q_t(i) = \frac{1}{K}, \quad i = 1, ..., K$$
Choose $J_t \sim q_t$ and observe the reward $x_t(J_t)$.
For $j = 1, 2, ..., K$
$$\hat{x}_t(j) = \begin{cases} \frac{x_t(j)}{\delta_t(1/K)}, & j = J_t \\ 0, & o.w., \end{cases}$$
$$\omega_{t+1}(j) = \omega_t(j) \exp(\eta \hat{x}_t(j)),$$
Go back to step 1.

3. Set
$$p_t(i) = \frac{\omega_t(i)}{\sum_{j=1}^{K} \omega_t(j)}, \quad i = 1, ..., K$$
Attack channel $I_t \sim p_t$ and accumulate the unobservable reward $x_t(I_t)$.
For $j = 1, 2, ..., K$
$$\hat{x}_t(j) = 0, \quad \omega_{t+1}(j) = \omega_t(j),$$
Go back to step 1.

Proof of Theorem 1. The regret at time t is a random variable equal to,

$$r(t) = \begin{cases} x_t(j^*) - \mathbf{E}_{POLA,A}\left[x_t(I_t)\right] & \text{, Attack} \\ x_t(j^*) - \mathbf{E}_{POLA,O}\left[x_t(I_t)\right] = x_t(j^*) & \text{, Observe} \end{cases}$$

where j^* is the index of the best channel. The expected value of regret at time t is equal to

$$\mathbf{E}\left[r(t)\right] = (1 - \delta_t)\left(x_t(j^*) - \mathbf{E}_{POLA,A}\left[x_t(I_t)\right]\right) + \delta_t\, x_t(j^*), \tag{8}$$

where the expectation is w.r.t. to the randomness in the attacker's attack policy. Expected value of accumulated regret, R, is

$$\begin{aligned} \mathbf{E}\left[R\right] &= \mathbf{E}\left[\sum_{t=1}^{T} r(t)\right] \\ &\leq \sum_{t=1}^{T} x_t(j^*) - \sum_{t=1}^{T} \mathbf{E}_{POLA,A}\left[x_t(I_t)\right] + \sum_{t=1}^{T} \delta_t. \end{aligned} \tag{9}$$

The inequality results from the fact that $\delta_t \geq 0$ and $x_t(j^*) \leq 1$. It is also assumed that $x_t(i) \leq x_t(j^*)$ for all i. The regret in equation (9) consists of two parts. The first one consists of the first two terms in this equation and it arises as a result of the attacker not attacking the most rewarding channel all the time but attacking some other low rewarding channels. The second part which consists of the last term in the equation is due to the observations made by the attacker in which it gains no reward. We derive an upper bound on each part separately, then add them together.

δ_t plays the key role in minimizing the regret. First of all, δ_t needs to be decaying since otherwise it leads to a linear growth of regret. So the key idea in designing a no-regret algorithm is to choose an appropriate decaying function for δ_t.

From one side, slowly decaying δ_t is desired from learning point of view; however, it results in a larger value of regret since staying more in the observation phase precludes the attacker from launching attacks and gaining rewards. On the other hand, if δ_t decays too fast, the attacker is very likely to settle in a wrong channel since it does not have enough time to learn the most rewarding channel. We choose $\delta_t = \min\left\{1, \sqrt[3]{\frac{K \ln K}{t}}\right\}$ and our analysis shows the optimality of this function in minimizing the regret.

Below is the derivation of the upper bound for the first term of regret in equation (9). For a single t

$$\begin{aligned} \frac{W_{t+1}}{W_t} &= \sum_{i=1}^{K} \frac{\omega_t(i)}{W_t} \exp(\eta \hat{x}_t(i)) \\ &\leq \sum_{i=1}^{K} p_t(i)[1 + \eta \hat{x}_t(i) + (e-2)\eta^2 \hat{x}_t^2(i)] \\ &\leq \exp\left(\eta \sum_{i=1}^{K} p_t(i)\hat{x}_t(i) + (e-2)\eta^2 \sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i)\right), \end{aligned} \tag{10}$$

where the equality follows from the definition of $W_{t+1} = \sum_{i=1}^{K} \omega_{t+1}(i)$ and $\omega_{t+1}(i)$ in Algorithm 1. Also the last inequality follows from the fact that $e^x \geq 1 + x$. Finally, the first inequality holds by definition of $p_t(i)$ in Algorithm 1 and since $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$. In this case, we need $\eta \hat{x}_t(i) \leq 1$. Based on our algorithm, $\eta \hat{x}_t(i) = 0$ if $i \neq J_t$ and for $i = J_t$ we have $\eta \hat{x}_t(i) = \eta \frac{x_t(i)}{\delta_t \frac{1}{K}} \leq \eta K \sqrt[3]{\frac{T}{K \ln K}}$, since $x_t(i) \leq 1$ and $\delta_t \geq \sqrt[3]{\frac{K \ln K}{T}}$ for $T \geq K \ln K$. This is equivalent to $\eta \leq \frac{1}{K \sqrt[3]{\frac{T}{K \ln K}}} = \sqrt[3]{\frac{\ln K}{K^2 T}}$.

By taking the ln and summing over $t = 1$ to T on both sides of equation (10), the left hand side (LHS) of the equation will be equal to

$$\begin{aligned} \sum_{t=1}^{T} \ln \frac{W_{t+1}}{W_t} = \ln \frac{W_{T+1}}{W_1} &\geq \ln \omega_{T+1}(j) - \ln K \\ &= \eta \sum_{t=1}^{T} \hat{x}_t(j) - \ln K. \end{aligned} \tag{11}$$

By combining (11) with (10),

$$\eta \sum_{t=1}^{T} \hat{x}_t(j) - \ln K$$

$$\leq \eta \sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\hat{x}_t(i) + (e-2)\eta^2 \sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i). \tag{12}$$

We take the expectation w.r.t. the randomness in $\hat{x}$, substitute j by j^* since j can be any of the actions, use the definition of $E_{POLA,A}[x_t(I_t)]$, and with a little simplification and rearranging the equation we get the following,

$$\sum_{t=1}^{T} x_t(j^*) - \sum_{t=1}^{T} E_{POLA,A}\left[x_t(I_t)\right] \leq (e-2)\eta K \sum_{t=1}^{T}\frac{1}{\delta_t} + \frac{\ln K}{\eta}. \tag{13}$$

The upper bound on $\sum_{t=1}^{T}\frac{1}{\delta_t}$ is equal to $\frac{3}{4}\sqrt[3]{\frac{(T+1)^4}{K\ln K}} + \frac{K\ln K}{4}$ which gives us,

$$\sum_{t=1}^{T} x_t(j^*) - \sum_{t=1}^{T} \mathbf{E}_{POLA,A}\left[x_t(I_t)\right] \leq \frac{\ln K}{\eta} + (e-2)\eta K\left(\frac{3}{4}\sqrt[3]{\frac{(T+1)^4}{K\ln K}} + \frac{K\ln K}{4}\right). \tag{14}$$

The upper bound on the second term of equation (9) is

$$\sum_{t=1}^{T}\delta_t = \sum_{t=1}^{T}\min\left\{1, \sqrt[3]{\frac{K\ln K}{t}}\right\} \leq \frac{3}{2}\sqrt[3]{KT^2\ln K}. \tag{15}$$

Summing up the equation (14) and equation (15) gives us the regret upper bound. ■

By choosing appropriate values for η, the above upper bound on the regret can be minimized.

Corollary 1. *For any $T > 2.577K\ln K$, we consider the input parameter*

$$\eta = \sqrt{\frac{\ln K}{(e-2)K\left(\frac{3}{4}\sqrt[3]{\frac{(T+1)^4}{K\ln K}} + \frac{K\ln K}{4}\right)}}.$$

Then

$$G_{max} - \mathbf{E}[G_{POLA}] \leq (\sqrt{3(e-2)} + \frac{3}{2})\sqrt[3]{T^2 K\ln K}$$

holds for any arbitrary assignment of rewards.

Proof of Corollary 1. We sketch the proof as follows. By getting the derivative from the statement in Theorem 1 with respect to η, we find the optimal value for η. Since $\eta \leq \sqrt[3]{\frac{\ln K}{K^2 T}}$, in order for the regret bound to hold, we need $T \geq \frac{8}{3\sqrt{3(e-2)^3}} K\ln K = 2.577K\ln K$. By plugging in the value of η and some simplifications the regret bound in the corollary is achieved. ■

Next is a theorem on the regret lower bound for this problem under which attacking and observation are not possible simultaneously.

Theorem 2. *For $K \geq 2$ and for any player strategy A, the expected weak regret of algorithm A is lower bounded by*

$$G_{max} - \mathbf{E}\left[G_A\right] \geq v\sqrt[3]{KT^2}$$

for some small constant v and it holds for some assignment of rewards for any $T > 0$.

Proof of Theorem 2. The proof follows from the lower bound analysis in [11]. The construction is given in the Appendix. ■

Based on Theorems 1 and 2, the algorithm's regret upper bound matches its regret lower bound which indicates POLA is an optimal online learning algorithm.

B. Attacking Strategy 2: PROLA Learning Algorithm

In this section, we propose another novel online learning algorithm that can be applied by the PUE attacker with one observation capability within the attacking slot. The proposed optimal online learning algorithm, called PROLA, at each time, chooses two actions dynamically to play and observe, respectively. The action to play is chosen based on an exponential weighting, while the other action for observation is chosen uniformly at random excluding the played action. The proposed algorithm is presented in Algorithm 2.

Algorithm 2 : PROLA, Play and Random Observe Learning Algorithm

Parameters: $\gamma \in (0,1)$, $\eta \in \left(0, \frac{\gamma}{2(K-1)}\right]$.

Initialization: $\omega_1(i) = 1, \quad i = 1, ..., K.$
For each $t = 1, 2, ..., T$

1. Set $p_t(i) = (1-\gamma)\frac{\omega_t(i)}{\sum_{j=1}^{K}\omega_t(j)} + \frac{\gamma}{K}, \qquad i = 1, ..., K.$

2. Attack channel $I_t \sim p_t$ and accumulate the unobservable reward $x_t(I_t)$.

3. Choose a channel J_t other than the attacked one uniformly at random and observe its reward $x_t(J_t)$ based on equation (1).

4. For $j = 1, ..., K$

$$\hat{x}_t(j) = \begin{cases} \frac{x_t(j)}{(1/(K-1))(1-p_t(j))}, & j = J_t \\ 0, & o.w., \end{cases}$$

$$\omega_{t+1}(j) = \omega_t(j)\exp(\eta\hat{x}_t(j)).$$

Figure 3 shows the observation policy governing the actions for $K = 4$ in a feedback graph format. In this figure, $Y_{ij}(t)$ is an observation indicator of channel j when channel i is attacked at time t. We define $Y_{ij}(t) \in \{0, 1\}$ such that at each time for the chosen action i to be played,

$$\sum_{j=1, j\neq i}^{K} Y_{ij}(t) = 1, \quad i = 1, \ldots, K \quad t = 1, \ldots, T. \tag{16}$$

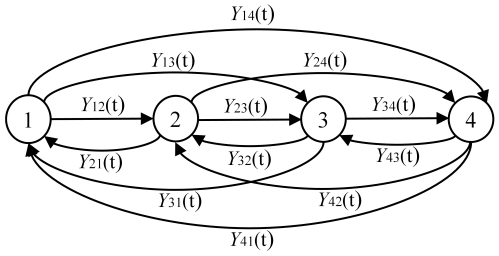


Figure 3: Channel observation strategy, $K = 4$

In other words, there is a policy based on which for any channel i played, only one of the observation indicators takes a value of one and the rest take a value of zero. For example, if channel 2 is attacked ($i = 2$), only one of the three outgoing edges from 2 will be equal to one. This edge selection policy represents the channel selection process for observation. We define it as a uniform random distribution equal to $\frac{1}{K-1}$. We call this feedback graph, a time-variable random feedback graph. Our feedback graph fits into the time-variable feedback graphs introduced in [11] and based on the results derived in that work, the regret upper bound of our algorithm is $\tilde{O}(\sqrt[3]{T^2})$. However, based on our analysis, the upper bound on the attacker's regret is in the order of $\tilde{O}(\sqrt{T})$ which shows a significant improvement. In other words, despite the fact that the agent makes only partial observation on the channels, it achieves a significantly improved regret order compared to the no observation in the attacking slot case. This has been possible due to the new property we considered in the partially observable graphs which is adding randomness. In the long run, randomness makes full observation possible to the agent.

In Step 4 of Algorithm 2, in order to create $\hat{x}_t(j)$, an unbiased estimate of the actual reward $x_t(j)$, we divide the observed reward, $x_t(J_t)$, by $(1/(K-1))(1-p_t(J_t))$which is the probability of choosing channel J_t to be observed. In other words, channel J_t will be chosen to be observed if it has not been chosen for attacking, with probability $(1-p_t(J_t))$, and second if it gets chosen uniformly at random from the rest of the channels, with probability $(1/(K-1))$.

Theorem 3. *For any $K \geq 2$ and for any $\eta \leq \frac{\gamma}{2(K-1)}$, for the given randomized observation structure for the attacker the upper bound on the expected regret of Algorithm 2,*

$$G_{max} - \mathbf{E}[G_{PROLA}] \leq (e-2)(K-1)\frac{\eta}{1-\gamma}T + \frac{\ln K}{\eta(1-\gamma)}$$

holds for any assignment of rewards for any $T > 0$.

Proof of Theorem 3. By using the same definition for $W_t = \omega_t(1) + \cdots + \omega_t(K) = \sum_{i=1}^{K} \omega_t(i)$ as in Algorithm 1, at each time t,

$$\frac{W_{t+1}}{W_t} = \sum_{i=1}^{K} \frac{p_t(i) - \gamma/K}{1-\gamma} \exp(\eta \hat{x}_t(i))$$
$$\leq \exp\left(\frac{\eta}{1-\gamma}\sum_{i=1}^{K} p_t(i)\hat{x}_t(i) + \frac{(e-2)\eta^2}{1-\gamma}\sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i)\right). \tag{17}$$

The equality follows from the definition of W_{t+1}, $\omega_{t+1}(i)$, and $p_t(i)$ respectively in Algorithm 2. Also, the inequality follows from the fact that $e^x \leq 1 + x + (e-2)x^2$ for $x \leq 1$ and $e^x \geq 1 + x$. When $\eta \leq \frac{\gamma}{2(K-1)}$, the result, $\eta\hat{x}_t(i) \leq 1$, follows from the observation that either $\eta\hat{x}_t(i) = 0$ or $\eta\hat{x}_t(i) = \eta\frac{x_t(i)}{\frac{1}{K-1}(1-p_t(i))} \leq \eta(K-1)\frac{2}{\gamma} \leq 1$, since $x_t(i) \leq 1$ and $p_t(i) = (1-\gamma)\frac{\omega_t(i)}{\sum_{j=1}^{K}\omega_t(j)} + \frac{\gamma}{K} \leq 1 - \gamma + \frac{\gamma}{2} \leq 1 - \frac{\gamma}{2}$.

By taking the logarithm of both sides of equation (17) and summing over t from 1 to T, we derive the following inequality on the LHS of the equation,

$$\sum_{t=1}^{T} \ln \frac{W_{t+1}}{W_t} = \ln \frac{W_{T+1}}{W_1} \geq \eta \sum_{t=1}^{T} \hat{x}_t(j) - \ln K. \tag{18}$$

Combining (17) and (18), we can get

$$\sum_{t=1}^{T}\hat{x}_t(j) - \sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\hat{x}_t(i)$$
$$\leq \gamma\sum_{t=1}^{T}\hat{x}_t(j) + (e-2)\eta\sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i) + \frac{\ln K}{\eta}. \tag{19}$$

Let $\acute{x}_t(i) = \hat{x}_t(i) - f_t$ where $f_t = \sum_{i=1}^{K} p_t(i)\hat{x}_t(i)$. We make the pivotal observation that (19) also holds for $\acute{x}_t(i)$ since $\eta\acute{x}_t(i) \leq 1$, which is the only key to obtain (19).

We also note that,

$$\begin{aligned}\sum_{i=1}^{K} p_t(i)\acute{x}_t^2(i) &= \sum_{i=1}^{K} p_t(i)(\hat{x}_t(i) - f_t)^2 \\ &= \sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i) - f_t^2 \\ &\leq \sum_{i=1}^{K} p_t(i)\hat{x}_t^2(i) - \sum_{i=1}^{K} p_t^2(i)\hat{x}_t^2(i) \\ &= \sum_{i=1}^{K} p_t(i)(1-p_t(i))\hat{x}_t^2(i). \end{aligned} \tag{20}$$

Substituting $\acute{x}_t(i)$ in equation (19) and combining with (20),

$$\sum_{t=1}^{T}(\hat{x}_t(j) - f_t) - \sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)(\hat{x}_t(i) - f_t)$$
$$= \sum_{t=1}^{T}\hat{x}_t(j) - \sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\hat{x}_t(i)$$
$$\leq \gamma\sum_{t=1}^{T}\left(\hat{x}_t(j) - \sum_{i=1}^{K} p_t(i)\hat{x}_t(i)\right)$$
$$+ (e-2)\eta\sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)(1-p_t(i))\hat{x}_t^2(i) + \frac{\ln K}{\eta}. \tag{21}$$

Observe that $\hat{x}_t(j)$ is similarly designed as an unbiased estimate of $x_t(j)$. Then for the expectation with respect to the sequence of channels attacked by the horizon T,

$$\mathbf{E}[\hat{x}_t(j)] = x_t(j), \quad \mathbf{E}\left[\sum_{i=1}^{K} p_t(i)\hat{x}_t(i)\right] = \mathbf{E}[x_t(I_t)],$$

and

$$\mathbf{E}\left[\sum_{i=1}^{K} p_t(i)(1-p_t(i))\hat{x}_t^2(i)\right] = \mathbf{E}\left[\sum_{i=1}^{K} p_t(i)(K-1)x_t(i)\hat{x}_t(i)\right] \leq K-1.$$

We now take the expectation with respect to the sequence of channels attacked by the horizon T in both sides of the last inequality of (21). For the left hand side,

$$\mathbf{E}\left[\sum_{t=1}^{T}\hat{x}_t(j)\right] - \mathbf{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{K} p_t(i)\hat{x}_t(i)\right] = G_{max} - \mathbf{E}[G_{PROLA}]. \quad (22)$$

and for the right hand side,

$$\mathbf{E}[\text{R.H.S}] \leq \gamma(G_{max} - \mathbf{E}[G_{PROLA}]) + (e-2)(K-1)\eta T + \frac{\ln K}{\eta}$$

Combining the last two equations we get,

$$(1-\gamma)(G_{max} - \mathbf{E}[G_{PROLA}]) \leq (e-2)(K-1)\eta T + \frac{\ln K}{\eta},$$

and since G_{max} can be substituted by T, by rearranging the relation above the regret upper bound is achieved. ■

Similarly, we can minimize the regret bound by choosing appropriate values for η and γ.

Corollary 2. *For any $T \geq \frac{8(K-1)\ln K}{e-2}$ and $\gamma = \frac{1}{2}$, we consider the following value for η*

$$\eta = \sqrt{\frac{\ln K}{2(e-2)(K-1)T}}.$$

Then

$$G_{max} - \mathbf{E}[G_{PROLA}] \leq 2\sqrt{2(e-2)}\sqrt{T(K-1)\ln K}$$

holds for any arbitrary assignment of rewards.

Proof of Corollary 2. We sketch the proof as follows. By getting the derivative from the statement in Theorem 3 with respect to η, we find the optimal value for η. Since $\eta \leq \frac{\gamma}{2(K-1)}$, in order for the regret bound to hold, we need $T \geq \frac{8(K-1)\ln K}{e-2}$. Replacing the value of η gives us the result on the regret upper bound. ■

The important observation is that, based on [11] such an algorithm, Algorithm 2, is expected to achieve a regret in the order of $\tilde{O}\left(\sqrt[3]{T^2}\right)$ since it can be categorized as a partially observable graph. However, our analysis gives a tighter bound and shows not only it is tighter but also it achieves the regret order of fully observable graphs. This significant improvement has been accomplished by introducing randomization into feedback graphs.

The following theorem provides the regret lower bound for this problem.

Theorem 4. *For any $K \geq 2$ and for any player strategy A, the expected weak regret of algorithm A is lower bounded by*

$$G_{max} - \mathbf{E}[G_A] \geq c\sqrt{KT}$$

for some small constant c and it holds for some assignment of rewards for any $T > 0$.

Proof of Theorem 4. The proof follows closely the lower bound analysis in [14]. Details are provided in the Appendix. ■

C. Extension to Multiple Action Observation Capability

We generalize Algorithm PROLA to the case of an agent with multiple observation capability. This is suitable for an attacker with multiple observation capability when the attacker after the attack phase is able to observe multiple other channels within the same time-slot. At least one and at most $K-1$ observations are possible by the agent (attacker). In this case, m indicates the number of possible observations and $1 \leq m \leq K-1$. Then, for m observations, we modify equation (16) as follows,

$$\sum_{j=1, j\neq i}^{K} Y_{ij}(t) = m, \quad i = 1, \ldots, K \quad t = 1, \ldots, T. \quad (23)$$

The probability of a uniform choice of m observations at each time, $\sum_{j=1, j\neq i}^{K} Y_{ij}(t) = m$ is equal to $\frac{1}{\binom{K-1}{m}}$. Corollary 3 shows the result of the analysis for m observations.

Corollary 3. *If the agent can observe m actions at each time, then the regret upper bound is equal to*

$$G_{max} - \mathbf{E}\left[G_A\right] \leq 4\sqrt{(e-2)}\sqrt{T\frac{K-1}{m}\ln K}.$$

This regret upper bound shows a faster convergence rate when making more observations.

Proof of Corollary 3. We substitute $1/(K-1)$ by $m/(K-1)$ in Step 4 of Algorithm PROLA since at each time, m actions are being chosen uniformly at random. Then the regret upper bound can be derived by similar analysis. ■

As our analysis shows, making multiple observations does not improve the regret in terms of its order in T compared to the case of making only one observation. This means that only one observation and not more than that is sufficient to make a significant improvement in terms of regret order reduction compared to the case of no observation within the attacking slot. The advantage of making more observations however is in reducing the constant coefficient in the regret. Making more observations leads to a smaller constant factor in the regret upper bound. This relationship is non-linear though. i.e., the regret upper bound is proportional to $\sqrt{1/m}$ which means that in order to make a large reduction in the regret in terms of its constant coefficient impact, only a few observations would suffice. Our simulation results in Section V-C provide more details on this non-linear relationship.

V. Performance Evaluation

In this section, we present the simulation results to evaluate the validity of the proposed online learning algorithms applied by the PUE attacker. All the simulations are conducted in

MATLAB and the results achieved are averaged over 10,000 independent random runs.

We evaluate the performance of the proposed learning algorithms, POLA and PROLA, and compare them with their theoretical regret upper bounds $\tilde{O}(\sqrt[3]{T^2})$ and $\tilde{O}(\sqrt{T})$, respectively. POLA and PROLA correspond to an attacker with no observation and one observation capability within the attacking time slot, respectively. We then examine the impact of different system parameters on the attacker's performance. The parameters include the number of time slots, total number of channels in the network, and the distribution on the PU activities. We also examine the performance of an attacker with multiple observation capabilities. Finally we evaluate a secondary user's accumulated traffic with and without the presence of a PUE attacker.

K primary users are considered, each acting on one channel. The primary users' on-off activity follows a Markov chain or i.i.d. distribution in the network. Also, the PU activities on different channels are independent from each other. K idle probabilities are generated using MATLAB's rand function, each denoting one PU activity on each channel if PUs follow an i.i.d. Bernoulli distributions. $pI = [0.85\ 0.85\ 0.38\ 0.51\ 0.21\ 0.13\ 0.87\ 0.7\ 0.32\ 0.95]$ is a vector of K elements each of which denotes the corresponding PU activity on the K channels. If the channels follow Markov chains, for each channel, we generate three probabilities, $p01$, $p10$, and $p1$ as the transition probabilities from state 0 (on) to 1 (off), from 1 to 0, and the initial idle probability, respectively. The three vectors below, are examples considered to represent PU activities. $p01 = [0.76\ 0.06\ 0.3\ 0.24\ 0.1\ 0.1\ 0.01\ 0.95\ 0.94\ 0.55]$, $p10 = [0.14\ 0.43\ 0.23\ 0.69\ 0.22\ 0.59\ 0.21\ 0.58\ 0.34\ 0.73]$, and $p1 = [0.53\ 0.18\ 0.88\ 0.66\ 0.23\ 0.87\ 0.48\ 0.44\ 0.45\ 0.88]$.

The PUE attacker employs either of the proposed attacking strategies, POLA, or PROLA. Throughout the simulations, when we talk about an attacker employing PROLA, we assume the attacker's observation capability is one within the attacking slot, unless otherwise stated.

Since the goal here is to evaluate the PUE attacker's performance, for simplicity we consider one SU in the network. Throughout the simulations, we assume the SU employs an online learning algorithm called Hedge [13]. The assumption on the Hedge algorithm is that the secondary user is able to observe the rewards on all the channels in each time slot. Hedge provides the minimum regret in terms of both order and the constant factor among all optimal learning-based algorithms. As a result the performance of our proposed learning algorithms can be evaluated in the worst case scenario for the attacker. As explained in Section II, even though in our analysis we considered an oblivious environment, in the simulations we can consider an SU that runs Hedge (an adaptive opponent). This experimental setup adds value since it shows that our proposed online learning-based algorithms perform reasonably even against adaptive opponents by keeping the occurred regret between the theoretical upper and lower bounds derived.

A. Performance of POLA and the impact of the number of channels

In this section, we evaluate the performance of the proposed algorithm, POLA and compare it with the theoretical analysis. Figure 4(a) shows the overall performance of the attacker for both cases of i.i.d. and Markovian chain distribution of PU activities on the channels for $K = 10$. Regret upper bound and lower bound from the analysis in Corollary 1 and Theorem 2, respectively are also plotted in this figure. Then we examine the impact of the number of channels in the network on it's performance. We consider K variable from 10 to 50. The attacker's regret when PUs follow i.i.d. distribution and Markovian chain for different number of actions are shown in Fig. 4(b) and (c), respectively. Since POLA has a regret with higher slope, in order to better observe the results, we have plotted the figures for T from 1 to $20,000$. We can observe the following from the figures.

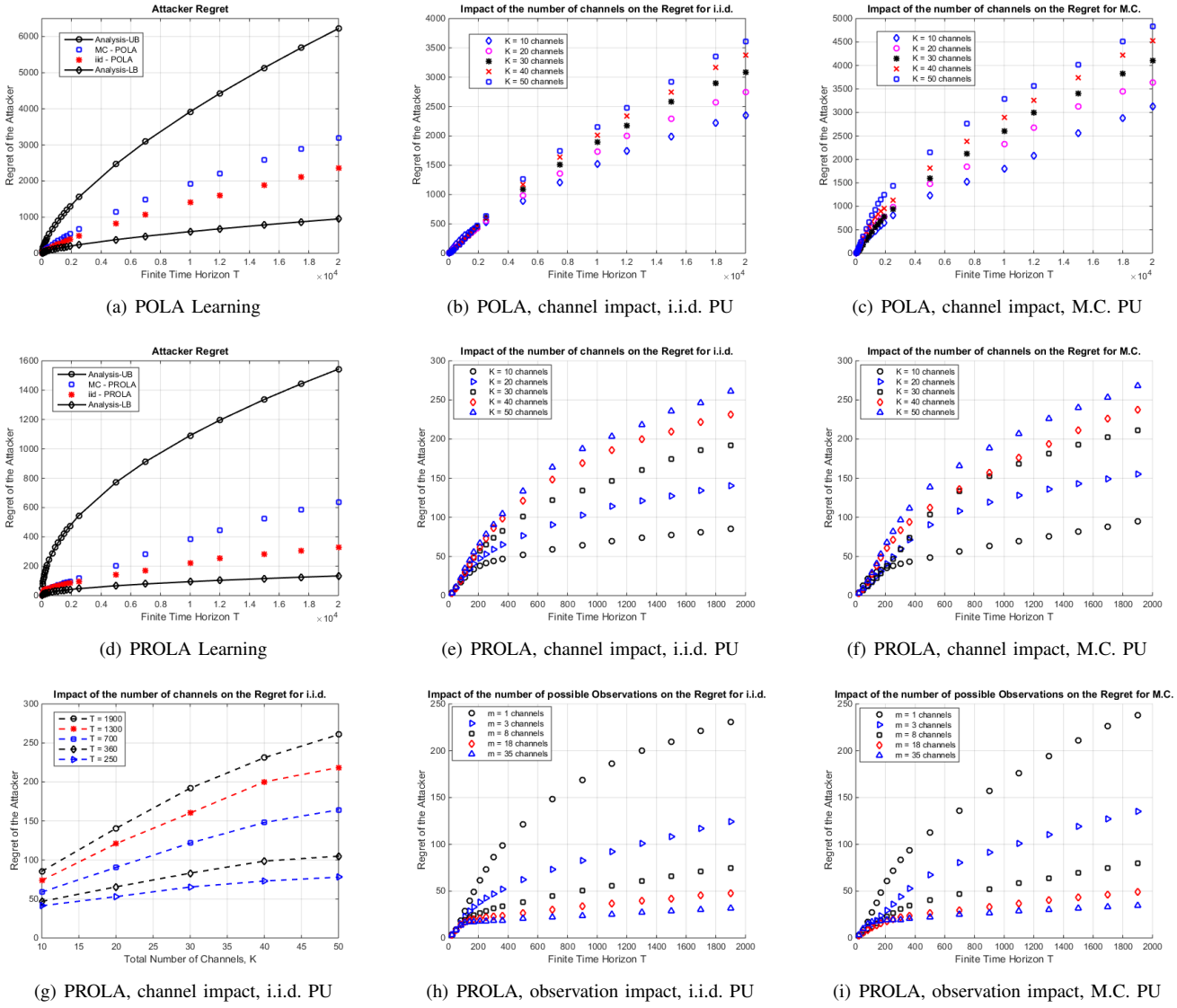
- Regardless of the PUs activity type, the occurred regret is below the regret upper bound achieved from the theoretical analysis.
- The regret on all three figures has a higher slope compared to the results for PROLA in Fig. 4(d)-(f) which also complies with our analysis. The higher slope in these figures can be seen by comparing their x and y axises.
- As the number of channels increases the regret increases as is expected based on the analysis from Corollary 1.
- As the number of channels increases, the regret does not increase linearly with it. Instead, the increment in the regret becomes marginal which complies with the theoretical analysis. Based on Corollary 1 the regret is proportional to $(\sqrt[3]{K \ln K})$. The dependency on K can be represented by plotting the regret versus K as is shown in the next subsection.

B. Performance of PROLA and the impact of the number of channels

We compare the performance of the proposed learning algorithm, PROLA, with the theoretical analysis from section IV. We consider a network of $K = 10$ channels. Figure 4(d) shows the simulation results as well as analytical results. From the simulations, we observe that the actual regret occurred in the simulations, is between the bounds achieved from the analysis regardless of the type of PU activity which complies with our analysis. We also note that when we derived the theoretical upper bound we did not make any assumption on the PU activity. The regret is only dependent on the K and T.

Then, we examine the impact of the number of channels in the network on the attacker's performance when it applies PROLA. Figure 4(e) and (f) show the attacker's regret when PUs follow i.i.d. distribution and Markovian chain respectively for K variable from 10 to 50. The same discussion on the system parameters and the results hold as in subsection V-A. In another representation, we plot the regret versus the number of channels, K, as the x axis in Fig. 4(g).

Moreover, comparing regret values in Fig. 4(a) with those in Fig. 4(d), we observe the huge difference in regret amount between no observation capability within the attacking slot and one observation capability.



(a) POLA Learning (b) POLA, channel impact, i.i.d. PU (c) POLA, channel impact, M.C. PU

(d) PROLA Learning (e) PROLA, channel impact, i.i.d. PU (f) PROLA, channel impact, M.C. PU

(g) PROLA, channel impact, i.i.d. PU (h) PROLA, observation impact, i.i.d. PU (i) PROLA, observation impact, M.C. PU

Figure 4: Simulation results under different PU activity assumptions

C. Impact of the number of observations in each time slot

We consider PROLA algorithm with $K = 40$ channels in the network. The number of observing channels, m, varies from 1 to 35. Figure 4(h) and (i) show the performance of the PROLA for $m = 1, 3, 8, 18, 35$ when the PUs follow i.i.d. distribution and Markovian chain on the channels, respectively. We can observe the following from the simulation results.

- As the observation capability of the attacker increases, it achieves a lower regret. This observation complies with the Corollary 3 provided in Section IV-C based on which we expect smaller constant factor as the observation capability increases.
- In the beginning, even adding a couple of more observing channels (from $m = 1$ to $m = 3$), the regret decreases dramatically. The decrement in the regret becomes marginal as the number of observing channels becomes sufficiently large (e.g., from $m = 18$ to $m = 35$). This observation implies that, in order to achieve a good attacking performance (smaller constant factor in regret upper bound), the attacker does not need to be equipped with high observation capability. In the simulation, when the number of observing channels ($m = 10$) is $\frac{1}{4}$ of the number of all channels ($K = 40$), the regret is approaching to the optimal.

D. Accumulated Traffic of SU with and without attacker

We set $K = 10$ and measure the accumulated traffic achieved by the SU with and without the presence of the attacker. The attacker employs the PROLA algorithm. Figure 5 shows that the accumulated traffic of the SU is largely decreased when there is a PUE attacker in the network for both types of PU activities.

VI. Conclusions and Future Work

In this paper, we studied the optimal online learning algorithms that can be applied by a PUE attacker without any prior knowledge of the primary user activity characteristics

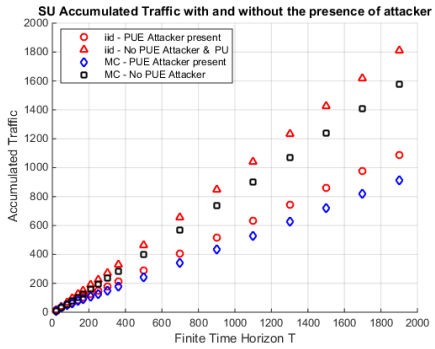


Figure 5: Accumulated Traffic of an SU

and secondary user channel access policies. We formulated the PUE attack as an online learning problem. We identified the uniqueness of PUE attack that a PUE attacker cannot observe the reward on the attacking channel, but is able to observe the reward on another channel. We proposed a novel online learning strategy called POLA for the attacker with no observation capability in the attacking time slot. In other words, this algorithm is suitable when simultaneous attack and observation is not possible within the same time slot. POLA dynamically decides between attack and observation, then chooses a channel for the decision made. POLA achieves a regret in the order of $\tilde{\Theta}(\sqrt[3]{T^2})$. We showed POLA's optimality by matching its regret upper and lower bounds. We proposed another novel online learning algorithm called PROLA for an attacker with at least one observation capabilities. For such an attacker, the attack period is followed by an observation period during the same time slot. PROLA introduces a new theoretical framework under which the agent achieves an optimal regret in the order of $\tilde{\Theta}(\sqrt{T})$. One important conclusion of our study is that with no observation at all in the attacking slot in the POLA case, the attacker loses on the regret order, and with the observation of at least one channel in the PROLA case, there is a significant improvement on the performance of the attacker. This is in contrast to the case where increasing the number of observations from one to $m \geq 2$, does not make that much difference, only improving the constant factor. Though, this observation can be utilized to study the approximate number of observations required to get the minimum constant factor. The attacker's regret upper bound has a dependency on the number of observations m as $\sqrt{1/m}$. That is, the regret decreases overall for an attacker with higher observation capability (larger m). However, when the number of observing channels is small, the regret decreases more if we add a few more observing channels. While, the decreased regret will become marginal when more observing channels are added. This finding implies that an attacker may only need a small number of observing channels to achieve a good constant factor. The regret upper bound also is proportional to $\sqrt{K \ln K}$ which means the regret increases when there are more channels in the network. The proposed optimal learning algorithm, PROLA, also advances the study of online learning algorithms. It deals with the situation where a learning agent cannot observe the reward on the action that is taken but can partially observe the reward of other actions. Before our work, the regret upper bound is proved to be in the order of $\tilde{O}(\sqrt[3]{T^2})$. Our algorithm achieves a tighter regret bound of $\tilde{O}(\sqrt{T})$ by randomizing the observing actions which introduces the concept of time-variable random feedback graphs. We show this algorithm's optimality by deriving its regret lower bound which matches with its upper bound.

As for future work, we believe that our work can serve as a stepping stone to study many other problems. How to deal with multiple attackers will be interesting, especially when the attackers perform in a distributed manner. One other interesting direction is to study the equilibrium between the PUE attacker(s) and secondary user(s) when both of them employ learning based algorithms. Integrating non-perfect spectrum sensing and the ability of PUE attack detection into our model will also be interesting especially that the PUE attacker may interfere with the PU during spectrum sensing period which can make it detectable by the PU. From theoretical point of view, our result shows only one possible case that the feedback graph despite being partially observable, achieves a tighter bound. A particular reward structure may allow for $\tilde{O}(\sqrt{T})$ regret even in partially observable graphs.

ACKNOWLEDGMENT

We would like to express special thanks to Dr. Tyrus Berry from Department of Mathematics at George Mason University for valuable comments in regret upper bound analysis of POLA algorithm.

APPENDIX

A. *Proof of Theorem 2*

Proof of Theorem 2. We sketch the proof as follows. The problem of the PUE attacker with no observation capability within the attacking time slot can be modeled as a feedback graph. Feedback graphs are introduced in [11] based on which the observations governing the actions are modeled as graphs. More specifically, in a feedback graph, the nodes represent the actions and the edges connecting them demonstrate the observations made associated with taking a specific action. Figure 6 shows how our problem can be modeled as a feedback graph.

In this figure, nodes 1 to K represent the overall number of actions. There are K more actions however in this figure. These K extra actions are required to be able to model our problem with feedback graphs. Actions $K+1$ up to $2K$ represent the observations; i.e., any time the agent decides to make an observation, it is modeled as an extra action. There are K channels and we model any observation on each channel with a new action which adds up to $2K$ actions overall. The agent gains a reward of zero if it chooses the observation action since it is not a real action. Instead, it makes an observation on the potential reward of its associated real action.

So, this problem can be modeled as a feedback graph and it in fact turns out to be a partially observable graph [11]. In [11], it is proved that the regret lower bound for partially observable graphs is $\Omega(\nu\sqrt[3]{KT^2})$ which completes the proof. ■

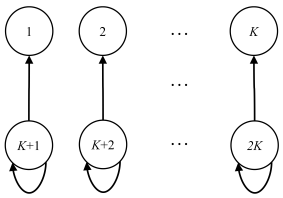


Figure 6: Modeling the Attacker with no observation capability within the attacking slot with a feedback graph

B. Proof of Theorem 4

The proof here is similar to the lower bound analysis of EXP3 given in proof of Theorem 5.1, Appendix, in [14]. We mention the differences here. For this analysis the problem setup is exactly the same as the one in [14]. The notations and definitions are also the same as given in the first part of the analysis in [14]. Below we bring some of the important notations and definitions used in the analysis here to keep the clarity of our analysis and to make our analysis self contained; however, the reader is referred to [14] for more details on the definitions of notations.

The reward distribution on the actions are defined as follows. One action is chosen uniformly at random to be the good action. Below is the reward distribution on the good action, i for all $t = 1, ..., T$,

$$x_t(i) = \begin{cases} 1, & 1/2 + \epsilon \\ 0, & o.w., \end{cases} \tag{24}$$

where $\epsilon \in (0, 1/2]$. The reward distribution on all other actions is defined to be one or zero with equal probabilities. $\mathbf{P}_* \{\cdot\}$ is used to denote the probability w.r.t. this random choice of rewards to play. $\mathbf{P}_i \{\cdot\}$ represents the probability conditioned on i being the good action. Also, $\mathbf{P}_{unif} \{\cdot\}$ shows the probability with respect to uniformly random choice of rewards on all actions. $\mathbf{O}$ notation is also used to denote the probability w.r.t. observation. Analogous observation probability $\mathbf{O}_* \{\cdot\}$, $\mathbf{O}_i \{\cdot\}$, $\mathbf{O}_{unif} \{\cdot\}$ and expectation notations, $\mathbf{E}_*[\cdot]$, $\mathbf{E}_i[\cdot]$, $\mathbf{E}_{unif}[\cdot]$ are used.

The agent's access/attack policy is denoted by A. $r_t = x_t(i_t)$ is a random variable which its value shows the reward gained at time t. $\mathbf{r}^t =< r_1, \ldots, r_t >$ is a sequence of rewards received till t and $\mathbf{r}$ is the entire sequence of rewards. $G_A = \sum_{t=1}^{T} r_t$ is the gain of the agent and $G_{max} = \max_j \sum_{t=1}^{T} x_t(j)$. The number of times action i is chosen by A is a random variable denoted by N_i.

Lemma 1 *Let* $f : \{0,1\}^T \longrightarrow [0, M]$ *be any function defined on reward sequences* $\mathbf{r}$. *Then for any action* i,

$$\mathbf{E}_i[f(\mathbf{r})] \leq \mathbf{E}_{unif}[f(\mathbf{r})] + \tfrac{M}{2}\sqrt{-\mathbf{E}_{unif}[\mathbf{O}_i]\ln(1-4\epsilon^2)}.$$

Proof.

$$\begin{aligned} \mathbf{E}_i[f(\mathbf{r})] - \mathbf{E}_{unif}[f(\mathbf{r})] &= \sum_{\mathbf{r}} f(\mathbf{r})(\mathbf{O}_i\{\mathbf{r}\} - \mathbf{O}_{unif}\{\mathbf{r}\}) \\ &\leq \sum_{\mathbf{r}:\mathbf{O}_i\{\mathbf{r}\} \geq \mathbf{O}_{unif}\{\mathbf{r}\}} f(\mathbf{r})(\mathbf{O}_i\{\mathbf{r}\} - \mathbf{O}_{unif}\{\mathbf{r}\}) \\ &\leq M \sum_{\mathbf{r}:\mathbf{O}_i\{\mathbf{r}\} \geq \mathbf{O}_{unif}\{\mathbf{r}\}} (\mathbf{O}_i\{\mathbf{r}\} - \mathbf{O}_{unif}\{\mathbf{r}\}) \\ &= \frac{M}{2}\left\|\mathbf{O}_i - \mathbf{O}_{unif}\right\|_1. \end{aligned} \tag{25}$$

We also know from [26, 14] that,

$$\left\|\mathbf{O}_{unif} - \mathbf{O}_i\right\|_1^2 \leq (2\ln 2) KL(\mathbf{O}_{unif} \,||\, \mathbf{O}_i). \tag{26}$$

From chain rule for relative entropy we derive the following,

$$\begin{aligned} KL(\mathbf{O}_{unif} \,||\, \mathbf{O}_i) &= \sum_{t=1}^{T} KL(\mathbf{O}_{unif}\left\{r_t|\mathbf{r}^{t-1}\right\} \,||\, \mathbf{O}_i\left\{r_t|\mathbf{r}^{t-1}\right\}) \\ &= \sum_{t=1}^{T} (\mathbf{O}_{unif}\{i_t \neq i\} KL(\frac{1}{2} \,||\, \frac{1}{2})) \\ &\quad + (\mathbf{O}_{unif}\{i_t = i\} KL(\frac{1}{2} \,||\, \frac{1}{2} + \epsilon)) \\ &= \sum_{t=1}^{T} \mathbf{O}_{unif}\{i_t = i\}(-\frac{1}{2}\lg(1-4\epsilon^2)) \\ &= \mathbf{E}_{unif}[\mathbf{O}_i](-\frac{1}{2}\lg(1-4\epsilon^2)). \end{aligned} \tag{27}$$

The lemma follows by combining (25),(26), and (27). ■

Proof of Theorem 4. The rest of the analysis in this part is similar to the analysis in Theorem A.2 in [14], except that when we apply lemma 1 to N_i, we reach the following inequality,

$\mathbf{E}_i[N_i] \leq \mathbf{E}_{unif}[N_i] + \frac{T}{2}\sqrt{-\mathbf{E}_{unif}[\mathbf{O}_i]\ln(1-4\epsilon^2)}$.

where $\sum_{i=1}^{K} \sqrt{\mathbf{E}_{unif}[\mathbf{O}_i]} \leq \sqrt{KT}$. By considering the observation probability and making a little simplification the upper bound is achieved. Following similar steps as in Theorem A.2 in [14] for the rest of the analysis gives us the regret lower bound equal to $\omega(c\sqrt{KT})$ which completes the proof. ■

REFERENCES

[1] "Report and order and second further notice of proposed rulemaking, federal communications commission 15-47 gn docket no. 12-354," 2015.

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

[3] Z. Gao, H. Zhu, S. Li, S. Du, and X. Li, "Security and privacy of collaborative spectrum sensing in cognitive radio networks," *Wireless Communications, IEEE*, vol. 19, no. 6, pp. 106–112, 2012.

[4] Q. Yan, M. Li, T. Jiang, W. Lou, and Y. Hou, "Vulnerability and protection for distributed consensus-based spectrum sensing in cognitive radio networks," in *INFOCOM, 2012 Proceedings IEEE*, March 2012, pp. 900–908.

[5] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part i: Known channel statistics," *Wireless Communications, IEEE Transactions on*, vol. 9, no. 11, pp. 3566–3577, November 2010.

[6] ——, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems; part ii: Unknown channel statistics," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 1, pp. 274–283, January 2011.

[7] M. Dabaghchian, A. Alipour-Fanid, K. Zeng, and Q. Wang, "Online learning-based optimal primary user emulation attacks in cognitive radio networks," in *Communications and Network Security, 2016 CNS 2016. IEEE*, Oct 2016.

[8] R. Chen, J.-M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.

[9] R. Chen, J.-M. Park, and J. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 1, pp. 25–37, Jan 2008.

[10] M. Clark and K. Psounis, "Can the privacy of primary networks in shared spectrum be protected?" in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[11] N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren, "Online learning with feedback graphs: Beyond bandits," *CoRR*, vol. abs/1502.07617, 2015. [Online]. Available: http://arxiv.org/abs/1502.07617

[12] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, May 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1013689704352

[13] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "Gambling in a rigged casino: The adversarial multi-armed bandit problem," in *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, Oct 1995, pp. 322–331.

[14] ——, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1137/S0097539701398375

[15] K. Wang, Q. Liu, and L. Chen, "Optimality of greedy policy for a class of standard reward function of restless multi-armed bandit problem," *Signal Processing, IET*, vol. 6, no. 6, pp. 584–593, August 2012.

[16] Y. Gai and B. Krishnamachari, "Distributed stochastic online learning policies for opportunistic spectrum access," *Signal Processing, IEEE Transactions on*, vol. 62, no. 23, pp. 6184–6193, Dec 2014.

[17] M. Dabaghchian, S. Liu, A. Alipour-Fanid, K. Zeng, X. Li, and Y. Chen, "Intelligence measure of cognitive radios with learning capabilities," in *Global Communications Conference, 2016 GLOBECOM 2016. IEEE*, Dec 2016.

[18] Q. Wang and M. Liu, "Learning in hide-and-seek," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–14, 2015.

[19] M. Dabaghchian, A. Alipour-Fanid, S. Liu, K. Zeng, X. Li, and Y. Chen, "Who is smarter? intelligence measure of learning-based cognitive radios," *CoRR*, vol. abs/1712.09315, 2017. [Online]. Available: http://arxiv.org/abs/1712.09315

[20] H. Su, Q. Wang, K. Ren, and K. Xing, "Jamming-resilient dynamic spectrum access for cognitive radio networks," in *Communications (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–5.

[21] Q. Wang, K. Ren, and P. Ning, "Anti-jamming communication in cognitive radio networks with unknown channel statistics," in *19th IEEE International Conference on Network Protocols*, 2011, pp. 393–402.

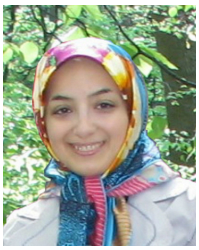
[22] Q. Wang, P. Xu, K. Ren, and X. y. Li, "Delay-bounded adaptive ufh-based anti-jamming wireless communication," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 1413–1421.

[23] K. Ezirim, S. Sengupta, and E. Troia, "(multiple) channel acquisition and contention handling mechanisms for dynamic spectrum access in a distributed system of cognitive radio networks," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*, Jan 2013, pp. 252–256.

[24] P. Lin and T. Lin, "Optimal dynamic spectrum access in multi-channel multi-user cognitive radio networks," in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2010, pp. 1637–1642.

[25] Y. Liao, T. Wang, K. Bian, L. Song, and Z. Han, "Decentralized dynamic spectrum access in full-duplex cognitive radio networks," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 7552–7557.

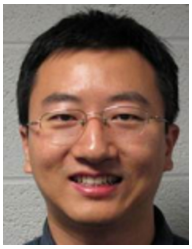
[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 2006.



Monireh Dabaghchian received the B.S. and M.S. degrees both in Electrical Engineering-Communications from University of Tabriz, Tabriz, Iran in 2006 and 2009, respectively. She is currently a Ph.D. student in the Electrical and Computer Engineering Department at George Mason University, Fairfax, VA, USA. Her research interests are broadly in theoretical machine learning, online learning algorithms, Multi-armed-bandit, statistical learning, information theory and their applications in security and privacy of wireless communication and spectrum sharing networks.

Amir Alipour-Fanid received the B.S. degree in Electrical Engineering-Power from Islamic Azad University of Ardabil, Ardabil, Iran in 2005, and the M.S. degree in Electrical Engineering-Communication from University of Tabriz, Tabriz, Iran, in 2008. Currently, he is working toward his Ph.D. in Electrical and Computer Engineering Department at George Mason University, Fairfax, VA, USA.

His research interests include machine learning applications in wireless networks, device identification, cyber-physical systems, vehicle-to-vehicle communication, security and privacy in Internet of Things.



Kai Zeng is an associate professor in Department of Electrical and Computer Engineering, Department of Computer Science, and Center for Secure Information Systems at George Mason University. He received his Ph.D. degree in Electrical and Computer Engineering at Worcester Polytechnic Institute (WPI) in 2008. He was a postdoctoral scholar in the Department of Computer Science at University of California, Davis (UCD) from 2008 to 2011. He worked in the Department of Computer and Information Science at University of Michigan - Dearborn as an assistant professor from 2011 to 2014. He was a recipient of the U.S. National Science Foundation Faculty Early Career Development (CAREER) award in 2012. He won Excellence in Postdoctoral Research Award at UCD in 2011 and Sigma Xi Outstanding Ph.D. Dissertation Award at WPI in 2008. He is an editor of IEEE Transactions on Wireless Communications. His current research interests are in cyber-physical system security and privacy, physical layer security, network forensics, and cognitive radio networks.

Qingsi Wang received the PhD degree in electrical engineering from the University of Michigan, Ann Arbor, in 2014. His primary research interest is in networked systems and statistical learning, and the most of his past work is particularly on resource allocation problems in wireless networks, with the application of stochastic control, game theory and online learning theory. He was affiliated with Qualcomm Research from 2014 to 2017, focusing on the design, evaluation and standardization of 4G/5G systems on the unlicensed spectrum. Since 2017, he is affiliated with Google Inc.

Peter Auer Peter Auer received his Ph.D. degree in mathematics from the Vienna University of Technology, Austria, in 1992.

From 1988 to 1992 he was Research Assistant with the Institute for Statistics and Probability Theory and the Institute for Applied Computer Science at the Vienna University of Technology. After receiving his Ph.D. degree he was Assistant Professor and since 1997 Associate Professor with the Institute for Theoretical Computer Science at the Graz University of Technology, Austria. From 1995 to 1996 he was Visiting Research Scholar with the University of California at Santa Cruz, USA. Since 2003 he is Full professor for Information Technology at the Montanuniversitaet Leoben, Austria. He is Action Editor for the *Journal of Machine Learning Research* and Editorial Board Member of the *Machine Learning Journal*. He has worked in probability theory and symbolic computation, and his current research interest are machine learning algorithms that interact with their environment, in particular multi-armed bandit problems and reinforcement learning.

Prof. Auer is a member of the ACM and has been President of the Association for Computational Learning for six years.