Mitigating selfish primary user emulation attack in multi-channel context: A surveillance game approach

Duc-Tuyen Ta\textsuperscript{1}, Nhan Nguyen-Thanh\textsuperscript{1}, Patrick Maille\textsuperscript{2}, Philippe Ciblat\textsuperscript{1}, and Van-Tam Nguyen\textsuperscript{1,3}  \\
\textsuperscript{1}LTCI, CNRS, Télécom ParisTech, Université Paris Saclay, 75013, Paris, France  \\
\textsuperscript{2}Telecom Bretagne/IRISA, Institut Mines-Telecom, Rennes, France  \\
\textsuperscript{3}Department of EECS, University of California at Berkeley, USA

Abstract—Primary User Emulation Attack (PUEA), in which attackers emulate primary user signals causing restriction of secondary access on the attacked channels, is a serious security problem in Cognitive Radio Networks (CRNs). If the attacker performs a PUEA for occupying the attacked channel selfishly, it is called a selfish attacker. Network managers could adopt a surveillance process on disallowed channels for identifying illegal channel occupation of selfish PUE attackers and hence mitigating selfish PUEA. Determining surveillance strategies, particularly in multi-channel context, is necessary for ensuring network operation fairness. In this paper, we adopt a sequence-form representation of the game between the selfish attack and the surveillance process, called surveillance game, to formulate the relation between the selfish attack and the surveillance process in multi-channel context and explore the Nash Equilibrium (NE). We show that performing the obtained NE surveillance strategy significantly mitigates PUEA.

I. INTRODUCTION

Cognitive radio (CR) is introduced as a solution to improve spectrum utilization by enabling secondary access to licensed spectrum. Discovering spectrum opportunities, there are two approaches: spectrum sensing and database-driven. In the spectrum sensing approach, the primary users’ activity is explored by measuring the spectrum environment, while in the database-driven approach, the information of spectrum usage for CR users is provided by a database server. Compared to the database-driven approach, the spectrum sensing approach is cheaper and more flexible for a wide range of networks\cite{1}. However, spectrum sensing faces a serious security risk: Primary User Emulation Attack (PUEA)\cite{2}–\cite{8}. In PUEA, the attacker transmits an emulated Primary User (PU) signal which could lead to disallowed state on the attacked channels (i.e., the channels which spectrum sensing system claims to be busy after the sensing period). The attacker therefore gains exclusive spectrum right. Depending on goals, PUEA can be categorized into two types: selfish and malicious. A malicious PUEA targets at ruining the operation of networks, hence it is similar to the jamming attack or the denial-of-service attack. A selfish PUEA, however, aims at selfishly occupying the attacked channel for data transmission. In that sense, selfish PUEA is associated with an illegal benefit degrading the fairness of CR networks (CRNs) and possible interference threatening to primary systems. In this study, we therefore focus on how to mitigate the selfish PUEA on CRNs.

Several earlier works have investigated the issues of selfish PUEA. Most of them adopt detection methods based on additional information, such as the locations of both primary and secondary users\cite{2} or the frequency deviation feature of FM signal\cite{4} to identify the PUE attackers. However, those methods are only applicable on special cases where added information, or the imperfection of attacked signals is available. The other works applied the game-theoretical framework to analyze the risk of PUEA\cite{5}–\cite{10} since there are the opposing objectives between the attacker and the network manager. In\cite{5}, the authors formulate a non-cooperative multistage game between a selfish PUEA attacker and a secondary node on the data transmission phase. A dogfight spectrum game between a PUE attacker and a CR user is formulated in\cite{9}. In that work, PUE attacking signals are treated as jamming signals and channel hopping is proposed as a solution for mitigating PUEAs. However, there is still vulnerability if the attacker conducts multiple channel attacks.

A successful selfish PUEA is usually followed by selfishly using of the attacked channel by the attacker. Therefore, it is possible to determine the illegal accessing in any communication link through the user’s identification. In our previous works\cite{7}, \cite{8}, we have proposed a surveillance process to mitigate the influence of selfish PUEA by monitoring data traffic at the beginning of the data frame on an occupied channel. To determine the surveillance strategy of the network manager, we use a game theoretic approach to formulate the relation between the attack and the surveillance process in a single channel. The best strategies of the attacker and the network manager are figured out in closed-form, as a Nash equilibrium (NE) point. However, CRNs usually work on multiple frequency bands, and because of the rapid expansion of software-defined radio, the attacker can launch multi-channel selfish PUEAs. For such a case, a sequential monitoring plan can be used, however, at the cost of long surveillance time. In this paper, we therefore consider the multi-channel surveillance process to mitigate the influence of the selfish PUEA in CRNs.

The multi-channel surveillance model is more complicated but more realistic than the single-channel model. We formulate
the relation between the selfish PUEA and the surveillance process as a two-player game in extensive form and consider the Nash Equilibrium (NE) for the surveillance strategy. This approach can be extended to mitigate the influence of malicious PUEA in CRNs.

Typically, the network manager observes the attacker’s action only indirectly, through the sensing results. Hence, the formulated game is an incomplete and imperfect information game. Finding a Nash equilibrium solution in this game is more complicated than in perfect-information games. We employ the sequence-representation method [11], [12] instead of the conventional (benchmark) strategic-representation method [13] to determine the best strategy for the defender and the attacker. We prove that the sequence-form representation is much more efficient than the strategic-representation method. We then analyze and interpret the impact of the PU’s presence probabilities, the network demand, as well as the penalty factor on the obtained NE strategies.

II. SYSTEM MODEL

We call the attacker, the representative of PUEA users and the defender, the resource manager of the network which monitors the traffic on disallowed channels to detect selfish attackers. Let $N$ be the number of available channels, $M$ be the maximum number of channels that the attacker can attack, and $L$ be the maximum number of channels that the defender can monitor.

The timing frame for network operation is the same as our previous work (Figure 1 in [7]). Before the transmission (data phase), a sensing phase, possibly followed by a surveillance phase, is carried out. For each channel, because of the non-ideal sensors, two possible sensing results could be obtained: “allowed”, and “disallowed”. An attacker can implement a selfish PUEA by transmitting the emulated primary user signals during the sensing time. We assume the sensors cannot distinguish the emulated and authentic primary signals. Consequently, the PUEA will not be detected in the sensing time. During a PUEA, the attacker cannot know the true status of the primary user on the attacked channel since it cannot sense for PU signal at the same time. This means the PUEA attacks the channel in a blind condition regarding the primary signal status. Concerning the defense against selfish PUEA threats, we assume that a fixed format data frame is used for exchanging data with all CR users including selfish users. The format contains the identifying information of a user such as the medium access control address. Therefore, CR users can be identified by observing transmitted signals in data time. Channel surveillance processes which are conducted on monitored secondary access channels can identify the selfish attackers if they are presented. Once a selfish attacker has been detected, punishments such as network isolation or bandwidth limitation can be adopted to penalize the attacker. This surveillance process is assumed to be implemented by the resource manager.

Before the analysis, we summarize the specific notations that will be used throughout the paper (Table I).

### Table I: Notations (at $i$ channel)

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_i$</td>
<td>The presence probability of PU.</td>
</tr>
<tr>
<td>$p^{i}_{NS}$</td>
<td>The probability that the sensor answers disallowed channel when the attacker does not attack.</td>
</tr>
<tr>
<td>$p^{i}_{A}$</td>
<td>The probability that sensor answers disallowed channel when the attacker attacks.</td>
</tr>
<tr>
<td>$p^{i}_{N}$</td>
<td>The probability that the channel is not used by the PU while the sensor claims disallowed and the attacker does not attack.</td>
</tr>
<tr>
<td>$p^{i}_{DA}$</td>
<td>The probability that the channel is not used by the PU while the sensor claims disallowed and the attacker attacks.</td>
</tr>
<tr>
<td>$C_i$</td>
<td>The implementing cost of the selfish PUEA.</td>
</tr>
<tr>
<td>$G^DA_i$</td>
<td>The using gain of selfish PUEA attacker at one data frame.</td>
</tr>
<tr>
<td>$C^M_i$</td>
<td>The monitoring cost of the resource manager at the channel.</td>
</tr>
<tr>
<td>$G^M_i$</td>
<td>The capturing gain for detecting illegal attack during the surveillance process of data frame.</td>
</tr>
<tr>
<td>$p_i$</td>
<td>The penalty value for being captured at the channel.</td>
</tr>
</tbody>
</table>

![Fig. 1: Two-channel surveillance game](image)

III. GAME FORMULATION

We formulate a two-player game in the extensive form, called the multi-channel surveillance game (MSG), to present the relationship between the multi-channel surveillance process and the selfish PUEA in CRNs. There are two players: **Attacker** (player 1) who represents selfish PUEA attackers, and **Defender** (player 2) who represents the resource manager.

Figure 1 illustrates the MSG game for a CRN with two channels ($N = 2$), the attack capability of the attacker $M = 1$ and the monitoring capability of the defender $L = 1$.

#### 1) Strategies: We denote by $A_0, A_1, A_2, \ldots, A_{K_1}$ the actions of the attacker, where $A_0$ corresponds to the action **No Attack** and $A_1$ corresponds to the action **Attack** of a non-empty subset of available channels. The attacker has $K_1$ actions leading to its pure strategy set $\Sigma_A$:

$$\Sigma_A = \{A_0, A_1, A_2, \ldots, A_{K_1}\}$$

(1)

where

$$K_1 = \sum_{i=1}^{\min(M,N)} \binom{N}{i}$$

(2)

Let $C$ be the set of the sensing result of all channels and $C$ includes $2^N$ elements. For a sensing result $C_k \in C$ ($k = 0, \ldots, 2^N - 1$), let $|C_k|$ be the number of disallowed channels. We denote by $D_0|C_k, D_1|C_k, D_2|C_k, \ldots, D_{|C_k|}|C_k$ the possible actions of the defender in which $D_0|C_k$ corresponds to the **No Surveillance** action and $D_j|C_k$ corresponds to
the Surveillance action of a disallowed channel subset belong $C_k$. The pure strategy set of the defender has $K_2$ actions leading to its pure strategy set $\Sigma_D$.

$$\Sigma_D = \bigcup_{k=0}^{2^N-1} \{D_0|C_k, D_1|C_k, D_2|C_k, \ldots, D_{|C_k|}|C_k\}_{C_k \in C}, \quad (3)$$

and

$$K_2 = 1 + \sum_{m=1}^{\min(N,L)} \sum_{n=1}^{m} \binom{m}{n} \times \binom{N}{m}, \quad (4)$$

We denote by $\Delta_A$ the mixed strategy set of the attacker and $\Delta_D$ the mixed strategy set of the defender.

2) Expected payoff: After sensing phase, there are two possible sensing results for $t$ channel: 1) disallowed and 2) allowed. Depending on the action of two players, their payoffs at $t$ channel are presented in Table III. These payoffs are computed by considering the present of PU which is represented by the values of $\rho_A^t$ and $\rho_D^t$.

<table>
<thead>
<tr>
<th>Table II: Payoffs of the attacker and the defender at $t$ channel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[Defender/Attacker]</strong></td>
</tr>
<tr>
<td>Disallowed</td>
</tr>
<tr>
<td>Surveillance</td>
</tr>
<tr>
<td>Allowed</td>
</tr>
</tbody>
</table>

We denote by $U_A^{t,D_k|C_j}$ the obtained payoff of the attacker by performing the Attack of $t$ channel while the defender plays $D_k|C_j$. Similarly, we denote by $U_D^{A_i,D_k|C_j}$ the obtained payoff of the defender by monitoring $t$ channel while the attacker plays $A_i$. For the action pair $\{A_i, D_j|C_k\}$, the payoffs of the defender $U_D^{A_i,D_k|C_j}$ and the attacker $U_A^{A_i,D_k|C_j}$ are given by

$$U_A^{A_i,D_k|C_j} = \sum_{t \in A_i} U_A^{t,D_k|C_j}, \quad (5)$$

$$U_D^{A_i,D_k|C_j} = \sum_{t \in D_k} U_D^{A_i,t}, \quad (6)$$

We denote by $\Omega_A$ the expected payoff of the attacker and $\Omega_D$ the expected payoff of the defender, respectively. We have

$$\Omega_D = \sum_{i=1}^{K_1} \delta_a(A_i) \sum_{j=0}^{2^{|C_j|}-1} p_{C_j|A_i} \delta_d(D_k|C_j) U_D^{A_i,D_k|C_j}, \quad (7)$$

$$\Omega_A = \sum_{i=1}^{K_1} \delta_a(A_i) \sum_{j=0}^{2^{|C_j|}-1} p_{C_j|A_i} \delta_d(D_k|C_j) U_A^{A_i,D_k|C_j}, \quad (8)$$

where $p_{C_j|A_i}$ is the probability of the sensing result $C_j$ under the attacker’s action $A_i$ and $\delta_a(A_i) = \in \Delta_A$ and $\delta_d(D_k|C_j) = \in \Delta_D$ are the mixed strategy of action $A_i$ and $D_k|C_j$, respectively.

IV. NASH EQUILIBRIUM

For the MSG game, we explore the Nash equilibrium (NE) point, i.e., the point where each player has selected a best response (BR) strategy to other players’ strategies. The BR are the strategies on which the player gains the highest payoff given other players’ strategies. A NE strategy may be a “pure” or a “mixed” strategy. To determine the NE point, two approaches are considered: 1) the conventional strategic-form representation and 2) the sequence-form representation.

A. Strategic-form representation

We first consider the conventional strategic-form representation, which is based on the Harsanyi transformation [13] and the Lemke-Howson (L-H) algorithm [14]. The Harsanyi transformation models all possible actions of a player, which are affected by the other players’ actions and the nature choices. For the MSG game, however, the method results in an exponential increment in the size of the game. In particular, the size of payoff matrix in the MSG game adopting the strategic-form representation is $(K_1 + 1) \times K_3$, where $K_3$ is given by

$$K_3 = \prod_{k=0}^{2^N-1} \left( \min\left(M, |C_k|\right) \right) \quad (9)$$

For the case that $M = L = 1$, $K_3$ is

$$K_3 = K_3^1 = \prod_{k=0}^{N} (k + 1)^{N_k}, \quad (10)$$

where $N_k$ denote a binomial coefficient indexed by $N$ and $k$. For $M = L = 1$, the payoff matrix is $3 \times 12$ if $N = 2$ and $5 \times (5 \times 12^6$ if $N = 4$. It is significantly larger when $M$ and $L$ bigger than 1. Consequently, it is very complicated to find the NE points of the game for the large $N$.

B. Sequence-form representation

In game theory, a perfect recall game is given by a tree with a finite number of nodes, change moves (with positive probabilities), payoffs for both players at the leaves and information sets partitioning the set of remaining decision nodes. Each node has a unique path from the root. A sequence is defined as a string listing the action choices of a particular player. In detail, for each node $h$ of player $i$, we define $\sigma_h$ as the sequence of player $i$ at $h$. Suppose $C_h$ is the set of choices at an information set of $h$. For each choice $c \in C_h$, the corresponding sequence is $\sigma_h c$. Hence, the set of sequences $\Sigma_i$ for player $i$ is given by

$$\Sigma_i = \{\emptyset\} \cup \{\sigma_h c | h \in H_i, c \in C_h\} \quad (11)$$

Since the MSG game is a perfect recall, we will adopt the sequence-form representation to solve the game. The trick here is that we consider the sensing results as the elements of the attacker’s sequence. The sequence strategy set of the attacker then is

$$\Sigma_A^{seq} = \{\sigma_{A_i}, i = 1 \ldots K_4\} = \{\emptyset, A_0, A_1, \ldots, A_{K_1}, A_0, c_0, A_0, c_1, \ldots\} \quad (12)$$

where $K_4$ is given by

$$K_4 = 1 + (K_1 + 1) + (K_1 + 1) \times 2^N \quad (13)$$
The corresponding sequence strategy set of the defender is
\[
\Sigma^\text{seq}_D = \{\sigma_D j = 1 \ldots (K_2 + 1)\}
\]
\[
= \{\emptyset, D_0|C_k, D_1|C_k, \ldots, D_{|C_k||C_k|} \}_{C_k \in C}
\] (14)
The mixed strategy for the sequence-form representation is presented by the probability of each sequence in the sequence strategy set. Let \(\Phi_A\) and \(\Phi_D\) denote the mixed sequence strategies for the attacker and the defender, respectively. From the definition of the sequence strategy set, we have
\[
\Phi^\text{seq}_D = \{\phi_D^i j = 1 \ldots (K_2 + 1)\}
\] (15)
\[
\Phi^\text{seq}_A = \{\phi_A^i j = 1 \ldots K_2\}
\] (16)
where \(\phi_A^i\) is the probability of the attacker’s \(i^{th}\) sequence, \(\phi_D^j\) is the probability of the defender’s \(j^{th}\) sequence.

The relation between these mixed strategies is called the realization plan. By default, for an empty sequence, the probability is 1. For any node \(h\), the mixed strategy of the sequence at \(h\) is the sum of all mixed strategies from \(h\).

Therefore, we obtain the realization plans for the defender and the attacker:

\[
\begin{cases}
\phi_A(\emptyset) = 1 \\
\phi_A(D_o|C_k) + \sum_{i=1}^{K_2} \phi_A(D_i|C_k) = 1 \forall C_k \in C \\
0 \leq \phi_A^i \leq 1, i = 1 \ldots K_2
\end{cases}
\] (17)

\[
\begin{cases}
\phi_D(\emptyset) = 1 \\
\phi_D(D_0|C_k) + \sum_{i=1}^{K_2} \phi_D(D_i|C_k) = 1 \\
\sum_{i=0}^{2^{K_2}-1} \phi_D(A_{|C_k|} C_k) = \phi_D(A_{|C_k|})
\end{cases}
\] (18)

In general, these realization plans can be re-written using the following matrix form.

\[
\begin{cases}
E\Phi_A = e, \quad & F\Phi_D = f \\
\Phi_A \geq 0, \quad & \Phi_D \geq 0
\end{cases}
\] (19)

where \(E\) and \(F\) are called the constraint matrices, and \(e\) and \(f\) are vectors in which the first element is 1, and the other elements are 0.

Let \(\Pi_A\) and \(\Pi_D\) denote the payoff matrix of the attacker and the defender in the sequence-form representation. For each pair of the player’s sequence strategy, the payoffs of the attacker and the defender are given in (4) and (5), respectively. The expected payoff of the attacker \(\Omega_A\) and the expected payoff of the defender \(\Omega_D\) are computed by
\[
\begin{align*}
\Omega_A &= \Phi^T_A \Pi_A \Phi_D \\
\Omega_D &= \Phi^T_D \Pi_D \Phi_D
\end{align*}
\] (20)

(21)
An equilibrium is a pair \((\Phi_A, \Phi_D)\) of mutual best responses. In particular, if the realization plan \(\Phi_D\) is fixed, then \(\Phi_A\) is the best response to \(\Phi_D\) if and only if it is an optimal solution of the linear program

\[
\begin{align*}
\text{maximize} & \quad \Phi^T_A (\Pi_A \Phi_D) \\
\text{subject to} & \quad E\Phi_A = e \\
& \quad 0 \leq \Phi_A \leq 1
\end{align*}
\] (22)

The dual form of linear program (21) is as follows.

\[
\begin{align*}
\text{minimize} & \quad e^T p \\
\text{subject to} & \quad E^T p \geq \Pi_A \Phi_D \\
& \quad \Phi^T_A (-\Pi_A \Phi_D + E^T p) = 0
\end{align*}
\] (23)

A similar program is established for the attacker strategy.

In conclusion, the problem of finding a Nash equilibrium can be formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad e^T p \\
\text{minimize} & \quad f^T q \\
\text{subject to} & \quad E^T p \geq \Pi_A \Phi_D, \\& \quad F^T q \geq \Pi_D \Phi_D \\
& \quad \Phi^T_A (-\Pi_A \Phi_D + E^T p) = 0 \\
& \quad \Phi^T_D (-\Pi_D \Phi_D + F^T q) = 0 \\
& \quad E\Phi_A = e, \\
& \quad F\Phi_D = f \\
& \quad \Phi_A \geq 0, \\
& \quad \Phi_D \geq 0
\end{align*}
\] (24)

The value of \(p, q, \Phi_A, \Phi_D\) that satisfies the constraints in (24) can be found through the Linear Complementary Programming (LCP) by introducing the non-negative vector \(z = (\Phi_A, \Phi_D, p, p^T, q, q^T)\), where \(p, p^T, q, q^T\) are non-negative vectors of the same dimension as \(p, p^T\) and \(q, q^T\). Moreover, we let

\[
M = \begin{bmatrix}
0 & -\Pi_A & E^T & -E^T & 0 & 0 \\
-\Pi_D^T & 0 & 0 & 0 & 0 & -F^T \\
-E & 0 & 0 & 0 & 0 & 0 \\
-E & 0 & 0 & 0 & 0 & 0 \\
0 & -F & 0 & 0 & 0 & 0 \\
0 & -F & 0 & 0 & 0 & 0
\end{bmatrix}
\] (25)

and \(b^T = (0, 0, e, -e, f, -f)^T\). Then, we have the LCP problem as follows:

\[
\begin{align*}
\text{find} & \quad z \\
\text{s.t} & \quad Mz + b \geq 0 \\
z^T (Mz + b) = 0 \\
z \geq 0
\end{align*}
\] (26)

The LCP problem above could be solved by the Lemke algorithm [11, 12, 15]. The main idea of the Lemke algorithm is to apply the pivoting operation in the complementary problem. A more detailed description can be found in [15]. Since a feasible solution at least exists for the formulated LCP problem [11], hence the solution of (24) could be found by considering the solution of (26). This solution is the NE point of the game. We have the following propositions.

**Proposition 1:** In the MSG game, the payoff matrix’s size in the sequence-form representation is much smaller than the payoff matrix’s size in the strategic-form representation.

**Proof:** We observe that the payoff matrix in the sequence-form representation is much smaller than one in the strategic-form because \(K_4 \ll K_3\). In particular, the payoff matrix in the sequence-form is linear in the size of the game whereas the payoff matrix in the strategic-form is generally exponential. For example, we consider for a CRN with the number
of available channels \( N = 4 \) where the attack capability of the attacker and the surveillance capability of the defender \( M = L = 1 \). The size of the corresponding payoff matrix is \( 86 \times 48 \), which is much smaller than one with the strategic-form representation \( (5 \times (5 \times 12^6)) \).

Remark 1: Below are some results from Proposition 1.

- The strategy space of the sequence-form representation is exponentially smaller than the strategy space of the strategic-form representation. Since the two methods operate similarly \([11], [12]\), the run time of each algorithm depends on the size of the input. Thus, it is exponentially faster to run the Lemke algorithm on the sequence-form than the \( \text{L}-\text{H} \) algorithm on the strategic-form.
- Since the sequence-form representation is much more compact than the strategic-form representation, we therefore adopt the sequence-form representation approach to determine the NE strategy of the MSG game.
- If the MSG game is a zero-sum game, (i.e., \( \Omega_A = -\Omega_D \)), the optimization problem \([24]\) could be reduced to a linear programming problem.

Proposition 2: If \( L \geq N \), the NE point of the MSG game is unique and corresponds to the combination result of \( N \) independent single-channel surveillance games (results in our previous work \([7]\)), each with one channel in the set of available channels.

Proof: (sketch) We consider the CRN with \( N \) available channels. When \( L \geq N \) m the defender can perform to surveillance on all disallowed channels. In this case, it is equivalent to perform single-channel surveillance game of all disallowed channels independently. For each single-channel surveillance game, the NE point is unique and has been figured out in \([7]\). Therefore, the NE point of the MSG game with \( L \geq N \) is unique and could be determined by combining \( N \) unique NE points of \( N \) single-channel surveillance games.

V. NUMERICAL RESULTS

In order to analyze the impact of system parameters to the NE point of the MSG game, the numerical computations are conducted in Matlab with the Gambit toolbox \([16]\) for game theory. We also assume the sensing system adopts the energy detection method and the average SNRs of the game theory. We also assume the sensing system adopts the energy detection method and the average SNRs of the game theory. We also assume the sensing system adopts the energy detection method and the average SNRs of the game theory. We also assume the sensing system adopts the energy detection method and the average SNRs of the game theory. We also assume the sensing system adopts the energy detection method and the average SNRs of the game theory. We also assume the sensing system adopts the energy detection method and the average SNRs of the game theory.

TABLE III: The computational time to determine the NE point.

<table>
<thead>
<tr>
<th></th>
<th>( N = 2 )</th>
<th>( N = 3 )</th>
<th>( N = 4 )</th>
<th>( N = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategic-form</td>
<td>2 s</td>
<td>1960.7 s</td>
<td>( \infty )</td>
<td>( \infty )</td>
</tr>
<tr>
<td>Sequence-form</td>
<td>2 s</td>
<td>32.4 s</td>
<td>11564 s</td>
<td>( \sim 12 \text{ h} )</td>
</tr>
</tbody>
</table>

Next, for illustrating the effect of parameters on NE strategy of the MSG game, we consider a CRN with two available channels \((N = 2)\), the attack capability of attacker \( M = 1 \) and the monitoring capability of the defender \( L = 1 \). We assume that \( k_A = 0.2 \), \( k_S = 0.1 \), \( \pi_1 = 0.2 \) and \( \pi_2 = 0.5 \).

Fig. 2a, Fig. 2b and Fig. 2c present the NE strategy of the MSG game for low penalty \((k_C = 0.4)\), medium penalty \((k_C = 2)\) and high penalty \((k_C = 10)\), respectively. To provide a better view, we only plot the NE strategies of the attack actions and the surveillance actions. We observe that for each the penalty factors \( k_C \), the distribution of NE points is separated into three regions along with the increase of the network demand \( k_b \). First, when the network demand \( k_b \) is low, the defender does not need to monitor disallowed channels. Consequently, the attacker could perform the attack to the best channel (channel with low operation of PUs). Second, when the network demand \( k_b \) is medium, the attacker has a trend of moving the attack from the best channel to the worst channel. The reason of this trend is that the defender performs its surveillance on the best channel due to the increase of its spectrum demand. Third, when the network demand \( k_b \) is very high, the defender increases the surveillance rate at the both channels. As a result, the attacker has a corresponding response by adjusting its attacking rate on the both channels. For each region, the strategy of defender depends on the relation between the network demand \( k_b \) and the penalty factor \( k_C \). For a fixed penalty factor \( k_C \), the defender performs monitoring the disallowed channel with a constant probability for each region. In addition, the size of each region also depends on the penalty factor \( k_C \) where the last region corresponding with high network demand \( k_b \) is enlarged with penalty factor values. The results mean that the NE strategies of the MSG game are affected by both the penalty factor and the network demand. We conclude that in order to reduce the influence of PUEA, the CRNs should set a high penalty.
between NE and \textit{uniform surveillance} (\textit{UniS}) strategies, (b) between NE and \textit{random} (\textit{Rand}) strategies.

Figure 3 shows the differences on the defender’s expected payoffs when the defender plays (a) NE and \textit{uniform surveillance} (\textit{UniS}) strategies, (b) NE and \textit{random} (\textit{Rand}) strategies.

Figure 3 shows the differences on the defender’s expected payoffs when the defender plays (a) NE and \textit{uniform surveillance} (\textit{UniS}) strategies, (b) NE and \textit{random} (\textit{Rand}) strategies.

VI. CONCLUSION

We have discussed the multi-channel surveillance process to deal with the selfish PUEA in multi-channel cognitive radio networks. Performing a multi-channel surveillance process on disallowed channels helps to identify selfish PUE attacker. The relation between attack strategies and the surveillance process has been formulated through an extensive-form game. Sequence representation method has been used for obtaining Nash equilibrium. The numerical results have shown the influence of the network demand and the penalty factor on controlling NE strategies and the effectiveness of using NE on mitigating PUEA.

Besides, the introduced method has a more efficient computational time compared to the conventional one. We will generalize this method to deal with other PUE attacker types such as malicious and unknown-attacking type.

REFERENCES