Research Article

A Game Theoretic Model for Wireless Sensor Networks with Hidden-Action Attacks

Yuxin Mao, Ping Zhu, and Guiyi Wei

School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

Correspondence should be addressed to Yuxin Mao; maoyuxin@zjgsu.edu.cn

Received 3 May 2013; Accepted 9 July 2013

Academic Editor: Jaime Lloret

Copyright © 2013 Yuxin Mao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The unattended nature of wireless sensor networks makes them very vulnerable to an adversary’s malicious attack. In this paper, we propose to apply game theory into solving the network security problem of wireless network. We explore game theory algorithms to model situation for wireless network with malicious nodes and investigate the attack and detection problem by modeling it as pairwise simultaneous game and spatial structured game. We consider the relationship between the nodes in a wireless sensor network to formulate the game and give the game theory algorithms in detail. We also evaluate the approach with a simulation experiment and analyze the simulation results in detail. We argue that the approach is able to support secure end-to-end communication in wireless sensor networks.

1. Introduction

The unattended nature of wireless sensor networks (WSNs) makes them very vulnerable to an adversary’s malicious attacks. An adversary can physically compromise a subset of nodes in a WSN to eavesdrop information. The compromised nodes (or malicious nodes) become black holes [1] in the network. Those black holes in a WSN raise hidden-action attacks [2] to reduce the performance of the network or even destroy the network. Therefore, network security is an important issue for WSNs. In a WSN with malicious nodes, there are obvious conflicts between malicious nodes and normal nodes. As a branch of applied mathematics, game theory is concerned with how rational entities make decisions in a situation of conflict, which has been used primarily in economics. Game theory aims to model situations in which decision makers have to make specific actions that have mutual—possibly conflicting—consequences [3]. In the context of WSNs, game theory may be used to form cooperation schemes among entities in a competitive environment, for example, power control, routing, and resource allocation. Game theory can be applied to model the situation where there are malicious nodes in a WSN.

In this paper, we try to explore game theory to model situation for wireless network with malicious nodes and solve the problem of secure wireless communications by using a game-based approach. The main contributions of this paper are summarized as follows: (1) a formal representation model for simulating a WSN with malicious nodes and hidden-action attacks by using game theory is given; (2) several efficient game-based algorithms are proposed to support reliable and secure wireless communications against the attacks of malicious nodes in WSN. The remaining of the paper is organized as follows. In Section 2, we first illustrate the system model including the network model as well as the problem statement. In Section 3, we present the game theoretic formulation for a WSN with malicious nodes. Game theory is used to solve the problem of reliable and secure wireless communications against the active attacks of malicious nodes in WSN. Moreover, we present the game theory algorithms in Section 4. In Section 5, we also evaluate the approach with a simulation experiment and analyze the simulation results in detail. Section 6 gives an overview of the related works. Section 7 concludes the paper with an outlook on future research directions.

2. Network Model and Problem Statement

2.1. Network Model. In this paper, we consider a relatively simple WSN. Consider $I = \{1, 2, \ldots, k\}$, and assume a WSN
with \( k \) nodes (see Figure 1). Each node has a unique identity \( i (i \in I) \) in the network. We assume the locations of the nodes are relatively static rather than dynamic. Each node in the network has at least one neighbor node. Each node in the network is battery powered and has limited computation and wireless communication capabilities. Assume the initial energy value of each node in the network is \( \eta \). Each normal node in the network has four kinds of actions.

1. **Forward** means a node sends a data packet to another node in the network.
2. **Receive** means a node receives a data packet from another node in the network.
3. **Detect** means a node receives a data packet from another node in the network and analyzes the pattern of the packet to find out any abnormality.
4. **Sleep** means a node turns off its antenna and does not take any other actions.

Each action (except sleep) of a node will consume certain energy. Therefore, either a normal node or a malicious node intends to sleep periodically, in order to save energy. An adversary is able to compromise a node or even physically capture a node. Therefore, there are a number of malicious nodes in the network. Assume the number of the malicious nodes is \( h \) (\( 0 < h \ll k \)). Each malicious node in the network has four kinds of actions.

1. **Forward**.
2. **Receive**.
3. **Jam** means a malicious node sends an abnormal data packet to another node in the network, in order to block the channel of the node.
4. **Sleep**.

Normal nodes do not have knowledge of the instantaneous channel state information (CSI) of malicious nodes, but they know their distribution, and vice versa. Moreover, we assume that the number of the malicious nodes is much smaller than the number of the sensor nodes in the network; that is, \( h \ll k \). It makes sense that a network is out of control if the number of the malicious nodes is very large. Another important assumption is that the type or identity (normal or malicious) of the nodes is unknown or blind. A node only knows the identity of itself.

### 2.2. Problem Statement

The major task of a normal node in the network is to transmit data to other nodes by routing. However, there is an additional task for normal nodes in a WSN with malicious nodes, that is, to detect the abnormal behaviors of malicious nodes. The major task of a malicious node is to block as many data packets of normal nodes as possible by attacking. A malicious node has two kinds of attack methods, passive methods like eavesdropping and active methods like jamming and DoS attack. In WSN, passive attacks of malicious nodes are hard to be detected, while active attacks have a risk of being detected. In this work, we mainly focus on the positive ones. Malicious nodes, in order to allay suspicions or save energy, only take attacks intermittently.

Moreover, we assume that detecting malicious nodes is an accumulative process to a normal node. Every detecting action of a normal node may gain a piece of evidence. When a normal node allocates enough evidence, it could then perform intrusion detection [4, 5] and locate malicious nodes.

### 3. Game Theoretic Formulation

In this section, we investigate the attack and detection problem by modeling it as a simultaneous game. In this game, we consider the relationship between every pair of nodes in the network. We try to formulate different cases for this game in detail.

#### 3.1. Pairwise Simultaneous Game

Let us consider a WSN with two nodes, a normal node \( n \) and a malicious node \( m \). It is a pairwise simultaneous game between the two nodes, where both players move simultaneously. Each player has four strategies (see Table 1). Here \( \gamma \) is the direct income for a node to send out a data packet successfully; that is, the packet is received by another node. \( \alpha_\gamma \) is the probability that a normal node sends out a packet successfully (\( 1 > \alpha_\gamma > 0 \)). \( \alpha_\gamma \) is the probability that a malicious node sends out a jamming packet successfully (\( 1 > \alpha_\gamma > \alpha_\gamma \)). It makes sense that jamming is more likely to be successful than forwarding. Here \( p \) is a punishment to a detected jamming action of the malicious node, and \( s \) is a stimulus to a successful jamming detection of the normal node (\( p > 0, s > 0 \)).

We formulate the energy consumption in the game by some constants. \( \beta_1 \) is the cost to receive a packet (receiving or detecting, \( \beta_1 > 0 \)), and \( \Delta \beta_1 \) is the additional cost for detecting attacks (\( \Delta \beta_1 > 0 \)). \( \beta_2 \) is the cost to send out a packet (forwarding or jamming, \( \beta_2 > 0 \)), and \( \Delta \beta_2 \) is the additional cost for jamming because jamming action consumes more energy compared with normal forwarding action (\( \Delta \beta_2 > 0 \)).
Table 1: The result of two nodes’ game (one normal node versus one malicious node).

| Malicious node (m) | Forward       | Detect          | Receive         | Sleep         |
|--------------------|---------------|-----------------|-----------------|---------------|
| Forward            | \{-\beta_2, -\beta_2\} | \{-\beta_2, - (\beta_1 + \Delta \beta_2)\} | \{\alpha_{F} - \beta_2, -\beta_1\} | \{-\beta_2, 0\} |
| Jam                | \{- (\beta_2 + \Delta \beta_2), -\beta_2\} | \{- p - (\beta_2 + \Delta \beta_2), s - (\beta_1 + \Delta \beta_1)\} | \{\alpha_{J} - (\beta_2 + \Delta \beta_2), -\beta_1\} | \{- (\beta_2 + \Delta \beta_2), 0\} |
| Receive            | \{-\beta_1, \alpha_{F} - \beta_2\} | \{-\beta_1, -\beta_1\} | \{-\beta_1, -\beta_1\} | \{-\beta_1, 0\} |
| Sleep              | \{0, -\beta_2\} | \{0, -\beta_1\} | \{0, -\beta_1\} | \{0, 0\} |

Figure 2: Different patterns for local subgames.

As we have mentioned before, nodes do not know the identity of each other in our network model. Even a malicious node is not able to confirm one of its neighbor nodes as its ally. Therefore, there is also a game between two malicious nodes, \(m_1\) and \(m_2\). The result of two malicious nodes’ game is shown in Table 2.

Similarly, there is a game between two normal nodes, \(n_1\) and \(n_2\). The result of two normal nodes’ game is illustrated in Table 3.

3.2. Spatial Structured Game. Let us consider a more complex situation. There are a number of subgames between different pairs of nodes (either normal or malicious) in the whole network. We could reduce the complete game into a number of two nodes’ games. For each two nodes’ game, we could get a result by the formulation in Section 2.2 and then synthesize the result of the complete game by the result of all the subgames. Considering a part of a WSN with three nodes \((n_1, n_2, n_3)\), we could get a series of patterns according to the network topology (see Figure 2).

The strategy set of a normal node is \(S_{normal} = \{F, D, R, S\}\), and the strategy set of a malicious node is \(S_{malicious} = \{F, J, R, S\}\). The meanings of \(F, D, J, R,\) and \(S\) are represented in Table 1. We use \(\ast\) to denote any action in \(S_{normal}\) or \(S_{malicious}\). Moreover, we use \(-X\) to denote any action except \(X\) in \(S_{normal}\) or \(S_{malicious}\). The payoff function for \(n_i\) in the subgame between \(n_i\) and \(n_j\) is denoted by \(u_{n_i/n_j}\).
Table 2: The result of two nodes’ game (one malicious node versus one malicious node).

| Malicious node (m₁) | Forward | Jam | Receive | Sleep |
|---------------------|---------|-----|---------|-------|
| Forward             | [-β₂, -β₁] | [-β₂, - (β₁ + Δβ₁)] | [αᵣᵧ - β₂, -β₁] | [-β₁, 0] |
| Jam                 | [-β₁, - (β₁ + Δβ₁)] | [-β₁, -β₁] | [-β₁, -β₁] | [-β₁, 0] |
| Receive             | [-β₁, αᵣᵧ - β₂] | [-β₁, αᵣᵧ - (β₂ + Δβ₂)] | [-β₁, -β₁] | [-β₁, 0] |
| Sleep               | [0, -β₁] | [0, - (β₂ + Δβ₂)] | [0, -β₁] | [0, 0] |

Table 3: The result of two nodes’ game (one normal node versus one normal node).

| Normal node (n₁) | Forward | Detect | Receive | Sleep |
|------------------|---------|--------|---------|-------|
| Forward          | [-β₁, -β₂] | [-β₁, - (β₂ + Δβ₂)] | [αᵣᵧ - β₁, -β₂] | [-β₂, 0] |
| Detect           | [-β₁, -β₂] | [-β₁, -β₁] | [-β₁, -β₁] | [-β₁, 0] |
| Receive          | [-β₁, αᵣᵧ - β₂] | [-β₁, αᵣᵧ - (β₂ + Δβ₂)] | [-β₁, -β₁] | [-β₁, 0] |
| Sleep            | [0, -β₁] | [0, -β₁] | [0, -β₁] | [0, 0] |

Assume that n₁ is always a normal node in this part and there are four cases for the identities of n₂ and n₃.

In pattern (a) and pattern (b), both neighbors of n₁ are normal. There is a subgame between n₁ and n₂. There is another subgame between n₁ and n₃. Due to the broadcast character of WSN, we shall not calculate the cost of a node for more than one time when we compute the payoff of the node that plays subgames with several nodes at the same time.

In these patterns, we have the following exception rules to the game result in Table 3.

1. If \( u_{n₁,n₂}(F, R) = αᵣᵧ - β₂ \), then \( u_{n₁,n₃}(F, *) = 0 \), and vice versa (Rule 1).
2. If \( u_{n₁,n₂}(F, *) = -β₂ \), then \( u_{n₁,n₃}(F, -R) = 0 \) and \( u_{n₁,n₃}(F, R) = αᵣᵧ \), and vice versa (Rule 2).
3. If \( u_{n₁,n₃}(D, *) = -(β₁ + Δβ₁) \) or \( u_{n₁,n₃}(D, *) = −β₁ \), then \( u_{n₁,n₃}(D, *) = 0 \), and vice versa (Rule 3).
4. If \( u_{n₁,n₃}(R, *) = -β₁ \), then \( u_{n₁,n₃}(R, -F) = 0 \), and vice versa (Rule 4).

In pattern (c) and pattern (d), n₁ has only one malicious neighbor (n₃), and the other normal neighbor (n₂) is normal. Besides the constraint in patterns (a) and (b), there is one more constraint for the three nodes in these two patterns. In pattern (c), the data that flows from n₂ to n₁ is affected by n₃. Therefore, when we compute the subgame between n₁ and n₂, we should consider this constraint. In pattern (d), there is one more edge between n₂ and n₃. That is, n₃ is a common neighbor of n₁ and n₂. It means that n₃ has effects on both n₁ and n₂, or both the data that flows from n₁ to n₂ and the one from n₂ to n₁ are affected by n₃.

In pattern (c), we have the following exception rules to the game results in Tables 1 and 3.

1. Rule 1.
2. Rule 2.
3. If \( u_{n₁,n₃}(D, *) = -(β₁ + Δβ₁) \) or \( u_{n₁,n₃}(D, *) = −β₁ \), then \( u_{n₁,n₃}(D, *) = 0 \), and vice versa (Rule 5).
4. If \( u_{n₁,n₃}(R, *) = -β₁ \), then \( u_{n₁,n₃}(R, -F) = 0 \), and vice versa (Rule 6).
5. If \( u_{n₁,n₃}(J, R) = αᵣᵧ - (β₂ + Δβ₂) \), then \( u_{n₁,n₃}(F, R) = -β₂ \) (Rule 7).
6. In pattern (d), we have one more exception rule.

In pattern (e) and pattern (f), both neighbors of n₁ are malicious. Similarly, with pattern (c) and pattern (d), a malicious node will affect the subgame of its neighbor. In pattern (e), the data that flows from n₃ to n₁ is affected by n₁, while the data that flows from n₃ to n₁ is also affected by n₂. In pattern (f), both the data flow from n₁ to n₂ and the one from n₂ to n₁ are affected by n₃, while both the data that flows from n₁ to n₃ and the one from n₃ to n₁ are affected by n₂.

In pattern (e), we have the following exception rules to the game results in Table 2.

1. Rule 1.
2. Rule 2.
3. If \( u_{n₁,n₃}(D, *) = -(β₁ + Δβ₁) \) or \( u_{n₁,n₃}(D, *) = −β₁ \), then \( u_{n₁,n₃}(D, *) = 0 \), and vice versa (Rule 9).
4. If \( u_{n₁,n₃}(J, R) = αᵣᵧ - (β₂ + Δβ₂) \), then \( u_{n₁,n₃}(F, R) = -β₂ \), and vice versa (Rule 10).
5. In pattern (f), we have one more exception rule.

If we take the whole network as an undirected graph, each edge denotes a subgame between two nodes. We could compute the game result of each edge in the graph by traversing the graph with a certain traversal algorithm (e.g., BFS or DFS).
4. Game Theory Algorithms

We could formulate the game for a WSN given in Section 3 by the following algorithms. First, let us consider the case for one specific wireless node (either normal node or malicious node). Algorithm 1 is proposed to compute the payoff of a specific wireless node.

The algorithm is trivial. The payoff of a specific node is impacted by the roles of its neighbors. In order to compute the payoff of a specific node, we have to consider the relationship between the node and each of its neighbors.

Then we could compute the payoffs of the whole network based on Algorithm 1. The algorithm for computing global payoffs is given as follows.

Algorithm 2 is also easy to be understood. The process of computing global payoffs is reduced to traversing an undirected graph by BFS. For each node, we just perform a single-node payoff computation, and then we get the information for the whole network. To a given WSN, we shall perform a series of computations to reach convergence.

5. Simulation and Evaluation

5.1. Simulation Setting. In this section, we construct simulation to evaluate the performance of the proposed approach. We consider a WSN with \( N \) players (wireless nodes). Player \( i \) is represented as a vertex \( v_i \) of a graph \( G(V, E) \), with \( v_i \in V \). An interaction between two players \( i \) and \( j \) is represented by an undirected edge \( e_{ij} \in E \). The number of neighbors of player \( i \) is the degree \( k_i \) of vertex \( v_i \). The average degree of the network is denoted by \( \langle k \rangle \). The terms vertex, individual, participant, and player are used interchangeably in this section, likewise for edge, interaction, and link. Each malicious node can take one of the four strategies: \{F, D, R, S\}, while each normal node can take one of the four strategies: \{F, J, R, S\}.

The BA scale-free network [6] is adopted to represent the population structure of our WSN, which is constructed according to the “growth” and “preferential attachment” mechanisms. Starting from \( m_0 \) fully connected nodes, a new node with \( m (m \leq m_0) \) edges is added to the system at every step. The new node links to \( m \) different nodes by a “preferential attachment” mechanism. The probability of connecting to an existing node \( i \) is proportional to its degree; that is, \( p_i = k_i / \sum j k_j \), where \( j \) runs over all existing nodes and \( k_i \) is the degree of node \( i \). After \( t \) time steps, this algorithm produces a graph with \( N = t + m_0 \) vertices and \( mt \) edges.

Let \( S(t) = (s_1(t), s_2(t), \ldots, s_N(t)) \) denote a configuration of the population strategies \( s_i(t) \in \{F, D, J, R, S\} \) at time step \( t \); the global synchronous system dynamics leads to \( S(t+1) \) by simultaneously updating all the players’ strategies according to the chosen rule, such as Fermi update rule [7]. Here, by synchronous, we mean that player’s strategy will not change (even if it had already changed during its strategy updating process at time step \( t \)) until all the other \( N-1 \) players complete their strategy updating processes.

The game process is the same as the standard evolutionary game. At each step, all nodes are synchronously updated according to a strategy update rule. Note that in realistic WSNs, there may be environmental noise, which influences individuals’ decisions (e.g., strategy mutation). Therefore, we adopt the Fermi updating rule, which considers environmental noise. When player \( i \) updates its strategy, it will first select a neighbor \( j \) out from all its \( k_i \) neighbors at random and then adopt \( j \)'s strategy with the probability \( H_{i \rightarrow j} = 1/(1 + e^{(P_j - P_i)/\kappa}) \).

Here, \( \kappa \) characterizes the environmental noise, including bounded rationality, individual trials, and errors in decision; \( \kappa \rightarrow \infty \) leads to neutral (random) drift whereas \( \kappa \rightarrow 0 \) corresponds to the imitation dynamics, where player \( j \)'s strategy replaces player \( i \)'s whenever \( P_j > P_i \). For finite value of \( \kappa \), the smaller the \( \kappa \) is, the more likely the fitter strategy is to replace the less fit one; thus the value of \( \kappa \) indicates the intensity of selection.

Finally, Tables 1–3 describe the payoffs of normal nodes and malicious nodes; Figure 2 depicts all the six exception rules in the network.

5.2. Experiment Results. In the following simulations, we study the effect of \( p_j \) (cost of receiving a package), \( \beta_j \) (cost of sending out a package), \( \Delta \beta_j \) (additional cost of detecting attacks), \( \Delta \beta_j \) (additional cost of detecting jamming), \( p \) (punishment to a detected jamming action of the malicious node), and \( s \) (stimulus to a successful jamming detection of the normal node). \( \alpha_F \) is set to 75%, and \( \alpha_J \) is set to 100% (i.e., whenever a malicious node decides to jam the network, it will succeed). Besides, \( \gamma \) the direct income for a node to send out a data packet successfully is set to unit payoff. All the simulations were carried out on a BA scale-free network with network size \( N = 1000 \) and \( m = m_0 = 4 \); therefore, the average degree \( \langle k \rangle = 8 \). Initially, normal nodes and malicious nodes are randomly distributed among the population with probability of 90% and 10%, respectively. In addition, for normal nodes, the four strategies, that is, \( F, D, R, \) and \( S \), are assigned the same probability, that is, 25%; for malicious nodes, the four strategies, that is, \( F, J, R, \) and \( S \), are also assigned the same probability. The equilibrium frequencies of malicious jamming nodes are obtained by averaging over 3000 generations after a transient time of 10,000 generations. Each piece of data is averaged over 1000 runs on 1000 different networks.
Input: A WSN with a collection of nodes \( N = \{n_0, n_1, \ldots, n_k\} \), a source node \( n_0 \), and a collection of malicious nodes \( M = \{m_1, m_2, \ldots, m_h\} \), where \( n_0 \in N, M \subset N \).

Output: A collection of payoff values \( P = \{p_0, p_1, \ldots, p_k\} \) for \( N \), where \( p_i \) is the payoff of \( n_i \) (\( i = 0, 1, \ldots, k \)).

Assign an initial action to each \( n_i \) in \( N \) and the action of \( n_0 \) is Forward
Set \( n_0 \) as the root node
Perform BFS by starting with \( n_0 \)
for each intermediate node \( n_i \) in BFS
    compute payoff for \( n_i \) by using Algorithm 1
get \( p_i \) for \( n_i \)
end loop
Return \( P = \{p_0, p_1, \ldots, p_k\} \)

Algorithm 2: The game algorithm for a WSN with malicious nodes.

5.2.1. Action Cost Effect. Figure 3 shows the relationship between network action \( (F, D, J, R, S) \) cost and \( \rho_f \), where \( \rho_f = N^f_m/N_m \) is the ratio of the number of malicious nodes that take the jamming action \( N^J_m \) to the total number of malicious nodes \( N_m \) in the equilibrium state. As previously referred, the initial proportion of malicious nodes \( N_m/N = 10\% \) and of \( N^F_m \) (number of initial malicious forwarding nodes), \( N^R_m \) (number of initial malicious receiving nodes), \( N^J_m \) (number of initial malicious jamming nodes), and \( N^S_m \) (number of initial malicious sleeping nodes) is assigned 25\%, respectively. We can see that by increasing package-receiving cost or additional detecting cost, \( \rho_f \) increases; however, when increasing package-sending cost or additional jamming cost, \( \rho_f \) decreases. We argue that this result is expected. It makes sense that normal nodes incline to perform package sending rather than detecting when the detecting cost increases. As a result, malicious nodes are more likely to perform jamming because there are not enough normal nodes detecting their attacks. Moreover, malicious nodes incline to be (or pretend to be) “normal” or when the cost of attacks increases.

5.2.2. Coaction Effect. To further explore the effect of network action cost, we jointly study the coaction of forwarding and receiving. The result is shown in Figure 4. The value of \( \rho_f \) is represented in the form of color depth. As Figure 3 indicated, malicious nodes that take the jamming action will emerge when \( \beta_1 \) is large and \( \beta_2 \) is small. On the other hand, when \( \beta_1 \) is small and \( \beta_2 \) is large, the self-evolved network can inhibit hidden-action attack, for example, jamming, effectively. Figure 4 just confirms our prediction. As we can see, the deepest blue area locates in the northeast corner of the axes and the lightest blue area locates in the southwest corner of the axes. We can also find that \( \beta_1 \) performs a bigger weight than \( \beta_2 \) on the equilibrium proportion of malicious nodes that take the jamming action. We also find that when \( \beta_1 \) is large enough, variation of \( \beta_2 \) has little effect on \( \rho_f \) (see the red-circled area in Figure 4). This result is in accordance with Figure 3, where the \( \beta_1 - \rho_f \) curve is steeper than the \( \beta_2 - \rho_f \) curve, indicating that change of \( \beta_1 \) is much more influential.

5.2.3. Stimulation and Punishment Effect. Next, we explore the impacts of detecting stimulation \( s \) and jamming punishment \( p \) on \( \rho_f \). The result is shown in Figure 5. It is indicated
that, for detecting stimulation, as $s$ increases, $\rho_j$ firstly increases to a local maximum and then decreases to a stable value. For jamming punishment, as $p$ increases, $\rho_j$ firstly decreases to a local minimum and then increases to about 50%. We argue that both of two curves are just as we expected. For detecting stimulation, when $s$ is small, it is not enough to encourage normal nodes to detect malicious jamming nodes until it reaches a threshold value, after which more normal nodes joined in for detection, and thus $\rho_j$ decreases. However, when $s$ is large, increase of detecting stimulation makes no sense. Because the number of malicious jamming nodes is small, the total detecting payoff is limited, which is unable to attract more normal nodes to join and perform detecting. For jamming punishment, when $p$ is small, the malicious nodes that take the jamming action are efficiently constrained. However, when $p$ is large, the punishment cost exceeds the income for punisher; thus the number of normal nodes that take the detecting action decreases and results in the increase of $\rho_j$.

Similarly, we also study the joint effect of $s$ and $p$ on the proportion of malicious nodes that take jamming action. The result is given in Figure 6. According to the result of Figure 5, a large $\rho_j$ will appear when $p$ is large and $s$ is around 3. However, as Figure 6 depicts, the maximum of $\rho_j$ has an apparent drift towards larger value of $s$ (see the blue-circled area). This can be explained as follows: when the jamming punishment $p$ is large, normal nodes are unwilling to perform detecting, because the punishment cost exceeds the income for punisher. Therefore, these normal nodes will wait until the detection stimulation increases to an acceptable threshold. The orbit of drift is shown as an arrowed curve in Figure 6. The black-circled area indicated the smallest $\rho_j$, which is also in accordance with Figure 5. Like that in Figure 4, here we argue that the jamming punishment $p$ performs a bigger weight than detecting stimulation $s$ on the equilibrium proportion of malicious nodes that take the jamming action.

6. Related Works

Applying game theory into improvement the QoS of WSN or wireless network is a new research direction, especially the games for the security in WSN or wireless network. There have been a few research efforts in this field. However, there are still many issues worthy of further exploration.

Kodialam and Lakshman in [8] model a zero-sum game between the intruder and service provider of the network. The objective of the intruder is to inject a malicious packet in the network at some node with node $t$ as the target. The intrusion is successful when the packet reaches the target and unsuccessful when it does not. To protect the nodes from the attack, the service provider is allowed to sample the packets flowing through the links on the network. The optimal solution for this game is the min.-max. optimal solution, which is the Nash equilibrium for a zero-sum game. Agah et al. in [9] study a game formulation at the routing layer of a wireless network between malicious nodes that do not forward incoming packets and an intrusion detector residing at the base station. They model this scenario as a repeated game, where the IDS uses the history of nodes' collaboration to determine paths comprising malicious nodes. The proposed protocol for the repeated game shows a correlation between network size and successful intrusion detection, where detection success rate increases with higher percentage of malicious nodes. Kamhoua et al. in [10] investigate a situation that it is cost effective to freely participate in the security mechanism or protect its privacy depending for each node on the fact that if that node believes or trusts that all other nodes or at least a minimum number of other nodes will do the same. They model a trustable dilemma for autonomous multihop networks by using the mathematical framework of game theory and evolutionary game theory. The well-known stag hunt game is used as their basic game model. They present the interconnection between cooperation, trust, privacy, and security in a network. However, they only
consider the game for normal nodes in the network rather than the game between normal nodes and malicious nodes.

Khouzani et al. in [11] develop a zero-sum dynamic game model and investigate the structural properties of the saddle-point strategies. They consider a game with two players (normal nodes and malicious nodes) and three different states with normal nodes. Their research efforts show that saddle-point strategies are simple threshold-based policies, and hence a robust dynamic defense is practicable. They assume there is no competition among normal nodes or malicious nodes in the network, which is not feasible to many situations in WSNs. Sagduyu et al. in [12] present a class of jamming games played at the MAC layer of wireless network among a set of transmitters and jammers. They address the incomplete information or uncertainty in games compared with existing works. The equilibrium strategies resulting from the jamming games characterize the expected performance under DoS attacks and motivate robust network protocol design to support secure wireless communications. Although they consider the conflicts among malicious nodes to be incomplete information, they have not taken the conflicts of normal nodes into consideration. Jaramillo and Srikant in [13] discuss the problem that packet collisions and interference may make cooperative nodes appear selfish sometimes, generating unnecessary and unwanted punishments in wireless ad hoc networks. They present a robust mechanism to imperfect measurements, which is collusion resistant and can achieve cooperation among nodes. However, their method does not satisfy the situation where there are malicious nodes in the network. Yan et al. in [14] proposed a penalizing mechanism to prevent the noncooperative selfish behavior of decreasing the contention window without permission based on repeated game theory for WSNs. A Contention Window Select Game is defined, and in this game each sensor node selects its own contention window to control the access probability.

Moreover, there are some research efforts about using other evolutionary algorithms rather than game theory to solve the security problem in WSNs or wireless networks. For example, Alrajeh et al. in [15] present an adaptive secure routing protocol based on bioinspired technique termed as ant colonization for WSNs. Their approach is able to select optimal paths from source to destination by ensuring adaptability, robustness, and security. However, their approach cannot suppress hidden-action attacks in WSNs. Hortos in [16] presents a cross-layer approach to WSN protocol design by applying a bioinspired evolutionary computational method to the functions of each protocol layer to improve the intrusion detection identification (IDID) performance. Genetic algorithms and ant colony optimization are used to solve the problem in the proposed approach.

Although many existing approaches have taken game theory to solve the security problem in WSNs or wireless networks, compared with existing works in this field, our approach considers the relationship between nodes in a WSN and gives a formal theoretic model for the game. We discuss different patterns in this game and represent them as rules to generate a more complete game model. We argue that such a representation model is relatively lacking in existing research efforts. Moreover, the algorithms proposed in this paper are not very complex and easy to be implemented in resource-constrained WSNs, compared to the existing approaches. In general, our approach is able to support secure wireless communication in WSNs with malicious nodes.

7. Conclusion

In this paper, we mainly present an approach to apply game theory into solving the network security problem of WSN. We try to explore game theory algorithms to model the situation for WSN with malicious nodes, in order to support reliable and secure wireless communications against the attacks of malicious nodes in the network. We consider the relationship between the nodes in a WSN to formulate the game and illustrate the game theory algorithms in detail. In order to verify the proposed approach, we also evaluate it with a simulation experiment and analyze the simulation results in detail. We argue that the approach is able to support secure end-to-end communication in WSNs with malicious nodes.

Future works may include (1) improving the efficiency of the algorithms to support different kinds of attacks rather than jamming, (2) evaluating the cost of the game theory algorithm for WSN, and (3) considering a more complex WSN model to evaluate the approach.

Acknowledgments

This work is partially supported by Grants from the NSFC Program (no. NSFC61003309), the Science and Technology Department of Zhejiang Province Program (no. 2010C13005, 2011C14024 and 2011C13006-1), the Key Science and Technology Innovation Team of Zhejiang Province Program (no. 2010R50041-03), the ZJNSF Program (no. LR12F02002). This work is also partially supported by the National Development and Reform Commission, China under Special Grants “The Operation System of Multimedia Cloud Based on the Integration of Telecommunications Networks, Cable TV Networks and the Internet”.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–105, 2002.
[2] T. Moore, “Countering hidden-action attacks on networked systems,” in Proceedings of the 4th Workshop on the Economics of Information Security, Cambridge, Mass, USA, June 2005.
[3] D. Fudenberg and J. Tirole, Game Theory, MIT Press, 1991.
[4] Y. Zhang and W. Lee, “Intrusion detection in wireless ad-hoc networks,” in Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM ’00), pp. 275–283, August 2000.
[5] K. Ioannis, T. Dimitriou, and F. C. Freiling, “Towards intrusion detection in wireless sensor networks,” in Proceedings of the 13th European Wireless Conference, 2007.
[6] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” Science, vol. 286, no. 5439, pp. 509–512, 1999.
[7] G. Szabó and C. Tóke, “Evolutionary prisoners dilemma game on a square lattice,” Physical Review E, vol. 58, no. 1, pp. 69–73, 1998.

[8] M. Kodialam and T. V. Lakshman, “Detecting network intrusions via sampling: a game theoretic approach,” in Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM ’03), pp. 1880–1889, April 2003.

[9] A. Agah, K. Basu, and S. K. Das, “Prevention of DoS attack in sensor networks using repeated game theory,” in Proceedings of the International Conference on Wireless Networks (ICWN ’05), pp. 29–36, 2006.

[10] C. A. Kamhoua, N. Pissinou, and K. Makk, “Game theoretic modeling and evolution of trust in autonomous multi-hop networks: application to network security and privacy,” in Proceedings of the IEEE International Conference on Communications (ICC ’11), pp. 1–6, June 2011.

[11] M. H. R. Khouzani, S. Sarkar, and E. Altman, “A dynamic game solution to malware attack,” in Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM ’11), pp. 2138–2146, April 2011.

[12] Y. E. Sagduyu, R. A. Berry, and A. Ephremides, “Jamming games in wireless networks with incomplete information,” IEEE Communications Magazine, vol. 49, no. 8, pp. 112–118, 2011.

[13] J. J. Jaramillo and R. Srikant, “A game theory based reputation mechanism to incentivize cooperation in wireless ad hoc networks,” Ad Hoc Networks, vol. 8, no. 4, pp. 416–429, 2010.

[14] M. Yan, L. Xiao, L. Du, and L. Huang, “On selfish behavior in wireless sensor networks: a game theoretic case study,” in Proceedings of the 3rd International Conference on Measuring Technology and Mechatronics Automation (ICMTMA ’11), pp. 752–756, Shanghai, China, January 2011.

[15] N. A. Alrajeh, M. S. Alabed, and M. S. Elwahiby, “Secure ant-based routing protocol for wireless sensor network,” International Journal of Distributed Sensor Networks, vol. 2013, Article ID 326295, 9 pages, 2013.

[16] W. S. Hortos, “Bio-inspired, cross-layer protocol design for intrusion detection and identification in wireless sensor networks,” in Proceedings of the 37th Conference on Local Computer Networks Workshops (LCN ’12), pp. 1030–1037, 2012.