JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 32, 559-573 (2016)

A Game Theory Approach for Malicious Node Detection in MANETs

YASER TAHERI[1], HOSSEIN GHARAEE GARAKANI[2] AND NASER MOHAMMADZADEH[1]
[1]*Department of Computer Engineering*
Shahed University
Tehran, 3319118651 Iran
[2]*Department of Network Security and Information Technology Research Center*
Tehran, 1439955471 Iran
E-mail: {y.taheri; mohammadzadeh}@shahed.ac.ir; gharaee@itrc.ac.ir

One of the main features of nodes in mobile ad hoc networks (MANETs) is their cooperation with neighbors to propagate data. Misusing this feature, malicious nodes cooperate with normal nodes to disrupt network operation and reduce its efficiency. These nodes attack other network nodes and prevent being detected by other nodes through using the mobility characteristic of nodes in MANETs. One of the effective ways to detect malicious nodes is using game theory. Focusing on this issue, in this paper, an approach is presented for malicious node detection using Games Theory. Data exchange between two nodes forms a game, and the history of the successful and unsuccessful sending/receiving of data, is stored to be used for future decisions. Malicious nodes are detected by the game components, sent and received data as well as information that have been stored in different stages. The experimental results indicate that if the density of malicious nodes approximately reaches 20%, and the game is repeated more than four times between neighboring nodes, the proposed approach raises the malicious node detection rate to 91%. However, if the density of malicious nodes is more than 30%, the algorithm can detect more than 87% of them after being repeated six times.

Keywords: network security, MANETs, malicious node, intrusion detection, game theory

1. INTRODUCTION

Mobile ad hoc network is a collection of self-organizing wireless nodes communicating with each other [1]. Nodes in the network do everything such as routing. This means that nodes are responsible for the routing, sending data, receiving data, network security controlling, recognizing malicious nodes, and other factors discussed in the network security issues [2]. This issue implies the important role of securing each node. Nodes of mobile ad hoc networks are classified into three categories "selfish," "malicious," and "normal." The *normal nodes* perform their routine activities in the network without any destruction. The *selfish nodes* themselves are divided into three types [3]; Type 1 includes the nodes that contribute in finding and maintaining the best route between source and destination, but they do not involve in sending packets to the next hop. Type 2 contains the nodes that do not participate in routing or sending the packet to the next hop. The nodes that act like nodes Type 1 or 2 depending on their energies are of Type 3. The *malicious nodes* include nodes that have destructive goals in the network. They try to maximize disturbing the network performance and efficiency, to deceive the

Received May 18, 2015; revised October 1, 2015; accepted November 19, 2015.
Communicated by Xiaohong Jiang.

normal nodes, to ignore, not sent, or eliminate the packets received from neighbors, to conduct received packets into an improper route, and to waste the energy of other nodes. A malicious node adopts the following actions: (1) It cooperates with normal nodes to gain their trust; (2) It attacks to degrade the network performance or achieve other destructive goals; (3) It attempts to escape before being detected; (4) It enters the network as a new comer. Under some circumstances, a malicious node may sense that other nodes' cooperation with it is less than before. It may happen because other nodes have guessed the malicious node's nature. Therefore, it is not beneficial that the malicious node stays in the network and may decide to change its ID and tries to enter the network as a new comer.

Identifying of malicious nodes plays an important role in enhancing network security and performance. Thus, this paper provides an algorithm to detect these nodes. Several methods have been proposed to detect malicious nodes. These methods include monitoring the behavior of neighbors [4], weighing a node's behavior and measuring confidence based on the weight [5], using the sequential hypothesis testing to assess nodes' reaction [6], and using the activity records gathered by neighbors in different times [7]. Games Theory is another powerful tool used by many researchers to detect malicious nodes or encourage collaboration with other participants. In reference [8], Gharaee *et al.* have proposed a strategy based on the Games Theory to provide a way in order to encourage malicious nodes to cooperate with others in heterogeneous wireless sensor networks. In references [9, 10], a method based on Bayesian game theory is suggested to model the interaction of malicious and normal nodes in order to identify malicious nodes and to understand how they act against being detected.

The other method proposed in this field [11] is based on Bayesian game theory that tries to conserve energy. Bayesian game refers to a game, with incomplete information and different types of players. In this game, each player's action depends on the opponents' previous action. The selected action is based on using a possibility value calculated at each step.

In this paper, the interaction between two nodes, which one of them is normal and another one type is unknown, is modeled as a game. Due to certain conditions of Bayesian game, and it's compatibility with the presented algorithm we used it in this paper. The sending and receiving results and neighbor's responses to inquiries are saved in nodes' recording table and used in the next decision-making. These interactions are continued to be stored at appropriate times to be used to identify malicious nodes.

The rest of this paper is organized as follows. The related research is discussed in Section 2. Section 3 includes the most important attacks carried out by malicious nodes. The basic concepts of the game theory are mentioned in Section 4. Section 5 contains the proposed approach. Experimental results are given in Section 6. Finally, Section 7 concludes and summarizes the paper.

2. RELATED WORK

As mentioned above, malicious nodes in the network have destructive purposes. Therefore, identifying and reporting these nodes can enhance network performance and reduce network downtime. Cooperation between nodes in the mobile ad hoc networks

and motivating them to cooperate with other nodes were completely analyzed and reviewed in [12]. Malicious nodes are also introduced, discussed, and studied in this paper. In reference [14], a method based on the game theory was introduced to detect malicious nodes in wireless sensor networks. In this method, it is assumed that the number of malicious nodes is much less than the number of normal nodes and each node is not aware of the other nodes' type. The purpose of this paper was to detect abnormal behavior of malicious nodes.

Reference [15] presented TPP-Game algorithm that aims at modeling, analyzing the cooperation and trustable behavior between the nodes. The proposed algorithm in reference [15] is based on a trust value, which is collected through direct and indirect evidence of the activities of the other nodes in the network and is used to measure trustworthy of network members. Because the trust value is calculated using collected data of all network nodes, it has a high accuracy, as the advantage of this parameter. Reference [16] used a non-zero sum and non-cooperative game [13]. Since this game uses the probability parameters to select strategies, it is a Bayesian game. Reference [17] provided a way to avoid dropping messages on P2P networks. The game presented in this paper is an iterative and two-player game between two neighboring nodes in the P2P network. In fact, the data exchanging between neighboring network nodes is modeled as a game. As a result, the game is iterative and will be presented at all stages of the network. In reference [18], some methods based on game theory were proposed to maintain the security of mobile ad hoc network. These algorithms are designed for networks whose members adopt restrictions on the use of its resources. These algorithms cannot be used in networks whose members have a high degree of mobility without any restrictions.

All aforementioned methods and algorithms are important and cannot be ignored; however, each of them has weaknesses in some circumstances that must be improved. To provide an efficient algorithm to detect malicious nodes in mobile ad hoc networks, the strengths of them can be beneficial. Focusing on this issue, we present a new game theory approach for malicious node detection in MANETs in this paper.

3. MALICIOUS ATTACKS IN MANET

One of the factors that may violate security of ad hoc networks is a malicious node regarded as a member of the network. Attacks in mobile ad hoc networks caused by malicious nodes are summarized in Table 1.

4. GAME THEORY

Game theory is one of the most powerful tools for modeling transactions between nodes and predicting strategies. It uses mathematical and computational support for the nodes to decide at each stage of the game. In this theory, players use a series of agreements between themselves to run the game using the rules adopted [13].

4.1 Definition

A game consists of a set of players, a set of moves or strategies, and results given

Table 1. Various attacks in MANET area done by malicious nodes.

Attacks in MANET		
Attack name	Active/passive	Attack layer
Modification	active	Multi-layer
Fabrication	active	Multi-layer
Impersonation (Spoofing)	active	Physical-layer
Black hole	active	Network-layer
Gray hole	active	Network-layer
Wormhole (Tunneling)	active	Network-layer
Rushing	active	Multi-layer
Denial of service	active	Multi-layer
Sybil	active	Network-layer
Reply	passive	Multi-layer
Location Disclosures	active	Network-layer

for each combination of strategies [19]. If we are to present a precise mathematical definition of game theory, G-game can be defined as follows.

$$G = <N, A, \{u_i\}>,$$

where N denotes the number of players, A is a set of action profile caused by the multiplying Cartesian product operations

$$A = A_1 * A_2 * A_3 * \ldots * A_n,$$

and u is a set of utility functions $\{u_i\} = \{u_1 \ldots u_n\}$ such that the resulting benefit for node i after selecting the action a_i is shown by u_i [19]. Also $u_i(x, y)$ represents the operating profit of the player i doing the action x while his opponent is doing the action y.

4.2 Classification of Game Theories

Games have many aspects. Hence, they can be categorized according to different aspects. The classifications of games are as follows.

- **Static or dynamic game:** In a dynamic game, the nodes are aware of their own or others' previous moves and benefit from them at different stages of the game, while in the static game, players make decisions simultaneously without any knowledge about strategies chosen by other players.
- **Number of repetitions:** A game may be done once, or it may be repeated several times. Each repetition can be done with the same or different players.
- **Complete or incomplete information game:** A record of the opponent and current player's own moves during a game may be available. This type of game is called "complete information game." If all information is not available, it is referred to as "incomplete information game."
- **Cooperative or non-cooperative nature of the game:** players may choose a strategy during the game with the agreement. If agreement between the players is applicable and practical, the game is called "cooperative" and if not, the game is referred to as "non-cooperative."

4.3 Nash Equilibrium

In game theory, Nash equilibrium [19] is a solution from the game theory, which gives the best gain to all players. No player can change his strategy to earn more gain while the other's gains are not increased. In such circumstances, the Nash equilibrium is composed of a set of strategies that players choose and their related payoffs [6].

4.4 Bayesian Game

Bayesian game is a game with incomplete information and includes the following components:

- A finite set of players, $i \in \{1, 2, \ldots, I\}$
- A finite set of actions for each player, $a_i \in A_i$
- A finite set of player types $\theta_i \in \Phi_i$
- The probability distribution for each type of the players
- Payoff function for players is:

$$u_i = A_1 * A_2 * \ldots * A_i * \Phi_1 * \Phi_2 * \ldots * \Phi_i \rightarrow R$$

To put it more simply, Bayesian game is a game played with incomplete information in which players' type is different from each other. Each player can choose an action according to the opponent's previous behavior and the conditions under which they are acting. Selected action would be placed using a possible value for each stage of the calculation. The Nash equilibrium for Bayesian game is the same as that for common games. The difference is that players need to consider their beliefs about the type of opponent with respect to the acquisition of the previous games. This game is used in the proposed algorithm due to some characteristics such as: different types of nodes as a player (normal and malicious), the set of strategies for each player selected in each specific situation and incomplete information about the players as well as the type of the opponent node.

5. DETECTION ALGORITHM

5.1 Specifications of the Proposed Game

- It uses a dynamic game model. A dynamic game is the one in which the nodes are aware of their own or other's previous moves and benefit from them at different stages of the game. Using the obtained profits, the can update its information on how its opponents perform.
- It uses a Bayesian game model. Because of uncertainty about the neighboring nodes and their types, there is always the possibility that repeated and emphatic decision fails. Therefore, it is better to have a probability parameter to recognize the best move made at any stage and to have the best possible use of your own previous behavior and other node's behavior. If you play once, the obtained benefits cannot be used and there will be no training parameters. Each repetition of game leads to having more experienced

nodes at any stage and best using of injuries, decision benefits, and previous information.

- **Number of players:** The game is considered between two nodes of the network. One of these nodes could be a malicious node and the second node is assumed to be a normal node.
- **Non-cooperative:** The goal is recognizing, punishing, or expelling malicious nodes. The nodes are also not aware of the existence and location of malicious nodes. Therefore, the game is non-cooperative.
- **Incomplete Information:** Obviously, the information is incomplete because nodes are not cognizant of the destructive nature of a node.

The set of strategies designed for the game are $\{A, C, D, R\}$, where A represents the attack, C represents the cooperation, R represents reporting a node as a malicious node, and D is indicative of indifference. Each player (node) should select one of the strategies at each stage of the game. After the adoption of one strategy, each player gains its payoff whose amount is dependent on the opponent's strategy and reaction. Malicious nodes have the choice to select the strategy C to deceive other players, attack, or prevent being recognized. To track the behavior of their neighbors for correct detection of malicious nodes, nodes monitor the behavior and performance of their neighbors and store these important messages, including successful and unsuccessful data transfer as parameters to be used later.

5.2 Malicious Node Detection Algorithm

In this section, the game payoff table between two nodes labeled 1 and 2 is formed, and the Nash equilibrium for each node is calculated. This table is shown in Fig. 1. It is assumed that node 1 is a normal node, and node 2 can be a normal or a malicious node. Due to lack of information about nodes at the beginning of the network, it is better to put the same initial values for parameter node type. Therefore, the probability for a node to be malicious or non-malicious is equal for all nodes at the beginning of the game.

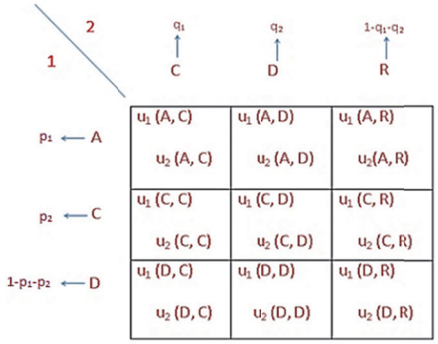

Fig. 1. Payoff functions for player 1 and 2.

In this section, we calculate the expected payoff for node in the proposed algorithm. As the table in Fig. 1 indicates, Eq. (1) is used for player 1's expected payoff.

$$\pi_1(A, C, D) = p_1q_1u_1(A, C) + p_1q_1u_1(A, D) + p_1(1 - q_1 - q_2)u_1(A, R) + p_2q_1u_1(C, C) + p_2q_2u_1(C, D) + p_2(1 - q_1 - q_2)u_1(C, R) + (1 - p_1 - p_2)q_1u_1(D, C) + (1 - p_1 - p_2)q_2u_1(D, D) + (1 - p_1 - p_2)(1 - q_1 - q_2)u_1(D, R) \quad (1)$$

The expectation that player 1 can choose any of the operations is calculated as follows:

$$E_1(A, q) = q_1u_1(A, C) + q_2u_1(A, D) + (1 - q_1 - q_2)u_1(A, R), \quad (2)$$
$$E_1(C, q) = q_1u_1(C, C) + q_2u_1(A, D) + (1 - q_1 - q_2)u_1(C, R), \quad (3)$$
$$E_1(D, q) = q_1u_1(D, C) + q_2u_1(D, D) + (1 - q_1 - q_2)u_1(D, R). \quad (4)$$

Using the topic of the best response, intersection points of the graphs of these formulas are equal to a compound of the game Nash equilibrium. In other words, the compound Nash equilibrium is achieved putting each of the Eqs. (2)-(4) as equivalent pairs. The probability value for each action of player 1 is obtained by calculating a compound Nash equilibrium. Obtaining these values at each stage of the game, player 1 can perform the best reaction.

Similarly, we have the following for the player 2:

$$\pi_2(C, D, R) = q_1p_1u_2(A, C) + q_1p_2u_2(C, C) + q_1(1 - p_1 - p_2)u_2(D, C) + q_2p_1u_2(A, D) + q_2p_2u_2(C, D) + q_2(1 - p_1 - p_2)u_2(D, D) + (1 - q_1 - q_2)p_1u_1(A, R) + (1 - q_1 - q_2)p_2u_2(C, R) + (1 - q_1 - q_2)(1 - p_1 - p_2)u_2(D, R). \quad (5)$$

The expectation that player 2 can choose any of the operations is calculated as follows:

$$E_2(C, p) = p_1u_1(A, C) + p_2u_2(C, C) + (1 - p_1 - p_2)u_2(D, C), \quad (6)$$
$$E_2(D, p) = p_1u_2(A, D) + p_2u_2(C, D) + (1 - p_1 - p_2)u_2(D, D), \quad (7)$$
$$E_2(R, p) = p_1u_2(A, R) + p_2u_2(C, R) + (1 - p_1 - p_2)u_2(D, R). \quad (8)$$

In this section, using the topic of best response, intersection points of these three formulas is equal to a compound of the game Nash equilibrium for node 2. Placing equivalent each pair of Eqs. (6)-(8), combination Nash equilibrium is achieved. The probability value for each action of player 2 is obtained by calculating a compound Nash equilibrium. Obtaining these values at each stage of the game, player 2 can perform the best reaction.

In the algorithm, sending a packet between two nodes is considered as a two-player game. This game is iteratively played between all nodes of the network. Malicious nodes are specified by using data collected from all nodes of the network. The detection procedure is as follows:

- All neighboring nodes in the network play many times with one another.
- Each node stores successful and unsuccessful data transfers with neighboring nodes as well as those of the nodes located on the data destination path and the result of his sending.
- At each stage of data sending, the nodes update their information related to their neighbors.

- Profitability values change with respect to time and the previous game information. This allows the players to play more cautiously. In this manner, malicious nodes cannot predict the strategy of normal nodes.
- At each step, the node that is going to send the packet to the next hop starts playing game with another node (neighbor node). At this time, the node checks and investigates its saved list. The node updates his profitability rates considering the number of the opponent's successful and unsuccessful data transfers, acquires probability values for each action, and selects its own action accordingly.
- At the specified times, a set of neighboring nodes or nodes that trust each other compare their saved lists, detect, and report malicious nodes on the path that have led to the greatest number of failures by sharing information.

Our approach uses more strategies in game for all of the normal and malicious nodes in comparison to the previous ones. Moreover, our method stores data in a concise and compact manner to be used in different stages and cooperation of neighboring nodes.

5.3 Time Complexity

Suppose a MANET with n nodes which each node could have at most k neighbors and assume each node plays g games with its neighbors during the network lifetime. There-Therefore, the time complexity to detect nature of all neighbors of a node is

(Number of node's neighbors) ∗ (Number of games played with neighbors) ∗ (Time complexity of one game)

The time complexity of a single game consists of finding the best probabilities for choosing strategies and then selecting one of them. The Eqs. (2)-(4), (6)-(8) are used in this paper to find these probability parameters. There are two unknown parameters in these three equations. Using Gauss-Jordan elimination, time complexity of finding these two parameters choosing one strategy from three strategies is $O(1)$. Therefore, the time complexity of a single game is $O(1)$. The time complexity of finding the nature of all neighbors of a node is

$$O(1) * O(g) * O(k) = O(kg).$$

The time complexity of the whole algorithm consists of exchanging reports between nodes in the MANET and detecting nature of all neighbors of the nodes which is equal with

(Time complexity of detecting nature of all neighbors of a node ∗ Number of nodes) + (Number of exchanged reports in the network ∗ Repetition of exchanging reports)

If all nodes in the MANET send reports to their neighbors, $n * k$ reports will be exchanged. If t is the number of repetitions, the time complexity of the whole algorithm is

$O(kg) + O(tnk)$.

It is worthwhile to note that the calculated time complexity is too smaller than $O(n^3)$ and $O(n^2)$ because t, k and g are constant and small coefficients.

5.4 Space Complexity

We assume that network contains n nodes, and each node has at most k neighbors. Each node needs to accumulate its neighbor's information. In this condition, a number or a string is presumed as storage unit. Each node should keep the following items for each neighbor:

1. Identifier
2. Number of interactions with it
3. Number of successful interactions with it
4. Number of unsuccessful interactions with it
5. Number of tolerable penalties assigned to it
6. Number of remaining tolerable penalties related to it
7. A field which determines maliciousness of a neighbor from beginning up to now.
8. A field that, determines nature of a neighbor

According to above items, if the number of common neighbors of each node with neighbor node is k', required space for each node will be $kk' + 7k$. Therefore, the total required space for all nodes is $n(kk' + 7k)$. For example, if a network contains 100 nodes, each node has 10 neighbors, each neighbor has 4 common neighbors with considered node, and required space for a field is 1 byte, then required space for each node will be equal to 110 bytes. Therefore, the total required space for all nodes will be 1100 bytes.

6. EXPERIMENTAL RESULT

6.1 Experimental Setup

Java was used to implement the proposed algorithm. The algorithm was run on a Dual core 2.53 GHz processor with 4 GB of RAM.

6.2 Dataset

To have close results to real world, datasets were used with characteristics look like to real MANET characteristics such as the number of neighbors that a node could have, percentage of malicious members of MANET, MANET scope and diversity of the nodes in the MANET. MANET members, their information about neighbors, and the other needed information were produced in a uniform random way. List of used datasets are collected in Table 2. The scope for MANET was considered between 10000 and 2250000 units. The number of members was changed from 100 to 1500 nodes. The malicious per-

centage of nodes was varied from 10% to 40%. Each member node has 5 to 16 neighbors. Datasets are available in http://cld.persiangig.com/download/XRRYIT/dl.

Table 2. Used datasets features.

	# of Nodes	# of Malicious nodes	Malicious nodes percentage	Maximum links	Network area
100_10.txt	100	10	10	5	100 * 100
100_20.txt	100	20	20	5	100 * 100
100_30.txt	100	30	30	5	100 * 100
100_40.txt	100	40	40	5	100 * 100
500_10.txt	500	50	10	10	500 * 500
500_20.txt	500	100	20	10	500 * 500
500_30.txt	500	150	30	10	500 * 500
500_40.txt	500	200	40	10	500 * 500
1000_10.txt	1000	100	10	14	1000 * 1000
1000_20.txt	1000	200	20	15	1000 * 1000
1000_30.txt	1000	300	30	16	1000 * 1000
1000_40.txt	1000	400	40	13	1000 * 1000
1500_10.txt	1500	150	10	11	1500 * 1500
1500_20.txt	1500	300	20	13	1500 * 1500
1500_30.txt	1500	450	30	15	1500 * 1500
1500_40.txt	1500	600	40	13	1500 * 1500

6.3 Efficiency Analysis of the Proposed Algorithm

To evaluate the algorithm it was compared to algorithms proposed in [20, 22]. To evaluate the detection rate of malicious nodes and the misdetection rate of normal nodes, the algorithms were run in different conditions. The experimental results are summarized in Figs. 2-9.

Figs. 2, 4, 6, and 8 show the detection rate of malicious nodes when MANET has 500 members, the percentage of the malicious nodes varies from 10% to 40%, each node plays from 10 to 40 times with its neighbors, and each node has at most 13 neighbors. The results show that the proposed algorithm has better efficiency in detecting of malicious nodes than two other algorithms. As the results indicate the algorithms proposed in this paper and in [20] that utilize the game theory have a better result than the algorithm proposed in [22] which doesn't use it. The figures show that when the percentage of malicious nodes increases the detection rate decreases. However, the slope of these diagrams for the proposed algorithm is less than the other ones. On the other hand, by increasing the number of plays (interactions) between a node and its neighbors, the detection rate increases. The proposed algorithm increases the detection rate more than the other two algorithms as shown in Fig. 8. When the number of plays increases, the effective information about neighbors is exchanged between nodes, therefore, each node has more complete and more robust information than previous step to decide whether a neighbor is malicious or not. This fact is the reason of increasing the detection rate when the number of plays increases. However, Figs. 2, 4, 6, and 8 refer to reference [22] shows that by increasing number of interaction between nodes, detection rate does not have effective change because, this algorithm does not utilize previous interactions information effectively.

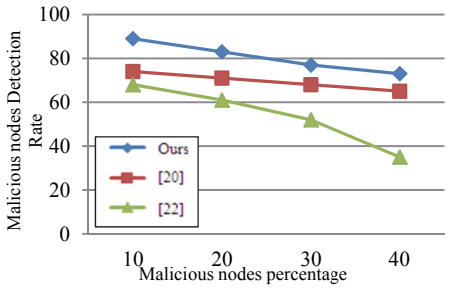

Fig. 2. Detection rate for 500 nodes. Each node plays 10 times with each neighbor.

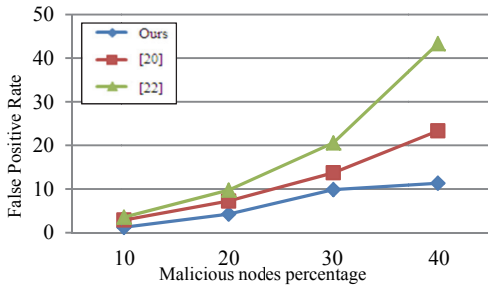

Fig. 3. False positive rate for 500 nodes. Each node plays 10 times with each neighbor.

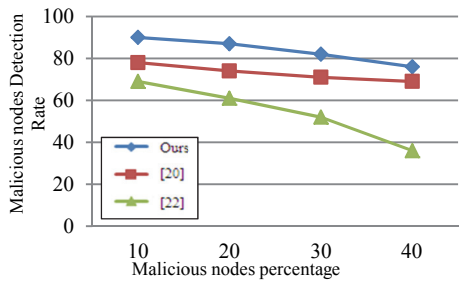

Fig. 4. Detection rate for 500 nodes. Each node plays 20 times with each neighbor.

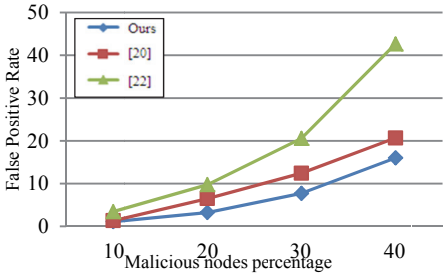

Fig. 5. False positive rate for 500 nodes. Each node plays 20 times with each neighbor.

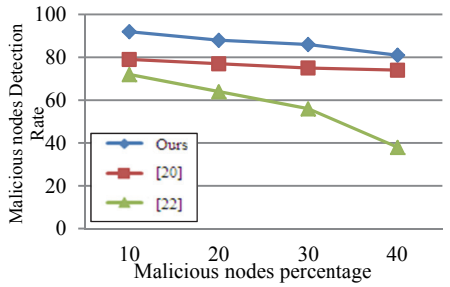

Fig. 6. Detection rate for 500 nodes. Each node plays 30 times with each neighbor.

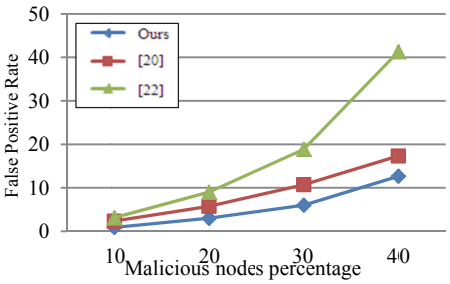

Fig. 7. False positive rate for 500 nodes. Each node plays 30 times with each neighbor.

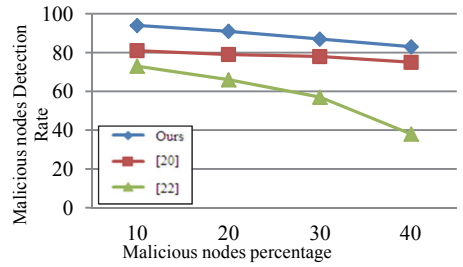

Fig. 8. Detection rate for 500 nodes. Each node plays 40 times with each neighbor.

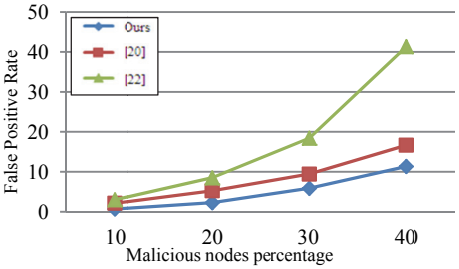

Fig. 9. False positive rate for 500 nodes. Each node plays 40 times with each neighbor.

6.4 Time Complexity Analysis

Fig. 11 compares the execution time of the proposed algorithm, algorithm [20], and algorithm [22]. In this condition, the number of plays (interactions) for each node is 30 and the average number of links for each node is 14. The figure shows that the required time for algorithm [22] is better than two others. By increasing the number of nodes, the number of plays (interactions) for each node increase and consequently, the execution time increases. If the number of nodes is more than 500, the execution time is more than the condition which the numbers of nodes are less than 500. This is because the number of links and nodes increase. The algorithms proposed in this paper and in [20] use game theory. Therefore, the proximity of their runtimes is reasonable.

Fig. 12 compares the execution time of the proposed algorithm, algorithm [20], and algorithm [22]. In these comparisons, number of nodes are 1000 and 30 percent of nodes are malicious. Number of plays (interactions) for each node varied from 10 to 40 and maximum link for each node is 14. As we expected, by increasing number of plays (interactions) execution time also increases.

False positive rate or ratio is the ratio of normal nodes, which are falsely detected as malicious nodes. The false positive rate for the proposed algorithm and algorithms suggested in [20] and [22] in various execution situations is presented in Figs. 3, 5, 7 and 9. The false positive rate for algorithms is calculated using the following equation.

$$FPR = \frac{FP}{FP + TN}$$

Where FP is the number of normal nodes that miss detected as malicious nodes and TN is the number of normal nodes which correctly detected as normal nodes. As the figures show, it is obvious that the false positive rate of the proposed algorithm is lower than the others. Therefore, the proposed algorithm has the lower detection fault. As the percentage of malicious nodes in MANET increases, the false-positive rate increases but as can be seen in the figures it for the proposed algorithm is lower than it for the other algorithms.

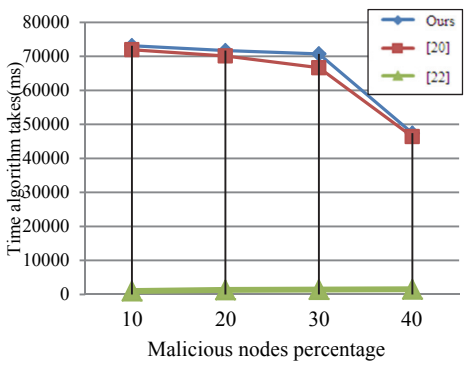

Fig. 10. Algorithm average consumed time diagram. 1000 nodes, nodes average link is 14 and each node plays 30 times.

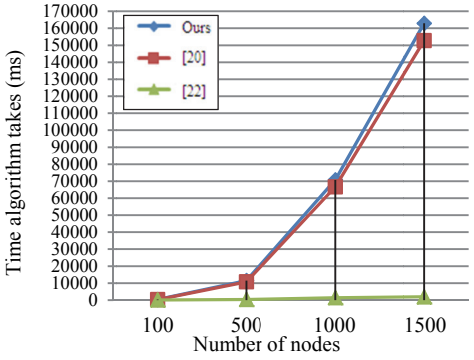

Fig. 11. Algorithm average consumed time diagram. 30% malicious nodes, nodes average link is 14 and each node plays 30 times.

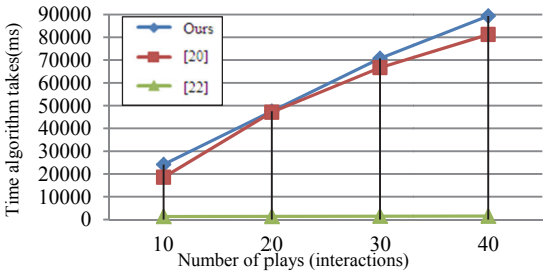

Fig. 12. Algorithm average consumed time diagram. 1000 nodes, 30% malicious nodes and nodes average link is 14.

7. SUMMARY AND CONCLUSION

Mobile ad hoc networks are always faced with many security threats because of mobility of nodes and lack of a central coordinator. In this paper, we proposed an algorithm based on the game theory that models the interactions between nodes in the mobile ad hoc networks. Sending and receiving interactions were considered as a two-player game. In this algorithm, all nodes playing in the game save some information about the interactions with neighbor nodes in a table. The data used in the next sending interaction helps the node to interact with his neighbors through a rational approach and do not send and receive packages blindly. In particular specified times, this information is gathered and used to detect malicious nodes. All experiments were done by using various datasets. Experiments on these datasets were used for evaluation of performance, fault, and execution time of the proposed algorithm. We compared our results with the results obtained by [20] and [22]. The results show that our algorithm behaves better in detecting of malicious nodes and its misdiagnosis rate is less than two other algorithms. We can increase the quality of detection by increasing algorithm cost. In addition, we see that in the runtime of the proposed algorithm is a bit more than two other algorithms but its high efficiency in the detection of malicious nodes compensates this flaw. The experimental results showed that if the density of malicious nodes approximately reaches 20% of the network, and the game is repeated more than four times between neighbor nodes, the detection rate of our method is over 91%. However, if the density of malicious nodes increases to more than 30%, the algorithm can detect more than 87% of them after being repeated for 40 times. In this case, the false-positive rate average is about 4%. With some modifications, the game proposed in this paper can be generalized to more than two players. In our future work, we will explore it.

REFERENCES

1. W. Liu, H. Nishiyama, N. Ansari, J. Yang, and N. Kato, "Cluster-based certificate revocation with vindication capability for mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, 2013, pp. 239-249.
2. Y. Wang *et al.*, "A mean field game theoretic approach for security enhancements," *IEEE Transactions on Wireless Communications*, Vol. 13, 2014, pp. 1616-1627.

3. K. Balakrishnan, J. Deng, and P. K. Varshney, "TWOACK: preventing selfishness in mobile ad hoc networks," in *Proceedings of Wireless Communications and Networking Conference*, 2005, pp. 2137-2142.
4. S. Tomasin, "Consensus-based detection of malicious nodes in cooperative wireless networks," *IEEE Communications and Letters*, Vol. 15, 2011, pp. 404-406.
5. O. Seo, H. Chan, O. Hong, and Y. Hwa Choi, "A malicious and malfunctioning node detection scheme for wireless sensor networks," *Wireless Sensor Network*, Vol. 4, 2012, pp. 84-90.
6. H. Jun-Won, M. Wright, and S. Das, "Distributed detection of mobile malicious node attacks in wireless sensor networks," *Ad Hoc Networks*, Vol. 10, 2012, pp. 512-523.
7. Y. Sung-Jib and Y. Choi, "Neighbor-based malicious node detection in wireless sensor networks," *Wireless Sensor Network*, Vol. 4, 2012, pp. 219-225.
8. M. Shamani, *et al.*, "Adaptive energy aware cooperation strategy in heterogeneous multi-domain sensor networks," *Procedia Computer Science*, Vol. 19, 2013, pp. 1047-1052.
9. W. Wang *et al.*, "A game theoretic approach to detect and co-exist with malicious nodes in wireless networks," *Computer Networks*, Vol. 71, 2014, pp. 63-83.
10. L. Feng, Y. Yang, and J. Wu, "Attack and flee: game-theory-based analysis on interactions among nodes in MANETs Systems," *IEEE Transactions on Systems*, *Man*, *and Cybernetics*, *Part B: Cybernetics*, Vol. 40, 2010, pp. 612-622.
11. J. Roles, H. Elaarag, and E. Friedman, "A Bayesian game approach to coexistence with malicious and selfish nodes in wireless ad-hoc networks," in *Proceedings of the 17th Communications and Networking Simulation*, 2014, pp. 50-57.
12. M. Felegyhazi, P. Hubaux, and L. Buttyan, "Nash equilibria of packet forwarding strategies in wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, Vol. 5, 2006, pp. 463-476.
13. D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, Cambridge, MA, 1991.
14. Y. Mao, P. Zhu, and G. Wei, "A game theoretic model for wireless sensor networks with hidden-action attacks," *International Journal of Distributed Sensor Networks*, Vol. 2013, 2013, pp. 1-9.
15. X. Jin *et al.*, "Modeling cooperative, selfish and malicious behaviors for trajectory privacy preservation using Bayesian game theory," in *Proceeding of the 38th Conference on Local Computer Networks*, 2013, pp. 835-842.
16. M. K. Rafsanjani, L. Aliahmadipour, and M. Javidi, "A hybrid intrusion detection by game theory approaches in MANET," *Indian Journal of Science and Technology*, Vol. 5, pp. 2123-2131.
17. J. Chen *et al.*, "Game theory analysis for message dropping attacks prevention strategy in mobile P2P live streaming system," in *Proceedings of IEEE 3rd International Conference on Software Engineering and Service Science*, 2012, pp. 359-363.
18. T. Alpcan and S. Buchegger, "Security games for vehicular networks," *IEEE Transactions on Mobile Computing*, Vol. 11, 2011, pp. 280-290.
19. R. J. Aumann and S. Hart, *Handbook of Game Theory with Economic Applications*, Elsevier, 1994.
20. B. Paramasiva and K. Pitchai, "Modeling intrusion detection in mobile ad hoc networks as a non-cooperative game," in *Proceedings of International Conference on*

Pattern Recognition, *Informatics and Mobile Engineering*, 2013, pp. 300-306.
21. S. Shamshirband, *et al.*, "Cooperative game theoretic approach using fuzzy Q-learning for detecting and preventing intrusions in wireless sensor networks," *Engineering Applications of Artificial Intelligence*, Vol. 32, 2014, pp. 228-241.
22. Y. Singh and S. Jena, "Intrusion detection system for detecting malicious nodes in mobile ad hoc networks," *Advances in Parallel Distributed Computing*, Vol. 203, 2011, pp. 410-419.

Yaser Taheri received the M.S. degree in Information Technology from Shahed University, Iran. His research interests include communication networks.

Hossein Gharaee Garakani received the B.S. degree in Electrical Engineering from Khaje Nasir Toosi University, M.S. and Ph.D. degree in Electrical Engineering from Tarbiat Modares University, Iran. Since 2009, he has been with ITRC. His research interests include general area of VLSI with emphasis on basic logic circuits for low-voltage low-power applications, DSP algorithm, crypto chip and instrusion detection and prevention systems.

Naser Mohammadzadeh received the B.Sc. and M.Sc. degrees in Computer Engineering from Sharif University of Technology, Iran. He received the Ph.D. degree in Computer Engineering from Amirkabir University of Technology, Iran. His research interests include optimization and quantum design automation.