IJCAT - International Journal of Computing and Technology, Volume 3, Issue 8, August 2016
ISSN: 2348 - 6090
www.IJCAT.org

A Secure Algorithm against Selective Forwarding Attack in Wireless Sensor Networks

[1] Sara Kamari, [2] Mojtaba Jamshidi

[1] Department of Computer Engineering, Kermanshah Branch, Islamic Azad University
Kermanshah, Iran

[2] Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University
Qazvin, Iran

Abstract - Selective forwarding attack is a dangerous attack in wireless sensor networks, in which a malicious node attempts to drop some incoming packets. This means that it refuses to forward them to destination. Selective forwarding attack has a highly destructive impact on multi-hop routing protocols in wireless sensor networks and decreases the packets delivery ratio to sinks. This paper proposes a secure routing protocol against selective forwarding attack in wireless sensor networks. The main idea of the proposed algorithm is to establish a cell structure within the network so that W≥1 node plays the monitoring role in each cell that monitors activities of the remaining nodes in the cell. By overhearing traffic within the cell, the observer nodes find out whether the nodes are maliciously drop packets. If the observer nodes recognize that a node is malicious and does not forward the incoming packet to the destination, they will alter the path of the packets. The proposed algorithm was implemented and its performance was compared with four existing algorithms in terms of packet delivery ratio to the sink. The results of the comparison show that the proposed algorithm has better performance.

Keywords **- Wireless Sensor Networks, Overhear Mechanism, Cellular, Observer Nodes, Selective Forwarding Attack.**

1. Introduction

Today, wireless sensor networks are used in many fields such as environment, military operation and exploration. Since sensor nodes have low computing, memory and radio capabilities and with regard to the use of the networks in critical environments, especially military contexts, providing security in such networks are of high importance and have been considered by many researchers [1, 2].

Selective forwarding attack is one of the common attacks in the wireless sensor networks that could disrupt many monitoring and surveillance missions. In this attack, malicious nodes often behave as legal nodes but sometimes drop reporting packets and therefore it is hard to detect such an attack. The attack greatly disrupts multi-hop routing algorithms in wireless sensor networks and decreases packet delivery ratio to the sinks. Selective forwarding attack usually will have the greatest impact when attacker node is directly on the path of data flow. The malicious node can be easily detected if it drops all incoming packets but if the malicious node tries selective forwarding; that is, it does not drop all incoming packets, it will be difficult and challenging to be detected [3,4].

So far, many mechanisms have been proposed to defend against selective forwarding attack. The most common mechanisms include multi-hop acknowledgement-based algorithms [5, 6] or multipath-based methods algorithms [2, 10]. In general, disadvantages of such algorithms are as follows: high delay in packet delivery to the sinks, high computational overhead, security problems and lack of scalability. This paper proposes a cell structure-based routing algorithm using observer nodes for securing the network against selective forwarding attacks so that it removes disadvantages of the previous algorithms [7, 8]. The paper is organized in the following sections. Section 2 and section the paper is organized in the following sections: Section 2 and section 3 respectively present previous works and observer nodes. Section 3 deals with hypotheses of system and attack model, section 4 introduces the proposed algorithm and section 5 shows performance evaluation and simulation results. The last section concludes the paper.

2. Related Work

Selective forwarding attack was first raised in [2] and the use of multi-path routing protocols was stated as the first strategy to defend against this attack. In this method, the packets are routed from source to destination through fully separate *n* paths. Disadvantages of the strategy include low security, lack of detection of malicious node, increased energy consumption and communication overhead. Another protocol was provided in [3], in which a multi-hop acknowledgement algorithm is used according to the responses received from intermediate nodes in order to disseminate alarm messages across the network. Another technique was presented in [4] in order to detect malicious nodes against selective forwarding attack. In fact, it is the improved version of the technique proposed in [3]. A centralized intrusion detection algorithm was demonstrated in [9] that is based on support vector machines and sliding window technique in order to defend against black hole and selective forwarding attacks. Another protocol, based on multi-data flow topologies (MDT), was proposed to defend against selective forwarding attack in [1].

The main idea of MDT is to divide sensor nodes into several groups or completely separate data flows so that data sensed by the source nodes are forwarded to the base station via the separate data flows. A reliable fuzzy-based data delivery algorithm was provided in [11] that is actually the improved form of multipath routing technique reported by [2]. Another lightweight algorithm was presented in [12] where only information provided by neighbors is used to detect the selective forwarding attack. Moreover, [13] presented an algorithm to defend against selective forwarding attack that used heterogeneous sensor network model to detect the selective forwarding attack. The algorithm acts only in cluster-based sensor networks. Moreover, [14] proposed another way to defend against selective forwarding attack, in which a multinomial-based approach is used. The main idea is that the sensed data is divided into several pieces and the pieces are forwarded to the base station via separate routes. A traffic monitoring-based algorithm was shown to detect selective forwarding attack in [15].

In this approach, eavesdropper and monitor (EM) nodes were used to overhear and monitor all network traffic. In addition, [16] introduced a sequential mesh test-based algorithm to detect selective forwarding attack in sensor networks. The algorithm is naturally centralized and is used for cluster-based networks. Furthermore, [17] presented a defensive lightweight algorithm that uses neighboring nodes as monitoring nodes.

The effects of the selective forwarding attack on data flows and ACKs flows were first simulated at the network level in [18]. Then, an ack-based multipath algorithm was studied to defend against the selective forwarding attack. Moreover, a fully distributed, dynamic, lightweight and learning automata-based algorithm was proposed by [19] in order to defend against selective forwarding attack in sensor networks. The algorithm uses overhear mechanism with the learning automata model to select secure path for forwarding packets in multi-hop routing protocols. Each node is equipped with learning automata whose mission is to choose the next node (upstream node) to forward data to the base station and monitor its performance.

Furthermore, a secure routing algorithm based on monitoring and reputation mechanism was proposed in [20]. In this algorithm, the amount of reputation is set based on the forwarding rate of the packets and residual energy of the node. Such detection and routing mechanism is common because it considers both security and network lifetime.

3. Observer Nodes

In a sensor network, nodes can be homogeneous or heterogeneous. Homogeneity or heterogeneity of the network nodes can be expressed in terms of hardware or software. In the case of hardware homogeneity, all nodes are identical in terms of hardware resources (such as memory, processing power, battery capacity, etc.). In the software homogeneity, all nodes are identically programmed and play the same role in the network. However, in hardware or software heterogeneity, nodes will be different from one another. For example, in the case of software heterogeneity, some nodes in the network play specific roles in the network that include intrusion detection, detection of a specific attack, leading a group, and so forth. The nodes usually make up a small part of a network and the remaining nodes should perform the network mission.

So far, many algorithms such as [21-25] have been proposed to defend against various attacks in wireless sensor networks, which rely on specific nodes under the "observer" node or other categories. The nodes can be different from other nodes in terms of software and play roles to detect a specific attack in addition to their usual duties in performing the network mission. For example, [21] used four detector nodes in the network to estimate the nodes location and detect Sybil node. In addition, [22] employed the monitoring node mechanism to remove the misbehaving nodes from the routing in individual mobile networks. Furthermore, [23] used beacon or watchdog

nodes to provide a reputation evaluation system. Moreover, [24] used four monitoring nodes to monitor network traffic and detect sinkhole attack. Finally, [25] employed a series of beacon nodes to detect the wormhole attack.

This paper used the same type of specific nodes called monitoring nodes to provide a new and efficient algorithm to defend against selective forwarding attack in wireless sensor networks.

4. Assumptions of the Systems and Attack Model

Sensor network is divided into two general categories of source nodes (SN) and forwarding nodes (FN). The source nodes actually generate reporting packets. Packets generated by the source nodes are delivered hop-by-hop to the sink node through forwarding nodes. For example, sensor nodes can be expanded in border areas or enemy environments and if they observe enemy forces activities, they will provide the necessary reports and deliver the reports to the base station or sink with a few hops.

The nodes are assumed to have unique identifier (ID). The required operational environment is a two-dimensional environment where the source nodes and intermediate nodes respectively are specifically and randomly distributed. All nodes have a constant radio range that is equal to *r*. Moreover, it is assumed that the nodes can communicate with each other through wireless radio channels and use omni-directional broadcast. Communication links are bi-directional; that is, if node *u* can get a message from node *v*, it can forward a message to *v*. In addition, it is assumed that nodes are aware of their location and remain fixed after development in the operating environment (nodes are not mobile). Sensor nodes and communication links are not safe; the enemy can capture some nodes, reprogram them as malicious nodes, and deploy them in the network.

It is also assumed that our network model uses key establishment protocol LEAP [17] to establish cluster keys between the nodes in the network. A cluster key is shared by a node and all its neighbors, and is mainly used to secure local broadcast messages, such as routing control information. This type of key is used to perform overhear operation. In other words, this type of key makes overhear operation possible. In LEAP algorithm, each node *u* shares a unique cluster key with all its neighbors in order to secure its messages. Its neighbors also use the same key to decrypt and authenticate messages of the node *u*.

The attack model considered in this paper is derived from the attack model in [5, 6]. A malicious node in the attack model can refuse forwarding the incoming packets to the base station and easily drop the incoming packet. Here, it is assumed that enemy can capture and reprogram normal nodes within the network and inject them as malicious nodes into the network or inject external malicious nodes into the network. It is also assumed that malicious nodes do not cooperate with each other and have no limit on the rate of dropping packets.

5. The Proposed Algorithm

The main idea of the proposed algorithm is to create a cellular structure in the network and use the overhear mechanism to monitor traffic and performance of the nodes with the help of monitoring nodes in order to defend against selective forwarding attack. In short, $W \geq 1$ node plays the monitoring role in each cell that monitors activities of the remaining nodes in the cell. When monitoring nodes observe a malicious behavior (dropping the packets) of the forwarding nodes, they alter the path of the packets; that is, they forward the packet to destination through another neighbor. The proposed algorithm consists of three phases that will be described below.

5.1. Determination of the Level of Nodes

Now, the level of nodes is briefly determined on the routing tree. In other words, the distance between each node to sink is determined in terms of hop. Each sensor node has a neighborhood table in its memory as shown in Figure 1. Identifiers (IDs) of the neighboring nodes are stored in the column *NID* and the number of neighboring nodes in the routing tree. In other words, its distance to the sink by hop is stored in *Hop-Count* column.

After the deployment of nodes in the network environment, the sink node as the root of the routing tree will generate a packet of "route-generator" with content << *NodeID* = 0, *HopCount* = 0 >>, encrypt it with its cluster key K_{k_0} and broadcast it. All sensor nodes of v_i that are in the radio range of the sink will receive this packet as the nodes of Level 1 in the routing tree, decrypt it with its cluster key K_{k_0} and update its neighborhood table. This means that it adds one row in the neighborhood table. The sink node identifier (the identifier was assumed to be 0) is stored in the field *NID* of the added row, and distance between the node forwarding the route-generator packet to the sink is stored in the field *HopCount*.

Then, each of the neighboring nodes of the base station; that is v_is, generates a new route-generator packet with content <<*NodeID* =v_i, *HopCount* = 1>>, encrypts it with the cluster key and broadcasts it to the neighbors. The process continues until the route-generator packet is delivered to all accessible nodes in the network. After the end of the phase, each node in the neighborhood table will have the identifier of all its neighbors with their distance to the sink node in terms of hop.

NID	*Hop-Count*

Figure 1: structure of neighborhood table

5.2. Configuration Phase

Partitioning, selection of the monitoring nodes and the next hop node are performed to forward packets to the sink.

The sensor nodes calculate their location via GPS or positioning algorithms. Then, each node determines that in what cell it has been deployed and stores its ID in its memory. Assuming that the network environment is *L*L* and the size of each side of each cell is *Grid_Size*, and then the network environment is partitioned as Figure 2.

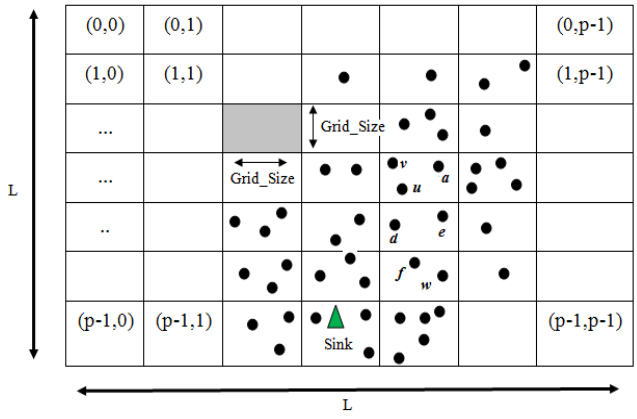

Figure 2: partitioning of the network environment

Figure 2 shows that the network environment has been transformed in the form of a cellular structure that contains *p*p* cells. The size of each cell is *Grid_Size*Grid_Size*. Each cell is determined by two components (i, j). Each node *u* calculates its cell ID from equation (1).

$$i = \left\lfloor \frac{x}{GridSize} \right\rfloor \qquad (1)$$

$$j = \left\lfloor \frac{y}{GridSize} \right\rfloor$$

Where, x and y are the coordinates of the node *u*.

Then, at most $W \geq 1$ node will be selected as the monitoring node in each cell. A variety of mechanisms can be taken to select the monitoring nodes. One simple mechanism is that the monitoring nodes are randomly selected in each cell. Another mechanism is that those nodes are selected as the monitoring nodes in each cell that have more neighbors. This mechanism will be more effective because it can monitor and overhear more other nodes. To implement the mechanism, each node broadcasts a number of its neighbors in the cell. Hence, each node becomes aware of each of the number of neighbors of its neighboring nodes. Therefore, *W* nodes that have the highest number of neighbors are selected as the monitoring nodes.

In this phase, each node *u* selects the next one of his neighbors who has a shorter distance to the sink as the next-hop node. Henceforth, the node *u* will forward its packets or incoming packets through this next-hop node to the sink.

5.2. Data Forwarding Phase

After ending the first two phases of the proposed algorithm, the source nodes generate particular reporting packets when they view the events in question and forward them to the next hop. Figure 3 shows the form of the reporting packets.

SrcID	**PacketID**	**SenderID**	**NextHopID**	**Payload**

Figure 3: structure of reporting packets

Source node identifier generating the reporting packet is stored in the field *SrcID* and packet identifier is stored in the field *PacketID*. Moreover, the node identifier forwarding packet is stored in the field *SenderID* and the node identifier of the next hop to which the packet is forwarded, is stored in the field *NextHopID*. Finally, data is stored in the Payload.

Assuming that the source node SN_k is going to generate a reporting packet after viewing an event and forward it to the sink, the source node SN_k will generate a packet with

the following content and forward it to the next hop node (e.g., node *u*):

<<*SrcID=SN*$_k$, *PacketID=eID, SenderID= SN*$_k$, *NextHopID=u, Payload=Data*>>

Here, *eID* is a unique *ID* that *SN*$_k$ source node assign to each of its reporting packets. Moreover, the forwarding node *u* should deliver this packet to the next hop (e.g., node *w*). In this case, the node *u* modifies and forwards the reporting packet as follows:

<<*SrcID=SN*$_k$, *PacketID=eID, SenderID= u, NextHopID=w, Payload=Data*>>

The process continues until the above-mentioned reporting packet is delivered to the sink node. However, if the enemy injects malicious node in order to establish a selective forwarding attack, the reaction of the proposed algorithm will be as follows:

Consider Figure 2 and assume that node *u* is a malicious node and node *a* is going to forward a packet to it. In fact, node *a* selects node *u* as its own next hops. Moreover, assume that node *v* is selected as the monitoring node. When node *a* encrypts its packet with a cluster key and forwards it to *u*, the monitoring node *v* will overhear the packet if it is located at the radio range of the node *u* and keeps it in its buffer for time *t*. If the node *u* attempts to forward the packet during the time *t*, the monitoring node will delete the packet from its buffer but the malicious node *u* refrain from forwarding the packet, the monitoring node will forward the packet to the next hop (a node other than *u*) and forwards a message to the node *a* in order to change its next-hop node. In this case, if the node *a* has another candidate for the next hop, it will choose it. For example, it chooses the node *e* as the next hop in Figure 2, and henceforth will forward its packets via the new next hop node to the sink. Therefore, if a monitoring node is in the neighborhood of both sender and receiver nodes of a packet, the monitoring node will notice malicious behavior of the receiving node and take necessary actions. Thus, if the number of monitoring nodes increases, resistance to selective forwarding attack will be increased.

5.3. The Proposed Algorithm Overhead

Memory overhead: the memory overhead that is imposed by the proposed algorithm to the sensor nodes concerns only to the neighbor table. Assuming that the average number of neighbors of each node is *d*, the memory overhead equals to 2*d* and would be of the order of O (*d*).

Communication overhead: unlike algorithms of [2, 10], the packets in the proposed algorithm are forwarded to the sink through single-path and if the packet is dropped by a malicious node and such a drop is observed by the monitoring node, the packet is forwarded to the sink through more than one path. Therefore, the communication overhead of the proposed algorithm is less than two [2, 10] and thus it consumes less energy. Of course, little communication overhead imposes on the monitoring nodes for overhear operation in the proposed algorithm. However, it is very obvious that the overhear operation has a very little overhead and consumes very small energy compared to the sending of a packet. In the first phase of the proposed algorithm, each node also sends a route-generator packet. In the second phase, each node broadcasts a message (containing the number of its neighbors) to its neighbors in order to select the monitoring node.

Processing overhead: overhead processing of the proposed algorithm returns to the second phase of the algorithm where each node should arrange the number of the neighbors of its own neighbors in a descending order list and examines whether it is on the list *W* of the monitoring node. Assuming that *d* is the average number of neighbors, processing overhead of the process is $o(d \times \log d)$ (using the merge sort algorithm).

6. Simulation Results

In order to examine the performance of the proposed algorithm against selective forwarding attack, we first implemented it in C++, evaluated its performance by doing some experiments, and compared the results with a few other algorithms.

Our simulation model is derived from [5] and [19] that is as follows: *n* sensor nodes are randomly distributed in a 100 × 100 square meters environment. Of these nodes, one node is a sink, which is constantly on coordinates (50, 30), and 20 other nodes are considered as the source nodes. *M* is also the number of malicious nodes that are randomly selected. The number of the monitoring nodes in each cell is *W*. The sizes of the cells are *Grid_Size* = 20 meters. All nodes have a constant radio range that is equal to 10 meters. In all experiments, the probability of packet drop by a malicious node is $Drop\ Probability = 0.5$. The source nodes generate and forward one reporting packet every five seconds. Each of the experiments would run for 10,000 seconds and the result of each experiment was obtained from the average of 100 different Runnings. Evaluation criterion is the packet delivery ratio to the sink; that is, percentage of the

packets generated by the source nodes that arrive at the sink.

The simulation results of the proposed algorithm were compared with four other algorithms in Table 1 in order to show the proposed algorithm performance. Moreover, the amounts of security parameters of each of the four algorithms have been set in the simulations, as described in Table 1.

Table 1. List of existing algorithms that compared to the proposed algorithm

Algorithm	Security parameters
Single Path Forwarding	-
Multi-Path based [2]	uses at most 5 distinct paths for leading each packet toward the base station
MDT [10]	uses two distinct data topology
Ack-based [5]	$ACK_{span} = 2$, $ACK_{TTL} = 4$, $t = 1$

First experiment: this experiment evaluated the effect of the number of malicious nodes in the network on the performance of the proposed algorithm. The results were compared with other algorithms. We changed total number of the nodes in the network n = 300 and the number of malicious nodes in the network from 0, 20, 40, 100 and evaluated the the results for W = 3. Figure 4 shows the results of the experiment.

As reported by the experiment results, the proposed algorithm performance in the packet delivery ratio to destination is higher than other algorithms. For example, when there are 50 malicious nodes in the network, the packet delivery ratio is 82% in the proposed algorithm while the amount is less than 70% for other algorithms. In addition, the experiment shows that when the number of malicious nodes in the network increases, the packet delivery ratio to the sink decreases because more packets are dropped in the network and the probability that some nodes (especially sink) are surrounded by a few malicious nodes increases.

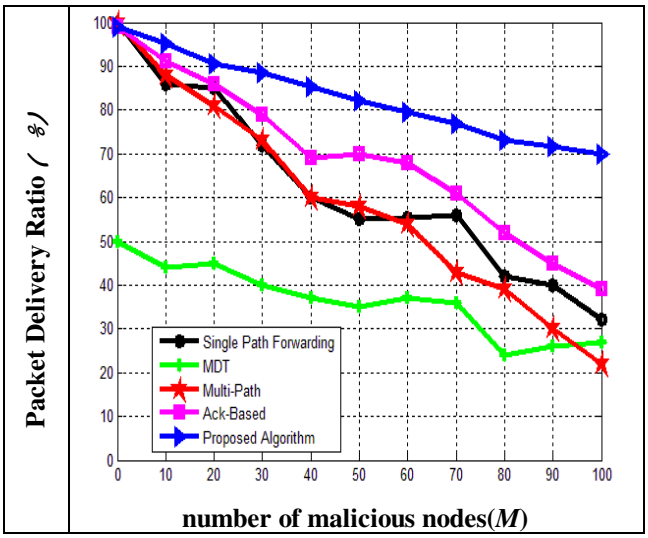

Figure 4: Comparison of proposed algorithm and the other algorithms in term of the packets delivery ratio

Second experiment: this experiment determined the effect of the number of total nodes in network n on the performance of the proposed algorithm. We changed the parameter W = 3 and the number of malicious nodes in the network from 0 to 100, and evaluated the experiment results for n = 300, 400, and 500. Figure 5 shows the results of the experiment. The results demonstrate that the increased density of the network increases the packet delivery ratio because the probability that a node is surrounded by a few malicious nodes or all nodes of the next hop are malevolent would be reduced. Hence, the packet delivery ratio to the sink is increased.

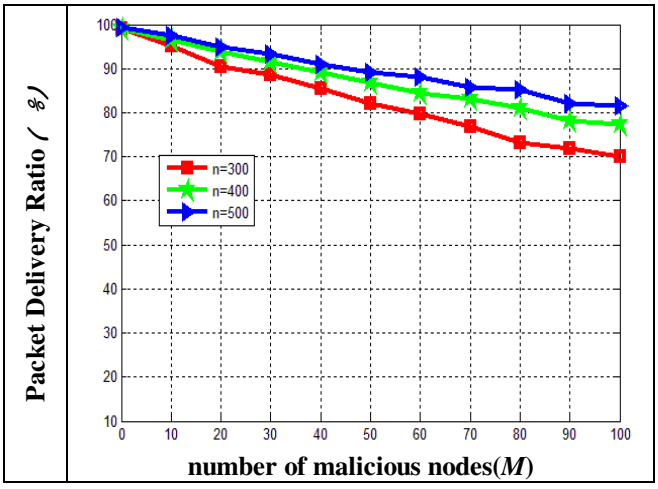

Figure 5: effect of the number of total nodes on packet delivery ratio of proposed algorithms

7. Conclusion

This paper proposed a new routing algorithm to be resistant against selective forwarding attack. The proposed algorithm uses a cell structure with overhear mechanism through the monitoring nodes mechanism to remove malicious nodes from data routes to the sink. The performance of the proposed algorithm was evaluated in terms of memory, communication and processing overhead. Its performance was also evaluated by simulation in terms of packet delivery ratio to the sink. The results were compared with four other algorithms that indicate the superiority of the proposed algorithm.

References

[1] Akyildiz Ian F. and Kasimoglu Ismail H.,"Wireless sensor and actor networks: research challenges", in: Proceedings of the Ad Hoc Networks 2, pp. 351–367, 2004.

[2] Karlof C. And Wagner D., "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", in: Proceedings of the Ad Hoc Networks, pp. 299-302, 2003.

[3] Sharma K. and et al., "A Comparative Study of Various Security Approaches Used in Wireless Sensor Networks", in: Proceedings of the International Journal of Advanced Science and Technology, Vol. 17, April, 2010.

[4] Mohammadi S., Atani R. E. and Jadidoleslamy H., "A Comparison of Link Layer Attacks on Wireless Sensor Networks", in: Proceedings of the Journal of Information Security, pp. 69-84, April 2011.

[5] Yu B. and Xiao B., "Detecting selective forwarding attacks in wireless sensor networks", In: Proceedings of the Second International Workshop on Security in Systems and Networks (IPDPS Workshop), pp. 1-8, April 2006.

[6] Xiao B., Yu B. and Gao C., "CHEMAS: identify suspect nodes in selective forwarding attacks", In: Proceedings Of the Journal of Parallel and Distributed Computing, Vol. 67, No. 11, pp. 1218-1230., June 2007.

[7] Bysani L. K. and Turuk A. K., "A Survey On Selective Forwarding Attack in Wireless Sensor Networks", In: Proceedings of the International Conference on Device and Communications (ICDeCom), Mesra, india, February 2011.

[8] Kkan W. Z., Xiang Y. and Aalsalem M. Y., "Comprehensive Study of Selective Forwarding Attack in Wireless Sensor Networks", In: Proceedings of the I.J. Computer Network and Information Security, p. 1-10, February 2011.

[9] Kaplantzis S., Shilton A, Mani N. and Sekercioglu Y., "Detecting selective forwarding attacks in wireless sensor networks using support vector machines", in: Proceedings of the IEEE 3rd International Conference Intelligent Sensors, Sensor Networks and Information(ISSNIP), pp. 335 –340, December 2007.

[10] Sun H.-M., Chen C.-M. and Hsiao Y.-C., "An efficient countermeasure to the selective forwarding attack in wireless sensor networks", in: Proceedings of the IEEE TENCON 2007, pp. 1-4, October 2007.

[11] Lee H. Y. and Cho T. H., "Fuzzy-based reliable data delivery for countering selective forwarding in sensor networks", in: Proceedings of the Ubiquitous Intelligence and Computing, pp. 535-544 ,Hong Kong, China, Springer-Verlag, pp. 535-544, 2007.

[12] Hai T. H. and Huh E.-N., "Detecting Selective Forwarding Attacks in Wireless Sensor Networks Using Two-hops Neighbor Knowledge", In: Proceedings of the Seventh IEEE International Symposium on Network Computing and Applications, pp.325-331, 2008.

[13] Brown J.and Du X., "Detection of selective forwarding attacks in heterogeneous sensor networks", in: Proceedings of the IEEE International Conferece on Communications, pp. 1583-1587, May 2008.

[14] Lei X., Yong-jun H., Yong P. and Yue-Fei Z., "A Polynomialbased Countermeasure to Selective Forwarding Attacks in Sensor Networks", In: Proceedings of the International Conference on Communications and Mobile Computing, pp.455- 459, 2009.

[15] Tumrongwittayapak C. and Varakulsiripunth R., "Detecting Sinkhole Attack And Selective Forwarding Attack In Wireless Sensor Networks", in: Proceedings of the 7th International Conference on Information, Communications and Signal Processing (ICICS 2009), pp. 1-5, December 2009.

[16] Li G., Liu X. and Wang C., "A Sequential Mesh Test based Selective Forwarding Attack Detection Scheme in Wireless Sensor Networks", in: Proceedings of the International Conference on Networking, Sensing and Control (ICNSC), pp. 554-558, April 2010.

[17] Xin-sheng W., Yong-zhao Z., Shu-ming X. and Liangmin W., "Lightweight defense scheme against Selective forwarding attacks in wireless sensor networks", in: Proceedings of the IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC '09), pp. 226-232, October 2009.

[18] Zhang Y., and Minier M.," Selective Forwarding Attacks against Data and ACK Flows in Network Coding and Countermeasures", Journal of Computer Networks and Communications Volume 2012 (2012).

[19] Jamshidi, M., Esnaashari, M. and Meybodi, M. R., "Utilizing Learning Automata for Defending Against Selective Forwarding in Wireless Sensor Networks", The CSI Journal on Computer Science & Engineering, Vol. 11, No. 3, pp. 31-39, 2013.

[20] Hu Y., Wu Y. and Wang H.," Detection of Insider Selective Forwarding Attack Based on Monitor Node and Trust Mechanism in WSN", Wireless Sensor Network,Vol. 6,pp. 237-248, 2014.

[21] S. Zhong, L. Li, Y. G. Liu, and Y. R. Yang. "Privacy-preserving locationbased services for mobile users in Wireless Networks", in: Proceedings of the Technical Report YALEU/DCS/TR-1297, Yale Computer Science, July 2004.

[22] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", in: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking, pp. 255-265, 2000.

[23] Srinivasan A. and et al., "DRBTS: Distributed Reputation-based Beacon Trust System", Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06), 2006.

[24] Tumrongwittayapak C. and Varakulsiripunth R., "Detecting Sinkhole Attacks In Wireless Sensor Networks", in: Proceedings of the International Joint Conference ICROS-SICE, August 18-21, Fukuoka International Congress Center, Japan, 2009.

[25] Ronghui H. and et al., "Detecting and Locating Wormhole Attacks in Wireless Sensor Networks Using Beacon Nodes", World Academy of Science, Engineering and Technology (55), 2009.

[26] Zhu S., Setia S. and Jajodia S., "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks", n: Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03), pp.62-72, 2003.

Author Profile:

Sara Kamari received the both B.S. and M.S. degrees in Computer Engineering from the Islamic Azad University, Kermanshah, Iran, in 2012 and 2015, respectively. Her research interests include wireless Sensor Networks and Security.

Mojtaba Jamshidi received the B.S. degree in Computer Engineering from the Academic Center of Education, Kermanshah, Iran, in 2009, and M.S. degree in Computer Engineering from the Islamic Azad University, Qazvin, Iran, in 2012. His research interests include Computer networks, learning systems, Security, Data Mining, and Recommender Systems.