VOL 3 (2019) NO 4

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Advanced Extremely Efficient Detection of Replica Nodes in Mobile Wireless Sensor Networks

Mehdi Safari[#], Elham Bahmani[*], Mojtaba Jamshidi[**], Abdusalam Abdulla Shaltooki[***]

[#] *Department of Computer Engineering, Sanandaj Branch, Islamic Azad University, Sanandaj, Iran*
[*] *Department of Computer Engineering, Malayer Branch, Islamic Azad University, Malayer, Iran*
[**] *Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
[***] *Department of Information Technology, University of Human Development, Sulaymaniyah, Iraq*

E-mail: safari.92@chmail.ir, bahmani.elham66@gmail.com, jamshidi.mojtaba@gmail.com, salam.abdulla@uhd.edu.iq

***Abstract*— Today, wireless sensor networks (WSNs) are widely used in many applications including the environment, military, and explorations. One of the most dangerous attacks against these networks is node replication. In this attack, the adversary captures a legal node of the network, generates several copies of the node (called, replica nodes) and injects them in the network. Various algorithms have been proposed to handle replica nodes in stationary and mobile WSNs. One of the most well-known algorithms to handle this attack in mobile WSNs is eXtremely Efficient Detection (XED). The main idea of XED is to generate and exchange random numbers among neighboring nodes. The XED has some drawbacks including high communication and memory overheads and low speed in the detection of replica nodes. In this paper, an algorithm is presented to improve XED. The proposed algorithm is called Advanced XED (AXED) in which each node observes a few numbers of nodes and whenever two nodes meet, a new random number is generated and exchanged. The efficiency of the proposed algorithm is evaluated in terms of the memory and communication overheads and its results are compared with existing algorithms. The comparison results show that the proposed algorithm imposes lower overheads to the nodes. In addition, the proposed algorithm is simulated and the simulation results show that the proposed algorithm is able to detect replica nodes faster than XED.**

***Keywords*— Wireless Sensor Network, Mobile Nodes, Security, Replication Attack, XED Algorithm.**

I. INTRODUCTION

A WSN consists of many tiny sensor nodes which cooperate with each other to monitor an area and have a wide variety of applications including battlefield monitoring, industrial applications, health applications and etc. Sensor nodes have many limitations including memory capacity, computational power, radio power, and energy. Regarding these limitations and the wireless nature of sensors, it is very important to provide them with a secure system, especially in military applications. This challenging field has recently attracted the attention of many researchers [1, 2].

Already, many attacks such as Sybil, Selective Forwarding, Sinkhole, Wormhole, and Node Replication attacks have been introduced in WSNs. Also, many algorithms have been proposed to defend against these attacks in both stationary and mobile WSNs. The algorithms proposed to defend against an attack in stationary WSNs usually cannot be employed in mobile WSNs because of the continuous movement of nodes in the network environment. Therefore, separate algorithms are designed to deal with them [3, 4].

In this paper, we have focused on the node replication attack which is one of the famous and serious attacks in WSNs. Because of the unattended deployment of sensor nodes in the network, an adversary can capture one (or more) legal node and extract its ID and key materials, then generates some replica nodes from the captured node. The replica nodes have the same ID and key materials as the captured node. Thus, they can establish a shared key with other legal nodes of the network. Then, the adversary injects the replica nodes on the network. Since the credentials of replica nodes are all the clones from the captured nodes, the replica nodes can be considered as legitimate members of the network, which make detection difficult. From the security point of view, the node replication attack is considered harmful to the networks, because having legitimate keys, the replica nodes controlled by the

adversary can easily launch the insider attacks, without easily being detected [5, 6].

Many algorithms such as [6-9] have been proposed to identify the replica nodes in stationary WSNs. However, these algorithms cannot be employed in mobile WSNs, since most of them rely on transmitting location claims to witness nodes or specific places in the network, or they are particularly designed for some specific topologies.

XED algorithm [10] has been presented to combat the replica node attack in mobile WSNs. The main idea of this algorithm is to generate and exchange random numbers among nodes. In XED, each node Sj requires a history table of size 3n to store random numbers as shown in Fig. 1. Node Sj stores the identity of meeting nodes in field NodeID. Also, it keeps the random numbers sent to and received from other nodes in fields SentRnd and RecivedRnd, respectively. In this algorithm, according to Fig. 2, whenever, for example at time t1, a node like SA appears in the neighborhood of another node like Sj, if Sj has not sent any random number to SA, it generates a random number (r) and sends it to Sj. Both Sj and SA store this random number in their history. At subsequent times, t2, Sj requests to return the random number, if a node with ID SA appears in the neighbourhood of it. SA is valid if it is the node met at t1 by Sj. But if this SA is one of the replica nodes, since it does not know the random number r, it is detected as a replica node by Sj. Each Sj, repeats this process for each of its neighbors in each round of movement in the environment.

Disadvantages of XED algorithm are as follows:

- High memory overhead (each node requires a memory of size 3n).
- High communication overhead (in each round of nodes' movement, each node should acquire the random numbers from its neighbors. In addition, it should respond to the transmitted messages for returning the random number).
- Security failure (if replica nodes cooperate and share their random, the algorithm fails).

TABLE 1.
THE *HISTORY* TABLE OF NODES IN XED ALGORITHM

NodeID	SentRnd	ReceivedRnd

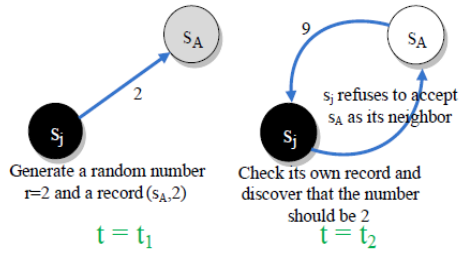

Fig. 1. Replica detection in XED at time t1 and t2. The gray and white nodes are different replicas and the black node is the legal node [10].

In this paper, an algorithm called Advanced XED (AXED) is presented to improve XED and resolve its shortcomings. In the proposed algorithm, each node has to observe a few numbers of nodes and whenever two nodes meet, a new random number is generated and exchanged. In designing the proposed algorithm, constraints of the sensor nodes in terms of memory, energy, and processing power are considered and it is tried to reduce the overhead of the algorithm and increase its security.

The rest of this paper is organized as follows. Section II presents related work, system assumption, and the proposed algorithm. Section III presents the simulation results. The paper is concluded in Section IV.

II. MATERIAL AND METHOD

In this section, we first present some existing algorithms to defend against Replication attack in WSNs. Then, we present the preliminaries of the proposed algorithm, including system assumptions and the attack model. Finally, the proposed algorithm is presented.

A. *Related Work*

Jamshidi et al. [6] proposed an algorithm based on a dynamic ID–assignment mechanism to defend against replica node attack in stationary WSNs. This algorithm uses a multi-tree architecture based on a multi-sink architecture for dynamic assignment of IDs to sensor nodes after deployment in a network. If this mechanism is used, replica nodes generated by the adversary cannot be easily attached to the network.

Randomized, Efficient, and Distributed (RED) [7] is a well-known and centralized algorithm to defend against replication attack in stationary WSNs. This algorithm transmits location claims (with digital signatures) to locations of the network that are selected based on a random number periodically issued by a central point.

Ding et al. [8] used similarity estimation with group deployment knowledge to detect cloned nodes in WSNs. They prevent replicas from generating false location claims without deploying localization techniques on the sensor nodes. An algorithm based on the received signal strength indicator (RSSI), link quality indicator (LQI), and packet sequence number (PSN) is proposed by Roy and Nene [9] to defend against replication attacks in stationary WSNs. The RSSI is used to obtain the amount of residual battery life, LQI is dependent on the distance between the transmitter and the receiver nodes, and the PSN is used to keep a check on any duplicate packet generated.

The main idea of XED algorithm [10] to detect replica nodes in mobile WSNs is that if node u meets another node, v, at time T1, it sends a random number, r, to v at the same time. When nodes u and v meet each other once again at time T2, u asks v for the random number sent to it at time T1, and expects v to send it the same number, r. If v is not a replica node, it returns r, but if it is a replica node, it might return another random number.

In [11] an algorithm is proposed to defend against mobile replica nodes which its mechanism was inspired by the fact that a legal node should not move faster than the maximum speed of the configured system. Replica nodes of u make other nodes think that node u moves faster than a predefined speed. In this case, they can mark u as a replica node.

Zhou and Wang [12] proposed an algorithm to detect clone nodes in mobile WSNs. This algorithm has two levels: local detection and global detection. The local detection is performed in a local area much smaller than the whole deployed area to increase the probability of finding

contradictory locations. The global detection over a longer time period assigns an epoch verity location claim with every node it meets.

Two distributed and decentralized algorithms are developed by Dimitriou et al. [13] to detect clone nodes in mobile WSNs. The main behind these algorithms was taken from the consideration that two sensor nodes generate random nuances and send them to each other at their first meeting to be used as the values they exchanged in a previous meeting. At future meetings, if a node cannot reply correctly, or replies with the wrong nuance, it is treated as a malicious node, and the ID of that node is considered cloned.

Conti et al. [14] proposed two protocols to detects mobile clone nodes, based on historical information exchange and its optimized version. Both of these algorithms employ the local information and node mobility to detect replica nodes. The two protocols differ in the amount of computation required.

In [15], an intelligent, and lightweight algorithm based on learning agents and watchdog nodes is proposed to detect clone nodes in mobile WSNs. This algorithm employs some watchdog nodes, each one equipped with a learning agent that monitors the network traffic and nodes' movements to identify potential replica nodes in the network.

B. System Assumptions and Symbols

The network contains n sensor nodes which are distributed randomly in a two-dimensional area and are not aware of their location. Each node has a unique ID and is unaware of its position. The radio range of all nodes is equal. All nodes are mobile and move according to mobility models, like two-dimensional mobility model (IID) [10, 14], throughout the network's lifetime. Nodes communicate with each other through a wireless radio channel and employ omnidirectional dissemination.

Additionally, since sensor nodes are mobile, they should periodically (after each t time unit, or when they reach a new location in the network) broadcast a Hello message, a route request, send data or send a keep-alive message [2, 3, 10, 15-17]. This broadcasting is one of the requirements of a mobile WSN. Therefore, each node can detect its current neighbors

Also, we assume that the sensor network is developed in an adversarial environment; thus, in such an insecure network, an adversary can capture nodes, create copies of them, and inject them into the network. We also assume that each replica node (like normal nodes) in each period t broadcasts a Hello message, a routing request, transmits data, or sends a keep-alive message.

C. The Proposed Algorithm

The main idea of the proposed algorithm, such as XED, is to generate and exchange random numbers among neighboring nodes during the lifetime of the network. The proposed approach aims to improve the efficiency of XED algorithm. The proposed algorithm is described in the following:

Each node only observes S other nodes (S<n).

At each meeting of nodes u and v, a new random number is generated and exchanged.

In the proposed algorithm, each node has two tables called send-table and receive-table according to Fig. 3 and Fig. 4, respectively. Each node u stores ID of the observed nodes and transmitted random numbers in its send_table and stores ID of the nodes from which it has received random numbers along with the received random numbers in its receive_table. The initial value of SentRnd and ReceivedRnd from these two tables is null.

TABLE II.
THE STRUCTURE OF SEND_TABLE OF THE PROPOSED ALGORITHM

	NodeID	*Sent-Rnd*
1		
...		
S		

TABLE III.
THE STRUCTURE OF RECEIVE_TABLE OF THE PROPOSED ALGORITHM

	NodeID	*Received-Rnd*
1		
...		
D		

In the proposed algorithm, in order to reduce the memory and communication overheads, each node has to observe only S other nodes. The parameter S should be smaller than n and $S=\log n$ is a proper selection. The list of nodes being observed by each node is done before deployment of the nodes in the environment randomly.

The size of *receive_table*, D, is calculated using equation (1):

$$Pe - \left(\frac{1}{n-1} + \frac{1}{n-2} + \cdots + \frac{1}{n-S}\right)$$
$$D = Pe \times (n-1) \qquad (1)$$

Here, P_e is the probability of selecting a specific node like u in the list of observed nodes of another node v.

After deployment of the nodes in the network, nodes start moving according to the selected movement model. Whenever node u meets node v in its neighborhood at time t_i, if v is in its observed list, one of the two following procedures is performed:

I. If it is the first meeting with node v, node u generates a random number r and stores it in the row corresponding to v in its *send_table* and transmits it to node v via a message. Upon receiving the message, node v inserts random number r in a row corresponding to node u in its *receive_table*.

II. If node u has met v before t_i, at $t_j < t_i$, that is, the *Sent-Rnd* field of node v in its *sent_table* contains a random number r, node u generates a new random number r' and transmits it to v via a message. Upon receiving this message, node v has to return the previous the random number received from node u at time t_i and store the new random number r' in the row corresponding to u in its *receive_table*. Upon receiving the response from node v, if the random number returned by node v is equal to r, node u verified v and replaced r' by the previous random number r in its *sent_table*. Otherwise, node v is considered as a replica node.

It should be noted that in the proposed algorithm since whenever the two nodes meet, new random numbers are generated if replica nodes cooperate, they can be detected again. In the proposed algorithm, whenever a node like u detects node v to be a replica, it informs all nodes of the network or the base station.

In the proposed algorithm, the required condition for detecting a replica node (v) by a legal node (u) is that *u* meets two different versions of replica nodes *v* at different times. If the adversary captures a legal node *v* in the network and broadcasts several copies of the node in the network, in the proposed algorithm, the node with ID *v* would not be in the list of any other node with probability P_f as a result of which it will not be detected. The P_f is calculated using equation (2):

$$P_f - (1 - Pe)^{n-1} \quad (2)$$

III. Discussion And Simulation Results

In this section, the efficiency of the proposed algorithm in terms of memory and communication overheads is investigated first. Then, the simulation results of the algorithm are presented and the results are compared with the results obtained from XED algorithm.

A. Overhead of the Proposed Algorithm

Memory Overhead: In the proposed algorithm, each node requires a space of 2×log *n* for its *send_table* and a 2×D space for its *receive_table*. While, in XED algorithm, each node requires a space of 3*n* for its history. It is clear that the memory overhead of the proposed algorithm is less than XED.

Communication Overhead: In the proposed algorithm, if each node has *d* neighbors in each movement round, unlike XED, it is not required to transmit a random number for all of them or requests a random number. It is only sufficient to perform the operation for neighbors which are in its observed list. Therefore, the communication overhead of the proposed algorithm is less than XED.

B. Simulation Model

In order to evaluate the efficiency of the proposed algorithm (AXED), a number of experiments are performed and the obtained results are compared with the results of XED. Evaluation metrics are as follows:

- **Detection speed (E_m)**: the number of expected rounds of execution of the algorithm (the number of movements) for detecting replica nodes.
- **Maximum rate per average movement ($H_{M/A}$)**: This metric is obtained by dividing the maximum number of movements on the average number of movements for detecting replica nodes.

In order to simulate the proposed algorithm, C++ is used. In simulations, it is assumed that the network contains *n* sensor nodes which are randomly deployed in a 200×200 m area. It is assumed that the adversary captures a node and generates *R* copies of it and injects them in the network. The movement model considered for the nodes is adapted from **Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.Error! Reference source not found.**. The radio range of nodes is adjusted such that the average number of neighboring nodes is *d*. In order to ensure the validity of the results, each simulation was repeated 500 times, and then, the results were averaged to obtain a final value.

Experiment 1: The purpose of this experiment is to evaluate the effect of the number of replica nodes, *R*, and the average number of neighbors, *d*, on the efficiency of the proposed algorithm and compare its results with XED algorithm. In this experiment, parameters of *n*=1000 and *d*=10, 20 are selected and the number of replica nodes, *R*, varies between 2 to 10 and the obtained results are presented in Fig. 5 and Fig. 6 in terms of E_m and $H_{M/A}$, respectively. As can be seen from the results in Fig. 5, when the number of replica nodes is 2, XED requires 130 movement rounds to detect replica nodes, for *d*=10, and 80 movement rounds when *d*=20. While the proposed algorithm requires 40 and 20 movement rounds to detect replica nodes, for *d*=10 and *d*=20, respectively. In the proposed algorithm, whenever a replica node is detected by node *u*, it informs the base station and the base station informs all other nodes via a multicast message. As a result, all nodes of the network are informed about the existence of replica nodes and their ID.

In addition, the results of this experiment show that as the number of replica nodes in the network increase, the detection speed of these two algorithms increases. Because the replica node would be present in more areas of the network and security algorithms are informed faster which is also satisfied in the proposed algorithm. By increasing the number of replica nodes like *v* in the network, the probability that a legal node which observes this node, meets two different versions of this replica node increases and replica nodes are detected faster.

In addition, increasing the number of neighbors of a node, *d*, detection speed of the proposed algorithm and XED increases. In the proposed algorithm, when the number of neighbors increases, nodes meet each other's more times and faster as a result of which replica nodes are detected faster. Also, it should be noted that by increasing the number of neighbors, communication overhead also increases because more random numbers have to be exchanged.

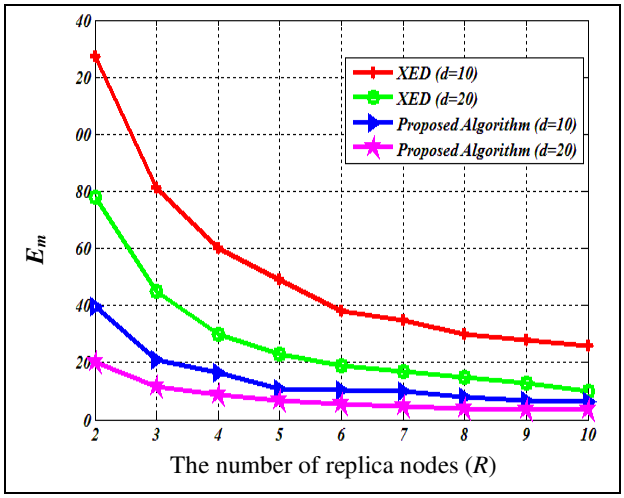

Fig. 2. The effect of parameters *R* and *d* on the efficiency of the proposed algorithm in terms of detection speed and comparison of the results with XED

In addition, the results of this experiment in Fig. 3 show that $H_{M/A}$ for both algorithms is between 2.2 to 4.2 which indicates that in the worst case, the maximum number of movements of nodes for detecting replica nodes is about 4 times the average. The results of this experiment show that $H_{M/A}$ of the proposed algorithm is smaller than XED. Because in the proposed algorithm, replica nodes are detected faster and even if malicious nodes cooperate with

each other, the proposed algorithm would be able to detect the replica nodes. While in such situation where malicious nodes cooperate with each other and exchange their history with each other, replica nodes would be detected with delay or they may not even be detected.

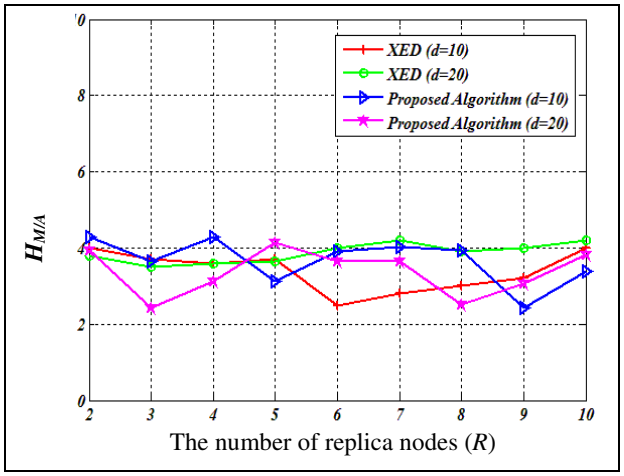

Fig. 3. The effect of parameters R and d on the efficiency of the proposed algorithm in terms of $H_{M/A}$ and comparison of the results with XED.

Experiment 2: This experiment evaluates the effect of the total number of nodes, n, on the efficiency of the proposed algorithm. In this experiment, parameters R=5, d=20, and n=500~3000 are selected and the obtained results are shown in Fig. 4. In this experiment, in order to preserve d=20 neighboring nodes for different values of n, it is required to change the radio range of the nodes. In fact, by increasing n, the radio range of the node should be reduced so that d=20 neighboring nodes is satisfied.

The results of this experiment show that by increasing the total number of nodes in the network, detection speed of the proposed algorithm is reduced. Because by increasing the total number of nodes and reducing the radio range of the node, the probability that replica nodes become neighbors is decreased as a result of which detection speed is also reduced. But, this reduction is tolerable against the increase in the number of nodes. For instance, when there are 500 nodes in the network, E_m=4 while as the number of nodes increases to 3000, E_m increases to 17.

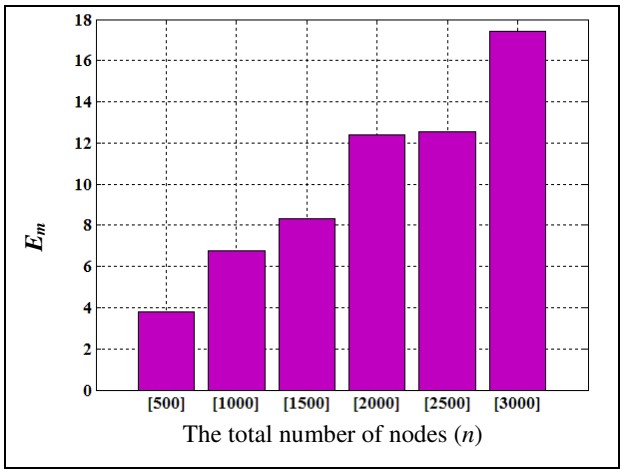

Fig. 4. The effect of the total number of nodes, n, on the efficiency of the proposed algorithm in terms of detection speed of replica nodes.

In addition, Fig. 5 shows that increasing the number of nodes in the network does not affect $H_{M/A}$ and it varies between 2.2 to 4.2. the results of this experiment show that the in the proposed algorithm (different number of nodes in the network), a condition in which malicious node is detected late or is not detected, does not occur. In the worst case, replica nodes are detected almost after 4.2 times the average movement of nodes.

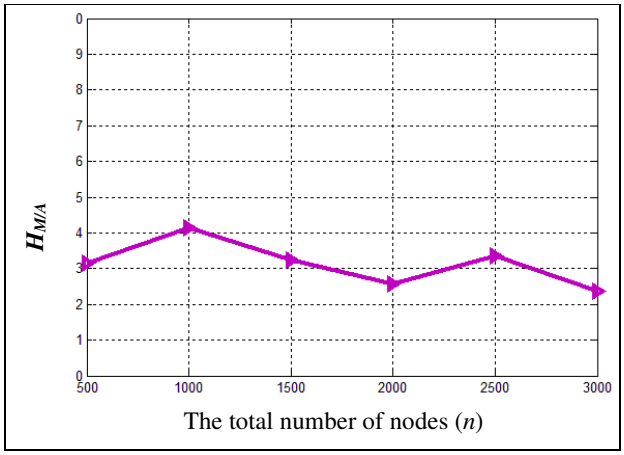

Fig. 5. The effect of the total number of nodes, n, on the efficiency of the proposed algorithm in terms of $H_{M/A}$

IV. Conclusion

In this paper, an algorithm is presented to detect replica nodes in mobile wireless sensor networks which resolves shortcomings of XED algorithm. XED has shortcomings like high communication and memory overheads, low detection speed and security failure. In the proposed algorithm, each node has to observe and verify a few numbers of nodes which reduces the communication and memory overheads imposed on sensor nodes. In addition, in the proposed algorithm, whenever two nodes meet, new random numbers are generated and exchanged to prevent failure of the algorithm against the cooperation of replica nodes. The proposed algorithm is simulated and the results showed that the proposed algorithm is better than XED in terms of detection speed of the replica nodes

References

[1] M. Jamshidi, H. Bazargan, A. A. Shaltooki, A. M. Darwesh, A Hybrid Key Pre-Distribution Scheme for Securing Communications in Wireless Sensor Networks. JOIV: International Journal on Informatics Visualization 3(1) (2019) 41-46.

[2] M. Jamshidi, E. Zangeneh, M. Esnaashari, and M. R. Meybodi, "A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks", Computers & Electrical Engineering, 64, 2017, pp. 220-232.

[3] M. Jamshidi, E. Zangeneh, M. Esnaashari, A. M. Darwesh and M. R. Meybodi. 2019. A Novel Model of Sybil Attack in Cluster-Based Wireless Sensor Networks and Propose a Distributed Algorithm to Defend It. Wireless Personal Communications, 105 (1) 145-173.

[4] M. Jamshidi, M. Ranjbari, M. Esnaashari, N. N. Qader, M. R. Meybodi, “Sybil Node Detection in Mobile Wireless Sensor Networks Using Observer Nodes,” JOIV: International Journal on Informatics Visualization, 2(3): 159-165, 2018.

[5] B. Parno, A. Perrig, V. D. Gligor, “Distributed Detection of Node Replication Attacks in Sensor Networks,” IEEE Symposium on Security and Privacy: 49-63, 2005.

[6] M. Jamshidi, A. A. Shaltooki, Z. D. Zadeh and A. M. Darwesh, “A Dynamic ID Assignment Mechanism to Defend Against Node Replication Attack in Static Wireless Sensor Networks,” JOIV: International Journal on Informatics Visualization. 3(1): 13-17, 2019.

[7] M. Conti, R. D. Pietro, L. V. Mancini, A. Mei, “Distributed Detection of Clone Attacks in Wireless Sensor Networks,” IEEE transactions on dependable and secure computing, 8(5): 685-698, 2011.

[8] C. Ding, L. Yang, M. Wu, “Localization-Free Detection of Replica Node Attacks in Wireless Sensor Networks Using Similarity Estimation with Group Deployment Knowledge,” Sensors, 17(1), 160-152, 2017.

[9] S. Roy, M. J. Nene, “Prevention of node replication in Wireless Sensor Network using Received Signal Strength Indicator, Link Quality Indicator and Packet Sequence Number,” IEEE International Conference on Green Engineering and Technologies (IC-GET), 1-8, 2016.

[10] C. M. Yu, C. S. Lu, S. Y. Kuo, “Mobile Sensor Network Resilient Against Node Replication Attacks,” IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 597-599, 2008.

[11] J. W. Ho, M. Wright, S. Das, “Fast Detection of Mobile Replica Node Attacks in Wireless Sensor Networks Using Sequential Hypothesis Testing,” IEEE transactions on mobile computing, 10(6): 767-782, 2011.

[12] C. Zhou, Z. Wang, “An Two Dimension detection to node replication attacks in mobile sensor networks,” 10th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), 63-69, 2016.

[13] T. Dimitriou, E. A. Alrashed, M. H. Karaata and A. Hamdan, “Imposter detection for replication attacks in mobile sensor networks,” Computer Networks, vol. 108, pp. 210-222, Oct. 2016.

[14] M. Conti, R. D. Pietro, A. Spognardi, “Clone wars: Distributed detection of clone attacks in mobile WSNs,” Journal of Computer and System Sciences, 80(3): 654-669, 2014.

[15] Jamshidi, M., Poor, S.S.A., Qader, N.N., Esnaashari, M. and Meybodi, M.R., 2019. A Lightweight Algorithm against Replica Node Attack in Mobile Wireless Sensor Networks using Learning Agents. IEIE Transactions on Smart Processing & Computing, 8(1), pp.58-70.

[16] M. Jamshidi, A. M. Darwesh, A. Lorenc, M. Ranjbari, M. R. Meybodi, “A Precise Algorithm for Detecting Malicious Sybil Nodes in Mobile Wireless Sensor Networks,” IEIE Transactions on Smart Processing & Computing, 7(6): 457-466, 2018.

[17] M. Jamshidi, M. Ranjbari, M. Esnaashari, N. N. Qader and M. R. Meybodi. 2018. Sybil Node Detection in Mobile Wireless Sensor Networks Using Observer Nodes. JOIV: International Journal on Informatics Visualization, 2(3): 159-165.