VOL 3 (2019) NO 2

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

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

Using One-hop and Two-hop Neighbouring Information to Defend Against Sybil Attacks in Stationary Wireless Sensor Network

Elham Bahmani[#], Sheida Dashtevan[*], Abdusalam Abdulla Shaltooki[**], Mojtaba Jamshidi[***]

[#] *Department of Computer Engineering, Malayer Branch, Islamic Azad University, Malayer, Iran*
[*] *Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran*
[**] *Department of Information Technology, University of Human Development, Sulaymaniyah, Iraq*
[***] *Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*
E-mail: bahmani.elham66@gmail.com, sheida.d6@gmail.com, salam.abdulla@uhd.edu.iq, jamshidi.mojtaba@gmail.com

***Abstract*— Considering the application of wireless sensor networks (WSNs) in critical areas like war fields, establishing security in these networks is of great challenge. One of the important and dangerous attacks in these networks is the Sybil attack. In this attack, a malicious node broadcasts several IDs simultaneously. Thus, the malicious node of the adversary attracts high traffic to itself and disrupts routing protocols and affects other operations of the network like data aggregation, voting, and resource allocation, negatively. In this paper, an efficient algorithm based on one-hop and two-hop neighborhood information is proposed to detect Sybil nodes in the stationary WSNs. The proposed algorithm is executed locally with the collaboration of neighboring nodes. The main purpose of the proposed algorithm is to increase the accuracy of detecting Sybil nodes under various conditions including the condition in which a malicious node broadcasts a few numbers of Sybil IDs which is the shortcoming of most existing algorithms. The proposed algorithm is simulated in MATLAB and its efficiency is compared with two similar algorithms in terms of true and false detection rates. The proposed algorithm not only reduces communication overhead but also increases the accuracy of detecting Sybil nodes compared to two similar algorithms.**

***Keywords*— Wireless Sensor Network, Security, Key Management, Cryptography, Pairwise Keys.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have become one of the most common and applicable networks in recent years which are applied in many areas like explorations, urban services, military, healthcare, monitoring, and etc. Unlike many other common networks which employ computers, laptops, routers, hub/switches, and cables, WSNs are comprised of a large number of sensor nodes which use wireless communications [1].

There are many problems in WSNs including security, routing, topology control, and coverage. Since, wireless networks are widely used in military applications (like monitoring boundaries) and considering nature of wireless and multi-hop transmission of data, constraints (energy, communications, memory, computational power), absence of diligence of nodes in the environment, establishing security in these networks is of great importance. In addition, considering constraints of sensor nodes, complicated and heavyweight algorithms presented in other networks (like local networks) cannot be applied to resource-limited sensor nodes. Therefore, any proposed algorithm for such networks should be lightweight and does not impose a large overhead to the sensor nodes [1-4].

One of the dangerous attacks in WSNs is Sybil attack [5] which affects routing algorithms significantly. In WSNs, each legal node has a unique ID. While, the malicious node establishing the Sybil attack broadcasts several IDs, simultaneously. As a result of which the malicious node attracts a large number of resources and high traffic to itself and affects many operations like routing, data aggregation, resource allocation, and validations, destructively [6].

Till now, various algorithms like [7-20] have been proposed against Sybil attacks in WSNs. Using neighboring information, cryptography protocols, and the received signal strength indicator (RSSI) are three common mechanisms to combat the Sybil attack is WSNs. Using RSSI mechanisms in noisy environments has an error. Using cryptography protocols imposes high overhead to sensor nodes and has low flexibility when new nodes are added to the network. On the contrary, using neighborhood information is more promising.

Algorithms [10-12] use neighborhood information to detect the Sybil attack. The efficiency of these algorithms relies on the assumption that "the number of Sybil IDs

broadcasted by a malicious node is more than the number of normal neighbor nodes in the network". This assumption indicates that algorithms [10-12] do not perform efficiently when malicious nodes broadcast a few numbers of Sybil nodes (less than the number of normal neighbor nodes). This assumption is acceptable for low-density networks but when the number of nodes increases in the network, this assumption is very difficult. Also, it should be noted that WSNs usually contains hundreds or thousands of nodes. In such condition, the average number of neighbor nodes might reach to tens of numbers (for instance, 30 or more). In such networks, algorithms like [10-12] cannot be efficient because there is no guarantee that each malicious node broadcasts more than 20 or 30 Sybil nodes.

In this paper, a new algorithm based on one-hop and two-hop neighborhood information is proposed to detect Sybil attacks in stationary WSNs such that shortcomings of algorithms [10-12] are resolved.

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 Sybil 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

Demirbas and Song [6] used four detector nodes and a simple RSSI mechanism to detect Sybil nodes in WSNs. Another RSSI-based algorithm is proposed by Misra and Myneni [8] which can detect Sybil attack even if the malicious node changes the transmission power for each of its Sybil nodes. Jamshidi et al. first proposed a novel model of Sybil attack in cluster-based WSNs [9]. They also proposed an RSSI-based algorithm which runs on cluster heads and detects Sybil nodes that joined multiple clusters at the same time.

Jamshidi et al. [10] also proposed another algorithm which employing some mobile observer nodes to detect Sybil nodes in stationary WSNs. In this algorithm, observer nodes first walk in the network and gather some data about suspicious areas. Then, each observer node runs a make-decision algorithm to analyze its collected data and identifies Sybil nodes in the network. Ssu et al. [11] proposed an algorithm which uses the neighboring information and network density to detect Sybil nodes. This algorithm is based on the assumption that the probability of two neighbor nodes having exactly the same set of neighbors is extremely low provided that the network has a high node density. Rafeh and Khodadad [12] has proposed a two-hop neighboring based algorithm to defend against Sybil attacks in static WSNs. This algorithm discovers the common neighbors between each couple of nodes which are two-hop neighbors by propagating some control messages. The number of neighboring neighbors has been used as a metric to identify the attack.

In [13], an algorithm based on evaluating trust values of neighbor nodes is proposed to combat Sybil nodes in WSNs. A message authentication algorithm is proposed by Dhamodharan and Vayanaperumal [14] to combat the Sybil attack in WSNs. This algorithm uses message authentication and passing procedure for authentication prior to communication. A rule-based anomaly detection system is proposed by Sarigiannidis et al. [15] which relies on an Ultra-Wide Band (UWB) ranging-based detection algorithm to combat Sybil attack. Also, some algorithms [16-19] have been presented to detect Sybil nodes in mobile WSNs [21-24]. These algorithms are based on nodes' mobility and hence they are not applicable in stationary WSNs.

B. System Assumptions And Symbols

Sensor network contains N sensor nodes which are distributed randomly in a two-dimensional area and are not aware of their location. Nodes are stationary and have a unique ID. Radio range of all nodes is constant and equal to r. It is assumed that nodes are aware of network's approximate density, d (or the average number of neighbors of a node), and if network's density changes, the base station informs all nodes securely. It is assumed that the nodes communicate with each other via a wireless radio channel and broadcast packets in an Omni-directional mode. It is also assumed that the sensor network is deployed in adversary environment, thus, the network is not secure and nodes might be captured by the adversary [10].

In this paper, according to taxonomies of [6], direct, simultaneous Sybil attack and fake or stolen identities are considered. The node captured by the adversary is called malicious node and other nodes of the network are called normal nodes. Each malicious node propagates S identities (Sybil nodes).

C. The Proposed Algorithm

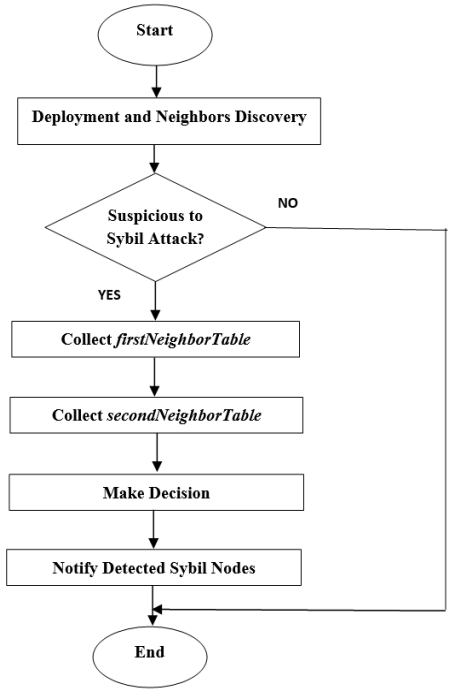

Fig. 1. Flowchart of the proposed algorithm

The main idea of the proposed algorithm is to detect Sybil nodes locally using one-hop and two-hop neighborhood information. The main purposes of designing the proposed algorithm are:

I. reducing the false detection rate.
II. detecting Sybil nodes while a few Sybil IDs are broadcast by the malicious node.
III. reducing the communication overhead.

A flowchart of the proposed algorithm is shown in Figure 1. In the following, the steps of the proposed algorithm are described.

C.1. NODES *DEPLOYMENT* AND DISCOVERING NEIGHBORS

First, sensor nodes are deployed in the environment randomly. Depending on the network and application, nodes might be deployed through helicopter or human force in the environment. Mainly, there is no initial knowledge of network topology and the position of nodes in the environment. On the other hand, sensor nodes employ short-range communication considering their constraints. Therefore, sensor nodes which have a large distance from the base station and the sink, cannot deliver their data to them directly and should deliver their packets using the multi-hop method, intermediate and neighboring nodes. Hence, after deployment in the environment, sensor nodes have to explore their neighbors to accomplish the mission of the network with the cooperation and deliver their data to the base station hop by hop.

Consequently, in most sensor networks, after deployment of nodes in the environment, each node becomes aware of its one-hop neighbors through sending a *Hello* message. This message is a broadcast method and all nodes inside radio range of the transmitter node (for example, node u) receive this packet and consider u as their one-hop neighbor. Thus, each node broadcasts a Hello message and explore its one-hop neighbors considering received *Hello* messages. Each node stores the ID of its one-hop neighbors in a list called *neighborList*.

Malicious nodes of the adversary can be present when the network starts or be injected to the network later. In both cases, Sybil nodes introduce themselves to the neighboring nodes through broadcasting a *Hello* message.

C.2. ATTACK *SUSPECTED* CONDITION

In this phase of the proposed algorithm, each node decides independently if it is inside an area suspected to Sybil attack or not. Each sensor node u see himself inside a suspicious area if the number of its one-hop neighbors to be greater than a threshold, $T_h=d+1$. Parameter d is the average number of neighbors which can be calculated as in Eq. (1) easily before deployment of the nodes in the network.

$$d = \left(\frac{N}{X \times Y} \times r^2 \times \pi\right) - 1 \qquad (1)$$

Where N is the total number of legal node in the network, X and Y are dimensions of the network environment and r is the radio range of the nodes. Since the presence of a malicious node which established a Sybil attack in a specific area of the network increases the number of neighbors, this reality can be used to detect areas suspected to attack.

Depending on the number of broadcasted Sybil IDs by the malicious node (means, S), the number of neighbors in the attacked area would be greater than average, d. A malicious node which establishes a Sybil attack should broadcast at least S=2 fake IDs for the Sybil attack to be meaningful. Therefore, in the proposed algorithm, if the number of neighbors is greater than d+1, Sybil attack is suspected.

If there is no suspicious node in the neighborhood, the rest of the proposed algorithm is not executed. But, if there exists a suspicious node in its neighborhood, the rest of the proposed algorithm is executed.

C.3. FIRST ROUND OF COLLECTING NEIGHBORHOOD INFORMATION

In this step, each node u which has suspected Sybil attack in its neighborhood broadcasts a message as detecting a suspicious area conveying the list of all of its one-hop neighbors, means *neighborList*, to its one-hop neighbors. If a neighboring node which receives this packet, for example, node v, is suspicious of attack, that is, the number of its neighbors is greater than T_h, it has to return the list of its neighbors to node u. Node u aggregates information of its *neighborList* and all *neighborList*s received from its neighbors. In the proposed algorithm, each node has a table entitled as *checklist* as shown in Fig. 2 which is comprised of two fields. The field *nodeID* keeps the ID of the one-hop neighbors and a numerical value called "aggregation value" is stored in the field tag. Node u assigns an aggregation value to each neighbor during the first and second rounds of neighborhood information collection. This field is used to detect Sybil nodes.

nodeID	*tag*

Fig. 2. Structure of *Checklist* table of the nodes in the proposed algorithm

Node u aggregates its own *neighborList* (entitled $neighborList_u$) and the *neighborList*s received from neighbors (entitled as $neighborList_k$) according to Algorithm I.

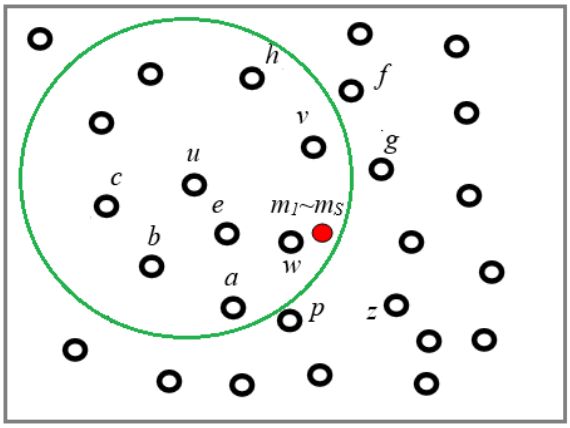

Fig. 3. A high-level view of executing the proposed algorithm

This phase of the proposed algorithm is described according to Fig. 3. In this figure, the malicious node broadcasts Sybil IDs m_1 to m_s. All nodes which are in the

neighborhood of the malicious node like *u, w, a, b, e, g, p, v, z,* and *f* execute the proposed algorithm if their number of neighbors is greater than threshold *d*. In the following, execution of the first phase of neighborhood information collection for node *u* is described. It is assumed that only *w*, *a*, *b, v*, and *e* among neighbors of *u* are suspicious of Sybil attack and other neighbors like *c* and *h* are not suspicious of attack

In this phase of the proposed algorithm, if node *u* receives *neighborList* from at least two of its neighbors, it performs aggregation. In other words, if two neighbors of node *u* are suspicious of Sybil attack, *u* aggregates information and continues detecting Sybil attack. This policy is to ensure the presence of Sybil attack in an area of the network and Sybil detection algorithm is performed only when the probability that an attack might occur in the network is high. Using this policy increases the accuracy of selecting Sybil nodes and reduces overhead imposed to the sensor nodes.

First, node *u* (Step 2) initializes its *checklist* table based on *neighborList* and initializes each neighbor in the *checklist* with an aggregation value of 1.

Then, node *u* (Step 3) scans its *checklist* for each of the *neighborList*s received from its neighbors *w*, *a*, *b*, *e*, and *v* and if there exists a common neighbor, one unit is added to aggregation value of the corresponding node (common neighbor) in the *checklist*.

In general, in this phase of the proposed algorithm, each node *u* assigns an aggregation value to its neighboring nodes with the help of its one-hop neighbors.

Since Sybil nodes are common neighbors of nodes *u*, *w*, *a*, *b*, *e,* and *v*, the same aggregation value is assigned to them (for example, the value of 6). Indeed, node *u* might assign aggregation value of 6 to some of its other neighbors like *w* and *e* because they might be common in the list of neighbors.

Up to this step, only one-hop neighborhood information is used to detect the Sybil attack. If node *u* only decides a node being Sybil or not based on this information, that is, it only employs information of common neighbors, like algorithms [10-12], many legal nodes are mistakenly marked as Sybil node. Although the true detection rate of Sybil nodes increases but the false detection rate increases also especially when the malicious node broadcasts a few numbers of Sybil IDs or some legal nodes are located near the malicious nodes.

In summary, in this step of the proposed algorithm, node *u* investigates if the elements of its *checklist* exist in *checklist* of its neighbors being suspicious of Sybil attack. This process is executed simultaneously by all neighboring nodes of the malicious node like *u*, *w*, *a*, *b*, *e*, *g*, *p*, *v*, *z*, and *f*. For example, in checklist of node *a*, information collected from neighbors like *u*, *w*, *b*, *e*, *p*, and *z* also exist. In addition, in checklist of node *v*, information collected from neighbors like *g* and others exist.

Algorithm I: The algorithm for the first round of collecting neighborhood information

1. Start
2. For i=1 to size ($neighborList_u$)
   - checklist[i][NodeID]= $neighborList_u$[i]
   - checklist[i][tag]= 1

End

3. If number of received $neighborList_k$ >= 2
   For each received $neighborList_k$

   For each node *v* in *u's* checklist, chechList[i][NodeID]

   If (*v* exsist in $neighborList_k$)

   - chechList[i][tag]= chechList[i][tag]+1

End

4. Finish

C.4. Second Round of Collecting Neighborhood Information

After collecting one-hop neighborhood information, node *u* collects its neighborhood information again to increase the accuracy of detecting Sybil nodes. At the end of the first round, legal nodes which are suspicious to Sybil attack in a specific area of the network, complete their *checklist* based on their one-hop neighbors.

In this phase, suspicious nodes which have executed the first round of neighborhood information collection transmit their *checklist* to their one-hop neighbors. For instance, nodes *w*, *a*, *b*, *e*, and *v* transmit their *checklist* to node *u*.

Node *u* performs aggregation on the *checkList*s received from its neighbors according to algorithm II. In fact, in this phase, node *u* employs two-hop neighborhood information to detect Sybil nodes. For instance, in Fig. 3, node *u* employs the aggregation value of its two-hop neighboring nodes like *g*, *p*, and *z* which discriminates the aggregation value assigned to the Sybil nodes from those assigned in the first round. For instance, in Fig. 3, when the first round of neighborhood information collection is finished, node *u* assigns the same aggregation value of 6 to Sybil nodes m_1 to m_s and legal nodes *e* and *w*. But, when the second round is finished, since nodes *e* and *w* are not in the list of common two-hop neighbors, for example, *e* does not exist in neighborhood list of *z*, their aggregation value would be different from the aggregation value of Sybil nodes. In fact, aggregation value of the Sybil nodes becomes different from the value of other nodes in *checklist* of the second round of node *u* which increases the accuracy of detecting Sybil nodes and reduces the false detection rate significantly.

Algorithm II: The algorithm for the second round of collecting neighborhood information

1. Start

2. For each received $checkList_k$

   For each node *v* in *u's* checklist, chechList[i][NodeID]

   If (*v* exsist in $checkList_k$)

   - chechList[i][tag]= chechList[i][tag]+1

End

3. Finish

C.5. DECISION-MAKING PHASE

In this phase of the proposed algorithm, each node detects Sybil node based on the *checklist* of its first and second rounds. The fact that all Sybil nodes belong to a unique hardware node (malicious node) and their position is also the same, is used to detect Sybil nodes. Considering this fact and the mechanism proposed in first and second phases of neighborhood information collection, it is clear that the aggregation value assigned to all Sybil nodes m_1 to m_s in the *checklist* of the *first* round (for example, value of α) and second round (for example, value of $\beta>\alpha$) would be the same. It is clear that the aggregation value assigned in the second round (β) is higher than the value assigned to each node in the first round (α). The aggregation value of the first round is taken from one-hop neighborhood information but aggregation value of the second round is taken from both one-hop and two-hop neighborhood information.

Decision-making phase of the proposed algorithm is very simple:

I. First, node u divides its first round *checklist* based on aggregation value to separate sets. That is, nodes with the same aggregation value are located in the same set.
II. Sets with less than three members are eliminated (assuming that malicious node of the adversary broadcasts at least three Sybil IDs).
III. For each set $L= \{N_1, N_2, \ldots, N_k\}$ (containing $k>2$ members), if all members of the set L have the same aggregation value in *checklist* of the second round, in other words, they are in the same set in the second round, they are detected as Sybil nodes. Otherwise, set L is considered as the set of legal nodes.

III. DISCUSSION AND SIMULATION RESULTS

A. *OVERHEAD OF THE PROPOSED ALGORITHM*

Memory Overhead: assuming that each node has an average of d neighbors (one-hop neighbors), then the memory overhead of the algorithms [11] and [12] is on the order of $O(d^2)$ and overhead of algorithm [10] and the proposed algorithm is on the order of $O(d)$. In the proposed algorithm, each node requires two *checklist* vectors for collecting neighborhood information in the first and second rounds. The magnitude of each *checklist* vector is equal to the average number of neighbors, d. Therefore, the memory overhead of the proposed algorithm for each node is $2d$ and on the order of $O(d^2)$. Thus, the memory overhead of the proposed algorithm is less than the two similar algorithms [11, 12] and is equals to algorithm [10].

Communication Overhead: considering the energy constraints of the sensor nodes, the energy consumption of the proposed algorithms for sensor nodes is an important issue. Since, packet transmission consumes more energy than packet processing and packet reception, calculating the number of transmitted packets which is imposed to the network due to employing a specific algorithm is an important measure for evaluating the efficiency of the proposed algorithms for sensor networks [11].

In the proposed algorithm, only nodes which are suspicious of Sybil attack in their neighborhood, run the detection algorithm. In the proposed algorithm, each suspicious node transmits two packets. In the first round of neighborhood information collection, a packet including *neighborlist* is transmitted and in the second round, a packet including *checklist* is transmitted. Therefore, communication overhead of the proposed algorithm for total nodes of the network is on the order of $O(N)$, while the communication overhead of algorithms [11] and [12] is on the order $O(N \times d^2)$ and $O(N \times d)$, respectively and the communication overhead of algorithm [10] is on the order of $O(R \times |MN| \times d)$ where, $|MN|$ is the total number of watchdog nodes in the network and R is the number of rounds in which the algorithm should monitor the nodes' traffic and mobility.

B. *SIMULATION MODEL*

The proposed algorithm is implemented with MATLAB simulator. In our simulation, it is assumed that the network includes N sensor nodes randomly distributed in a 100×100 m^2 area. The network contains M malicious nodes with random distribution, each of which broadcasts S fake Sybil identifiers. All nodes (normal and malicious) have the same radio range of r= 10 m. Each simulation was repeated 100 times and the mean of 100 repetitions has been calculated.

Our evaluation metrics are as follows:

- **True Detection Rate (TDR):** percentage of Sybil nodes which are detected by a security algorithm.
- **False Detection Rate (FDR):** percentage of normal nodes which are detected as Sybil nodes incorrectly.

C. *EXPERIMENTS RESULTS*

Experiment 1: This experiment investigates the effect of the number of nodes in the network, N, on detection accuracy of the proposed algorithm and obtained results are compared with algorithms [11] and [12]. In this experiment, parameters S=20 and N=150~400 are considered and the results are shown in Fig. 4 and Fig. 5.

The results of this experiment in Fig. 4 show that by increasing the number of nodes in the network, TDR in the proposed algorithm increases. For instance, when N=150, TDR would be 71% and when the number of nodes is increased to N=300 or N=400, TDR increases to 96.7% and 98.3%, respectively. Because as the number of nodes increases, the number of legal nodes in the neighborhood of the malicious node increases and the condition "each legal node u should receive suspicious *neighborlist* from at least two of its neighbors" is satisfied and more Sybil nodes are detected. Therefore, as the number of nodes increases, the TDR of the proposed algorithm increases. This is held for algorithm [12] also. For algorithm [11], the TDR is always greater than the proposed algorithm and algorithm [12] and it is about 99.5%.

Comparing the results, it can be seen that the algorithm [11] has higher TDR compared to the proposed algorithm and algorithm [12] only when there are a few numbers of nodes in the network.

But, considering the results of this experiment in terms of FDR, a better comparison can be presented. The FDR is an essential measure because its being high indicates that a large part of legal nodes is eliminated. For instance, the

results of this experiment in Fig. 5 show that when N=150, FDR of the algorithm [11] is 11% which is very high. While FDR of the proposed algorithm and algorithm [12] is about 3%. That is, the FDR of the algorithm [11] is 4 times the proposed algorithm and algorithm [12]. Indeed, as the number of nodes of the network increase, FDR of the three algorithms is reduces but for N>200, FDR of the proposed algorithm is always lower than the two other algorithms.

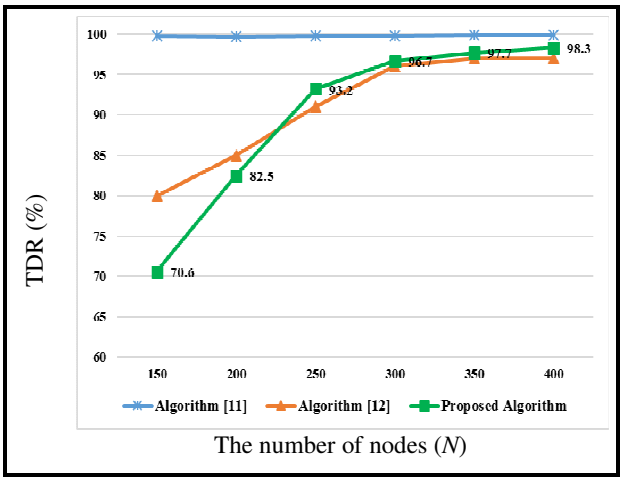

Fig. 4. Effect of parameter N (the number of nodes) on the TDR of the proposed algorithm and other algorithms.

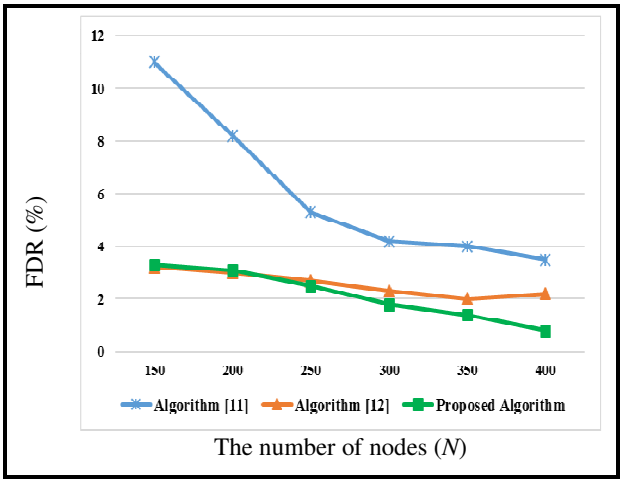

Fig. 5. Effect of parameter N (the number of nodes) on the FDR of the proposed algorithm and other algorithms.

In the proposed algorithm, as the number of nodes in the network increases, the number of legal nodes neighboring the malicious node increases as a result of which each node detects Sybil nodes in collaboration with more one-hop and two-hop neighbors which increases the accuracy of detecting Sybil nodes as a result of which TDR increases and FDR decreases.

Considering both TDR and FDR, the proposed algorithm has higher efficiency compared to algorithm [11]. Compared to Algorithm [12], when the number of nodes is less than 200 (N<200), the proposed algorithm performs weaker in terms of TDR and it outperforms in terms of FDR and TDR for N>200.

Experiment 2: this experiment investigates the effect of parameter S on detection accuracy of the proposed algorithm and the obtained results are compared with the results obtained from other algorithms. In this experiment, parameters N=300 and S=10~20 are considered.

The results of this experiment in Fig. 6 show that by varying the number of Sybil IDs from malicious nodes, TDR of the proposed algorithm does not change significantly and it is about 95%. While it is 99.5% for algorithm [11] and between 90% to 95% for algorithm [12].

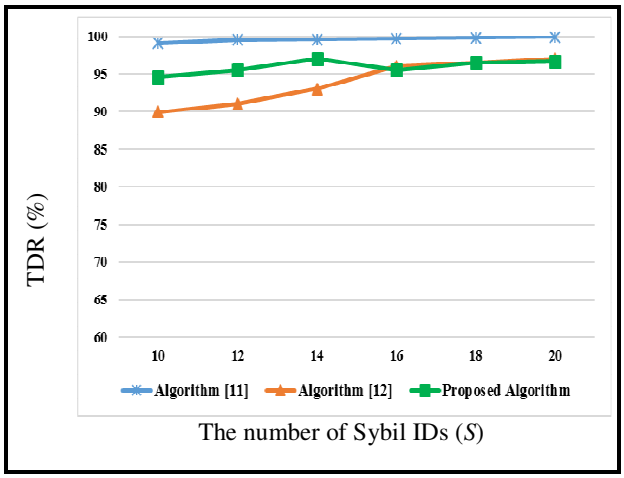

Fig. 6. Effect of parameter S (the number of Sybil IDs) on the TDR of the proposed algorithm and other algorithms.

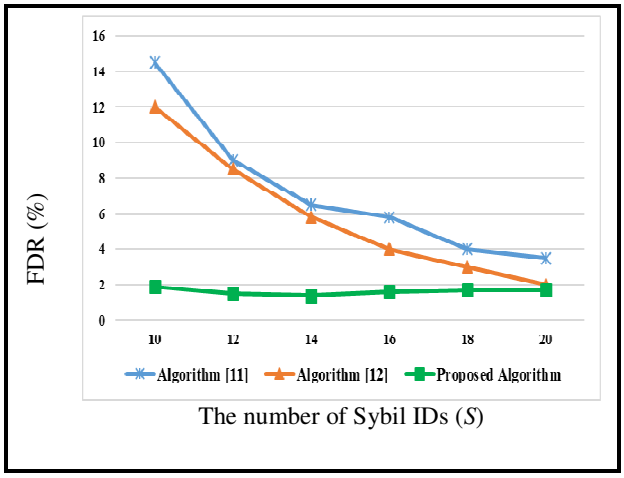

Fig. 7. Effect of parameter S (the number of Sybil IDs) on the FDR of the proposed algorithm and other algorithms.

In addition, the results of this experiment in Fig. 7 show that for S=10 to S=20, FDR of the proposed algorithm is less than 2% while it is very high for the other two algorithms. For instance, when S=10, FDR of the algorithm [11] is higher than 14% and FDR of the algorithm [12] is 12%. It is clear that algorithms [11] and [12] are efficient when malicious nodes broadcast a large number of Sybil IDs. But if the malicious node broadcasts a few numbers of IDs, these two algorithms cannot be employed. Because, algorithms [11] and [12] rely on the number of common neighbors and if the number of common neighbors exceeds a threshold, the

common neighbors are selected as Sybil nodes. While the proposed algorithm employs one-hop and two-hop neighborhood information and depends on the number of legal nodes neighboring malicious nodes. In other words, the proposed algorithm tries to detect Sybil attack in collaboration with legal nodes neighboring malicious nodes. Using this policy, it is tried to make the proposed algorithm independent of the number of Sybil IDs broadcast by each malicious node. because there is no knowledge available about the number of Sybil IDs broadcasted by a malicious node. But algorithms [11] and [12] rely on the assumption that the number of Sybil IDs broadcasted by each malicious node is larger than the average number of neighbors. But there is no such assumption in the proposed algorithm. In order to verify this claim, in the next experiment, the efficiency of the proposed algorithm for smaller values of *S* is investigated.

Experiment 3: Purpose of this experiment is to show the efficiency of the proposed algorithm under the most difficult Sybil attack establishment conditions. The most difficult condition is when each malicious node broadcasts a few numbers of Sybil IDs. In this experiment, the total number of nodes is *N*=300 and the number of broadcast Sybil IDs by each malicious node varies from *S*=3 to *S*=10 and its effect on TDR and FDR is studied. The results of this experiment in Fig. 8 and Fig. 9 show that as the number of Sybil IDs increase from *S*=3 to *S*=10, TDR of the proposed algorithm increases from 48% to 95% and FDR fluctuates between 1% and 2%. The results of this experiment indicate the desired efficiency of the proposed algorithm under difficult conditions for establishing a Sybil attack. While similar algorithms cannot be employed under such condition due to their high FDR.

IV. Conclusion

In this paper, an algorithm based on single-hop and double-hop neighborhood information is presented to detect Sybil attack in wireless sensor networks. The proposed algorithm is executed locally and legal nodes in the neighborhood of the malicious node try to detect Sybil attack in collaboration with each other. In the proposed algorithm, sensor nodes are first deployed in the environment and explore their single-hop neighbors by transmitting Hello messages. Then, each node whose number of neighbors exceeds the average number of neighbors under normal condition is suspicious of attack and tries to detect Sybil nodes in collaboration with its neighbors which are also suspicious of Sybil attack.

The efficiency of the proposed algorithm is evaluated in terms of memory overhead and communication overhead and the results are compared with two similar algorithms. Evaluation results show that the memory overhead of the proposed algorithm is similar to algorithms [11] and [12]. But, in terms of communication overhead, the proposed algorithm outperforms algorithms [11] and [12] with an overhead of order O(N). In addition, the efficiency of the proposed algorithm is evaluated in terms of false detection rate and true detection rate and the results are compared with algorithms [11] and [12]. Results of experiments and comparisons show that the proposed algorithm outperforms algorithms [11] and [12] in terms of false detection rate and true detection rate

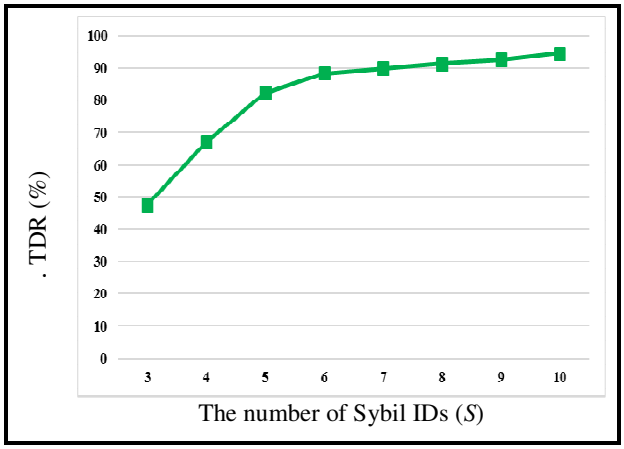

Fig. 8. Effect of parameter *S* (the number of Sybil IDs) on the TDR of the proposed algorithm.

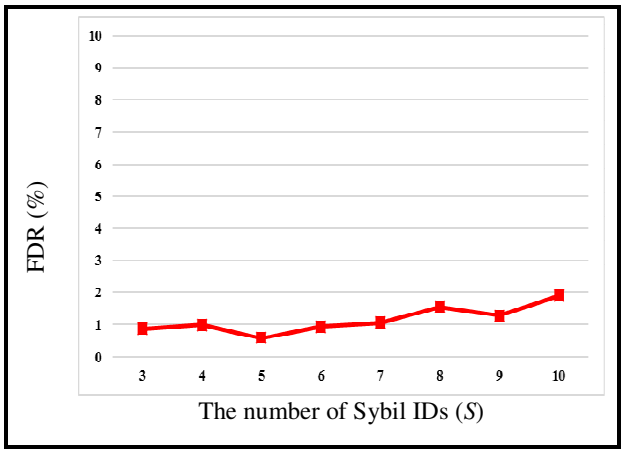

Fig. 9. Effect of parameter *S* (the number of Sybil IDs) on the FDR of the proposed algorithm.

References

[1] J. Yick, B. Mukherjee and D. Ghosal. 2008. Wireless sensor network survey. Computer Networks. 52(12): 2292–2330.

[2] 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, Vol. 3, No. 1, 2019.

[3] M Jamshidi, H Bazargan, AA Shaltooki, AM 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.

[4] 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.

[5] J. R. Douceur, "The Sybil attack", First International Workshop on Peer-to-Peer Systems (IPTPS '02), 2002.

[6] J. Newsome, E. Shi, D. Song and A. Perrig. 2004. The Sybil attack in sensor networks: analysis and defenses. Proc. International Symposium on Information Processing in Sensor Networks: 259–268

[7] M. Demirbas and Y. Song, "An RSSI-based scheme for Sybil attack detection in wireless sensor networks" IEEE Computer Society International Symposium on World of Wireless, Mobile and Multimedia Networks, pp. 570–574, 2006.

[8] S. Misra and S. Myneni, “On Identifying Power Control Performing Sybil Nodes in Wireless Sensor Networks Using RSSI”, IEEE Global Telecommunications Conference (GLOBECOM), pp. 1-4, 2010.

[9] 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.

[10] Jamshidi, M., Ranjbari, M., Esnaashari, M., Darwesh, A.M. and Meybodi, M.R., 2019. A New Algorithm to Defend Against Sybil Attack in Static Wireless Sensor Networks Using Mobile Observer Sensor Nodes. Adhoc & Sensor Wireless Networks, 43, pp. 213–238.

[11] K. F. Ssu, W. T. Wang, and W. C. Chang, “Detecting Sybil attacks in wireless Sensor Networks using neighboring information”, Computer Networks, Vol. 53, No. 18, pp. 3042–3056, 2009.

[12] R. Rafeh, and M. Khodadadi, “Detecting Sybil Nodes in Wireless Sensor Networks Using Two-hop Messages”, Indian Journal of Science and Technology, Vol. 7, No. 9, pp. 1359-1368, 2014.

[13] S. Rupinder, J. Singh, and R Singh, “TBSD: A Defend Against Sybil Attack in Wireless Sensor Networks”, International Journal of Computer Science and Network Security (IJCSNS), Vol. 16, No. 11, 2016.

[14] U. S. Dhamodharan and R. Vayanaperumal, “Detecting and Preventing Sybil Attacks in Wireless Sensor Networks Using Message Authentication and Passing Method”, The Scientific World Journal, Vol. 1, No. 1, pp. 13-17, 2015.

[15] Sarigiannidis, P., Karapistoli, E. and Economides, A.A., 2015. Detecting Sybil attacks in wireless sensor networks using UWB ranging-based information. expert systems with applications, 42(21), pp.7560-7572

[16] 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.

[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.

[18] A. Andalib, M. Jamshidi, F. Andalib and D. Momeni. 2016. A Lightweight Algorithm for Detecting Sybil Attack in Mobile Wireless Sensor Networks using Sink Nodes. International Journal of Computer Applications Technology and Research. 5(7): 433 – 438.

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