Wireless Personal Communications
https://doi.org/10.1007/s11277-018-6107-5

A Novel Model of Sybil Attack in Cluster-Based Wireless Sensor Networks and Propose a Distributed Algorithm to Defend It

Mojtaba Jamshidi[1] · Ehsan Zangeneh[2] · Mehdi Esnaashari[3] · Aso Mohammad Darwesh[1] · Mohammad Reza Meybodi[4]

Abstract
Today, Wireless Sensor Networks are widely employed in various applications including military, environment, medical and urban applications. Thus, security establishment in such networks is of great importance. One of the dangerous attacks against these networks is Sybil attack. In this attack, malicious node propagates multiple fake identities simultaneously which affects routing protocols and many other operations like voting, reputation evaluation, and data aggregation. In this paper, first, a novel model of Sybil attack in cluster-based sensor networks is proposed. In the proposed attack model, a malicious node uses each of its Sybil identity to join each cluster in the network. Thus, the malicious node joins many clusters of the network simultaneously. In this paper, also a distributed algorithm based on Received Signal Strength Indicator and positioning using three points to defend against the novel attack model is proposed. The proposed algorithm is implemented and its efficiency in terms of true detection rate, false detection rate, and communication overhead is evaluated through a series of experiments. Experiment results show that the proposed algorithm is able to detect 99.8% of Sybil nodes with 0.008% false detection rate (in average). Additionally, the proposed algorithm is compared with other algorithms in terms of true detection rate and false detection rate which shows that the proposed algorithm performs desirably.

Keywords Wireless sensor network · Sybil attack · Novel attack model · Clustering

1 Introduction

A Wireless Sensor Network (WSN) comprises a large number of limited resource sensor nodes (in terms of energy, memory, radio range, processing and etc.). Clustering is one of the effective scaling mechanisms for reducing traffic and energy consumption of the network. In this mechanism, the network is divided into several sections called clusters. Each

✉ Mojtaba Jamshidi
jamshidi.mojtaba@gmail.com

Extended author information available on the last page of the article

Published online: 15 December 2018

cluster has one node which is called cluster head. Each cluster head node aggregates data collected by its cluster members and if needed transmits data to the base station [1].

One of the challenges in clustered sensor networks is the security problem and many attacks, such as Sybil [2], Clone [3], and Sinkhole [4] have been introduced by researchers. Sybil attack is one of the most dangerous attacks in routing layer which might be established in cluster-based sensor networks also. In this attack, adversary node propagates multiple fake identities simultaneously which defrauds the neighboring normal nodes and normal nodes consider each Sybil node as a separate node, thus malicious node attracts high resource and traffic and disrupts routing protocols significantly, and this can affect operations like data aggregation, resource allocation, and voting or it might even inject false data into the network [4–7].

In [8, 9], a Sybil attack model in clustered WSNs is proposed. In this attack model, malicious node tries to introduce itself as the cluster head to affect the performance of the network. But, since cluster heads usually communicate with the base station directly and play an important role in the network, their behavior is controlled by the base station. Therefore, if malicious node is the cluster head, it is detected by the base station, because base station node is unlimited in terms of resources and might contain complex and robust security algorithms, and base station is usually aware of all network information, including total number of nodes, identity of other nodes, encryption keys and etc.

In short, the major contributions of this paper are as follows:

- A novel Sybil attack model in clustered WSNs is proposed. The proposed Sybil attack model is more dangerous and stronger than the model proposed in [8, 9]. In the proposed Sybil attack model, malicious node does not try to introduce itself as the cluster head but it tries to join multiple clusters simultaneously, where the base station cannot easily detect it and it can affect multiple clusters simultaneously (by injecting false data).
- A distributive algorithm based on RSSI and collaboration of cluster head nodes is proposed to defend against the proposed Sybil attack model.

The rest of this paper is organized as follows: In Sect. 2, related works are reviewed. In Sect. 3, network assumptions and attack model are discussed. Section 4 introduces the proposed algorithm to defend against the novel attack model, while Sect. 5 presents the simulation results. The paper is concluded in Sect. 6.

2 Related Work

Sybil attack was first proposed by Douceur [2] and it was mentioned that peer-to-peer networks are vulnerable to this attack. Karlof [5] suggested that this attack can affect routing protocols of Sensor Networks. Newsome et al. [7] analyzed Sybil attack in WSNs systematically. They also proposed taxonomies of Sybil attack in terms of identity generation, communicating Sybil nodes and simultaneity, to which most researchers refer. Moreover, approaches for (1) detecting Sybil nodes using radio resource testing, (2) Detecting Sybil nodes using random key predistribution, (3) defending against Sybil attack using identity registration mechanism, and (4) remote code verification or code attestation are proposed.

Chenet al. [8] proposed an RRSI-based algorithm for detecting Sybil nodes in WSNs based on LEACH routing protocol. They proposed a Sybil attack model in which

malicious node acts as a cluster head. Thus, the number of cluster heads exceeds an optimal number, C_{opt}. When the number of cluster heads exceeds C_{opt}, intrusion detection system is activated and base station node detects Sybil nodes considering RSSI of cluster head nodes. This mechanism is centralized and has problems in terms of scalability. On the other hand, the considered attack model is simple and can be easily detected by the base station.

Jangra and Priyanka [9] proposed an algorithm for detecting Sybil nodes in LEACH routing protocol which have used the attack model proposed in [8] and uses RSSI of cluster head nodes and Jakes channel to detect Sybil nodes. Vasudeva1 and Sood [10] proposed an algorithm for detecting Sybil nodes in mobile WSNs based on lowest ID clustering.

Zhong et al. [11] proposed a positioning algorithm based on RSSI which uses the ratio of RSSIs from multiple receivers to estimate the location of nodes. Demirbas and Song [12] used the positioning mechanism proposed in [11] to present an algorithm for detecting Sybil nodes. In this algorithm, four monitor nodes, which can overhear packets from all regions if the network is used so that whenever a packet is transmitted, monitor nodes can estimate its location collaboratively. This is sufficient to detect Sybil nodes because Sybil nodes are all in one location.

Ssu et al. [13] proposed a distributive algorithm for detecting Sybil nodes which do not require a hardware or information of signal strength and only employs the number of neighbors to detect Sybil nodes. Ramachandran and Shanmugan [14] proposed an algorithm for detecting Sybil attack in multicast routing protocols based on geographical location. Misra and Myneni [15] proposed a mechanism based on advanced RSSI to detect Sybil nodes, such that Sybil nodes cannot hide by adjusting power.

Muraleedharan et al. [16] proposed a method which can collect routing information using swarm intelligence and Sybil nodes are detected through energy variations. Wen et al. [17] proposed another approach for detecting Sybil attack which is based on Time Difference Of Arrival (TDOA) between a source node and beacon nodes and detects Sybil nodes. ZHANG et al. [18] proposed Trust Evaluation Based on Angle Of Arrival (AOA) called TEBA. Since the malicious node can create several identities but it has only one physical location, beacon node detects Sybil identities whose signal phase difference is lower than trust threshold (which is calculated by evaluating trust degree for adjacent sensor nodes).

Butler et al. [19] present an ID assignment protocol based on identity-based cryptography. This protocol tightly allows nodes to acquire an identity. This protocol is used to defend against Sybil attacks, such that it does not allow malicious nodes to acquire multiple identities. In Li et al. [20] a Sybil control distributive algorithm is presented to control the extent of Sybil attack. This algorithm is an admission control mechanism for nodes in a distributed system which should solve computational puzzles periodically. Another algorithm is proposed by Taol and MA [21] to defend against Sybil attack in which sensor node's identity information is validated for random pre-distribution of confidential information.

Zhang et al. [22] propose a light-weight identity certificate method to defend against Sybil attacks. This method exploits one-way key chains and Merkel hash trees to detect Sybil attack. Wang et al. [23] proposed a method which uses Jake channel model and operates in cluster-based WSNs. In this method, errors caused by fading and path loss are considered and other parameters like distance and power received from a data transmitter node are used to detect Sybil attack. Yanget al. [24] proposed an algorithm to defend against Sybil attack in which cluster analysis is used to detect Sybil attacks based on the spatial correlation between signal strength and physical locations. Another algorithm is proposed by WANG et al. [25], in which a cluster-based Merkel hash tree is used to detect Sybil

attack. This algorithm divides the network into clusters in which all nodes sustain a Merkle hash tree which is used to build the keys.

Jan et al. [26] proposed an algorithm for Sybil attack detection based on TDOA localization method, which detects the malicious behavior of the head node and member nodes in a cluster. Sweety and Sejwar [27] proposed a detection scheme for Sybil attack in a centralized clustering-based sensor network. This scheme requires a collaboration of nodes to analyze received signal strengths of neighboring nodes.

Almas et al. [28] proposed an algorithm to detect Sybil nodes in mobile WSNs which employs watchdog nodes and overhear Hello packets exchanges between nodes. Each watchdog node uses a co-presence state diagram to provide partial information. A designated watchdog node aggregates all partial information and uses a detection rule to detect the Sybil nodes.

Rupinder et al. [29] have proposed a mechanism based on evaluating trust values of neighbor nodes to against Sybil attack in wireless sensor networks. The nodes with the trust values less than a threshold value are detected as Sybil nodes. Dhamodharan and Vayanaperumal [30] proposed a message authentication algorithm for detecting Sybil nodes in WSNs. This algorithm uses message authentication and passing procedure for authentication prior to communication. If a sensor node is not authorized by the base station, it cannot communicate with other nodes in the network.

Amuthavalli and Bhuvaneswaran [31] has proposed a Random Password Generation (RPG) algorithm that concentrates on various traffic levels and security during data transmission in WSNs. RPG algorithm generates the routing table which keeps information about nodes. In this algorithm, the intermediate nodes in the route are discovered between the source and destination nodes. The intermediate nodes' information is compared with RPG database during communication, and then the comparison results are used to decide whether these intermediate nodes are Sybil or legal.

Shi et al. [32] have proposed a detection algorithm to detect Sybil attack in cluster-based WSNs. This algorithm takes advantage of the fact that every sensor node has its own unique identity, and puts forward a method for detecting Sybil nodes. Sinha et al. [33] have proposed a location-based Sybil attack detection method using the characteristics of received signal powers of the nodes. Rafeh and Khodadad [34] has proposed a distributed algorithm based on propagating two-hop messages to detect Sybil nodes in WSNs. In this algorithm, each node discovers its two-hop neighbors and the common neighbors between itself and each of its two-hop neighbors by propagating two-hop messages. The number of common neighbors is used to detect Sybil nodes.

A rule-based anomaly detection system is proposed by Panagiotis et al. [35], which relies on an Ultra-Wide Band (UWB) ranging-based detection algorithm. Hu et al. [36] have proposed a one-way key chain ID authentication algorithm to decrease the probability for attackers to lunch Replica and Sybil attacks. They used elliptic curve discrete logarithm problem and node neighbor relationship to authorized nodes.

Jamshidi et al. [37] also proposed a lightweight, dynamic algorithm for detecting Sybil nodes in mobile wireless sensor networks. This algorithm uses watchdog nodes first to label (bit label) mobile nodes based on their movement behaviors, and then detects Sybil nodes according to the labels, during detection phase.

Jamshidi et al. [38] also proposed another algorithm which uses some observer nodes to detect Sybil attacks in mobile wireless sensor networks. This algorithm has two phases: (1) monitoring phase in which observer nodes record the number of meeting occurrences of other nodes in a vector called history, for R time periods. (2) detection phase in which observer nodes cooperate and identify Sybil nodes based on the content of history vectors.

3 System Assumptions and Attack Model

In this section, system assumptions are described first. Then a new model of establishing Sybil attacks in clustered sensor networks is presented.

3.1 System Assumptions

Sensor network includes N sensor nodes which are deployed in a two-dimensional region randomly. Nodes are static and are not aware of their location. The network is homogeneous (all network nodes have equal hardware and software facilities) and each node has a unique identity. Nodes communicate with each other through wireless radio and employ omni-directional broadcast. Moreover, it is assumed that radio range of all nodes is the same and it covers the whole operational environment. Each Node is able to adjust its power. The network is clustered and the number of clusters is C_{opt}. Additionally, it is assumed that the sensor nodes are deployed in an adversary environment, therefore, they might be captured by the adversary. Nodes are not tamper-resistant against interference and if the adversary captures a node, it can access its secret information and reprogram it.

3.2 Attack Model

Here, Sybil attack models proposed in [7, 8] are considered and a new model is proposed to establish this attack in clustered sensor networks.

In the proposed model of the Sybil attack, a malicious node generates $1 < S \leq C_{opt}$ Sybil IDs during each clustering phase (where C_{opt} represents the number of clusters in the current phase) and then uses each generated ID to join a different cluster.

Unlike the Sybil attack model given in [8, 9], we assume that a malicious node would not try to become a cluster head. This is due to the fact that such a decision results in an extraordinary number of cluster heads (each Sybil ID becomes a separate cluster head), and such a situation could be easily detected by the sink node. Hence, in the proposed Sybil attack model, the malicious node does not try to become the cluster head but it tries to join each cluster with one of its Sybil identities.

Thus, the malicious node joins multiple or all clusters simultaneously. Consequently, malicious node affects operations and data of multiple clusters or all clusters. Figure 1 shows an example of such Sybil attack. In this example, the malicious node with identities A, B, C and D is a member of cluster head H1, H4, H2 and H3, respectively. It should be noted that malicious node with different powers transmits join message to cluster heads. Further when the cluster head from the malicious node, join message is sent to the cluster head with more power.

This model of Sybil attack has not yet been proposed, but considering its impact, it is introduced here and an approach is also proposed to defend against it in Sect. 4.

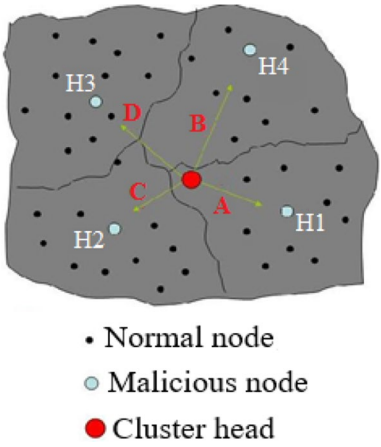


Fig. 1 An example of establishing a new Sybil attack model

4 The Proposed Algorithm to Defend Against the Novel Sybil Attack Model

The main idea of the proposed algorithm is inspired by this issue that all Sybil nodes are located in one geographical location of the network because all Sybil nodes belong to a single device (malicious node). As mentioned before, in cluster establishment phase, cluster head nodes broadcast a message in the network. Other nodes choose one of the cluster heads upon receiving these messages according to some criteria (like distance from cluster heads, their energy) to join to it. Then each node transmits a join message to the selected cluster head.

Each cluster head, stores identity of the node which has transmitted join message and its approximate distance in its *member-table* (Fig. 2). Cluster head node, stores its members' identity in *the member-ID* field and its approximate distance in the *est-distance* field. Cluster head nodes use RSSI to estimate the distance of member nodes from themselves, as given in [11]. Assuming that node i send a packet for node j with a power P_0. In this case, the power of the received packet at the receiver (node j) is obtained as Eq. (1).

$$R_{ij} = \frac{P_0 K}{\left(\hat{d}_{ij}\right)^{\alpha}} \tag{1}$$

In this equation, R_{ij} is RSSI at the receiver node, K is a constant, d_{ij} is the Euclidean distance between two nodes i and j, and finally $\alpha \in \{2, 4\}$ is the signal attenuation factor. Note that due to measurement errors, the distance (d_{ij}) can only be estimated ($\hat{d}_{ij}$) [15].

Fig. 2 The structure of the *member-table* of cluster heads in the proposed algorithm

member_ID	*est_distance*
a	**12**
…	**…**

It should be noted that cluster establishment might occur several times throughout network's lifetime. At each clustering round, cluster heads should estimate their distance from member nodes. After each round of clustering and estimating distances from cluster heads and member nodes, cluster head nodes perform the proposed algorithm for detecting Sybil nodes collaboratively. The proposed algorithm is comprised of two phases: scan phase and examination phase.

4.1 First Phase of the Proposed Algorithm: Scan Phase

In the first phase, each cluster head CH_i, first calculates mean distance from member nodes, as *avg-distance*. Then, it scans its *member-table* and records member nodes with an estimated distance greater than *avg-distance* as suspicious members in a list called suspicious list (SL). Because of the malicious node is simultaneously a member of $S > 1$ clusters, it might at least one of its Sybil identities located far from *avg-distance*s. Therefore, in the scan phase, each cluster head CH_i provides its SL to detect probable malicious nodes in the second phase. SL of cluster head CH_i is shown with SL_{CH_i}. The scan phase is performed by all cluster heads.

4.2 Second Phase of the Proposed Algorithm: Examination Phase

In this phase, each cluster head CH_i selects an Experimental Sample (ES) as a member of $u \in SL_{CH_i}$ which has the maximum distance from the cluster head. ES of CH_i is shown with ES_{CH_i}. Then, each cluster head sends its ES (including member's identity and its approximate distance) to other cluster heads. Thus, each cluster head would have ES of all other cluster heads. Each cluster head stores its ES and other cluster heads in a list called *ES_List*. Each member of this *ES_List* shows ES of a specific cluster head.

It should be noted that *ES_List* have equal values in all cluster heads. Now, cluster head with the maximum distance from its ES, ES_{CH_i}, is selected to run Malicious Node Detection Procedure (MNDP). If ES of more than one cluster heads has maximum distance, then cluster head with lower ID is selected to run the MNDP. Note that, ES with maximum distance is selected first because the probability of its being malicious is higher.

When cluster head CH_i is selected to run the MNDP, it sends a message to its ES,ES_{CH_i}. In this message, the ES_{CH_i} is forced to broadcast a message *Pkt* (with the same power strength as the join message which it had sent in clustering phase), so that other cluster heads receive it. It should be noted that the message broadcasted by ES_{CH_i} is only received by cluster heads which their distance from ES_{CH_i} is smaller than or equal to the distance of the ES_{CH_i} from cluster head CH_i, because the node selected as ES is bounded to broadcast its message with the same power strength as it has previously broadcast the join message.

In this stage, if ES sends its message with different power strength, cluster head CH_i understands and marks it as a malicious node. Each cluster head CH_j, calculates the distance of this node, ES_{CH_i}, from itself upon receiving *Pkt* according to Eq. (1) and stores it in a variable called *dis*. Then, cluster head CH_j scans its *member-table* and selects any member u which its distance is equal to *dis* and sends its ID (u) to cluster head CH_i as Probable Malicious Node (PMN).

If CH_j does not have any members with distance *dis*, then it does not respond to CH_i. if CH_i receives a response (means, PMNs) from at least T_{min} other clusters, then it stores the list of received PMNs from other cluster heads along with its ES, ES_{CH_i}, as Sybil malicious node in *Sybil-List*. Otherwise, no malicious node is detected at this stage.

According to "positioning with three points" [39], T_{min} should be selected with at least value of 2 such that detection accuracy is high. When this stage is finished (whether a malicious node is detected or not), cluster head CH_i eliminates ES_{CH_i} from its SL and selects another member of SL (if there exists one) as the new ES (ES'_{CH_i}) and sends information of the new ES to other cluster heads. Other cluster heads replace new ES (ES'_{CH_i}) with the old ES (ES_{CH_i}) in their *ES-List*, upon receiving this message.

Thus, *ES-List* of all cluster heads is updated for cluster head CH_i. At this stage, if SL of cluster head CH_i is empty, -1 is transmitted to other cluster heads as the ES_{CH_i}. The value -1 indicates that SL of CH_i has become empty and from now on, CH_i would not run MNDP. At this time, another cluster head which has ES with maximum distance (according to *ES-Lists*) is selected to run MNDP.

Examination phase is continued until all members of *ES-List* of all cluster heads become empty. At this time, execution of the proposed algorithm is finished and each cluster head notifies nodes of its *Sybil-List* as malicious nodes to the base station or other nodes.

4.3 False Detection in the Proposed Algorithm

In some cases, normal nodes (non-malicious nodes) might be detected as Sybil malicious nodes. Additionally, in some cases, they might not be detected as malicious nodes. Consider the following scenarios:

Scenario 1 consider a network which is clustered as Fig. 3. Here, we have three cluster heads, H1, H2, and H3. There is also a malicious node in the network which joins the cluster heads H1, H2, and H3 with identities *S*1, *S*2, and *S*3, respectively. In clustering phase, cluster heads H1, H2 and H3 estimate their approximate distance from *S*1, *S*2 and *S*3 as 11, 18 and 5, respectively.

In the first phase (scan phase) of the proposed algorithm, cluster heads build their SL. For example, SL of H2 contains SL = {*S*2, *u*, *v*} because their approximate distance is more than *avg-distance* = 10.75 (average distance of H2 from its members). In the second phase,

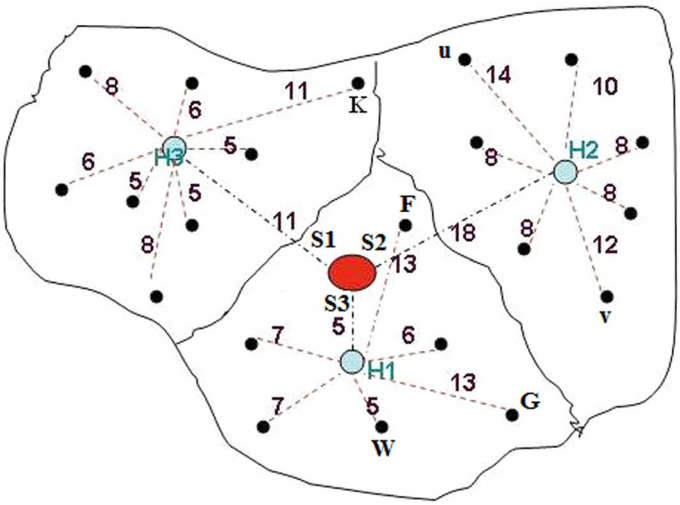


Fig. 3 A possible clustering of the network and incorrect detection of the proposed algorithm

each cluster head selects the member with maximum distance as its SL. Then cluster heads send their *ES*s to each other. In this example, member nodes of *S*1 (or *K*), *S*2 and *F* (or *G*) are selected as ES for cluster heads H1, H2, and H3, respectively, and they are recorded in *ES-List*s. Then, cluster head H2 is selected to run MNDP, because it has an ES with maximum distance. Cluster head H2 sends a message to its experimental sample, *S*2, to broadcast a message in the network. S2 broadcasts a message in the network with the same power strength which it had sent a join message for H2.

This way, cluster heads H1 and H3 also receive a message of *S*2 and calculate their approximate distance from this node. Cluster heads H1 and H3 estimate their distance from *S*2, $dis = 11$ and $dis = 5$, respectively. Cluster head H1 scans its *member-table* and finds out that distance from *K* and *S*1 is also 11, thus it selects these members as PMNs (response to the H2). Cluster head H3 also selects *W* and *S*3 as PMNs (response to the H2). Assuming that $T_{min} = 2$, cluster head H2 records *S*2 (its own ES), *K*, *S*1, *W* and *S*3 as Sybil malicious nodes in its *Sybil-List*. Thus, nodes *W* and *K* are incorrectly detected as Sybil malicious nodes. It should be noted that in this scenario, if $T_{min} > 2$, no malicious node is detected. In general, T_{min} should be smaller than the number of clusters in the network (C_{opt}).

Scenario 2 as mentioned, in some conditions, malicious node might not be detected. Figure 4 shows a possible clustering state. In this case, the malicious node is so close to all cluster heads that cannot be placed in SL of any of the cluster heads. For example, it can be seen in Fig. 4 that average distance of members from cluster heads H1, H2, and H3 is 7.16, 19.625, and 9.1, while the distance of the malicious node from these cluster heads is 5, 9, and 8, respectively. Thus the malicious node cannot be placed in any of the cluster heads' SL.

In this case, the proposed algorithm cannot detect the malicious node. Such case might rarely occur, especially when the malicious node joins more clusters. On the contrary, if the malicious node joins a smaller number of clusters, for instance, $S = 2$ or $S = 3$, the probability of being detected by the proposed algorithm decreases. Because the malicious node can join clusters which are closest, thus the malicious node will not be placed in SL of cluster

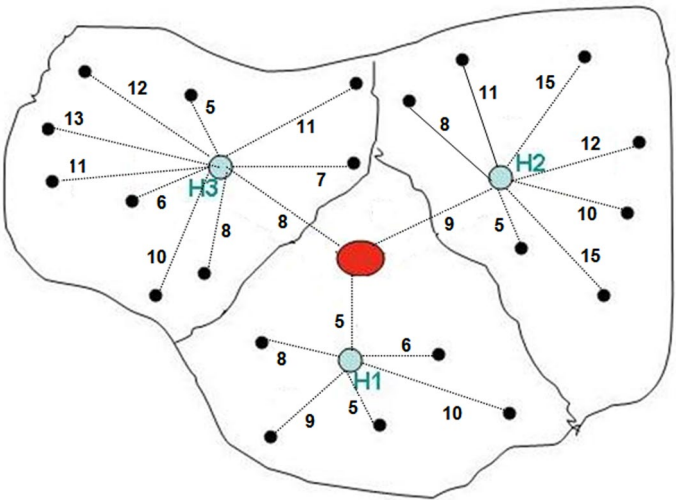


Fig. 4 A possible case of network clustering where malicious node is not detected by the proposed algorithm

heads (malicious node acts intelligently). But if the malicious node joins more clusters ($S>3$), probability that its distance from all these $S>3$ clusters is very low which caused the probability of not being placed in SL of any cluster head decreases. Indeed, it should be noted that as the number of clusters to which the malicious node has simultaneously joined is higher, its impact on the network would be more. In fact, the weakest case of Sybil attack establishment is when the malicious node joins two clusters simultaneously ($S=2$).

5 Performance Evaluation and Simulation Results

In this section, we first evaluate the overhead of the proposed algorithm in terms of memory, communication, and computation. Then we simulate our proposed algorithm and evaluate its performance through some experiments. The performance of the proposed algorithm also is compared with other existing algorithms in terms of true detection and false detection rates, on average case.

5.1 Overheads of the Proposed Algorithm

Memory overhead in the proposed algorithm, no memory overhead is imposed on nodes which are a member of the cluster. But each cluster head requires a $2m$ space to store *member-table*, a $m/2$ space (in average) to store SL and a C_{opt} space to store *ES-List*. Here, m is the average number of cluster members, obtained using Eq. (2).

$$m = \frac{N + M(S-1)}{C_{opt}} \tag{2}$$

Here M is the number of malicious nodes in the network, S is the number of Sybil identities propagate by each malicious node and $N+M(S-1)$ is the total number of nodes (considering Sybil nodes) in the network. In addition, since SL is the list of members whose distances from a cluster head is greater than the average distances of cluster heads from their members, the number of SL members, or better say the number of ESs of each cluster head, is equal to |SL| which could be obtained using Eq. (3).

$$|SL| = \frac{m}{2} = \frac{N + M(S-1)}{2 \times C_{opt}} \tag{3}$$

Therefore, the memory overhead of the proposed algorithm for nodes which are a member of the cluster is zero and for each cluster head node is $O\left(\frac{N}{C_{opt}}\right)$. Table 1 compares memory overhead of the proposed algorithm with other existing algorithms.

Communication overhead considering energy constraints of the sensor nodes, energy consumed by the proposed algorithms for sensor networks is an important issue. Since sending packets consumes more energy than processing packets or receiving packets, thus calculating the number of sent packets due to employing a specific algorithm is an important measure for evaluating the performance of the algorithm. Since the novel attack model is considered for clustered networks, on the other hand, the proposed algorithm is executed after clustering, thus communication overhead of clustering phase is not caused by the proposed algorithm. After clustering, each cluster head creates its SL. In this stage, no communication overhead is imposed on the nodes. Each cluster head obtains its ES. There is no overhead in this stage, also. Then all cluster heads broadcast their ES to each other. In

Table 1 Comparison of the proposed algorithm and the other algorithms in terms of memory overhead and communication overhead

Algorithm	Memory overhead (per node)	Communication overhead (per network)
Chen et al. [8]	$O(d \times q)$	$O(N)$
Ssu et al. [13]	$O(d^2)$	$O(N \times d^2)$
Misra and Myneni [15]	$O(d \times q)$	$O(N)$
Almas et al. [28]	$O(N^2)$	$O(N^2)$
Dhamodharan and Vayanaperumal [30]	$O(d)$	$O(N^2)$
Rafeh and Khodadadi [34]	$O(d^2)$	$O(N \times d)$
Jamshidi et al. [37]	$O\left(R \times N \times \left\lceil \log_2^q \right\rceil\right)$	$O(R \times q)$
Jamshidi et al. [38]	$O(N)$	$O(R \times d \times q^2)$
Proposed algorithm	$O\left(\frac{N}{C_{opt}}\right)$	$O(N)$

N number of total sensor nodes, d number of neighbors on average case, R number of rounds in which the algorithm should be executed, q number of observer nodes which are responsible for detecting Sybil attack, C_{opt} number of clusters in the network

this stage, since all cluster heads broadcast their ES, communication overhead would be C_{opt}. each cluster head constructs *ES-List* considering its ES and the other cluster heads' ES which has no communication overhead. Then the following operations should be performed for each ES:

1. Without any communication overhead, a proper cluster head, like CH_i, is selected to run the MNDP. Then CH_i sends a message to its ES and it also broadcasts a message. Therefore, two packets are transmitted in the network. Thus, for each ES, two packets are broadcasted in the network. The total number of ESs in the network could be obtained through Eq. (4):

$$|ES| = C_{opt} \times \frac{N + M(S-1)}{2 \times C_{opt}} = \frac{N + M(S-1)}{2} \tag{4}$$

2. Assuming that the packet sent from ES, reaches to $\frac{C_{opt}-1}{2}$ other cluster heads (as receiver cluster heads), in average, each receiver cluster head scans its *member-table* and sends a message containing a list of PMNs to the CH_i, if required. If the ES is a Sybil node (belong to a malicious node), and in addition to CH_i, it is simultaneously a member of a receiver cluster head, like CH_j, then CH_j sends a message to the CH_i. Indeed, despite no malicious node has joined receiver cluster heads, they might have a member in their *member-table* which its distance from the receiver is equal to its distance from the ES. According to scenario 1, in this case, receiver cluster head sends a message to the initiator cluster head. But since this case rarely occurs, it is neglected.
3. The probability of a node to be Sybil is $\frac{M \times S}{N + M(S-1)}$. On the other hand, a Sybil node which is a member of a cluster head, like CH_i, will be placed in SL of cluster head CH_i only when it is located further from the average distance of cluster head from members so that it is selected as an ES. Otherwise, this Sybil node will not be placed in CH_i's SL. Therefore, considering intelligent behavioral of the malicious node in selecting cluster

heads, it is probable that a Sybil node which is a member of a cluster head is not located in SL. Here, it is assumed that half of Sybil nodes which are members of cluster heads will be placed in SLs. Therefore, the probability that the ES is a malicious node (E_m) would be determined by the Eq. (5):

$$E_m = \frac{\frac{M\times S}{2}}{N + M(S-1)} \times \frac{N + M(S-1)}{2 \times C_{opt}} = \frac{M \times S}{(N + M(S-1))(2 \times C_{out})} \tag{5}$$

The probability that the malicious node E_m is also being a member of the receiver cluster head, referred to as E_c, can be calculated using Eq. (6):

$$E_c = \frac{S-1}{C_{opt} - 1} \tag{6}$$

In this case, means ES is a malicious node, $\frac{C_{opt}-1}{2} \times E_c \times 1$ packets are transmitted to the CH_i by the receiver cluster heads.

4. Then the CH_i determines whether it has detected a malicious Sybil node or not (without any communication overhead). In the next stage, CH_i eliminates the experimental sample selected from SL and sends the new ES in a message in order to other cluster heads update their ES-List. Therefore, for any ES, a message is also broadcasted in the network.

All the above mentioned stages will be repeated for all experimental samples, |ES|. Thus, the total communication overhead for executing the proposed algorithm is obtained using Eq. (7) given below.

$$\begin{aligned}
Comm_{\cos t} &= C_{opt} + |ES| \times 2 + |ES| \times \left(E_m \times E_c \times \frac{C_{opt-1}}{2} \times 1 \right) + |ES| \times 1 \\
&\Rightarrow Comm_{\cos t} = C_{opt} + |ES| \times (2+1) + |ES| \times \left(E_m \times \frac{S-1}{C_{opt}-1} \times \frac{C_{opt}-1}{2} \times 1 \right) \\
&\Rightarrow Comm_{\cos t} = C_{opt} + |ES| \times (2+1) + |ES| \times \left(E_m \times \frac{S-1}{2} \right) \\
&\Rightarrow Comm_{\cos t} = C_{opt} + |ES| \times \left[2 + 1 + \left(E_m \times \frac{S-1}{2} \right) \right] \\
&\Rightarrow Comm_{\cos t} = C_{opt} + \frac{N + M(S-1)}{2} \times \left[2 + 1 + \left(\frac{M \times S}{(N + M(S-1))(2 \times C_{opt})} \times \frac{S-1}{2} \right) \right]
\end{aligned} \tag{7}$$

A deeper look into the Eq. (7) reveals that the communication overhead of the proposed algorithm is in the order of $O(N)$. This is in coincidence with the results presented in Sect. 5.2, where it is indicated that among the parameters of the proposed algorithm, N has the most impact on the communication overhead. Table 1 also compares communication overhead of the proposed algorithm with other existing algorithms.

Computation overhead considering the above considerations, no computation overhead is imposed on member nodes. But some computation overhead is imposed on cluster head nodes. Each cluster head node requires $\left(\frac{N}{2} \times C_{opt} \right)$ time for each ES to determine which cluster head is proper to run the MNDP. Moreover, each cluster head requires $\left(\frac{N}{2} \times \frac{N}{C_{opt}} \right)$ time to scan its *member-table* to find PMNs. Therefore, computation

overhead of the proposed algorithm for member nodes is zero and for cluster head, it is of order $O\left(\frac{N^2}{C_{opt}}\right)$.

5.2 Simulation Results

The proposed algorithm was simulated using J-SIM simulator [40] and a number of experiments are conducted to evaluate its performance. The obtained results also compared with the algorithms proposed in [8, 13, 15, 28, 30, 34, 37, 38]. Four performance metrics were considered:

- *True Detection Rate (TDR)* is the percentage of Sybil nodes which are detected by a security algorithm.
- *False Detection Rate (FDR)* is the percentage of normal nodes erroneously marked as Sybil nodes by a security algorithm.
- *Communication Overhead* is the number of packets transmitted by all nodes for execution of a specific algorithm in the network. We have computed the communication overhead in both the mathematical, according to Eq. (6), and the simulation cases.
- *Average TDR/FDR* in order to calculate this measure, the proposed algorithm is performed for all cases (changing different parameters, like the total number of nodes, number of malicious nodes, and etc.) and possible conditions and the obtained results are averaged.

Simulation parameters are shown in Table 2. All these parameters can affect the performance of the proposed algorithm; thus we have tried to evaluate such effects in the simulations.

Experiment 1 In this experiment, parameters are adjusted as $N=100$, $M=1$, $C_{opt}=7$, and S vary from 2 to 7 and the results in terms of TDR, FDR and communication overhead are represented in Figs. 5, 6, and Table 3, respectively. As can be seen in the results (Fig. 5), as malicious node joins more clusters (larger S), TDR of the proposed algorithm increases. For example, if the malicious node joins 2 clusters, the detection rate is 86% and if it joins more than 3 clusters, TDR is higher than 99%. There are two reasons for this issue. First, in the proposed algorithm, cluster heads collaborate to detect the malicious node, thus the more clusters the malicious node joins, detection rate increases. The second reason, as mentioned in scenario 2, this is because if the malicious node joins fewer clusters, the probability of its being detected by the proposed algorithm decreases. Because the

Table 2 Simulation parameters

Parameter	Values
Network size	100×100 m^2
Network topology	Random
Total nodes	$N=100$–500
Number of malicious nodes	$M=1$–5
Number of Sybil identities propagated by each malicious node	$S=2$–7
Number of clusters in the network	$C_{opt}=2$–7
Threshold	$T_{min}=1$–5
Each simulation iteration	100

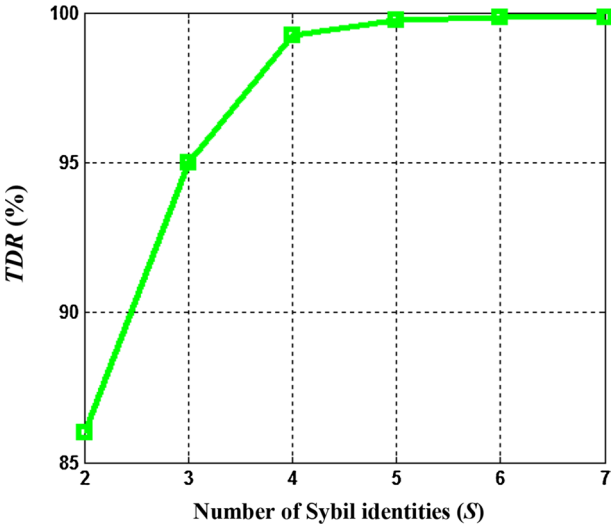


Fig. 5 Effect of number of Sybil identities, *S*, on true detection rate of the proposed algorithm

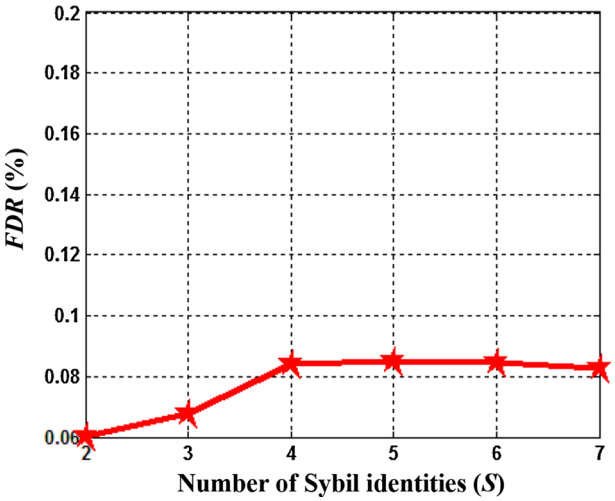


Fig. 6 Effect of number of Sybil identities, *S*, on false detection rate of the proposed algorithm

Table 3 Effect of number of Sybil identities, *S*, on communication overhead of the proposed algorithm

	$S=2$	$S=3$	$S=4$	$S=5$	$S=6$	$S=7$
Mathematical results	158	160	161	163	165	167
Simulation results	143	145	147	149	152	155

malicious node can choose clusters to join, which are located in minimum distance, then the malicious node will not be placed in SL of any cluster head. But if the malicious node joins more clusters simultaneously, the probability that its distance with these cluster heads is so small which it is not located in any cluster head's SL decreases. Thus, its detection probability increases.

In addition, results of this experiment in Fig. 6 show that FDR of the proposed algorithm for $S=2$ and $S=3$ is 0.06% and 0.07% respectively and for $S>3$, it is about 0.085%. As explained in Scenario 1, in each execution of the detection algorithm for an ES, some legal nodes might be incorrectly detected as Sybil nodes. Therefore, by increasing S (malicious node joins to more clusters), the number of ESs increase, thus FDR also increases.

Table 3 shows results of this experiment in terms of communication overhead for both simulation and mathematical (Eq. 7) cases. Mathematical results showed that by increasing S, communication overhead increases 1 or 2 units and simulation results also showed by increasing S, communication overhead increases 2 or 3 units. The result of this experiment shows that changing parameter S does not affect communication overhead of the proposed algorithm significantly.

Experiment 2 In this experiment, parameters are adjusted as $S=5$, $M=1$, $C_{opt}=7$, and $T_{min}=2$ and N vary from 100 to 500 (with step 100) and its impact on the performance of the proposed algorithm is evaluated. Figure 7 shows TDR, Fig. 8 shows FDR and Table 4 shows communication overhead of this experiment.

Experiment results show that increasing the number of nodes increases all TDR, FDR and communication overhead of the proposed algorithm, because increasing number of nodes increases the number of ESs, therefore the number of execution of MNDP also increases. This caused TDR and FDR increases slightly but communication overhead increases more sensible. Results of this experiments show that for different values of N, TDR is greater than or equal to 99.8% and FDR is smaller than or equal to 0.085%. Also, Mathematical/simulation results show that by increasing N (100 units), communication overhead increases 150/143 units.

Experiment 3 In this experiment, parameters are adjusted as $N=300$, $S=5$, $C_{opt}=7$, and $T_{min}=2$ and M vary from 1 to 5 and its impact on the performance of the proposed algorithm is evaluated. It is clear that increasing number of malicious nodes decreases TDR of the proposed algorithm and every algorithm. Results of this experiment in Fig. 9 show that by increasing number of malicious nodes in the network, TDR decreases. But this reduction in the proposed algorithm is negligible. Figure 9 shows that for varying M from 1 to 5,

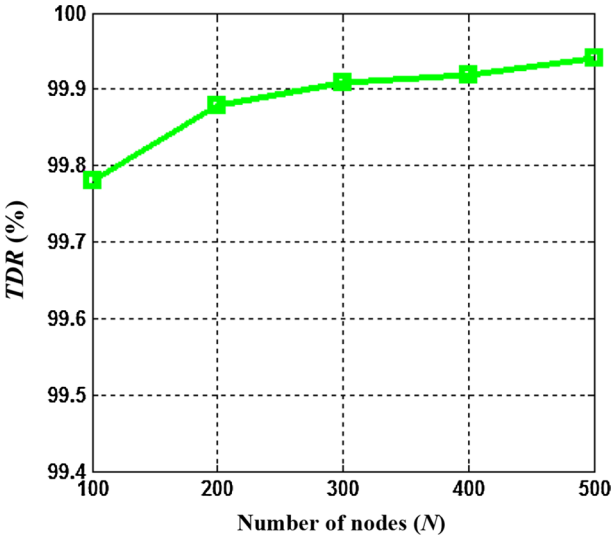


Fig. 7 Effect of number of total nodes, N, on true detection rate of the proposed algorithm

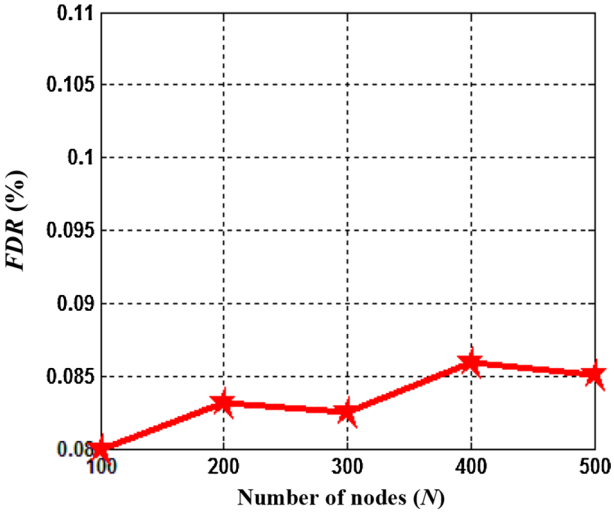


Fig. 8 Effect of number of total nodes, *N*, on false detection rate of the proposed algorithm

Table 4 Effect of number of total nodes, *N*, on communication overhead of the proposed algorithm

	$N=100$	$N=200$	$N=300$	$N=400$	$N=500$
Mathematical results	163	313	463	613	763
Simulation results	149	291	434	577	721

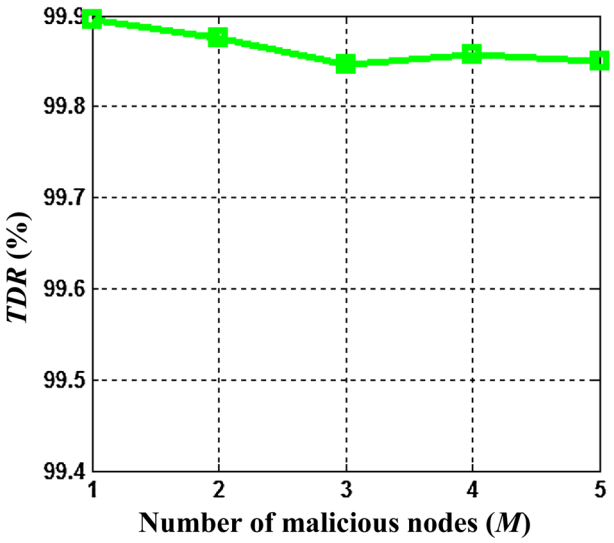


Fig. 9 Effect of number of malicious nodes, *M*, on true detection rate of the proposed algorithm

TDR of the proposed algorithm is higher than 99.85%. it should be noted that it is difficult and costly for the adversary to capture a lot of legal nodes and reprogram them as malicious nodes. On the other hand, as malicious nodes in the network increase, the network manager will detect them faster and takes required actions.

Table 5 Effect of number of malicious nodes, M, on communication overhead of the proposed algorithm

	$M=1$	$M=2$	$M=3$	$M=4$	$M=5$
Mathematical results	463	470	477	483	490
Simulation results	435	441	449	459	468

Moreover, results of this experiment shown in Fig. 10 and Table 5 show that increasing number of malicious nodes increases FDR and communication overhead, because of increasing number of malicious nodes in the network, increases the number of ESs, therefore MNDP is executed in more rounds. It is clear that number of execution of MNDP, increases communication overhead of the proposed algorithm and on the other hand according to scenario 1, increases FDR; for example, when there is only one malicious node in the network, $M=1$, FDR is about 0.1% and when $M=5$, FDR is about 0.4%. Table 6 also shows that by increasing parameter M by one, communication overhead increases about {6, 7} units in mathematical results and about {6, 8, 9, 10} units in simulation results.

Experiment 4 In this experiment, parameters are adjusted as $N=300$, $M=1$, $C_{opt}=7$, and $S=6$ and T_{min} varies from 1 to 5 and its effect on the performance of the proposed algorithm is evaluated. Figures 11, 12 and Table 6 show results of this experiment in terms of TDR, FDR and communication overhead, respectively. As mentioned, T_{min} determines when a cluster head, that runs MNDP, should marking the malicious nodes. If a cluster head receives a response from at least T_{min} other cluster heads (list of PMNs), it stores

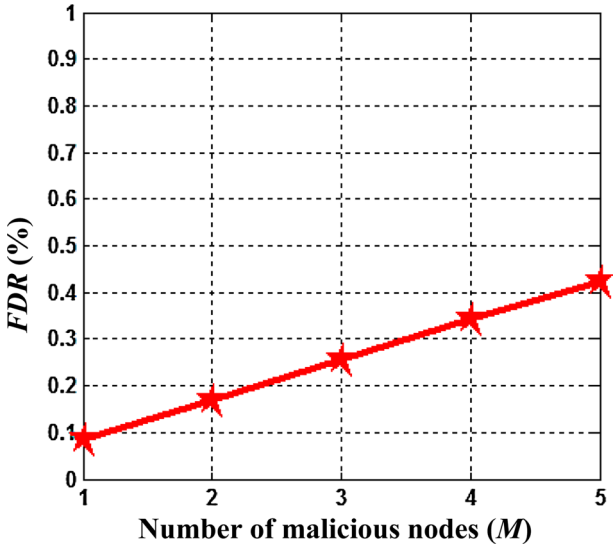


Fig. 10 Effect of number of malicious nodes, M, on true detection rate of the proposed algorithm

Table 6 Effect of threshold, T_{min}, on communication overhead of the proposed algorithm

	$T_{min}=1$	$T_{min}=2$	$T_{min}=3$	$T_{min}=4$	$T_{min}=5$
Mathematical results	465	465	465	465	465
Simulation results	436	435	435	435	436

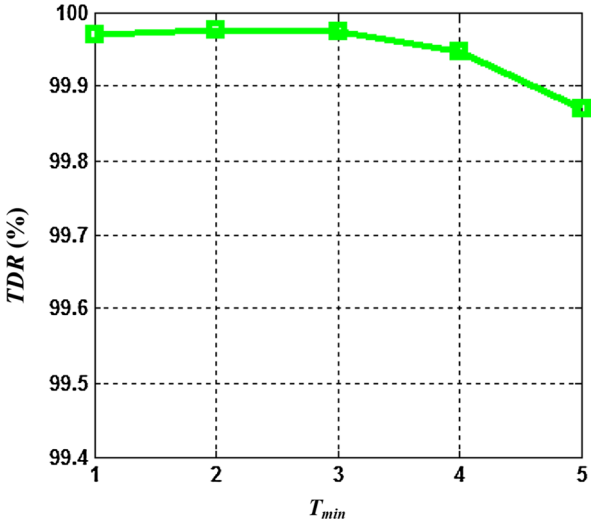


Fig. 11 Effect of threshold, T_{min}, on true detection rate of the proposed algorithm

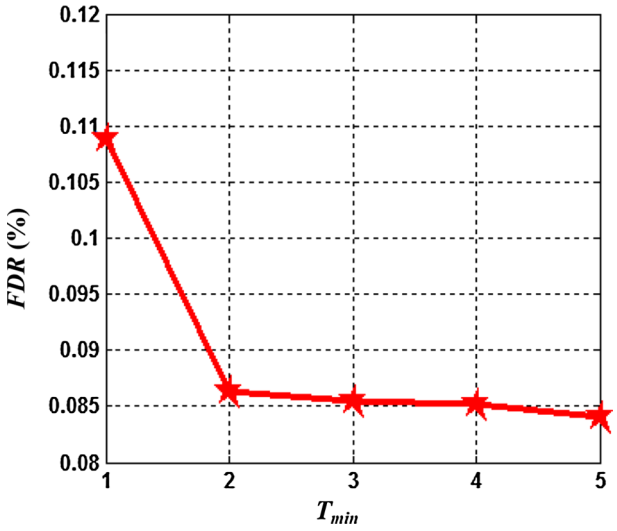


Fig. 12 Effect of threshold, T_{min}, on false detection rate of the proposed algorithm

the list of malicious nodes in its *Sybil-List*. Therefore, increasing T_{min} decreases TDR and FDR. But changing T_{min} does not affect communication overhead, because changing this parameter does not affect ESs, therefore it will not affect communication overhead. Figure 9 shows that for $T_{min} < 5$, TDR is above 99.95% and for $T_{min} > 1$, this measure is about 0.085%. In addition, experiment results in Table 6 show that changing T_{min} affects communication overhead, neither in simulation results nor in mathematical results.

Experiment 5 In this experiment, parameters are adjusted as $N = 200$, $S = 3$, $T_{min} = 2$, and $M = 1$ and C_{opt} vary from 3 to 7 and its impact on the performance of the proposed algorithm is evaluated and the results are given in Figs. 13, 14 and Table 7. In this experiment, a limited case of Sybil attack is considered ($S = 3$). Results of this experiment in Fig. 13 show that by increasing number of cluster heads in the network, TDR decreases. The reason was explained in scenario 2. By increasing number of cluster heads in the network,

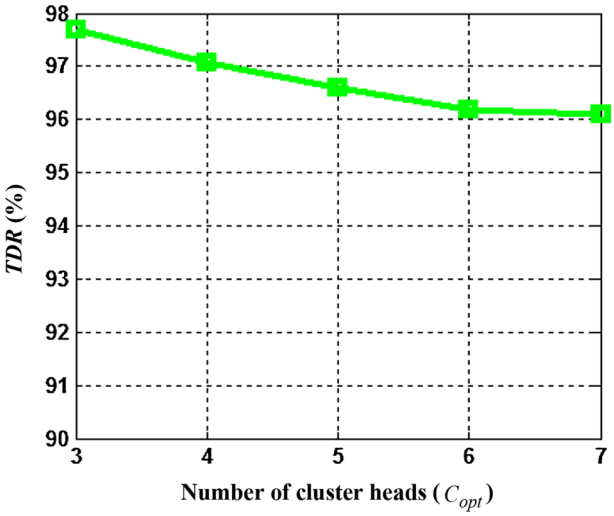


Fig. 13 Effect of number of cluster heads, C_{opt}, on true detection rate of the proposed algorithm

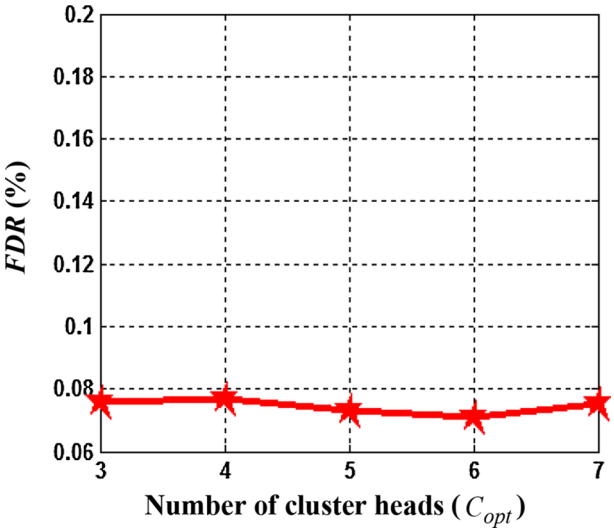


Fig. 14 Effect of number of cluster heads, C_{opt}, on false detection rate of the proposed algorithm

Table 7 Effect of number of cluster heads,C_{opt}, on communication overhead of the proposed algorithm

	$C_{opt} = 3$	$C_{opt} = 4$	$C_{opt} = 5$	$C_{opt} = 6$	$C_{opt} = 7$
Mathematical results	306	307	308	309	310
Simulation results	288	288	288	288	289

probability that the malicious node has three cluster heads in its proximity increases. A malicious node can remain safe from being detected by the proposed algorithm by joining three cluster heads in its proximity. However, TDR of the proposed algorithm for different values of C_{opt} is more than 96% which is an acceptable result.

Moreover, the result of this experiment in Fig. 14 and Table 7 show that by varying C_{opt}, FDR and communication overhead do not vary. For different values of C_{opt}, FDR is less than 0.08% and communication overhead is about 288 packets. According to Eq. (4), varying number of cluster heads does not affect the number of ESs and therefore it would not affect the number of execution of MNDP. Therefore, varying C_{opt} does not affect these two measures.

Experiment 6 In this experiment, the performance of the proposed algorithm is compared to that of other existing algorithms [13, 28, 34, 37, 38] in terms of TDR and FDR. These algorithms are from two categories:

1. Algorithms given in [13, 34] as well as the proposed algorithm which are suited for stationary WSNs.
2. Algorithms given in [28, 37, 38] which are suited for mobile WSNs.

Table 8 shows the adjusted parameters for each of these algorithms.

In the first section of this experiment, the effect of the parameter S (number of Sybil IDs propagated by each malicious node) on the performance of the algorithms has been evaluated. Results have been presented in terms of TDR (Fig. 15) and FDR (Fig. 16). For all algorithms, the parameters M and N were set to 1 and 100 respectively. Results show that the proposed algorithm has met at least 25% improvement over the other algorithms in terms of the TDR criterion. Likewise, in terms of the FDR, the proposed algorithm meets desired performance similar to the ones presented in [28, 37] with the rate of less than 0.5%.

In the second section of this experiment, we have tried to evaluate the effect of the parameter N (number of nodes) on the performance of the proposed algorithm in comparison to the other algorithms. The results of this evaluation have been presented in terms of TDR (Fig. 17) and FDR (Fig. 18). For all algorithms, we have set $M=1$ and $S=5$. Simulation results has implied that the proposed algorithm is at least 3% better than the other ones in terms of the TDR. Moreover, the FDR of the proposed algorithm as well as the FDR of the algorithms presented in [28, 37] are less than 0.5% which is the best among the compared algorithms.

Third section of the experiment compares the performance of the algorithms with the other algorithms for different values of the parameter M (number of malicious nodes in the network). Figures 19 and 20 illustrate the results in terms of TDR and FDR, respectively. For all the algorithms we set $S=5$ and $N=100$. Results indicate that the proposed

Table 8 Adjusted parameters for the algorithms being compared to the proposed algorithm

Algorithm	Parameters
Ssu et al. [13]	$r=10$ m, $T_s=0.7$
Almas et al. [28]	$r=10$ m, $q=4$, $T_s=1$, $R=100$
Rafeh and Khodadadi [34]	$r=10$ m
Jamshidi et al. [37]	$r=10$ m, $q=4$, $T_s=2$, $R=100$
Jamshidi et al. [38]	$r=10$ m, $q=10$, $Tmim=S$, $R=100$
Proposed algorithm	$C_{opt}=7$, $T_{min}=2$

r Transmission range of sensor nodes, T_s *and* T_{min} security thresholds, R number of rounds, q number of observer/watchdog nodes in the network C_{opt} number of clusters in the network

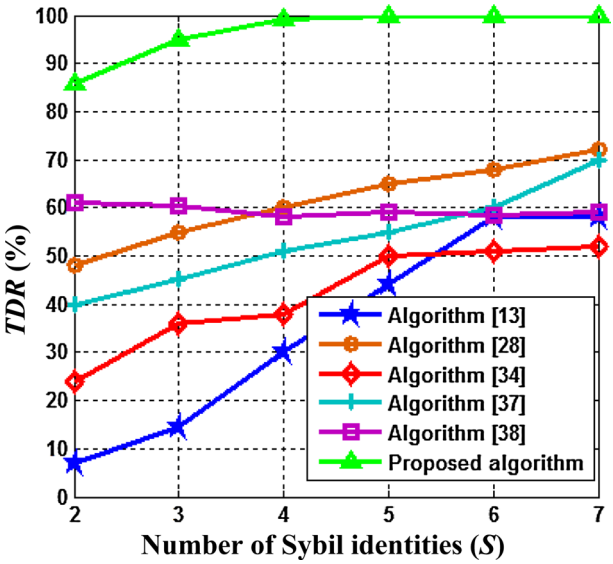


Fig. 15 Impact of the parameter S on the performance of the proposed algorithm in comparison to other algorithms in terms of TDR

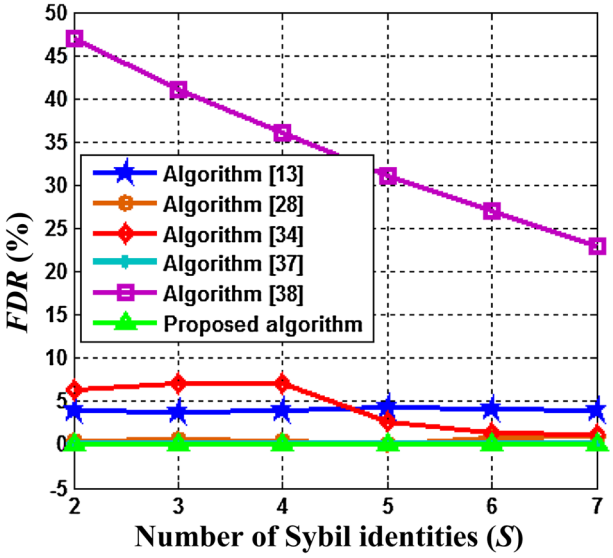


Fig. 16 Impact of the parameter S on the performance of the proposed algorithm in comparison to other algorithms in terms of FDR

algorithm has a significantly better TDR over the other mentioned algorithms with the rate of at least 4%. Like previous experiments, the FDR of the proposed algorithm is equal to that of algorithms given in [28, 37] which is an acceptable result.

Algorithms given in [13, 34] which are suited to static sensor networks, assume that the number of propagated Sybil IDs is more than the average number of neighboring nodes in the normal case. As a result, if an adversary node establishes fewer Sybil IDs, then it would not be easily detected by these algorithms. On the other hand, in the proposed algorithm, even if a malicious node propagates only two Sybil IDs ($S=2$), cluster heads would be able to detect it using RSSI.

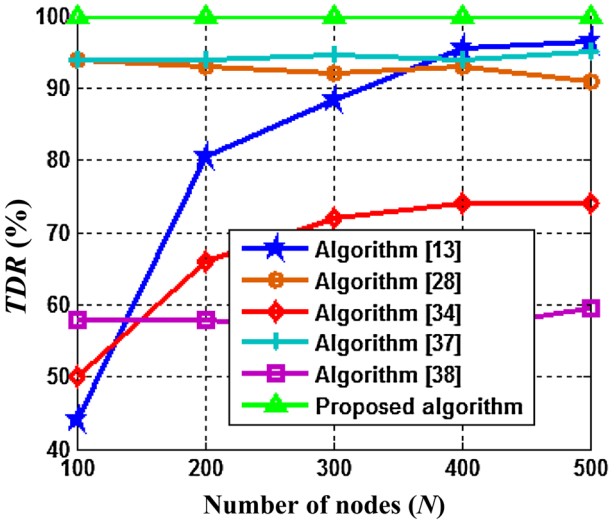


Fig. 17 Impact of the parameter *N* on the performance of the proposed algorithm in comparison to other algorithms in terms of TDR

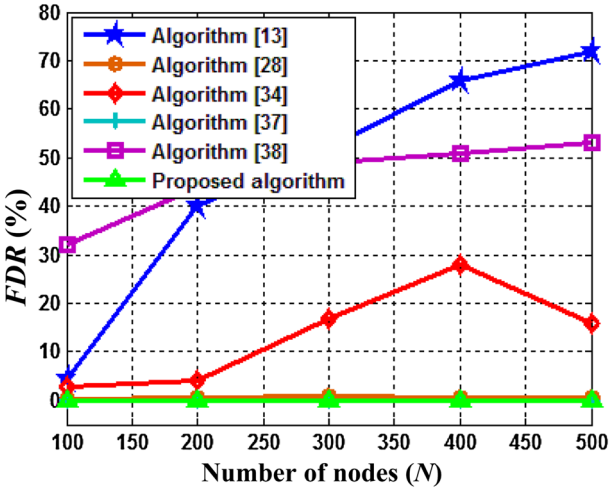


Fig. 18 Impact of the parameter *N* on the performance of the proposed algorithm in comparison to other algorithms in terms of FDR

Algorithms presented in [28, 37, 38] are suited for mobile sensor networks. The detection process in all these algorithms takes place in two phases: (1) monitoring and (2) detection. These algorithms are executed by a small fraction of special sensor nodes, referred to as Watchdog nodes. In the first phase, watchdog nodes monitor the activities of other nodes for *R* periods of time and collect all the required information in their own buffer. In the next phase, watchdog nodes begin to detect Sybil IDs, either solely or in cooperation with each

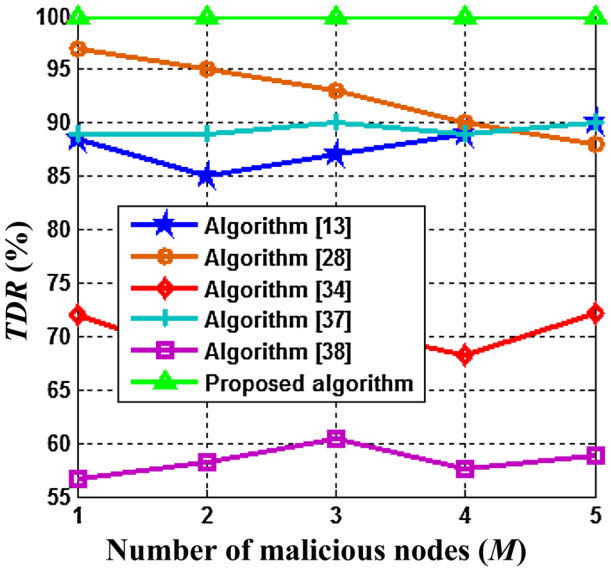


Fig. 19 Impact of the parameter M on the performance of the proposed algorithm in comparison to other algorithms in terms of TDR

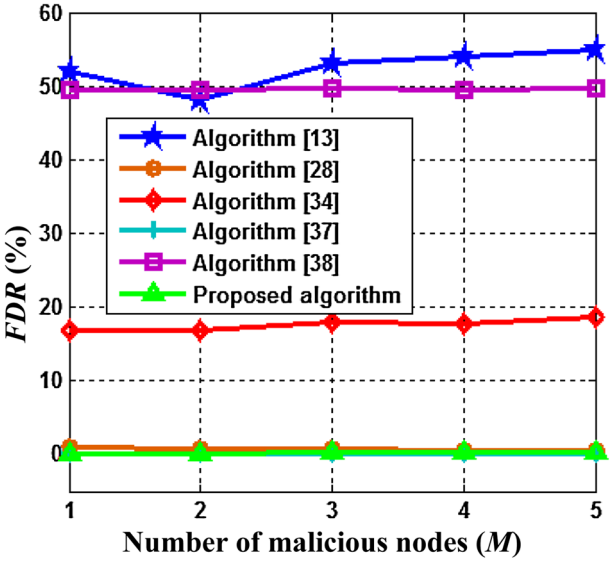


Fig. 20 Impact of the parameter M on the performance of the proposed algorithm in comparison to other algorithms in terms of FDR

other. In these algorithms, the parameter R needs to be set to a high enough value (e.g., $R > 100$) in order for watchdog nodes to collect enough information for the detection phase. On the other hand, performances of the mentioned algorithms—similar to the ones in [13, 34]—depend heavily on the parameter S. Thus, in cases where malicious nodes propagate only a few Sybil IDs, these algorithms have low rate of TDR.

Experiment 8 Since the considered attack model is different from other algorithms, the performance of the proposed algorithm cannot be compared with other algorithms in terms of TDR and FDR (for varying different parameters). But this comparison can be done in the

Table 9 Comparative analysis of various Sybil attack detection algorithms

Algorithm	Network type?	Location information?	Specific topology?	Centralized?	Mechanism?
Chen et al. [8]	Stationary	Yes	Yes (cluster-based)	Yes	Uses RSSI and clustering
Ssu et al. [13]	Stationary	Yes	No	No	Uses neighboring information and discovers common neighbors
Misra and Myneni [15]	Stationary	Yes	No	No	Uses RSSI to discover the location of nodes
Almas et al. [28]	Mobile	Yes	No	No	Employs watchdog nodes and movement information
Dhamodharan and Vayanaperumal [30]	Stationary	Yes	No	Yes	Combines "compare and match-position verification method" with "message authentication and passing"
Rafeh and Khodadadi [34]	Stationary	Yes	No	No	Uses neighboring information and broadcast two-hop packets
Jamshidi et al. [37]	Mobile	Yes	No	No	uses a distributed labeling mechanism to assign a bit label to nodes based on their movement
Jamshidi et al. [38]	Mobile	No	No	No	Employs observer nodes and neighboring information
Proposed Algorithm	Stationary	Yes	Yes (cluster-based)	No	Uses RSSI and positioning using three points

Table 10 Comparison of performance of the proposed algorithm and other algorithms in terms of average TDR/FDR

Algorithm	Average TDR (%)	Average FDR (%)
Chen et al. [8]	92	2
Ssu et al. [13]	99	5
Misra and Myneni [15]	98	7
Almas et al. [28]	98	1
Dhamodharan and Vayanaperumal [30]	80	0
Rafeh and Khodadadi [34]	98	2.5
Jamshidi et al. [37]	94	0
Jamshidi et al. [38]	99	2
Proposed algorithm	99.8	0.08

average case. Thus, here we first present a comparative analysis of various Sybil attack detection algorithms in Table 9, then average TDR and FDR of the proposed algorithm and other algorithms are compared. Table 10 shows the comparison result. As can be seen in the results, the proposed algorithm with an average TDR of 99.8% outperforms other algorithms. Also, the average FDR of the proposed algorithm is 0.08% which indicates its performance is desirable.

6 Conclusion

In this paper, a new Sybil attack model in cluster-based WSNs like LEACH is proposed. Then an algorithm based on RSSI and collaboration of cluster head nodes is proposed to defend against this new attack model. The proposed algorithm was simulated and its performance was evaluated in terms of TDR, FDR, and communication overhead. Experiment results showed that the proposed algorithm imposes less communication overhead on the network and can detect 99.8% of Sybil nodes with 0.08% FDR (in average). In addition, the performance of the proposed algorithm in terms of average TDR and average FDR was also compared with other existing algorithms, which indicates that the proposed algorithm outperforms other algorithms.

References

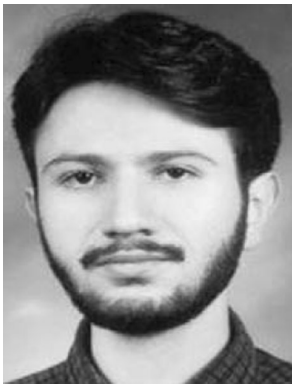
1. Muthukumaran, K., Chitra, K., & Selvakumar, C. (2017). An energy efficient clustering scheme using multilevel routing for wireless sensor network. *Computers & Electrical Engineering* (**in press**).
2. Douceur, J. R. (2002). The Sybil attack. *First international workshop on peer-to-peer systems (IPTPS '02)*.
3. Conti, M., Pietro, D. R., & Spognardi, A. (2014). Clone wars: Distributed detection of clone attacks in mobile WSNs. *Journal of Computer and System Sciences, 80*(3), 654–669.
4. Shafiei, H., Khonsari, A., Derakhshi, H., & Mousavi, P. (2014). Detection and mitigation of sinkhole attacks in wireless sensor networks. *Journal of Computer and System Sciences, 80*(3), 644–653.
5. And, K. C., & Wagner, D. (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. *AdHoc Networks, 1*(2), 299–302.
6. Liu, G., Yan, Z., & Pedrycz, W. (2018). Data collection for attack detection and security measurement in mobile ad hoc networks: A survey. *Journal of Network and Computer Applications, 105,* 105–122.
7. Newsome, J., Shi, E., Song, D., & Perrig, A. (2004). The Sybil attack in sensor networks: Analysis and defenses. In *International symposium on information processing in sensor networks* (pp. 259–268).
8. Chen, S., Yang, G., & Chen, S. (2010). A security routing mechanism against Sybil attack for wireless sensor networks. In *International conference on communications and mobile computing* (pp. 142–146).

9. Jangra, A., & Priyanka, S. (2011). Securing LEACH protocol from Sybil attack using jakes channel scheme (JCS). In *International conferences on advances in ICT for emerging regions (ICTer)*.
10. Vasudeva, A., & Sood, M. (2012). Sybil attack on lowest id clustering algorithm in the mobile ad hoc network. *International Journal of Network Security & Its Applications, 4*(5), 135–147.
11. Zhong, S., Li, L., Liu, Y. G., & Yang, Y. R. (2004). Privacy-preserving location based services for mobile users in wireless networks. *Technical report YALEU/DCS/TR*-1297, Yale Computer Science.
12. Demirbas, M., & Song, Y. (2006). An RSSI-based scheme for Sybil attack detection in wireless sensor networks. In *International symposium on world of wireless, mobile and multimedia networks* (pp. 564–570).
13. Ssu, K. F., Wang, W. T., & Chang, W. C. (2009). Detecting Sybil attacks in wireless sensor networks using neighboring information. *Computer Networks, 53*(18), 3042–3056.
14. Ramachandran, S., & Shanmugan, V. (2011). Impact of Sybil and wormhole attacks in location based geographic multicast routing protocol for wireless sensor networks. *Journal of Computer Science, 7*(7), 973–979.
15. Misra, S., & Myneni, S. (2010). On identifying power control performing Sybil nodes in wireless sensor networks using RSSI. In *InGlobal telecommunications conference* (pp. 1–5).
16. Muraleedharan, R., Ye, X., & Osadciw, L. A. (2008). Prediction of Sybil attack on WSN using bayesian network and swarm intelligence. In *Wireless sensing and processing, Orlando, FL, USA*.
17. Wen, M., Li, H., & Zheng, Y.-F. (2008). TDOA-based Sybil attack detection scheme for wireless sensor. *Journal of Shanghai University, 12*(1), 66–70.
18. Zhang, Y., Fan, K. F., Zhang, S. B., & Mo, W. (2010). AOA based trust evaluation scheme for Sybil attack detection in WSN. *Journal on Application Research of Computers, 27*(5), 1847–1849.
19. Butler, K. R., Ryu, S., Traynor, P., & McDaniel, P. D. (2007). Leveraging identity-based cryptography for node ID assignment in structured P2P systems. *Advanced Information Networking and Application Workshops, 20*(12), 1803–1815.
20. Li, F., Mittal, P., Caesar, M., & Borisov N. (2012). Sybil control: Practical Sybil defense with computational puzzles. In *Seventh* ACM *workshop on Scalable trusted computing* (pp. 67–68).
21. Taol, F., & Ma, J.-F. (2008). New approach against Sybil attack in wireless sensor networks. *Tongxin Xuebao/Journal on Communications, 29,* 13–19.
22. Zhang, Q., Wang, P., Reeves, D., & Ning, P. (2005). Defending against Sybil attacks in sensor networks. In *Second international workshop on security in distributed computing systems* (pp. 85–191).
23. Wang, J., Yang, G., Sun, Y., & Chen, S. (2007). Sybil attack detection based on RSSI for wireless sensor network. In *International conference on wireless communications, networking and mobile computing* (pp. 2684–2687).
24. Yang, J., Chen, Y., & Trappe, W. (2008). Detecting Sybil attack in wireless and sensor networks using cluster analysis. In *5th IEEE international conference on mobile ad hoc and sensor systems.* Atlanta, GA (pp. 834–839).
25. Wang, X.-D., Sun, Y.-Q., & Meng, X.-X. (2009). Cluster-based defending mechanism for Sybil attacks in wireless sensor network. *Computer Engineering, 15,* 47.
26. Ahmad, J. M., Nanda, P., He, X., & Liu, R. P. (2015). A Sybil attack detection scheme for a centralized clustering-based hierarchical network. *Trustcom/BigDataSE/ISPA, 1,* 318–325.
27. Sweety, S., & Sejwar, V. (2014). Sybil attack detection and analysis of energy consumption in cluster based sensor networks. *International Journal of Grid and Distributed Computing, 7*(5), 15–30.
28. Almas, S. R., Faez, K., Eshghi, F., & Kelarestaghi, M. (2017). A new lightweight watchdog-based algorithm for detecting Sybil nodes in mobile WSNs. *Future Internet, 10*(1), 1–17.
29. Rupinder, S., Singh, J., & Singh, R. (2016). TBSD: A defend against Sybil attack in wireless sensor networks. *International Journal of Computer Science and Network Security, 16*(11), 90.
30. Dhamodharan, U. S., & Vayanaperumal, R. (2015). Detecting and preventing Sybil attacks in wireless sensor networks using message authentication and passing method. *The Scientific World Journal, 1*(1), 13–17.
31. Amuthavalli, R., & Bhuvaneswaran, R. S. (2014). Detection and prevention of Sybil attack in wireless sensor network employing random password comparison method. *Journal of Theoretical & Applied Information Technology, 67*(1), 236–246.
32. Shi, W., Liu, S., & Zhang, Z. (2015). A lightweight detection mechanism against Sybil attack in wireless sensor network. *KSII Transactions on Internet & Information Systems, 9*(9), 3738–3749.
33. Sinha, S., Paul, A., & Pal, S. (2014). Use of spline curve in Sybil attack detection based on received signal power-new approach. *International Journal on Recent Trends in Engineering & Technology, 11*(1), 602–611.

34. Rafeh, R., & Khodadadi, M. (2014). Detecting Sybil nodes in wireless sensor networks using two-hop messages. *Indian Journal of Science and Technology, 7*(9), 1359–1368.
35. Panagiotis, S., Karapistoli, E., & Economides, A. (2015). Detecting Sybil attacks in wireless sensor networks using UWB ranging-based information. *Expert Systems with Applications, 42*(21), 7560–7572.
36. Hu, R.-H., Dong, X.-M., & Wang, D.-L. (2015). Defense mechanism against node replication attacks and sybil attacks in wireless sensor networks. *Acta Electronica Sinica, 43*(4), 744–752.
37. Jamshidi, M., Zangeneh, E., Esnaashari, M., & Meybodi, M. R. (2017). A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks. *Computers & Electrical Engineering, 64,* 220–232.
38. Jamshidi, M., Ranjbari, M., Esnaashari, M., Qader, N. N., & Meybodi, M. R. (2018). Sybil node detection in mobile wireless sensor networks using observer nodes. *International Journal on Informatics Visualization, 2*(3), 159–165.
39. Yale, P. B. (1968). Geometry and symmetry. Holden-Day.
40. JSIM Simulator. https://sites.google.com/site/jsimofficial/. Accessed March 21, 2017.

Mojtaba Jamshidi received the B.S. degree in Computer Engineering from the Iranian Academic Center for Education, Culture and research (ACECR), Kermanshah, Iran, in 2009, and M.S. degree in Computer Engineering from Islamic Azad University, Qazvin, Iran, in 2012. His research interests include computer networks, learning systems, security, data mining, meta-heuristic algorithms, and recommender systems.

Ehsan Zangeneh received the B.S. degree in Computer Engineering from the Iranian Academic Center for Education, Culture and research (ACECR), Kermanshah, Iran, in 2014. His research interests include computer networks, algorithms, and security.



Mehdi Esnaashari received the B.S., M.S. and Ph.D. degrees in Computer Engineering all from the Amirkabir University of Technology in Iran, in 2002, 2005, and 2011 respectively. He worked at Iran Telecommunications Research Center as an Assistant Professor from 2012 to 2016. Currently, he is an Assistant Professor in Computer Faculty of K. N. Toosi University of Technology. His research interests include computer networks, learning systems, soft computing, and information retrieval.

Aso Mohammad Darwesh received the B.S. in Mathematics in University of Sulaimani, Iraq 2001, M.S. degrees in Computer Science in University of Rene Descartes, France 2007 and Ph.D. in Computer Science, University of Pierre and Mari Curie, France 2010. Currently he is associate Professor in Information Technology Department, University of Human Development, Sulaimani, Iraq. His research interests include Serious Games, learning system, Computer Network, and Data mining.

Mohammad Reza Meybodi received the B.S. and M.S. degrees in Economics from the Shahid Beheshti University in Iran, in 1973 and 1977, respectively. He also received the M.S. and Ph.D. degrees from the Oklahoma University, USA, in 1980 and 1983, respectively, in Computer Science. Currently he is a Full Professor in Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran. Prior to current position, he worked from 1983 to 1985 as an Assistant Professor at the Western Michigan University, and from1985 to 1991 as an Associate Professor at the Ohio University, USA. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.

Affiliations

Mojtaba Jamshidi[1] · Ehsan Zangeneh[2] · Mehdi Esnaashari[3] · Aso Mohammad Darwesh[1] · Mohammad Reza Meybodi[4]

Ehsan Zangeneh
ehsan.zangneh@yahoo.com

Mehdi Esnaashari
esnaashari@kntu.ac.ir

Aso Mohammad Darwesh
Aso.darwesh@uhd.edu.iq

Mohammad Reza Meybodi
mmeybodi@aut.ac.ir

[1] Department of Information Technology, University of Human Development, Sulaimani, Iraq

[2] Information Technology Development Center, Industrial Development and Renovation Organization of Iran, Tehran, Iran

[3] Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran

[4] Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran