Computers and Electrical Engineering 000 (2016) 1–13

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks

Mojtaba Jamshidi[a,*], Ehsan Zangeneh[b], Mehdi Esnaashari[c], Mohammad Reza Meybodi[d]

[a] Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[b] Department of Computer Engineering, Academic Center of Education, Kermanshah, Iran
[c] Information Technology Department, Cyber-Space Research Institute, Tehran, Iran
[d] Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran

article info

Article history:
Received 1 June 2016
Revised 10 December 2016
Accepted 12 December 2016
Available online xxx

Keywords:
Wireless sensor network
Malicious node
Sybil node
Watchdog Node

abstract

Sybil attack is a well-known attack against wireless sensor networks (WSNs) in which a malicious node attempts to propagate multiple identities. This attack is able to affect routing protocols negatively as well as many other operations such as voting, data aggregation, resource allocation, misbehavior detection, etc. In this paper, a light weight, dynamic algorithm is proposed for detecting Sybil nodes in mobile wireless sensor networks. The proposed 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. As all Sybil nodes belong to a single device (malicious node), they move together, hence, they would have identical bit_label. This fact is used to detect Sybil nodes in the detection phase. The proposed algorithm is simulated using JSIM simulator and simulation results are compared with existing algorithms in terms of true detection and false detection rates. The results show that the proposed algorithm is able to identify more than 94% of Sybil nodes, while false detection rate is 0%

1. Introduction

Wireless sensor networks (WSN) consist of many tiny sensor nodes which cooperate with each other to monitor an area and have a wide variety of applications including battlefield monitoring, industrial applications, health applications and etc. WSNs are mostly used in environments where human attendance might be dangerous or costly. There are hundreds or thousands of sensor nodes deployed in an area when the mission is terminated. Therefore it is not often possible to recollect these sensor nodes. Sensors' small size limits memory capacity, computational power, radio power and energy. Regarding these limitations and wireless nature of sensors, it is very important to provide them with a secure system (especially in military applications). This challenging field has recently attracted attention of many researchers [1].

Sybil is one of the most famous attacks that affect network layers [2]. In Sybil attack, the adversary adds a malicious node to the network or captures a legal (normal) node, reprograms it and then sends it back to the network. Once this malicious

Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. R. Varatharajan.

* Corresponding author.
E-mail addresses: jamshidi.mojtaba@gmail.com (M. Jamshidi), ehsan.zangneh@yahoo.com (E. Zangeneh), esnaashari@itrc.ac.ir (M. Esnaashari), mmeybodi@aut.ac.ir (M.R. Meybodi).

http://dx.doi.org/10.1016/j.compeleceng.2016.12.011
0045-7906/

node joins the network, it begins to exhibit multiple identities, referred to as Sybil nodes, which might be either fabricated or stolen from legal nodes of the network. Showing multiple identities, the malicious node is able to make legitimate nodes believe that they have many neighbors. Consequently, the malicious node attracts more traffic to itself which disrupts the routing protocol and affects network operations such as data aggregation, voting, reputation evaluation, and fair resource allocation [3].

So far, many algorithms have been proposed against Sybil attack in static WSNs which employ different mechanisms such as random key pre-distribution, radio resource testing [4], Received Signal Strength Indicator (RSSI) [5], neighboring information [6], Time Difference Of Arrival (TDOA) [7], Angle Of Arrival (AOA) [8]. But none of them work for mobile wireless sensor networks (due to the nodes' mobility).

Vasudeval and Sood [9] and Piro et al. [10] proposed an algorithm for detecting Sybil nodes in mobile ad hoc networks. Sharmila and Umamaheswari [11] proposed an algorithm for detecting Sybil nodes in mobile wireless sensor networks. They introduced a technique that comprises three phases. During the first phase, in cooperation with Base station and also regarding the number of dropped packets, some nodes will be chosen as cluster head (Trust nodes). Then, these nodes detect Sybil nodes according to received signal strength from member nodes. In the second phase, two nodes which are close to Sybil node send packets to it at the same time. As a result, collision occurs because all Sybil nodes belong to a single device. Finally, in the third phase, the routing procedure is checked to verify if there are any hops between the identities. If true, they are not Sybil nodes; otherwise, the identities are considered as Sybil. This algorithm consists of three complicated and heavy phases. Since nodes' mobility has not been considered during detection mechanism, it seems not to be suitable for sensor networks.

Our goal is to present a practical and light-weight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks which eliminates drawbacks of previous algorithms. Proposed algorithm is not based on RSSI, random key pre-distribution, radio resource testing, or AOA mechanisms.

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

- Utilizing a few Watchdog Nodes (WNs) which monitor the network traffic passively as well as nodes' mobility to detect Sybil nodes.
- Assigning bitwise tags to mobile sensor nodes, using WNs, considering their movement behaviors.
- Detecting Sybil nodes in cooperation with WNs.
- Eliminating memory, computation, and communication overheads of sensor nodes for detecting Sybil nodes.

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

2. Related works

Douceur [2] first introduced the Sybil attack problem in peer-to-peer distributed systems. Karlof and Wagner [3] proved that Sybil attack could affect routing protocols in WSNs.

Newsome et al. [4] analyzed Sybil attack systematically and presented its taxonomy based on how Sybil identities are created and Simultaneity of their indication (indicating them). They also proposed several methods to defend against Sybil attack, such as: Radio Resource Test (RRT), Code Attestation (CA), Random Key Pre-distribution (RKP), Identity Registration (IR), and Position Verification (PV). In RRT mechanism, it is assumed that each node in a typical network is not capable to transmit more than one radio channel simultaneously. Whenever a node wishes to verify whether it is a victim of a Sybil attack or not, it assigns a unique channel to each of its neighbors and requests them to broadcast an acknowledgement (ACK) message to their allocated channel at a specified time. Then the node tunes its receiver to a particular channel randomly and waits to receive an ACK message. If no ACK message is received, the node infers that the node assigned to this particular channel is a Sybil node since the malicious node is unable to broadcast the ACK message to all of its false identities on multiple channels simultaneously. By repeating this procedure periodically, all fake nodes in the network can be recognized with a high probability. In RKP, each node picks k-fold keys from a large pool of m-tuple keys, randomly. Number m is chosen such that two nodes share at least one key with some probability after picking their keys. Then identity of the node is combined with a particular set of keys which it chooses. Thus any node can be authenticated by verifying some or all of the keys which it claims to possess.

Zhong et al. [12] offered an RSSI-based algorithm that uses a few receiver nodes to estimate location of each sender node. Demirbas and Song [5] have also used this mechanism by employing four detector nodes, in order to pinpoint sensor nodes in the environment. As all Sybil nodes belong to a single device (malicious node), their locations are the same. Ssu et al. [6] proposed a distributed algorithm which only uses neighboring information to detect Sybil nodes. The TDOA-based mechanism [7] associates the TDOA ratio with the sender's identity. Once there are two different identities with the same TDOA ratio, a Sybil attack is detected. Zhang et al. [8] proposed an AOA-based algorithm, named TEBA which takes advantage of the fact that, although malicious node could create multi-identities, they all have the same physical position. Beacon node identifies Sybil identities with signal phase difference below a trusted threshold for adjacent sensor nodes.

Two algorithms were proposed by Chen et al. [13] and Jangra and Priyanka [14] for detecting Sybil nodes in cluster-based sensor networks. Ramachandran and Shanmugan [15] proposed a geographical location based algorithm for detecting Sybil

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

attack. An advanced RSSI-based algorithm was proposed by Mirsa and Myneni [16] which can detect Sybil attack even if Sybil nodes change their transmission power for each Sybil node.

Jamshidi et al. [17] proposed a distributed dynamic algorithm based on learning automata model and the client puzzles theory to identify Sybil nodes in wireless sensor networks. In this algorithm, each node sends some puzzles to its neighbors periodically during the network's lifetime and tries to identify the Sybil nodes regarding their response times (puzzle solving times). This algorithm uses learning automata for reducing communication and computation overhead of sending, and solving the puzzles. Li et al. [18] proposed Sybil control distributed algorithm to take control over the extent of Sybil attack. It is an admission control mechanism for nodes in distributed systems in which nodes need to solve some computational puzzles periodically. Saxena and Sejwar [19] proposed an algorithm for Sybil attack detection based on TDOA localization method, which detects malicious behavior of head nodes and member nodes in a cluster based network.

Dhamodharan and Vayanaperumal [20] proposed a message authentication algorithm which is applied for checking the trustworthiness or detecting a Sybil node. The node acts as a Sybil node with duplicated ID, and information can happen only when the node has complete information about other nodes. CAM-PVM has to be applied in order to verify the node's needs. Instead of wasting time for CAM-PVM to check each and every node, message authentication and passing procedure is applied for authentication prior to communication. If a node is not authorized by the network or by the base station, it cannot communicate with any other node in the network.

Amuthavalli and Bhuvaneswaran [21] proposed a Random Password Generation (RPG) algorithm that focuses on various traffic levels and security during data transmission in WSN. RPG algorithm generates the routing table which holds information about deployed nodes. The intermediate nodes in the route are identified between source and destination. The intermediate node's information is compared with RPG database during communication, and then the comparison results are used to decide whether these intermediate nodes are Sybil or normal.

Shi et al. [22] proposed a detection algorithm to detect Sybil attack In 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 attack called LEACH-RSSI-ID (LRD).

Sinha et al. [23] proposed a new location based Sybil attack detection technique by describing the characteristics of Received Signal Powers (RSPs) of the nodes using spline curves. Here, the spline representation of RSPs is an innovation to track the Sybil identities even in a mobile environment.

3. Symbols, system assumptions and attack model

3.1. Symbols

- bit_label: a bitwise label assigned to each mobile sensor node, by WNs, considering its movement behavior in the network environment.
- W_i: the Watchdog Node with identity i.
- moving_history: a table in the memory of WNs to temporary store the SNs' bit label.
- P: the number of monitoring rounds that are carried out by WNs (the number of iterations in first phase of the proposed algorithm).
- R_i: the ith round of monitoring phase of the proposed algorithm.
- request_list: a list containing a set of node identities which are broadcasted by a WN to other WNs; and request them to send him back if they have information about those nodes.
- T_s: the threshold T_s which is determined regarding the minimum number of Sybil identities that could be exhibited by a single malicious node.
- r: radio range of the nodes.

3.2. System assumptions

Sensor network contains two sets of nodes: ordinary sensor nodes (SNs) and Watchdog Nodes (WNs) that are randomly distributed in a two-dimensional area. SNs do the usual network operations (like data gathering, sending data toward Base station) and WNs are responsible for detecting Sybil attack. Each node has its own unique ID and nodes are not aware of their location. Radio range of SNs is fixed and equal to r, while radio range of WNs is equal to the diameter of the network, such that WNs are able to communicate with each other directly. SNs are mobile and move in the network area according to mobility models like random waypoint, during the network's lifetime. After deployment of nodes, the WNs do not move and remain in their locations, unless the distance between two of them is less than or equal to 2r (they have common neighbors). In such situation, the two WNs begin to move (randomly) in order to lengthen distance such that it becomes greater than 2r (they do not have any common neighbors). It is assumed that the nodes communicate with each other via a wireless radio channel and broadcast information in an Omni-directional mode. WNs use a Group key [24] for communication.

It is assumed that SNs are not tamper proof, so if they are compromised by adversary, their confidential information will be revealed and the adversary is able to reprogram them, and put them back to the network. But it is assumed that WNs are tamper proof and adversary cannot reprogram them, if they are compromised. Since SNs are mobile, they need to send

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

Node_ID	*bit label*

Fig. 1. Structure of moving_history table.

a message such as “Hello message”, “keep alive message”, “route request” periodically (in either case, after a SN moves to a new location or after each t period of time) [10]. Each node is able to identify its neighbors at any time which is actually one of the mobile sensor network’s requirements.

3.3. Attack model

According to categorization of Newsome et al. [4], direct, fabricated and simultaneous Sybil attack is considered here. Also, according to the attack model considered by Ssu et al. [6], it is assumed that the network is insecure and the nodes can be compromised with a certain probability. The compromised node is referred to as a malicious node, while the other nodes in the network are referred to as legitimate nodes. Each malicious node cheats its neighbors by creating multiple identities (Sybil nodes). When a node sends a message to any of the Sybil nodes, the message will be received and replied (if needed) by the malicious node. The main mission of malicious node is to trick the legitimate nodes to believe that they have many neighbors. Since these nodes do not really exist, many of the network protocols are seriously disturbed and become impractical.

Adversary would disrupt network operations in two ways. First, the adversary tries to compromise some legal nodes and reprogram them, then puts them back into the network, so that all of these nodes exhibit a few (2 or 3) Sybil identities. In this case, it is too hard for security algorithms to detect Sybil nodes or some algorithms such as the one introduced by Ssu et al. [6]; they are not even able to discover Sybil attack. Of course, it takes the adversary so much time to compromise nodes and decode their information and take them under control. The second way, in which adversary is able to affect network significantly, is to compromise fewer nodes but let them exhibit more identities. Similarly, we suppose that our network is victim of the second type of Sybil attack. We also assume that malicious nodes are mobile like legal nodes. Finally, it is assumed once a malicious node moves to a new location, it shows all its Sybil identities by sending HELLO or routing request messages for each of them. This is a necessity regarding the simultaneous model of the attack; otherwise the attack is not able to significantly affect the network operations.

4. Proposed algorithm

As mentioned before, there are two types of sensor nodes, SNs which are ordinary nodes and do the typical operations of the network and WNs that are supposed to be responsible for detecting Sybil nodes. In fact, the proposed algorithm would be implemented on WNs. The main idea of this algorithm takes advantage of the nodes’ mobility. As all Sybil nodes belong to a single device (malicious node), they move together, therefore all of them would be neighbor of a WN at the same time. The proposed algorithm on the WNs takes advantage of this issue to detect Sybil nodes. When a certain SN moves to a certain WN’s neighborhood, the WN stores some information (that will be discussed later) about the SN in its memory.

The proposed algorithm consists of two phases: monitoring phase and the Sybil node detection phase, which are both performed by WNs. In order to conserve energy, the proposed algorithm could be executed periodically (e.g. once per half an hour). Each time that the algorithm is executed, WNs store information of the SNs’ movement in P rounds (monitoring phase). After round P (in other words, R_P) and based on the gathered information, WNs would be able to detect Sybil nodes (detection phase). In the proposed algorithm, there is a table in each WN’s memory, named moving_history (Fig. 1). This table contains two fields: Node_ID, in which the WN stores identity of a certain SN that appears in the WN’s neighborhood, and the bit_label which is used to keep the appearance label of the corresponding node. Each WN empties its moving_history table when the algorithm initiates.

In Fig. 2, S1–S10 are Sybil nodes belonging to a malicious node, W_1, W_2, W_3 are WNs and the others are SNs. Suppose that in the first round(R_1), the malicious node (the nodes with S1–S10 identities), and {x, y, z, u, v, d} which are SNs are all in the W_1’s neighborhood. Therefore, W_1 stores these nodes’ identities in its moving_history and assigns a particular bit label to them (in this example,“ 00”). If the number of WNs in our network is q, then we can assign a unique $\log_2 q$ bit label to each WN. In this example, bit labels for W_1, W_2, and W_3 are “00”, “01”, and “10”, respectively. Fig. 3 shows W_1’s moving_history table in R_1. All other WNs also do the same operation. Since, nodes “a” and “b” are not in any of WN’s neighborhood during R_1, their information is not stored in any of WN tables. This may happen for a large number of nodes during the first round, but as they move in the area over time, the number of nodes for which this may happen, decreases.

We suppose that during R_2, the nodes move and are positioned as shown in Fig. 4. In this round, Sybil nodes, x and y are out of W_1’s neighborhood and now they are in W_3’s neighborhood. Nodes d and z are still in W_1’s neighborhood, v has moved to W_2’s neighborhood, and nodes a and b, which were isolated during the first round, are now close to W_1 and W_2, respectively. From now on (from second round on), if some nodes appear in W_i’s neighborhood and there is no information about them in W_i’s moving_history table (i.e. if they are new neighbors of W_i), W_i generates a single packet and put all these

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

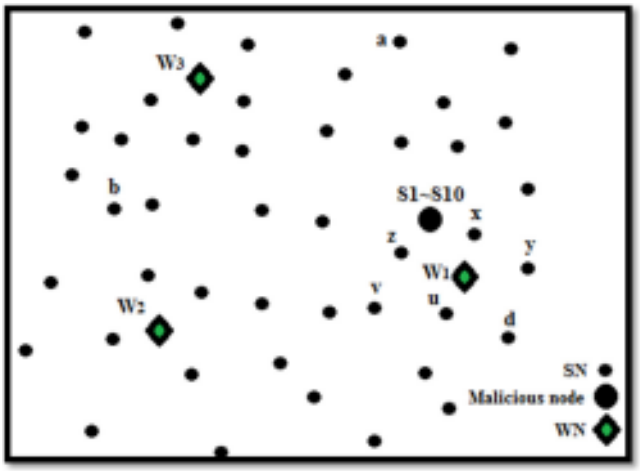

Fig. 2. An example of deploying nodes in the first execution round of the algorithm.

Node_ID	*bit_label*
S1-S10	**00**
x	**00**
y	**00**
z	**00**
u	**00**
v	**00**
d	**00**

Fig. 3. W_1's moving_history table after R_1.

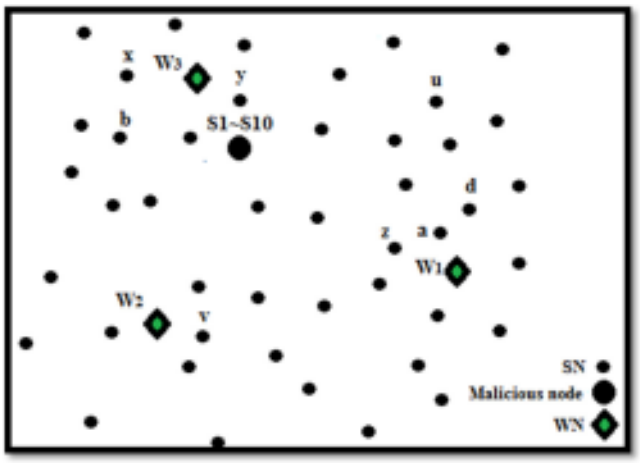

Fig. 4. Arrangement of nodes during the second round.

new identities into it, and sends this packet as a request_list to the other WNs. Then the node W_i sets a timer with value T_w and waits for response. Each WN that receives this request_list, searches its moving_history table and if it finds any of these identities (e.g. u), sends that node's (node u) information (Node_ID and bit_label) to W_i first and then removes this information from its Moving_history table. If there was no response about a certain identity, W_i realizes that bit_label of this node identity is null. After the timer is expired, W_i appends its own particular bit_label to its current neighbors' label.

Now let us discuss how the proposed algorithm operates regarding these concepts. As depicted in Fig. 4, in the second round (R_2), W_1 sees node a in its neighborhood while there is no information about it in W_1 moving_history table. Therefore,

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

(a)

Node_ID	*bit_label*
z	0000
u	00
d	0000
a	00

(b)

Node_ID	*bit_label*
…	…
x	0010
y	0010
S1-S10	0010
b	10

Fig. 5. The moving_history table of (a) W_1 and (b) W_3.

W_1 inserts the identity "a" in a request_list and broadcast it throughout the network and waits for response from other WNs. Node W_3 does the same for nodes {x, y, b, S1–S10} as there is no information about them in W_3's moving_history table. Each WN that has new nodes in its neighborhood broadcasts a request_list containing these new node identities to other WNs. In this example, since none of the identities in W_1 and W_3's request_list exist in W_2's moving_history table, W_2 does not respond to any request_list from them. Furthermore, since W_3 has no information about node a, when it receives W_1's request_list, it does not response. But when W_1 receives W_3's request_list, it sends this information (Node_ID and bit_label) back to W_3, and then removes this information from its moving_history, because its moving_history table contains information related to {S1–S10, x, y}. As soon as W_3 receives this packet containing information about {S1–S10, x, y}, puts this information in its moving_history table. When their timers are expired, W_1, W_2, and W_3 append their own particular bit label to neighbors' current bit_label. So at the end of R_2, moving_history table of W_1 and W_3 would be updated as shown in Fig. 5.

As depicted in Fig. 5(a), information related to node v has been eliminated from W_1's moving_history table because it has moved to the neighborhood of W_2 in R_2 and following that, W_2 broadcasted a request_list containing the identity of v and then received response from W_1. But information about node u has not been eliminated from W_1'smoving_history table. Although it has been moved out of W_1's neighborhood, it is not in any other neighborhoods of WNs (during R_2), so no request is sent to W_1 for information related to node u and that is why bit_label of u has not changed during round R_2. But bit_label for node z and d are changed into "0000" because they are still in W_1's neighborhood. So far, node a has not been in any of WN's neighborhoods, hence it gets its first bit_label from W_1. On the other hand, W_3 appends its own particular bit_label to {S1–S10, x, y}, so that their bit_label in R_2 is "0010". Since node b was not in any of WN's neighborhoods for the previous round (R_1), its first bit_label is assigned byW_3 (Fig. 5(b)).

This process keeps going for next rounds. After round P, R_P, detection phase of the proposed algorithm begins in order to detect Sybil nodes. The WNs execute this phase independently, and according to the contents of their moving_history. Each WN scans its moving_history table and creates some distinct sets of identities such that identities in each set have the same bit_label. Those sets in which the number of members is greater than a certain threshold, T_s, are considered as Sybil nodes' sets. The threshold T_s can be determined regarding the minimum number of Sybil identities that could be exhibited by a single malicious node. Since Sybil identities move together, they would have identical bit_label, therefore they would be categorized in the same set. Of course, it is possible for a few legal identities to be in the same set like Sybil nodes. This state depends on the number of rounds, which the proposed algorithm is supposed to be executed (depends on P). By increasing P, accuracy is increased.

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 experiments.

5.1. Performance evaluation

Memory overhead: since the proposed algorithm would be performed only by WNs, just WNs suffer from memory overhead. Each WN needs to allocate part of its memory to store moving_history table to keep track of SN's movement. It should also be considered that information related to a certain SN (e.g. node u) is being kept just by one WN (the one which has node u in its neighborhood), at a time. Information related to a certain SN consists of the SN's identity and its bit_label. Therefore after P round, memory overhead of all WNs is $|SN| \times Size_{ID} + |SN| \times P \times \log_2^q$ bits, in which, $|SN|$ is the number of SNs in the network, $Size_{ID}$ is the size of each SN's identity in terms of bit, and $|WN| = q$is the number of WNs in the network. Therefore, the imposed memory overhead for each WN equals $O(\frac{|SN| \times P \times \log_2^q}{q})$ and for each SN equals zero. It is worth mentioning that increasing the number of WNs in the network results in longer bit_labels which consequently increases the memory overhead. In general, if number of WNs increases from α to β, memory overhead increases $|SN| \times P \times (\log_2^\beta - \log_2^\alpha)$ bits.

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

Table 1
Simulation parameters.

Parameter	Values
Network size	$100 \times 100\ m^2$
Total nodes	N = 100–400
Number of malicious nodes	M = 1, 5, 10
Number of Sybil identities propagated by each malicious node	S = 6, 10, 14, 18
Number of WNs	q = 4, 5, 6
SNs' radio range	r = 10 m
Threshold	T_s = 4, 6, 8
Number of monitoring rounds	P = 20–100

After P rounds and executing the detection phase, all WNs release their memory and empty their moving_history table, for next execution of the algorithm.

Communication overhead: energy consumption of the algorithm is critical due to limitation of SNs' energy. Since sending packets consumes much more energy than other operations such as receiving or computing, number of transmitted packets imposed upon the network during execution of a certain algorithm is considered an important criterion. SNs broadcast "Hello" message whenever they move to a new location because it is essential for them to update their neighboring table or to request a new route for themselves. As mentioned above, this is actually one of the requirements for mobile sensor networks. Therefore "Hello" messages sent by SNs are not regarded as overhead. At the end of each round, each WN sends (broadcasts) a request_list if there is even one new SN in its neighborhood that does not exist in its moving_history table. This takes place by all WNs at the same time. Therefore maximum number of request_lists transmitting over the network in this step equals q which is the total number of WNs. Then, each WN which has information related to node(s) – in the received request_list – in its moving_history table responds to the corresponding WN. Therefore, at most 2q packets have been sent by WNs in each round, thus after P rounds, the communication overhead would be $O(P \times q)$. This overhead would increase if the number of WNs in the network increases, since more request_list packets will be transmitted throughout the network in such a case.

Computation overhead: Since only WNs are supposed to perform the algorithm, there is no SN which suffers from computation overhead. At the end of each round, each WN creates a list of new SNs identity in its neighborhood, about which there is no information in its (the WN's) moving_history table and sends it as the request_list. Time complexity for creating a request_list is $O(d \times k)$, in which d is the average number of a WN's neighbors and k is the number of the rows in a WN's moving_history table. Suppose $k = \frac{|SN|}{q}$. Moreover, it takes $O(q \times d \times k)$ time to investigate its own moving_history table to find information corresponding to Node_IDs in the request_list received from other WNs. Therefore, total computational overhead at the end of each round would be $q \times d \times k + d \times k$ which is from order $O(q \times d \times k)$. After P round, each WN divides its moving_history table into distinct sets, such that Node_IDs in each set have similar bit_labels. This operation is done in $O(k^2 \times \phi)$ times regarding that each couple of bit_labels would be compared with time ϕ, which is minimum length of these two bit_labels. Then the algorithm chooses sets which contain more than threshold T_s, as the sets which include Sybil nodes with time complexity O(k). Note that in the worst case, there is one unique bit_label for each identity in the moving_history table, which means there are k sets. By increasing the number of WNs, k decreases, thus the computation overhead would remain constant. As mentioned before, by increasing the number of WNs in the network, the length of bitwise tags would also increase, consequently the time required for comparing such tags increases. If number of WNs increases from α to β, computation overhead of each WN in second phase of the proposed algorithm increases $k^2 \times (\log_2 \beta - \log_2 \alpha)$ times.

5.2. Simulation results

Performance of the detecting algorithm was evaluated by performing a series of simulations. Two performance metrics were considered: true detection rate (TD), i.e. the percentage of detected Sybil nodes, and false detection rate (FD), i.e. percentage of normal nodes erroneously classified as Sybil nodes. Experiment results were compared to similar algorithms [5,6,13,14,16,17,19–21]. The proposed algorithm was simulated using JSIM simulator [25]. Simulation parameters are shown in Table 1. All these parameters can affect performance of the proposed algorithm, thus we have tried to evaluate such effects in the simulations (Experiments 1–5).

It has been assumed in the simulations that the network is comprised of N nodes randomly distributed in a $100 \times 100\ m^2$ area. The sensing field contained q Watchdog Nodes (WNs), M malicious nodes where each forged S Sybil nodes, and N− (q + M) normal nodes. Both normal and malicious nodes had a fixed and maximum radio range of r = 10 m, but radio range of WNs covers the whole area such that they are able to directly communicate with each other. We used the mobility model in [26] to verify the results. Each simulation was repeated 50 times and then the results were averaged to obtain a final value.

Experiment 1: In this experiment, we study the effect of threshold T_s on true detection and false detection rates, where our parameters are N = 300, M = 5, q = 4, S = 10 and the threshold value T_s varies from 4 to 8 (by incremental step of 2) and the accuracy of the algorithm is measured for round 20–100 (by incremental step of 20). Fig. 6 shows the results of

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

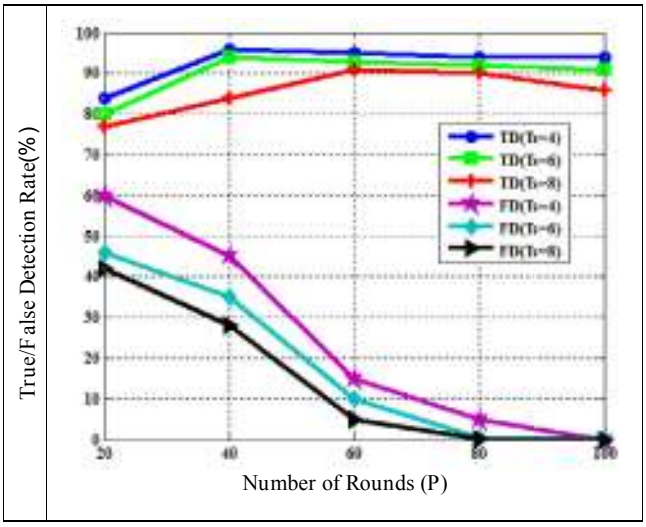

Fig. 6. Effect of threshold T_s on detection performance.

this experiment. Results show that true detection of the algorithm after 20 rounds is approximately 80% for different values of T_s, and it is 94% for T_s= 6 and T_s= 4, after 40 rounds.

It is clear from the results that, by decreasing threshold T_s, true detection rate increases. As it was discussed before, when nodes move to a new location, they send some messages such as "Hello", and then WNs register their appearances; but since a collision might occur, sometimes nodes' appearance might not be observed by WNs. It is an effective item for true detection rate. Although malicious node and all of its Sybil identities move together, but a few Sybil identities might be hidden from WNs due to the collision. Being hidden from WNs causes these Sybil identities (hidden identities) to get a different label from other Sybil identities and separate from the others. This problem is called "breaking-problem", which might happen in next execution of the algorithm and cause more nodes to get out of the group, which results in a Sybil group (the group containing Sybil IDs) with fewer members. Therefore, if a small value is assigned to T_s, then Sybil groups with few members would be detected, so true detection rate would rise. Fig. 6 also shows that after round 40 (means 60, 80 and so on), more breakings are might take place while running the algorithm, so true detection rate decreases. It is also possible for a malicious node not to appear in a WN's neighborhood for the first several rounds due to their random movement. That is why true detection rate is low for the first several rounds (specifically less than 40). In other words, Sybil nodes will not be detected unless they appear in the WNs' neighborhood.

Results also show that false detection rate strongly relies on the threshold T_S. It is clear that a number of normal nodes may accidentally move together in the area such that after round P, there are a few small groups containing normal nodes with identical label. Accordingly if we set T_S with a small value, then it is possible for the algorithm to determine these small groups (of normal nodes) as Sybil groups which means false detection rate has been increased. It is also possible that some normal nodes accidentally move in the same pattern as a certain malicious node which causes these normal nodes to be labeled like the Sybil ones. Like random model of the nodes' movement, it is expected that by increasing the number of rounds (means, P), these two cases, which cause the false detection, become ineffective. For example, it can be seen in Fig. 6 that false detection rate for different values of T_s after 80 rounds has dropped below 5% and after 100 rounds it is approximately 0%

Experiment 2: In this experiment, we study the effect of S (number of Sybil identities spread by a malicious node) on both false detection and true detection rates. Here the parameters are N = 300, M = 5, q = 4, T_S = 6. The parameter S varies from 6 to 18 (by the increment step of 4). We also study the accuracy of the algorithm for executive rounds from 20 to 100 (increment step = 20) and obtained results are illustrated in Fig. 7. Results show that true detection rate for S = 6 is less than 80% and for S ≥ 10 it is above 92% This result is due to the probability of the breaking-problem, which means some Sybil identities may get out of the group (due to the collision). Spreading fewer Sybil identities caused the number of identities in the Sybil group to reach below the threshold T_s (because of breaking-problem) and so the total Sybil group will not be detected. So the detection rate increases as the number of Sybil nodes increases.

As illustrated in Fig. 7, parameter S does not have any significant effects on false detection rate, because all Sybil identities belong to a single physical malicious node, so different values of this parameter do not affect the probability of a normal node to join a Sybil group. However, value of the parameter M (number of malicious nodes) affects the probability of a normal node to join a Sybil group (next experiment shows this issue).

Experiment 3: In this experiment, we will study the effect of M (number of malicious nodes) on performance of the proposed algorithm. In this experiment, parameters are N = 300, S = 14, q = 4, T_S = 6 and the accuracy of the proposed algorithm is evaluated for executive rounds varying from 20 to 100 (by increment step of 20). Result of the experiment is

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

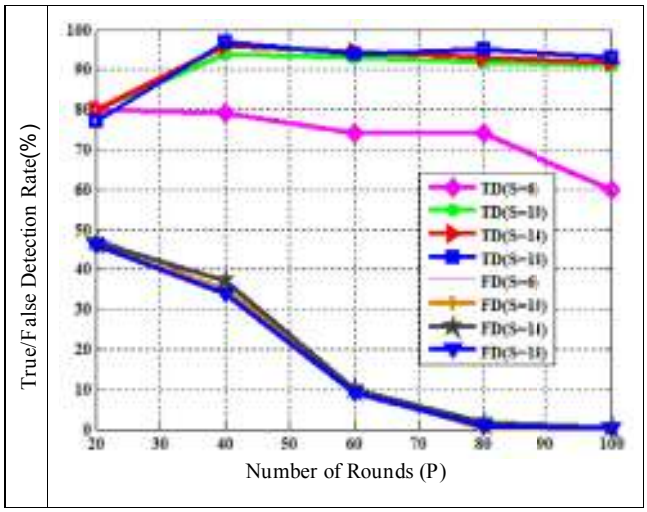

Fig. 7. Effect of number of Sybil nodes, S, on detection performance.

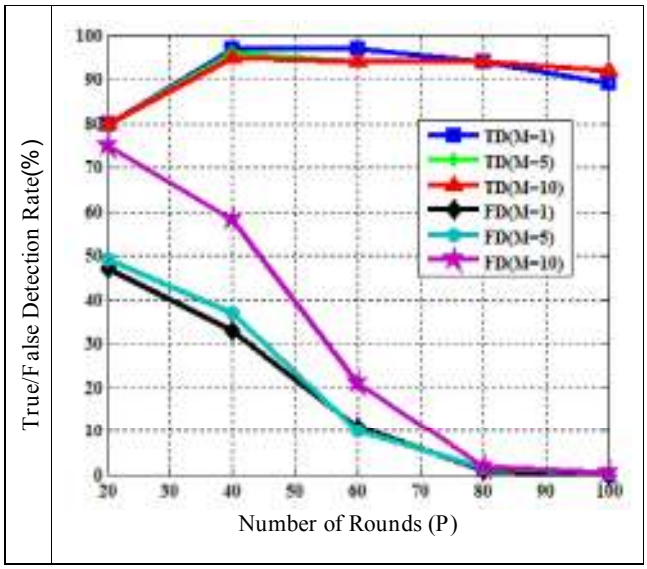

Fig. 8. Effect of number of malicious nodes, M, on detection performance.

illustrated in Fig. 8, for M = 1, 5, 10. As it is clear from the obtained results, number of malicious nodes, M, does not affect true detection rate of the proposed algorithm significantly and the rate is higher than 92% That is because the mechanism used in the proposed algorithm is based on nodes' movement. As each malicious node moves with its own Sybil identities independently, number of malicious nodes does not affect true detection rate considerably.

But as expected, the parameter M affects false detection rate, for example, increasing M will increase false detection rate, because by increasing parameter M (which means increasing number of adversary nodes), probability of a certain normal node to move with a certain malicious node during the initial several rounds would also increase. In other words, the probability of a certain normal node to be a member of a Sybil group would increase which causes false detection rate to rise for initial rounds. Note that for executive rounds greater than 60, probability of normal nodes to get out of a Sybil group would increase; this gives better results in higher rounds. Likewise, as illustrated in Fig. 8, for different values of M, by incrementing the number of executive rounds, false detection rate tends towards zero.

Experiment 4: In this experiment, we examine the effect of parameter q (number of Watchdog Nodes) on true detection and false detection rates. We set the parameters as N = 300, S = 14, M = 5, T_S = 6. We evaluate accuracy of the proposed algorithm for executive rounds varying from 20 to 100 (by increment step of 20) and parameter q = 4, 5, 6. Fig. 9 depicts the obtained results. As results indicate, for q = 4 (when there are 4 WNs in the network),true detection rate of the algorithm is approximately 94% and when there are 5 or 6 WNs in the network, true detection rate drops to about 92%

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

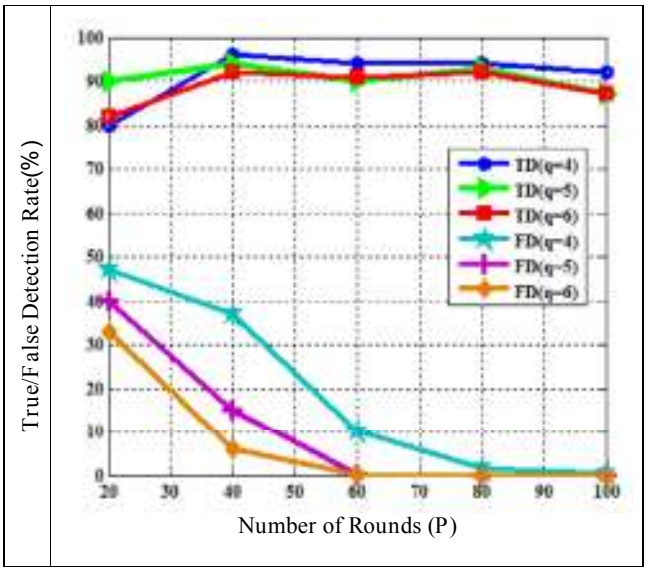

Fig. 9. Effect of number of Watchdog Nodes, q, on detection performance.

On the other hand, results show that increasing the number of WNs results in less false detection rate and vice versa. It means that by decreasing the number of WNs in the network we would have a higher percentage of false detection. As an example in this experiment, after 40 rounds of executing the algorithm for q = 4, 5, 6, we get 37%, 15% and 6% false detection rates respectively. Of course, it should be noted that as parameter P (number of executive round) increases, false detection rate tends to 0% Likewise, for great values of P, different values of q do not make any difference for false detection rate. Now, let us see the reasons beyond the obtained results. When a malicious node moves to a new location in the network and sends "Hello" message to some of its Sybil nodes e.g. S1, collision may occur so that appearance of S1 in this region of the network would remain hidden from nodes in S1's neighborhood. Now if there is a WN in this region, it would save information related to all Sybil nodes except for S1. Therefore, S1 gets out of the Sybil group (Breaking-problem) and S1 would not be detected as a Sybil node. However, if there are no WNs in the region, Sybil node S1 would not get out of the Sybil group (and true detection rate would also be increased). On the other hand, by increasing the number of WNs in the network, those normal nodes that have randomly been labeled as members of Sybil group, would exit from the Sybil group, this decreases false detection rate.

Experiment 5: In this experiment, the effect of parameter N (total number of nodes) on detection rate is studied. The goal of this experiment is to evaluate the scalability of the proposed algorithm. Here the considered parameters are q = 4, S = 14, M = 5, T_S = 6. Accuracy of the proposed algorithm is investigated for executive rounds of 60 and 80 and for parameterN = 100, 200, 300, 400, 500. As shown in Fig. 10, results of this experiment show that number of nodes does not affect true detection rate and it is about 94% But, false detection rate changes by varying the number of nodes in the network. By increasing the number of nodes in the network, false detection rate also increases. Because the higher is the number of nodes in the network, the higher would be the probability of normal nodes' movement along with malicious nodes. This increases false detection rate. Results also show that by increasing the number of executive rounds (P), false detection rate would decrease noticeably. For example, when N = 500, for P = 60, false detection rate is 19% and for P = 80, it is below 3%

Experiment 6: In this experiment, performance of the proposed algorithm is compared to other existing algorithms in terms of average true and false detection rates. As Table 2 illustrates, true detection rate (on average) in the algorithms proposed by Demirbas and Song [5], Ssu et al. [6] Mirsa and Myneni [16] and Jamshidi et al. [17] is approximately 99%, and in the algorithms proposed by Chen et al. [13] and Jangra and Priyanka [14], it is about 90%, and it is about 94% in the proposed algorithm. Of course, it should be considered that all other algorithms are designed for detecting Sybil nodes in stationary networks, whereas the proposed algorithm discovers Sybil nodes in mobile networks. Therefore, obtained results verify the desired performance of the proposed algorithm.

Second column in Table 2 shows average false detection rate compared to other algorithms. It also indicates that (on average) false detection rate is 0% which is better compared to other algorithms.

6. Summary

The experiments show that:

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

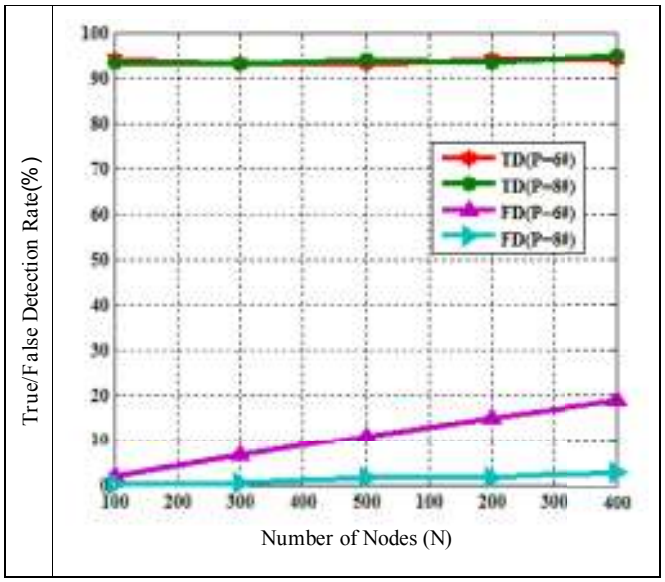

Fig. 10. Effect of number of nodes, N, on detection performance.

Table 2
Comparison of performance of the proposed algorithm and other algorithms in terms of false/true detection rate.

Algorithm	Average of true detection rate (%)	Average of false detection rate (%)
Demirbas and Song [5]	100	6
Ssu et al. [6]	99	5
Chen et al. [13]	92	2
Jangra and Priyanka [14]	90	1
Misra and Myneni [16]	98	6
Jamshidi et al. [17]	100	5
Saxena and Sejwar [19]	96	4
Dhamodharan and Vayanaperumal [20]	80	0
Amuthavalli and Bhuvaneswaran [21]	60	0
Proposed algorithm	94	0

- Decreasing the value of threshold T_s increases both false and true detection rates at the beginning, but after about P= 100 rounds in the monitoring phase, value of false detection rate decreases to 0%
- Increasing the number of propagated identities by each Sybil node, increases the value of true detection rate while it does not affect false detection rate.
- Increasing the number of malicious nodes in the network increases the value of false detection rate while it has no effect on the value of true detection rate. But after about P= 80 rounds in the monitoring phase, the value of false detection rate decreases to 0%
- Increasing the number of WNs decreases false detection rate noticeably, while it has slight negative effects on the value of true detection rate.
- Increasing the number of nodes in the network does not affect true detection rate while it has negative effects on false detection rate.

Table 3 summarizes the aforementioned results.

7. Conclusion

In this paper, we proposed a new efficient light-weight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks. The proposed algorithm is based on nodes' mobility in the area, and does not use mechanisms like pin-pointing, RSSI, etc. to detect Sybil nodes. The proposed algorithm consists of two phases: monitoring phase and detection phase, which are both performed by WNs.

In the first phase, Watchdog Nodes keep track of nodes' movement during different periods (P rounds) and assign them bitwise labels, which are stored among all WNs distributedly. In detection phase, each WN discovers Sybil nodes based on the information in its moving_history table, independently. In this phase, the main idea of identifying Sybil nodes originates from the fact that all of them have the same bit-label. Since all Sybil nodes belong to a malignant group, their movement

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

Table 3
Summary of results obtained from the conducted simulation studies.

Parameter↑	True detection rate	False detection rate
T_s	↑	↑
S	↑	–
M	–	↑
q	↓	↓
N	–	↑
P	⇕	↓

↑: increase
↓: decrease
–: no change
⇕increase and decrease

pattern would be identical. Each WN creates independent sets of nodes available in its moving_history table such that nodes of each set have identical bit-labels. Finally, sets in which number of members is more than specified threshold, T_s, are marked as Sybil nodes.

Simulation results indicate that the proposed algorithm outperforms existing algorithms Demirbas and Song [5], Ssu et al. [6], Chen et al. [13], Jangra and Priyanka [14], Misra and Myneni [16], Jamshidi et al. [17], Saxena and Sejwar [19] in terms of false detection rate and in terms of true detection rate, the proposed algorithm outperforms algorithms Chen et al. [13], Jangra and Priyanka [14], Dhamodharan and Vayanaperumal [20], and Amuthavalli and Bhuvaneswaran [21], but performs slightly worse than the other algorithms which is because it works in mobile sensor networks. One drawback of the proposed algorithm is "breaking-problem" which can be resolved in future works using aging mechanism.

References

[1] Yick J, Mukherjee B, Ghosal D. Wireless sensor network survey. Comput Netw 2008;52:2292–330.
[2] Douceur JR. The Sybil attack. In: Proceedings of the first international workshop on peer-to-peer systems (IPTPS); 2002. p. 251–60.
[3] Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. AdHoc Netw 2003;1(2):299–302.
[4] Newsome J, Shi E, Song D, Perrig A. The Sybil attack in sensor networks: analysis and defenses. In: Proceedings of the international symposium on information processing in sensor networks; 2004. p. 259–68.
[5] Demirbas M, Song Y. An RSSI-based scheme for Sybil attack detection in wireless sensor networks. In: Proceedings of the international symposium on world of wireless, mobile and multimedia networks; 2006. p. 564–70.
[6] Ssu K F, Wang W T, Chang W C. Detecting Sybil attacks in wireless sensor networks using neighboring information. Comput Netw 2009;53:3042–56.
[7] Wen M, Li H, Zheng Y F. TDOA-based Sybil attack detection scheme for wireless sensor. J Shanghai Univ 2008;12(1):66–70.
[8] Zhang Y, Fan K F, Zhang S B, Mo W. AOA based trust evaluation scheme for Sybil attack detection in WSN. J Appl Res Comput 2010;27(5):1847–9.
[9] Vasudeva1 A, Sood M. Sybil attack on lowest ID clustering algorithm in the mobile ad hoc network. Int J Netw Secur Appl 2012;4:135–47.
[10] Piro C, Shields C, Levine BN. Detecting the Sybil attack in mobile ad hoc networks. In: Proceedings of the second international ICST conference on security and privacy in communication network, SecureComm; 2006. p. 1–11.
[11] Sharmila S, Umamaheswari G. Detection of Sybil attack in mobile wireless sensor networks. Int J Eng Sci Adv Technol 2012;2:256–62.
[12] Zhong S, Li L, Liu Y G, Yang Y R. Privacy-preserving location based services for mobile users in wireless networks. Yale Computer Science; 2004. Technical report YALEU/DCS/TR-1297.
[13] Chen S, Yang G, Chen S. A security routing mechanism against Sybil attack for wireless sensor networks. In: Proceedings of the international conference on communications and mobile computing, 1; 2010. p. 142–6.
[14] Jangra A, Priyanka S. Securing LEACH protocol from Sybil attack using jakes channel scheme (JCS). In: Proceedings of the international conferences on advances in ICT for emerging regions; 2011.
[15] Ramachandran S, Shanmugan V. Impact of Sybil and wormhole attacks in location based geographic multicast routing protocol for wireless sensor networks. J Comput Sci 2011;7:973–9.
[16] Misra S, Myneni S. On identifying power control performing Sybil nodes in wireless sensor networks using RSSI. In: Proceedings of the global telecommunications conference; 2010. p. 1–5.
[17] Jamshidi M, Esnaashari M, Meybodi MR. An algorithm for defending Sybil attacks based on client puzzles and learning automata for wireless sensor networks. In: Proceedings of the eighteenth national conference of computer society of Iran. Tehran, Iran: Sharif University; 2013.
[18] Li F, Mittal P, Caesar M, Borisov N. SybilControl: practical Sybil defense with computational puzzles. In: Proceedings of the seventh ACM workshop on scalable trusted computing; 2012. p. 67–78. Networking and Internet Architecture.
[19] Saxena S, Sejwar V. Sybil attack detection and analysis of energy consumption in cluster based sensor networks. Int J Grid Distrib Comput 2014;7(5):15–30.
[20] Dhamodharan US, Vayanaperumal R. Detecting and preventing Sybil attacks in wireless sensor networks using message authentication and passing method. Sci. World J. 2015;1(1):13–17.
[21] Amuthavalli R, Bhuvaneswaran RS. Detection and prevention of Sybil attack in wireless sensor network employing random password comparison method. J Theor Appl Inf Technol 2014;67(1):236–46.
[22] Shi W, Liu S, Zhang Z. A lightweight detection mechanism against Sybil attack in wireless sensor network. KSII Trans Internet Inf Syst 2015;9(9):3738–49.
[23] Sinha S, Paul A, Pal S. Use of spline curve in Sybil attack detection based on received signal power-new approach. Int J Recent Trends Eng Technol 2014;11(1):602–11.
[24] Zhu S, Setia S, Jajodia S. LEAP, efficient security mechanisms for large-scale distributed sensor networks. In: Proceedings of the tenth ACM conference on computer and communications security; 2003. p. 62–72.
[25] JSIM Simulator, http://www.physiome.org/jsim/download [accessed 26.12.15].
[26] Yu C M, Lu C S, Kuo S Y. Mobile sensor network resilient against node replication attacks. In: Proceedings of the IEEE conference on sensor, mesh and ad hoc communications and networks (SECON), 5; 2008. p. 597–9.

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011

Mojtaba Jamshidi received the B.S. degree in Computer Engineering from the Academic Center of Education, Kermanshah, Iran, in 2009, and M.S. degree in Computer Engineering from Islamic Azad University, Qazvin, Iran, in 2012. His research interests include computer networks, learning systems, security, data mining, and recommender systems.

Ehsan Zangeneh received the B.S. degree in Computer Engineering from the Academic Center of Education, 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. Currently, he is an Assistant Professor in Cyberspace Research Institute, Tehran, Iran. His research interests include computer networks, learning systems and soft computing.

M.R. 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. His research interests include, channel management in cellular networks, learning systems, parallel algorithms, soft computing and software development.

Please cite this article as: M. Jamshidi et al., A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks, Computers and Electrical Engineering (2016), http://dx.doi.org/10.1016/j.compeleceng.2016.12.011