ISSN (Print) : 0974-6846
**Indian Journal of Science and Technology,** *Vol 10(3), DOI: 10.17485/ijst/2017/v10i3/110641 January 2017* ISSN (Online) : 0974-5645

# Detection of Sybil Nodes in Wireless Sensor Networks

**Kumar Debasis[1], M. P. Singh[1], Prabhat Kumar[1] and Sachin Bhaskar[2]**

[1]Department of Computer Science and Engineering, National Institute of Technology Patna - 800005, Bihar, India; devkunu@gmail.com, mps@nitp.ac.in, prabhat@nitp.ac.in
[2]IMPACT College, Patna – 801503, Bihar, India; sachinbhaskar007@yahoo.com

## Abstract

Wireless sensor networks are prone to different types of attacks from malicious sources. These attacks are aimed at generating incorrect data or modifying legitimate data that is in transit in the network. A Sybil attack is a type of attack on WSNs where a malicious node either fabricates some new identities for itself or steals identities of some legitimate nodes. It can be countered in three basic methods. The first method verifies if a particular identity actually belongs to a real unique entity. The second method manages the cost and profit of acquiring identities. The third method focuses on confining the effects of the Sybil attack. However, these methods may not be used in wireless sensor networks directly. This is so because there are restrictions on the computational and storage capabilities of sensor nodes. Currently some of the principal detection schemes that are used in wireless sensor networks are radio resource testing, verification of key sets for random key predistribution, registration and position verification. The radio resource verification defense can be broken with custom radio hardware, and validation may be costly in terms of energy. Position verification can only put a limitation on how many Sybil nodes an attacker can generate unless it is able to very precisely verify node positions. Node registration needs human effort to securely add nodes to the network, and involves a way to securely maintain and query the current known topology information. The proposed method proposes a new approach to authenticate a sensor node based on a cryptographic hash function. It detects Sybil nodes so that they are isolated and any type of Sybil attack is avoided. This method performs well in terms of energy and memory usage when compared to previous methods.

**Keywords:** Cluster Head, Hash, Random String, Sybil Attack, Sybil Node

## 1. Introduction

A Wireless Sensor Network is a wireless network that comprises independent devices that are committed and spatially distributed. These devices utilize sensors to analyze the physical or environmental conditions around them. In WSNs all the communications are based on the node identities. In a Sybil attack a malicious node acquires some fake identities so that the one-to-one mapping that exists between entity and identity is broken down. The majority of WSN applications rely on accuracy of the data, correct decisions made through voting schemes and fair allocation of resources. Hence identification of Sybil nodes is a major issue in WSNs. In this paper a model has been proposed based on a hashing based scheme to detect Sybil nodes in wireless sensor networks. A hash value is stored at both the ends (cluster head and sensor node) and is replaced after each communication by a new value. This scheme is basically verifying the trustworthiness of the nodes sending data to the cluster head. So, for each communication, cluster head will match the hash value at its end with the hash value calculated from the packet sent by the node.

## 2. Problem Definition

In wireless sensor networks packets are broadcasted in an open medium, and chances of altering them or eavesdropping on them are maximum. One of the common attacks is Sybil attack in which multiple fake

* *Author for correspondence*

identities (Sybil nodes) are acquired by a single malicious node. These Sybil nodes communicate with legitimate nodes with the motive to send erroneous packets to the cluster head, alter the contents of the packets forwarded by legitimate nodes, etc. Some of the consequences of such attacks are inaccurate data aggregation, biased voting results, unfair resource allocation etc. The challenge is to identify such Sybil nodes in WSNs and discard all the packets that originate from them or are altered by them.

# 3. Related Work

In a Sybil attack a malicious node acquires a number of fake identities, either by creating them or by stealing the identities of legitimate nodes that exist in the network. Eventually the malicious node which is a single physical device owns multiple identities. The extra identities that the node procures are called Sybil nodes. A Sybil attack may be launched in three possible ways which have been discussed below in detail.

## 3.1 Sybil Attack Taxonomies

### *3.1.1 Direct or indirect communication*

In case of direct communication Sybil nodes talk to legitimate nodes directly[3]. The messages sent by a legitimate node to a Sybil node are actually heard by a malicious node that owns the identity of that Sybil node. Similarly, messages from a Sybil node that arrive at a legitimate node are actually sent by a malicious node.

In case of indirect communication, the Sybil node communicates with the legitimate node via another malicious node. In other words, a legitimate node cannot talk to a Sybil node directly. A message directed towards the Sybil node is first received by a malicious node which then forwards it to the intended Sybil node.

### *3.1.2 Stolen or fabricated identities*

A Sybil node has two options for getting an identity for itself. The first option is that it fabricates a new identity for itself. In the second option the identity of a legitimate node is stolen by the Sybil node. The easiest way to get an identity is to get the identity of an impersonated node, if such a node exists in the network. The identity theft can remain undetected if the impersonated node is destroyed or temporarily disabled from the network.

### *3.1.3 Simultaneous or Non-simultaneous*

In first case the attacker attempts to make all his Sybil nodes participate in the network at the same time[3]. A single physical device can participate in the network using only one identity at a time. But here the malicious node switches between these identities to give an impression that all the identities are active at the same time.

In second case, some node identities are used in one time interval, and others are used in the next time interval. Also, if the attackers have several compromised nodes, then these nodes can swap their identities on a periodic basis and remain undetected.

## 3.2 Sybil Attack Applications

The following are some of the basic applications of Sybil attacks for wireless sensor networks:

### *3.2.1 Routing*

Sybil attacks have proved to be successful in damaging the routing protocols used in wireless sensor networks. The multi-path and disparity routing algorithms are particularly vulnerable if the path consisting of multiple segments goes via a malicious node that possesses multiple Sybil identities[3]. This attack can also affect geographical routing protocols when a Sybil node appears in several locations at once instead of appearing in one place.

### *3.2.2 Data Aggregation*

Sensor networks use query protocols, which instead of replying with the reading of each individual sensor node calculate the aggregate of the readings of multiple sensor nodes and return that value. A small number of malicious nodes returning incorrect readings are not able to affect the aggregate reading by a wide margin. But by means of a Sybil attack a node is able to contribute to the aggregate numerous times, thereby affecting the aggregate sensor reading.

### *3.2.3 Voting*

In some applications, sensors are used to participate in voting, in order to facilitate decision making. Because of the ability of Sybil nodes to replicate identities, such nodes can affect the outcome of any vote.

#### *3.2.4 Misbehavior Detection*

It may become difficult to isolate a misbehaving node when the network has Sybil nodes present in it. An attacker with numerous Sybil identities makes it difficult for the system to take action against any particular Sybil node by making no node misbehave enough to be isolated. Even if the system takes action against any felonious node the attacker is not really affected as he or she can continue using other Sybil identities to carry on the attacks.

#### *3.2.5 Fair Resource Allocation*

In networking, resources are often shared among the nodes and often called on a per node basis. For example, a wireless channel using TDMA MAC may assign the same channel to different users for short intervals of time (time slots). The Sybil attacker can disrupt the fair allocation of resources by assigning a resource to the same node several times by changing its identity[3].

### 3.3 Sybil Attack Defenses

A Sybil attack can be countered in three basic ways. The first method is to verify if a particular identity actually belongs to a real unique entity. It can be achieved either through a reactive or a proactive way. In the reactive way, the system checks if an entity has provided adequate features to differentiate it from every other entity present in the network [2, 4]. One way to accomplish this is a resource test in which it is verified if each identity owns the same number of resources as the single physical device it is linked to. If there is any divergence it is inferred that the corresponding node has been compromised. In the proactive way, the central authority assigns an identity to every entity before it joins the network. It is now the responsibility of the central authority to provide protection against Sybil attacks.

The second method to counter the Sybil attack is to manage the cost and profit of acquiring identities[5, 6]. The inspiration behind Sybil attack is to cause maximum damage by forging as many fake identities as possible. One way to demoralize an attacker is to inflict higher costs on acquiring an identity, and limit the profits of possessing multiple identities.

Finally, some studies moved their focus to confining the effects of the Sybil attack owing to the efforts in completely isolating it. One example of this is limiting the maximum number of Sybil identities that can be created [7, 1, 8]. Unfortunately, the above methods cannot be used in wireless sensor networks directly, due to the restrictions on the computational and storage capabilities of sensor nodes. Presently the chief detection schemes in WSN include radio resource testing, verification of key sets for random key predistribution, registration and position verification[3]. Many of these detection schemes do not perform well in terms of energy and memory usage.

## 4. Proposed Model

Security is a major concern in wireless sensor networks and Sybil attack is one of the major attacks on WSNs. It has very adverse effect on the network and can cause damage to the entire network. This motivates us to bring up some kind of secure communication which would be reliable and robust. The ability to detect and deal with Sybil nodes will enhance the accuracy of data and the efficiency and robustness of network. These are also the characteristics that motivate us to find an efficient technique to prevent Sybil attacks.

### 4.1 System Model and Preliminaries

This experiment considers a big sensor network with numerous sensor nodes that are densely deployed. Each of these sensor nodes is allotted a unique identification number. As the nodes are densely deployed their sensing regions overlap each other. This causes an event to be sensed by multiple nodes at the same time. Sensor nodes have restricted communication and computation abilities but base stations have no such limitations. The network is a collection of clusters. There is one hop communication in clusters, and hence no intermediate node exists between a node and cluster head.

### 4.2 Distributing Random Strings

The proposed solution is based on sharing of a random string for the first communication. A unique Random string will be sent by the cluster head to each sensor node present in that cluster. This string will be used by a sensor node for the first communication as shown below:

- Sensor nodes have tables with entries < *id, rs*> (*id* = identification & *rs* = random string) initially before any communication, and cluster head will keep entries of all sensor nodes present in the cluster with their corresponding random strings.
- While communicating first time, each sensor node will fetch *rs* and calculate hash of (*rs* + *data*) say *H1*.

Now sensor node will update the entry in the table as < *id, HI*> and packet will be transmitted to cluster head.

### 4.3 Verification Process

Sensor node has calculated message digest for the first communication and the packet's format is <*id, H1+data*>. Now sensor node will send the packet to the cluster head and the process shall continue as follows:

- After receiving packet from the node, cluster head will separate *H1* from data, and retrieve the *rs* corresponding to the node's *id*.
- Now cluster head will calculate hash of (*rs* + *data*) and resulting hash say *H2* will be compared with *H1*.
- For a sensor node to be legitimate, both the hash values should be equal, because hashing algorithm is shared between cluster head and sensor nodes.
- If *H1* and *H2* are equal then cluster head will accept the packet and update the entry table with the hash value.
- If *H1* and *H2* are not equal then the node will be declared as compromised node or Sybil node.

## 5. Simulation and Performance Analysis

The simulation and experimental results are shown in the following subsections.

### 5.1 Simulator

For simulation purpose NS2 simulator is used whose architecture is shown in Figure 1.

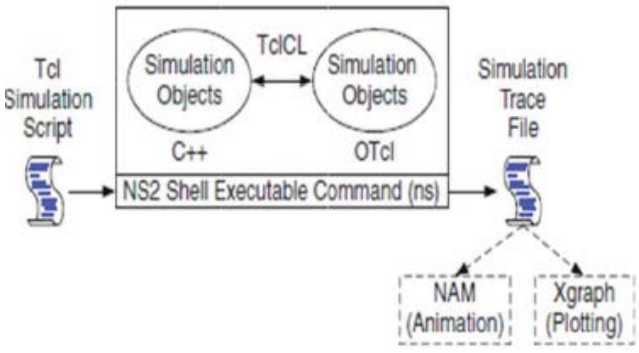


**Figure 1.** NS2 Simulator Architecture [9].

### 5.2 Performance Evaluation

#### *5.2.1 Energy Consumption*

##### *5.2.1.1 Energy Consumption When No Security Algorithm is Used*

In this paper, energy consumption has been calculated for 8 communications at a time interval of 0.5 sec. Initially all the nodes are presumed to possess the same energy level and are assigned 1 unit each. After 8 communications, the energy graph as obtained has been shown in Figure 2. It shows that sensor node 1 is having 1 unit of energy at t = 0 sec. At t = 1, t = 1.5, t = 3.5 and t = 4, sensor node 1 sends data to the cluster head and remaining nodes sense the data; energy consumption is found decreasing linearly almost. At t = 2 and t = 2.5 Sensor node 2 is sending data; energy consumption at sensor node 1 is observed to be minimum as it is not participating in communication with the cluster head. At t = 3, an adversary tries to send some data to cluster head by forging the identity of a legitimate node; energy consumption is found to be maximum.

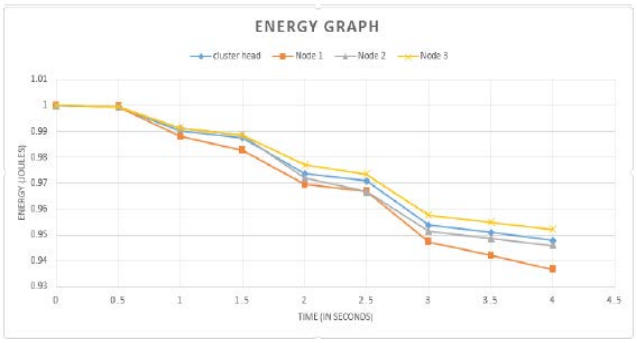


**Figure 2.** Energy Consumption in CH and Nodes When No Security Mechanism is Used.

##### *5.2.1.2 Energy Consumption in Proposed Solution*

In this paper, energy consumption is calculated for 8 communications at a time interval of 0.5 sec. Initially all the nodes are assumed to have same energy level and are assigned 1 unit each. After 8 communications, the energy consumptions graph as obtained has been shown in Figure 3. It shows that sensor node 1 has 1 unit of energy at t = 0 sec. At t = 1, t = 1.5, t = 3.5 and t = 4, sensor node 1 sends data to the cluster head and remaining nodes sense

the data; energy consumption is found decreasing linearly almost. At t = 2 and t = 2.5, Sensor node 2 is sending data; energy consumption at sensor node 1 is observed to be minimum as it is not participating in communication with the cluster head. At t = 3, an adversary tries to send some data to the cluster head by forging the identity of a legitimate node; energy consumption is slightly more when compared to the previous case though the proposed security algorithm is used.

#### *5.2.1.3 Energy Consumption Comparison*

After comparison, it is found that energy consumption is slightly more in the proposed solution. The Table given below shows that extra energy consumption is ranging from 2.7 % to 5.7 %.

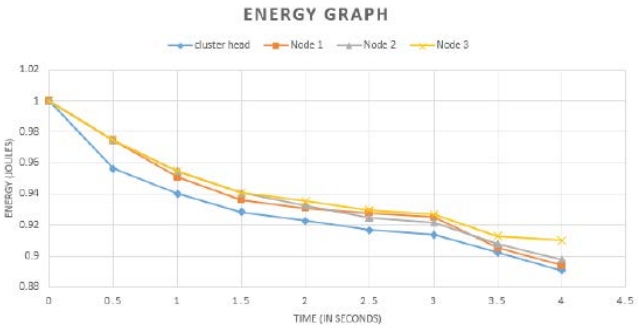


**Figure 3.** Energy Consumption in CH and Nodes in Proposed Security Mechanism.

**Table 1.** Energy Comparison

| | Energy E1 | Energy E2 |
|---|---|---|
| Cluster Head | 0.947888 | 0.890696 |
| Node 1 | 0.931192 | 0.893968 |
| Node 2 | 0.937908 | 0.906707 |
| Node 3 | 0.938036 | 0.910112 |

E1 - Energy at CH and Nodes when no security mechanism is used
E2 - Energy at CH and Nodes in Proposed Security mechanism

Figure 4. compares the energy consumption graphs of the cluster head and a node when they are not implementing the algorithm with the graphs when they are implementing the algorithm. It shows that when the algorithm is implemented the increase in energy consumption of the cluster head and the node is very less.

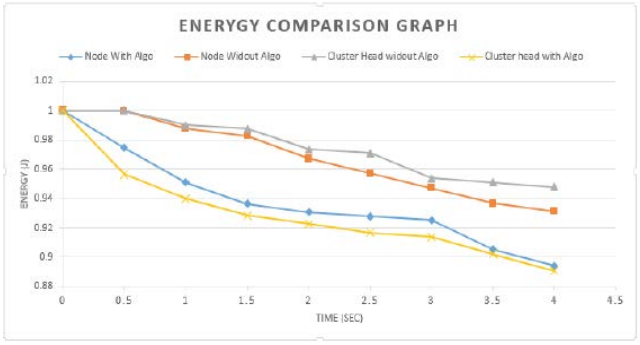


**Figure 4.** Final Energy Comparison.

### *5.2.2 Memory Consumption*

#### *5.2.2.1 Memory Consumption When No Security Algorithm is Used*

This paper presents the comparison on the basis of the assumption that node will send data of 3 bytes (assuming character data) and identity of each node is an integer i.e. 2 bytes in length. The size of the packet without any security algorithm will be as follows:

- Sample data size: 3 bytes
- Size of identity: 2 bytes
- Total size of the packet: 5 bytes = 40bits

#### *5.2.2.2 Memory consumption in Proposed Solution*

In Proposed solution size of the packet is larger as compared to the previous case because hash based approach is used. The extra space used is needed to store the hash value in the table. Here MD5SUM hashing algorithm is used that generates 128 bits of message digest.

- Sample data size: 3 bytes
- Size of identity: 2 bytes
- Size of the hash or message digest: 16 bytes
- Total size of the data: 3+2+16= 21*8= 168 bits

#### *5.2.2.3 Memory Consumption Comparison*

Since only hash is occupying extra space so extra memory used in proposed approach is 16 bytes.

## 6. Conclusion and Future Work

This paper discusses the types and effects of Sybil attacks and proposes a solution to detect and deal with them. The proposed security algorithm is a hash based approach in which the cluster head compares the hash value that it computes from the packet sent by the sensor node with the hash value stored in its own table. If both the hash values match then it is inferred that the packet has been received without any alteration from a legitimate node, in which case it is accepted. If both the hash values do not match then it is inferred that the packet has originated from a Sybil node or has been manipulated by it, in which case it is rejected. The energy and memory consumption of the proposed solution is slightly more when compared to the scenario where no security algorithm is used. However, there are a few issues that need to be resolved for e.g. this solution works only for one hop communication in clusters. In future, this algorithm can be modified to further reduce the energy and memory consumption and make it effective for multihop communication in clusters.

## 7. Acknowledgement

All authors have contributed equally in this paper. We thank the four B.Tech. students namely Md. Rijvan Khan, Ankur Agrawal, Omaram Khot and Ajeet Kumar for helping us in simulation.

## 8. References

1. Yu H, Kaminsky M, Gibbons PB, Flaxman A. Sybilguard: Defending Against Sybil Attacks via Social Networks. ACM SIGCOMM Computer Communication Review. ACM. 2006 September; 36 (4): 267–78.
2. Douceur JR. The Sybil Attack. International Workshop on Peer-to-Peer Systems. Springer Berlin Heidelberg. 2002 March; 251–60.
3. Newsome J, Shi E, Song D, Perrig A. The Sybil Attack in Sensor Networks: Analysis and Defenses. Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks. ACM. 2004 April; 259–68.
4. Freedman MJ, Morris R. Tarzan: A Peer-to-Peer Anonymizing Network Layer. Proceedings of the 9th ACM Conference on Computer and Communications Security. ACM. 2002 November; 193–206.
5. Cheng A, Friedman E. Sybilproof Reputation Mechanisms. Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems. ACM. 2005 August; 128–32.
6. Feldman M, Lai K, Stoica I, Chuang J. Robust Incentive Techniques for Peer-to-Peer Networks. Proceedings of the 5th ACM Conference on Electronic Commerce. ACM. 2004 May; 102–11.
7. Gatti R, Lewis S, Ozment A, Rayna T, Serjantov A. Sufficiently Secure Peer-to-Peer Networks. Workshop on the Economics of Information Security. 2004 May.
8. Kamvar SD, Schlosser MT, Garcia-Molina H. The Eigentrust Algorithm for Reputation Management in p2p Networks. Proceedings of the 12th International Conference on World Wide Web. ACM. 2003 May; 640–51.
9. Article title. http://www.ns2blogger.in/p/n.html. Date accessed: 09/02/2017.