An Approach to Mitigating Sybil Attack in Wireless Networks using ZigBee

Gunhee Lee, Jaesung Lim, Dong-kyoo Kim
School of Information and Communication
Ajou University
Suwon 443-749, South Korea
Email: {icezzoco, jaslim, dkkim}@ajou.ac.kr

SungHyun Yang
Dept. of Electrical Engineering
Kwangwoon University
Seoul 139-701, South Korea
Email: shyang@daisy.kw.ac.kr

MyungHyun Yoon
Ubiquitous Computing Research Center
Korea Electronics Technology Institute
Seongnam 463-816, South Korea
Email: yoon@keti.re.kr

***Abstract*—Various sensors employing ZigBee protocol send environment information to a home gateway. The home gateway manages every devices, sensors and services according to the information. However, if an attacker fabricates the information with malevolent purpose, the system may perform any inadvertent service, which may be harmful to users. Sybil attack is one of serious threats that uses such vulnerability. In this paper, we propose a Sybil attack detection method and its response method to mitigate the effect of the attack in wireless network using ZigBee. The proposed method employs a challenge-response approach. A verifier send a node a request that corresponds to an identity, and the receiver should respond the request properly within threshold time. The performance of the proposed method is reasonable for applying it to the practical networks. This will help the wireless networks using ZigBee to enhance its security.**

***Keywords*—Sybil Attack, ZigBee Protocol Suite, Wireless Sensor Network, Security**

1. INTRODUCTION

In these days, ubiquitous computing becomes one of the most considerable computing paradigms. Many information services are researched and developed based on the ubiquitous computing environment. According to this change, home network has been converted into smart home. In the smart home, various network technologies work separately and cooperatively. Among them ZigBee based on the IEEE standard 802.15.4 is an important suite of communication protocols that is used to control devices and equipments [1]. Various sensors in the home can employ ZigBee protocol suite in order to send a home gateway information about environment and systems such as temperature, illuminance and system status. The home gateway manages every devices, sensors and services according to the information. By using this kind of approaches, every user's purpose can be achieved with the minimum amount of manual user intervention.

However, if the information is incorrect or somebody fabricates the information with malevolent purpose, the system may perform any inadvertent service, which may be harmful to users. Sybil attack is a threat that uses such vulnerability [2]. A malicious attacker compromises a sensor which has virtually multiple identities in the network. By using it, the attacker can generate multiple legal messages having different sources without any doubt about DoS (Denial of Service) attack. This is a dangerous and important threat to any sensor networks. For example, there is a building having a fire alarm system that consists of some sensors. If an attacker compromises a sensor, he/she can register the sensor as a new node. After that, he/she is able to register several virtually created identities as well. The server is not aware of the truth that some of the newly joined identities are spoofed. After that the attacker can send several false alarms to the server, and then the server makes the wrong decision and responds to a fire that is not really happened.

In this paper, we propose a Sybil attack detection and response method in ZigBee networks. The proposed method uses resource testing approach [3]. The system measures how long it takes each node receiving a request to treat it properly. Additionally we employ a response method that isolates the attacker's node from the network by banning traffic from the malicious one. The proposed method is more effective and accurate, and it will enhance the quality of security in ZigBee networks. Furthermore, because the ZigBee network employs tree topology, a master node does not need to monitor every node in the network. Instead of it, each node detects Sybil attack occurring by one of their child nodes. Thus the burden task of detection will be distributed, and it will reduce master node's load and make the method efficient.

This paper is organized as follows. Section 2 summarizes the Sybil attack and previous approaches that detect the attack. We describe the proposed detection and response method against the Sybil attack in section 3. This is followed by the analysis of the proposed model in section 4. Section 5 concludes.

2. RELATED WORKS

2.1. ZigBee Network

The specification of ZigBee supports three types of topology such as star, tree, and mesh topologies [1]. Among them, we focus on the tree topology in this paper. In ZigBee protocol suite, the network is controlled by the ZigBee coordinator. It is responsible for initiating and maintaining the devices on the network. ZigBee routers may be employed to extend size of network. It maintains the network by cooperating with ZigBee coordinator. For the security of the network, a trust

center (TC) should be used in a network. ZigBee coordinator is able to become the trust center. The TC has three roles such as trust manager, network manager, configuration manager. A device trusts its TC to identify the device that configure and manage the network. The network manager is responsible for distributing and maintaining the network keys that shares with all nodes in a network. The configuration manager enables end-to-end security between devices by distributing master keys or link keys. Each device should share a master key with the TC, and two devices that want to communicate with each other should share a link key between them.

2.2. The Sybil Attack

The Sybil attack is launched by a malicious node that illegally acquires multiple identities. The malicious node is able to deceive the other nodes into thinking there are several normal nodes including itself. Figure 1 shows an example of the Sybil attack. After the malicious node m_1 joins the network, it continuously registers three virtual identities v_1, v_2, and v_3 as normal ones. As the result of it, the network believes four new nodes are joined. The network will provide any possible services to them and consume information from them. In the example, the malicious node and virtual identities send normal messages to a central node that decides whether a intrusion is launched or not, while other nodes send alert messages. Since number of alert messages is smaller than that of normal ones, the central node will decide there is no attack.

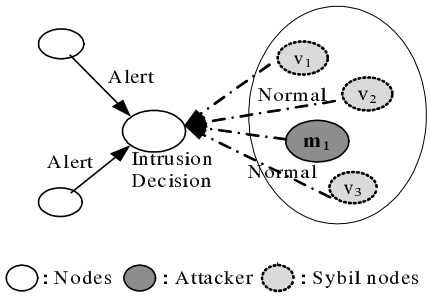

Figure 1. An example of the Sybil attack

In many application areas, the attacker can benefit from the Sybil attack such as distributed storage, routing, data aggregation, voting, intrusion detection, and resource sharing [3]. The summaries of threat in each application area are as follows.

- *Distributed storage*: In distributed storage system, the data will be kept by several nodes by using replication and fragmentation [2]. However, the attacker can easily harm the system since it could store large part of data in a node by launching Sybil attack.
- *Routing*: In multi-path routing protocol, the malicious node can make the protocol decide a falsified multi-path [4], [5], [6]. In fact, every selected path go through the malicious node.
- *Data aggregation*: In a kind of sensor network, a sink node or a base station gathers information from sensors, and it computes a aggregate with collected information [7]. However, a malicious node may be able to send a base station information several times by using multiple identities. As a result of it, the aggregate might be computed as intention of the attacker.
- *Voting*: In the voting system, it ballots nodes in a network on whether to perform a task [8]. The malicious node, however, may be able to make the system come to a wrong determination since it has many identities and each identity votes for negatives.
- *Intrusion detection*: In some cooperative malicious activity detection approaches, each intrusion detection agent (or sensor) can alert different agents to a intrusion [9]. According to the alarm, other nodes notice the intrusion. However, a malicious node having many identities generates many false alerts that have different source identities. As a result of it, other nodes believe that is true.
- *Resource sharing*: The Sybil attack may result in DoS situation in resource sharing application. If a malicious node and its virtual identities request the same resource, the other nodes may not be able to use the resource.

Figure 2 shows an example of the Sybil attack in ZigBee network using tree topology. A virtual identity may be able to join the network as a sibling or a children of the attacker. Thus, based on the position of the virtual identities in a tree, the attack can be classified as three types as shown in Figure 2.

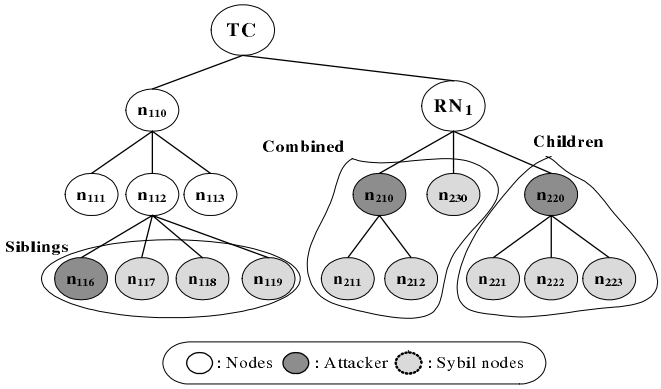

Figure 2. An example of Sybil attack in a ZigBee network

2.3. Previous Approaches

Douceur introduced the sybil attack and suggested a detection method [2]. In the method, the verifier tests whether identities is a physical entity or not. To do so, it verifies that each identity has enough tested resource to be classified as an physical entity. employs a resource testing approach. Since identities, which are used to the Sybil attack, correspond to a physical entity, it is difficult to respond to the test as a normal identity. Computation, storage, and communication are suggested to use for this purpose. However computation and storage is not suitable for current computing devices because the computing power and the storage capacity of the mobile devices are big enough nowadays. In addition to this, the test of communication is also not suitable for wireless ad

hoc networks since there is no significant difference between processing time of the respond and the propagation delay.

Newsome et al. suggested another resource testing approach [3]. In the method, a centralized administrative module assigns a radio channel to a node, and it sends a test message to a randomly selected node. The selected node should answer to the message. Since the attacker does not receive from other channel, the verifier can detect the malicious node. However, since the assumption that every node can use an assigned radio channel is not practical, it is not suitable for the wireless ad hoc network.

3. MITIGATING SYBIL ATTACK

3.1. Detecting the Sybil Attack

In order to detect the Sybil attack in a network, we should identify nodes having multiple identities. For this, in the proposed method, we employ and enhance the resource testing approach. That is, if a node receives a request for testing, it should respond to the request. Handling the request properly takes enough time to differentiate a malicious node from normal ones. Since the malicious node manages several identities, it should handle more requests than other nodes. Because of this reason, in the case of the malicious node, managing each request will take a little more time in comparison with other nodes. However the dissimilarity is too small to distinguish attack from normal, so we make the request difficult to respond to it in order to increase the difference.

In the proposed method, we assume that the ZigBee network employs the tree topology. With this assumption, the decision of testing point is easier than mesh topology and we can employ distributed approach. Since each parent node may be able to manage its child nodes in tree topology, the test can be performed by a parent node. However, the proposed approach is able to apply to the ZigBee network using mesh topology with slight change if the ZigBee coordinator takes charge of the test.

In addition to this, according to the specification of ZigBee, a unique link key should be shared between a child node and its parent in order to communicate in secure. For example, if a node has 5 child nodes, it has different 5 link keys. Each node also has a reputation table that contains number of bad activities of each child node. The table consists of tuples of the form (id_i, mal_i), where id_i is an identity of node n_i and mal_i is a counter that keeps the number of malicious activities of node n_i.

When a node p_i wants to detect a malicious node among its children $(c_1, c_2, ..., c_n)$, the node p_i send a request to randomly selected one of its child nodes. The request consists of an encrypted nonce r and a number k. The parent p_i encrypts the nonce k times with the link key of selected child node. The request message M_{req} is constructed as follows, where key_i is a link key of a child node c_i.

$$M_{req} = [REQUEST, k, E^k_{key_i}\{id_{p_i}, id_{c_i}, r\}]$$

In the meantime, when a child node c_i receives the request, it decrypts the encrypted nonce k times with its link key, and it encrypts a number $r+1$ k times with its link key. The node c_i sends the response to the node p_i. The structure of the response message is as follows.

$$M_{res} = [RESPONSE, k, E^k_{key_i}\{id_{p_i}, id_{c_i}, r+1\}]$$

The parent p_i waits for the response until a predefined timer t_h is expired. The timer t_h is calculated by following equation (1), where k is number of encryption or decryption, t_{enc} is an encryption time, t_{dec} is a decryption time, t_{delay} is the maximum network propagation delay, and α is the other processing delay.

$$t_h = k(t_{enc} + t_{dec}) + 2t_{delay} + \alpha \quad (1)$$

The response time t_{resp} is defined as (2), where t_{send} is the time of sending a request and t_{recv} is the time of receiving a response.

$$t_{resp} = t_{recv} - t_{send} \quad (2)$$

If the node c_i is a normal node, the response time will be lesser than the threshold time t_h. However, if the node c_i is a malicious node, the response time will be larger than the the threshold time t_h. Since the malicious node does not know what the correct key is, it tries to use each by each in order to decrypt the request properly. As a result of it, the t_{resp} of the malicious one is greater than the threshold t_h. Figure 3 shows the schematic of the proposed method. The left part of the Figure 3 shows the pattern of a response from the normal node, while The right part depicts the pattern of that from a malicious node.

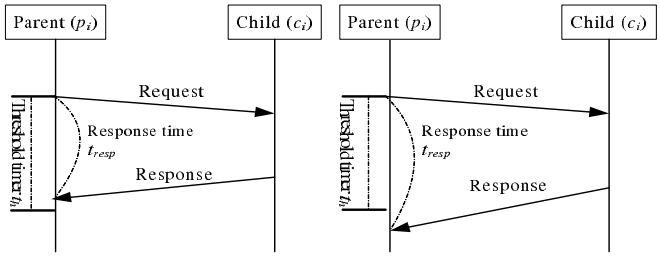

Figure 3. The comparison of the response time between normal node and malicious one in the proposed method

If the response is arrived within a valid time, the parent node p_i verifies that the response is valid. In order to verify the response, the node p_i acquires the random number by decrypting the response k times with the link key of the responder. If the number is the same as $r+1$, the response is valid. If the response is arrived after the threshold t_h or the response is not valid, the parent node p_i will drop the response and regard the node c_i as a candidate of the malicious node.

When a response from node c_i is categorized as invalid one, the node p_i searches the reputation table for the identity of the node c_i, and it increases the number of malicious activities of c_i, which is the object mal_i of a tuple, by 1. If the mal_i

is greater than a predefined threshold h, the node p_i regards c_i as a malicious node, and it reports on the fact to the trust center of the network.

3.2. Isolating a Malicious Node

Isolation of the detected attacker from the network will be performed at two different situation as follows. First, each node can detect the malicious node by performing the test we described previous section. Second, each node checks whether the sender is in its black list when it receives a packet from a node.

For the later one, in order to isolate the malicious node, each node should maintain a black list that contains information of the detected attackers. The information consists of MAC addresses of the attackers. With this list, when a node receives a message from another node, the receiver searches the black list for the identity of the sender. If there exists the identity in the list, the node drops the packet, and it sends *Device-Update* command containing the MAC address of the attacker to the trust center in order to remove the attacker from the network. The trust center sends a *Remove-Device* command to other nodes in the network.

For the former one, we consider whether the malicious node is a leaf node in the tree or not. If the attacker is a leaf node, the TC sends a *Remove-Device* command to its parent node, and inserts the MAC address of the malicious node into the black list. Otherwise, the TC makes the parent of the malevolent node ban traffic from the malicious one, and removes the bad node by sending a *Remove-Device* command. Then every child node of the attacker looses their connection, so they automatically try to join the network again. However their requests might be dropped because they are forwarded via the malicious node. Thus the TC makes neighbors of children broadcast the address of the attacker. Child nodes send a join request to neighbors except for the malicious node.

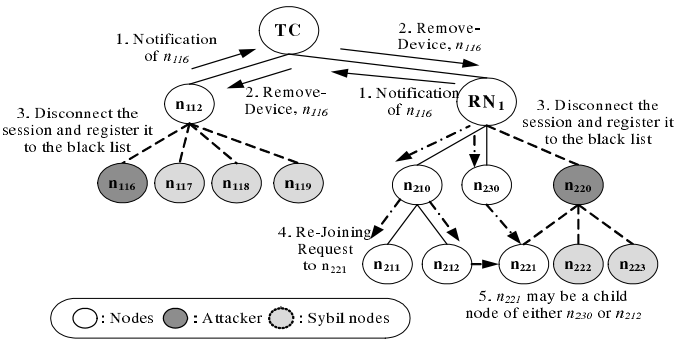

Figure 4. The example of the attacker isolation procedure.

Figure 4 shows the process of the isolating the malicious node. In the left sub-tree of the trust center, a node n_{112} detects the malicious node n_{116}, and the malicious node and Sybil identities are removed from the network. In the right sub-tree of the trust center, a router node RN_1 detects the malicious node n_{220}, and the trust center makes the router node eliminate the malicious node and its virtual identities from the network. However, the node n_{221}, which is one of child nodes of the malicious node, is a normal node, so it should join the network again. As mentioned above paragraph, it will rejoin the network by sending the joining request to other normal nodes such as either n_{230} or n_{212}.

4. Analysis of the Proposed Method

In order to analyze the effectiveness of the proposed method, we investigate the probability of detecting malicious node properly. Equation (3) is the probability P_f of the false negative, where n is the number of child nodes of a parent node, m is the number of virtual identities including the malicious node, and h is the threshold of the malicious activities. That is, the probability that the parent regard the response from an attacker as a valid one since the attacker decrypts the request with the first selected key.

$$P_f = \frac{m}{n}(\frac{1}{m})^h \tag{3}$$

Thus, the detection rate P_d is defined as (4).

$$P_d = 1 - \frac{m}{n}(\frac{1}{m})^h \tag{4}$$

Figure 5 shows the variation of the detection rate. There are 8 curves, and each curve means the variation of the detection rate according to the increasing number of tests under the specific number of Sybil identities including the malicious node. In each case, the number of child nodes is set to 10. As shown in Figure 5 the detection rate is at least 90% after the first test regardless of the number of Sybil identities. Furthermore, if there are two Sybil identities among ten normal nodes, the detection rate is greater than 99% after the fifth test. In the other cases except for the case that the number of Sybil identities is 3, the detection rate is greater than 99% after the third test. According to this result, we suggest that the threshold h for the counter of malicious activity is set to 3.

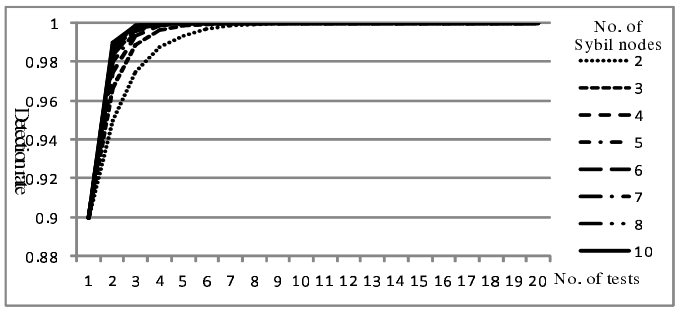

Figure 5. The variation of the detection rate according to the change of the number of Sybil identities and the number of tests.

For the proposed method, each node should maintain two data structures such as a reputation table and a black list. For the reputation table, the amount of memory that may be used by each entry of the table is 7 bytes (6 bytes for the MAC address and 1 byte for the counter). Thus the total amount of

memory is up to $7l$ bytes for the table, where l is the number of child nodes.

For the black list, the amount of memory is 6 bytes for each entry of the list. Thus the size of memory we need amounts to $6j$ bytes if the number of nodes in the list is j.

Thus the total amount of the memory for two structures is $7l + 6j$. For instance, if there are 10 child nodes and 4 malicious nodes in the black list, the size of the table is up to 70 bytes and the size of the list is up to 24 bytes. Even if we consider a sensor for the node in the network, 94 bytes is reasonable amount of memory.

5. CONCLUSION

In this paper, we propose an effective approach that mitigates the Sybil attack in wireless networks using ZigBee. The proposed method employs a challenge-response approach. A verifier send a node a request that corresponds to an identity. Since the malicious node handles several nodes, it is difficult to decide an identity of the request. Thus the processing time of the request in the malicious node will be greater than that in a normal node. After detecting the malicious node, it inserts the malicious one into the black list. In addition to this, the traffic of the malicious node will be banned by dropping the packets from it. The detection rate of the method is reasonable for applying it to the practical networks, and the response of the method will mitigate the effect of the Sybil attack efficiently. Moreover, it needs small amount of the memory so that any device can use the proposed approach. It will help the wireless networks using ZigBee to enhance its security.

ACKNOWLEDGMENT

This research was supported in part by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-(C1090-0602-0011)). This work was also supported in part by grant No. 10024724 from the Growth Engine Technology Development Program and funded by Ministry of Commerce, Industry and Energy, Korea

REFERENCES

[1] ZigBee Alliance, "ZigBee Specification v1.0", ZigBee Alliance, June 2005.

[2] J. Douceur, "The sybil attack", Peer-to-Peer Systems: First International Workshop (ITPTS 2002), Lecture Notes in Computer Science 2429, Springer-Verlag, March 2002, pp. 251-260

[3] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses", In Proc. of 3rd IEEE/ACM Information Processing in Sensor Networks (IPSN'04), April 2004, pp. 259-268

[4] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing With Guaranteed Delivery in Ad Hoc Wireless Networks", Wireless Networks, 7(6), Kluwer Academic Publishers, November 2001, pp. 609-616.

[5] B. Karp, and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", Proc. of International Conference on Mobile Computing and Networking (MobiCom 2000), August 2000, pp. 243-254.

[6] Y. B. Ko, and N. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks", Wireless Networks, 6(4), Kluwer Academic Publishers, July 2000, pp. 307-321.

[7] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation Service for Ad Hoc Sensor Networks", ACM SIGOPS Operating Systems Review, 36(SI), ACM Press, 2002, pp. 131-146.

[8] B. Hardekopf, K. Kwiat, and S. Upadhyaya, "A Decentralized Voting Algorithm for Increasing Dependability in Distributed Systems", Joint Meeting of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001), July 2001.

[9] F. Kargl, A. Klenk, S. Schlott, and M. Weber, "Advanced Detection of Selfish or Malicious Nodes in Ad Hoc Networks", Security in Ad-hoc and Sensor Networks, Lecture Notes in Computer Science 3313, Springer-Verlag, 2005, pp. 152-165.