Volume 3, Issue 7, July 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Catching Packet Updaters by Using Authentication Scheme in Path-by-Path Method to Detect Fake Inserted Data Packets in Wireless Sensor Networks

M.Rajitha [*]
M.Tech 2nd year,
Dept of CSE,
ASIT, GUDUR
India

P. Venkateswara Rao
Professor & HOD,
Dept of CSE,
ASIT, GUDUR
India

A. Rama Mohan Reddy
Professor, Dept of CSE,
S.V. U. College of Engineering,
Sri Venkatesawara University,
Tirupathi, India

Abstract - There are primarily two common attacks that interrupt the communication in wireless sensor networks, they are packet dropping and packet updating. To conquer these attacks many schemes are proposed, but only some are working capably to recognize the intruders that dropping packets but not the updaters. Criterion authentication mechanisms can't prevent this attack if the opponent has compromised one or small number of sensor nodes. For recognize updaters, we proposed an interleaved path-by-path AS that assures that the base station will recognize any fake insertion data packets when no excess of t nodes are compromised. Using AS at every node, Can recognize which packet is carrying Fake data. This scheme enables the base station to verify the authenticity of a report that it has received as long as the number of compromised sensor nodes does not go beyond a certain threshold. The scheme will filter out Fake data packets inserted into the network by compromised nodes before they arrive at the base station. So that we can stop the Fake data packets by using this scheme and can faintly modify the packets.

Keywords- Wireless sensor networks, packet updater, Authentication scheme (AS), Compromised node, Intruder.

I. INTRODUCTION

In wireless sensor network, sensor nodes will watch the atmosphere and sink will be the base station. Some nodes will be compromised by the intruder for some malicious purpose, after compromising one or several sensor nodes, an opponent may launch different attacks to interrupt the in-network communication. Among these attacks, two regular ones are packets dropping and updating packets, i.e., compromised nodes drop or update the packets that they are invented to forward. A simple yet useful method to hold both packet droppers and updaters, a routing tree fixed at the sink is established first. When sensor data are transmitted along the tree structure toward the sink, each packet sender or forwarder adds a small number of more bits, which is called packet grades, to the packet. The plan of the small packet grades is purposely planed such that the sink can gain very useful information from the grades. Specially, based on the packet grades, the sink will figure out the dropping ratio connected with every sensor node, and then runs node categorization algorithm to recognize nodes that are droppers/modifiers for sure or are unsure droppers/modifiers. As the tree structure vigorously changes every time period, behaviours of sensor nodes can be observed in a large variety of scenarios. As the information of node behaviours has been accumulated, the sink cyclically runs heuristic ranking algorithms to recognize most expected bad nodes from doubtful bad nodes. This way, most of the bad nodes can be slowly recognize with small fake positive.

But in the existing system only packet dropping has been discussed, for update our proposed scheme is interleaved path-by-path authentication scheme for filtering of inserted fake data in sensor networks. In this paper, we present a scheme for addressing this form of attack, which we call a *fake data insertion attack*. Our scheme enables the base station to verify the authenticity of a report that it has received as extended as the number of compromised sensor nodes does not exceed a certain threshold. Further, our scheme attempts to filter out fake data packets inserted into the network by compromised nodes before arrived the base station, thus saving the energy for relaying them. To defend against fake data insertion attacks, we present an authentication scheme in which at-least t +1 sensor nodes have to agree upon a report before it is sent to the base station. Further, all the nodes that are involved in relaying the report to the base station authenticate the report in *an interleaved, path-by-path* fashion. Here t is a security threshold based on the security requirements of the application under consideration and the network node density. Our scheme guarantees that if no more than t nodes are compromised, the base station will detect any fake data packets inserted by the compromised sensors. If every non compromised node on the path between a cluster head and the base station knows the ids of the nodes that are t + 1 paths away from it on the path, then B = t; otherwise, without this knowledge, B = (t ¡ 1)(t ¡ 2). We also propose a variant of this scheme which guarantees B = 0 but works for a small t.

In this paper, we focus on *fake data insertion attacks*, in which an attacker's aim is to cause fake alarms or to reduce already-constrained resources of forwarding nodes by injecting fake data. We assume that the compromised nodes can join together in their attacks. Our aim is to design an authentication scheme that can defend against fake data insertion attacks launched by up to t compromised nodes, where t is a system constraint. This scheme having some properties when there are no more than t compromised nodes. First, the base station should be able to detect any fake data packet inserted by a compromised node. Second, the number of paths before an inserted data packet is detected and discarded should be as small as possible. Third, the scheme should be proficient in calculation and communication with respect to the security it provides. Finally, the scheme should be strong to node failures.

II. Literature Survey

WSN security has in recent years been the subject of a number of proposals. Zhang and Lee [11] were among the first to study the problem of incursion detection in wireless ad hoc networks. The effectiveness to detect malicious packet droppers and updaters is limited without recognizing them and without them from the network. Researchers hence have proposed schemes to localize and recognize packet droppers. There are three types of existing approaches to detect packet dropping attacks. They are multipath forwarding approach, neighbour monitoring approach, and acknowledgement approach. Multipath forwarding [1], [2] is a broadly adopted countermeasure to moderate packet droppers, which is based on delivering disused packets along several paths. A simple yet useful method to hold both packet droppers and updaters, a routing tree fixed at the sink is established first. When a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet. The watchdog does this by listening promiscuous to the next node's transmit transmissions. If the next node does not broadcast the packet, it is misbehaving and the watchdog detects it[3].

III. Problem Statement

An opponent may compromise several sensor nodes, and then utilize the compromised nodes to inject fake data in to the network. This attack falls in the type of insider attacks. Criterion authentication scheme are not enough to prevent such insider attacks, since opponent knows all the keying material processed by the compromised nodes. so, for this a scheme is used to address the form of attack, which we call a fake data insertion attack. The scheme enables the base station to verify the authenticity of a report that it has received as long as the number of compromised sensor nodes does not exceed a certain threshold. The scheme will filter out fake data packets inserted into the network by compromised nodes before arrive the base station.

IV. Proposed Scheme

Here we focus on fake data insertion attacks, in which an attacker's aim is to cause fake alarms or to reduce already-constrained resources of forwarding nodes by injecting fake data. We assume that the compromised nodes can join together in their attacks. Our aim is to design an authentication scheme that can defend against fake data insertion attacks launched by up to t compromised nodes. Our scheme involves some steps.

1. *Node Initialization and Deployment:* Base station initiates the process by broadcasting a HELLO message, which is recursively forwarded to all nodes so that every node discovers the id's of the t+1 closest nodes that are on its path to the base station. On receiving a HELLO message from the BS, a node attaches its own id to the HELLO message before broadcasting it.
2. *Association Discovery*: The scheme restricts the max number of node id's that are included in a message to t+1. To achieve this, each node replaces the id of the node that is t+1 paths closer to the BS with its own id. On receiving the HELLO message, a CH assigns each of the t+1 id's in the message to one of its cluster nodes(including itself).After base station HELLO, the cluster head sends an ACK to the BS. ACK includes the cluster id, and id's of t+1 lower association nodes. When a node receives the ACK, it will check if all the node id's in the ACK are distinct if not drop the ACK. During the forwarding of ACK, the node id's are replaced in the opposite direction in BS HELLO step.

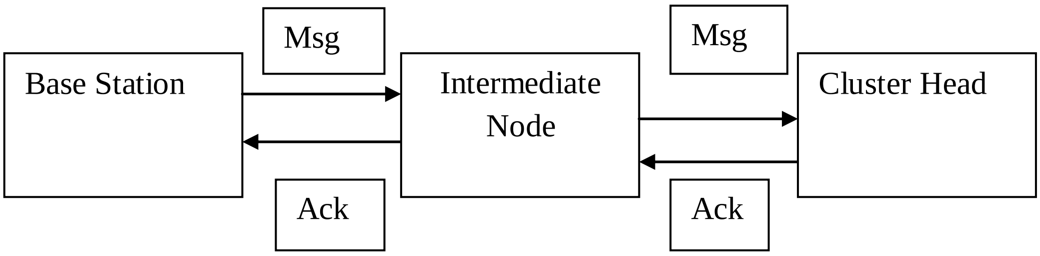

Fig. 1 System Architecture

CH first computes a MAC over S_n and the cluster id C_i, using its authentication key $K^a_{ch.}$CH generates an ACK, which includes its id CH, the above MAC, and an ordered list of id's of the t+1 cluster nodes that have discovered their upper associated nodes in BS HELLO phase. CH sends ACK to a node U_1, the node that previously forwarded the HELLO message to CH. The id list in the ACK message is{CH,V_3,V_2,V_1}. As a result, U_1 discovers that its lower association is V1, the last one in the list. Node U_1 then removes V_1 from the list and inserts its own id at the begining of the list. The id list it sends to U_2 is then {$U_{1,}$CH,V_3,V_2}.

In cryptography, a message authentication code is a small bit of information used to authenticate a message and to provide integrity and authenticity assurances on the message. Integrity assurances detect minor and intentional messages changes, while authenticity assurances affirm the messages origin.

$$mac=c(k, m)$$
$$m=\text{input message}$$
$$c=\text{mac function}$$
$$k=\text{shared secret key}$$

In Cryptography, (HMAC) is a specific construction for calculating a MAC involving a cryptographic has function in mixture with a secret cryptographic key. As with any MAC, it may be used to concurrently verify both the data reliability and authentication of a message.

$$hmac(k, m)=h((k\oplus opad) \,||\, h(k\oplus ipad) \,||\, m)$$
$$ipad=\text{inner padding}(1\times 5c5c5c...5c5c,\text{one block-long hexadecimal const})$$
$$opad=\text{outer padding}(0\times 363636....3636,\text{one-block-long hexadecimal const})$$

Algorithm

```
function hmac (key, message)
    if(length(key)>block size)then
        key=hash(key) //keys longer than block size are shortened
    end if
    if(length(key) <block size)then
        key=key||[$0\times 00$*block size - lenght(key)] //keys shorter than block size are zero-padded
    end if
    o-key-pad=[$0\times 5c$*block size] $\oplus$ key //where block size is that of underlying has function
    i-key-pad=[$0\times 36$*block size] $\oplus$ key
    return hash(o-key-pad||hash(i-key-pad||message)
end function.
```

3. *Report endorsement*: Sensor nodes generate a report when triggered by a special event 'E'(E typically contains an event type and a timestamp). When a node V agrees on an event E, it computes a MAC for E, using its authentication key K^{a}_{V} as the MAC key. In addition, node V computes another MAC for E, using pair-wise key shared with its upper association node U as the MAC key. Node V then sends an endorsement message to the CH that includes the two MAC's(Individual MAC and pair-wise MAC).CH collects endorsements from t+1 cluster nodes(including itself). It then compresses the t+1 individual MACs by XORing them to reduce size of the report. When a node receives R from its downstream node, it first verifies the authenticity of R based on its pair-wise key shared with that node

4. *Enroute-filtering*: CH collects endorsements from t+1 cluster nodes (including itself). It then compresses the t+1 individual MACs by XORing them to reduce size of the report. When a node receives R from its downstream node, it first verifies the authenticity of R based on its pair-wise key shared with that node. Then it checks the number of different pair-wise MAC's in R. If a node s(s<t+1) paths away from BS, it should see s pair-wise MACs otherwise t+1 pair-wise MACs. U will drop the report if any above check fails.

5. *Base station verification*: BS only needs to verify The base station BS only needs to verify the compressed MAC.BS computes t + 1 MACs over E using the authentication keys of the nodes in the id list, then XORs the MACs to see if it matches the one in the report. The BS can easily compute the authentication key of a node based on its id. If the report is authenticated and BS knows the location of every cluster node, it can locate these reporting nodes and then react to the event. On the other hand, if the verification fails, BS will discard the report.

V. Conclusion

Before various schemes have been proposed to hold both packet droppers and packet updaters, only some are good in finding packet droppers not for updaters and no schemes have been proposed. So, we presented a easy but useful authentication scheme called Authentication Code which is to prevent fake data insertion attacks in sensor networks. The scheme assures that the base station can sense a fake report when no excess of t nodes are compromised, where t is a security threshold. so that we can recognize the packets that consists of fake data by using authentication code at each and every node before packet arrive at the base station . By this scheme can indistinctly update the packets.

References

[1] M. Kefayati, H.R. Rabiee, S.G. Miremadi, and A. Khonsari, “Misbehaviour Resilient Multi-Path Data Transmission in Mobile Ad-Hoc Networks,” Proc. Frouth ACM Workspath Security of Ad-Hoc and Sensor Networks (SASN ’06), 2006.

[2] V. Bhuse, A. Gupta, and L. Lilien, "DPDSN: Detection of Packet- Dropping Attacks for Wireless Sensor Networks," Proc. Fourth Trusted Internet Workspath, 2005.

[3] S. Lee and Y. Choi, "A Resilient Packet-Forwarding Scheme Against Maliciously Packet-Dropping Nodes in Sensor Networks," Proc. Fourth ACM Workspath on Security of Ad Hoc and Sensor Networks (SASN '06), 2006.

[4] Mike Burmester and Tri van Le. Secure Communication in Ad hoc Networks, Proceedings of the IEEE Workshop on Information Assurance and Security, West Point, NY, pp. 234{241, June 2004.

[5] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *Proceedings of Advances in Cryptology (Crypto)*, pages 1–15, 1996.

[6] A. Tsirigos and Z.J. Haas, Analysis of Multipath Routing - Part I: The E®ect on the Packet Delivery Ratio, IEEE Transactions on Wireless Communications, Vol. 3, Issue 1, pp. 138-146, January 2004.

[7] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. IEEE SPNA*, May 2002, pp. 113–127.

[8] S. Zhu, S. Xu, S. Setia, and S. Jajodia. Establishing Pairwise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach. November 4-7, 2003.

[9] B. Yu, B. Xiao, Detecting selective forwarding attacks in wireless sensor networks, in: Proceedings of the Second International Workspath on Security in Systems and Networks (IPDPS 2006 Workspath), 2006, pp. 1–8.

[10] Y. Zhang, W. Lee, Incursion detection in wireless ad-hoc networks, in: Proceedings of ACM MobiCom, 2000, pp. 275–283.

[11] B. Xiao, B. Yu, and C. Gao, "Chemas: Recognize Suspect Nodes in Selective Forwarding Attacks," J. Parallel and Distributed Computing, vol. 67, no. 11, pp. 1218-1230, 2007.

[12] X. Zhang, A. Jain, and A. Perrig, "Packet-Dropping Opponent Identification for Data Plane Security," Proc. ACM CONEXT Conf. (CoNEXT '08), 2008.

[13] W. Du, J. Deng, Y. Han, and P. Varshney. A Pair-wise Key Pre-distribution Scheme for Wireless Sensor Networks. In
Proc. of 10th ACM Conference on Computer and Communications Security (CCS), Washington DC, October 27-31, 2003.