IOSR Journal of Electronics & Communication Engineering (IOSR-JECE)
ISSN(e) : 2278-1684 ISSN(p) : 2320-334X, PP 01-04
www.iosrjournals.org

Detection of Packet Droppers and Modifiers in Wireless Sensor Networks

Sruthi S[1], Devu M S[2]

[1] *First year M.Tech, ,Mohandas college of Engineering, Trivandrum.*
[2]*Assistant Professor, Mohandas College of Engineering and Technology, Trivandrum*

ABSTRACT: *Packet dropping and modification are two common attacks that can disrupt communication in wireless multi-hop sensor networks. Many schemes have been proposed to alleviate the attacks but none can effectively and efficiently identify the intruders. To address the problem, we propose a simple yet effective scheme, which can identify misbehaving forwarders that drop or modify packets.*
Keywords– *Topology, dynamic routing, heuristic ranking*

I. INTRODUCTION

Sensor nodes in a wireless sensor network, monitor the environment, detect events of interest, produce data and co-operate in forwarding the data towards a sink, which could be a gateway, base station, storage node, or querying user. To perform the monitoring and data collection tasks a sensor network may often established in an unattended and hostile environment, which lacks physical protection and is subject to node compromise. After compromising one or multiple sensor nodes, an adversary may launch various attacks to disrupt the in-network communication. Among these attacks, two common ones are *dropping packets* and *modifying packets*, i.e., compromised nodes drop or modify the packets that they are supposed to forward. Multi-path forwarding is a widely adopted counter-measure to deal with packet droppers. In multi-path forwarding each packet is forwarded along multiple redundant paths and hence packet dropping in only some path can be tolerated. This scheme introduces high extra communication overhead. Monitoring the behaviour of forwarding nodes is another scheme of countermeasures. However, these schemes are subject to high energy cost incurred by the promiscuous operating mode of wireless interface. To deal with packet modifiers, most of existing countermeasures [7]–[10] are to filter modified messages within a certain number of hops.

However, without identifying packet droppers and modifiers, these countermeasures cannot fully solve the packet modification problems because the compromised nodes can continue attacking the network without being caught. To identify packet modifiers, Ye et al. [11] recently proposed a probabilistic nested marking (PNM) scheme to identify packet modifiers with a certain probability. However, the PNM scheme cannot be used together with the false packet filtering schemes [7]–[10], because the filtering schemes will drop the modified packets which should be used by the PNM scheme as evidences to infer packet modifiers. This degrades the efficiency of deploying the PNM scheme.

In this paper, we propose a simple yet effective scheme to catch both packet droppers and modifiers. According to the scheme, a dynamic routing tree rooted at the sink is first established. When sensor data is transmitted along the tree structure towards the sink, each packet sender or forwarder adds a small number of extra bits, which is called packet marks, to the packet. The format of the small packet marks is deliberately designed such that the sink can obtain very useful information from the marks. Specifically, based on the packet marks, the sink can figure out the dropping rate associated with every sensor node, and then run our proposed *node categorization algorithm* to identify nodes that are droppers/ modifiers for sure or are suspicious droppers/modifiers. The tree structure dynamically changes in every time interval. One searching interval is called as a round. As the number of misbehaving node is increased, we have to find out the most suspicious node from among these nodes. So here we perform a *heuristic ranking algorithm* to find out the most suspicious bad node. This way, most of the bad nodes can be gradually identified with small false positive. Our system has the following unique characteristics compared with existing system: (1) being effective in identifying both packet droppers and modifiers, (2) low overhead in terms of both communication and energy consumption, and (3) being compatible with existing false packet filtering schemes [7]–[10]; that is, it can be established together

with the false packet filtering schemes, and therefore cannot only identify intruders but also filter modified packets immediately after the modification is detected.

I. SYSTEM MODEL

2.1 Network Assumptions

We consider a typical deployment of sensor network, as shown in Fig. 1, where a large number of sensor nodes are established in a two dimensional area. Each sensor node generates sensing data periodically and all these nodes co-operate to forward packets that contain the data hop by hop towards a sink, located at some place within the network. We assume that all sensor nodes and the sink are time synchronized [12], which is required by many applications. The sink shares a unique key with all the nodes. The sink is aware of the network topology, which can be achieved by requiring nodes to report their neighbouring nodes soon after deployment.

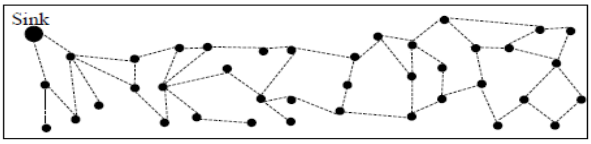

Figure 1: Network System Model

2.2 Security Assumptions and Attack Model

We assume the network sink is trustworthy and free of compromise, but regular sensor nodes can be compromised. Compromised nodes may or may not collude with each other. A compromised node can launch the following two attacks:

1) Packet dropping: a compromised node drops all or some of the packets that it is supposed to forward. It may also drop the data generated by itself for some malicious purpose such as accusing innocent nodes. 2) Packet modification: a compromised node modifies all or some of the packets that it is supposed to forward. It may also modify the data it generates to protect itself from being identified or to accuse other nodes.

II. THE PROPOSED SCHEME

In our scheme for identifying packet droppers and modifiers, a system initialization phase is followed by several equal duration rounds of intruder identification phases.

- In the initialization phase, sensor nodes form a dynamic routing tree rooted at the sink. The structure of the tree changes dynamically from round to round.
- In each round, data traffic is transmitted through the routing tree to the sink, and each packet sender/forwarder adds a small number of extra bits to the packet and also encrypts the packet. When one round finishes, based on the extra bits carried in the received packets, the sink runs the node categorization algorithm to identify nodes that must be droppers or modifiers and nodes that are suspiciously bad.
- The routing tree is reshaped every round. As a certain number of rounds have passed, the sink will have collected information about node behaviours in different routing topologies. The information includes which nodes are bad for sure, which nodes are suspiciously bad, and the nodes' topological relationship. To further identify bad nodes from the potentially large number of suspiciously bad nodes, the sink runs heuristic ranking algorithms.

3.1 Tree Establishment and Packet Transmission

Dynamic routing tree routed at the sink is first established. The sink knows the tree topology and shares a unique key with each node on the tree. When a node wants to send out a packet, it attaches a sequence number to the packet, encrypts the packet with the key shared with the sink, and then forwards the packet to its parent. When an innocent intermediate node receives a packet, it attaches a few bits to the packet to mark the forwarding path of the packet, encrypts the packet, and then forwards the packet to its parent. On the contrary, a misbehaving intermediate node may drop a packet it receives. On receiving a packet, the sink decrypts it, and thus finds out the original sender and the packet sequence number. The sink keeps tracking the sequence

numbers of received packets for every node, and for every certain time interval, which we call a *round*, it calculates the packet dropping rate for every node. Based on the dropping rate and the knowledge of tree topology, the sink identifies packet droppers based on rules we derive. In detail, the scheme includes the following components, which are elaborated in the following.

1) System Initialization: A dynamic routing tree routed at the sink is first established. The sink knows the tree topology structure, it setup a secret pair-wise keys between the sink and every regular sensor node and shares it with each node on the tree. The key facilitate packet forwarding from every sensor node to the sink.

2) Packet Sending and Forwarding: Each node maintains a counter *Cp* which keeps track of the number of packets that it has sent so far. When a sensor node *n* has a data item *D* to report, it composes and sends the following packet to its parent node *Pu*:

3.2. Node Categorization Algorithm

In every round, for each sensor node *n*, the sink keeps track of the number of packets sent from *n*, the sequence numbers of these packets and the number of flips in the sequence numbers of these packets, (i.e., the sequence number changes from a large number such as *Ns ¡* 1 to a small number such as 0). In the end of each round, the sink calculates the dropping rate for each node *n*. Suppose *n: max* is the most recently seen sequence number, *n:flip* is the number of sequence number flips and *n:rcv* is the number of received packets. Based on the dropping rate of every sensor node and the tree topology, the sink identifies the nodes that are droppers for sure and that are possibly droppers. For this purpose, a threshold μ is first introduced. We assume that if a node's packets are not intentionally dropped by forwarding nodes, the dropping rate of this node should be lower than μ. Note that μ should be greater than 0, taking into account droppings caused by incidental reasons such as collisions. The first step of the identification is to mark each node with "+" if its dropping ratio is lower than μ, or with "-" otherwise. After all nodes have been marked with "+" or "-".

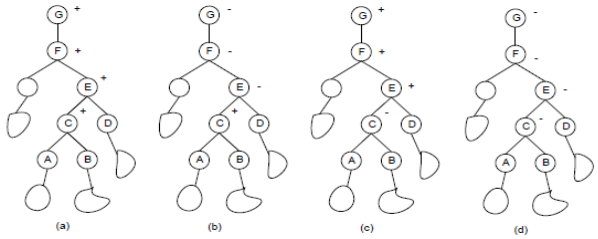

Figure 2: Node Status Pattern

- +{+}+: The node and its one or more continuous immediate upstream nodes are marked as "+".
- +{-}+: The node is marked as "+", but its one or more continuous immediate upstream nodes are marked as "-".
- -{+}+: The node is marked as "-", but its one or more continuous immediate upstream nodes are marked as "+".
- -{-}+: The node and its one or more continuous immediate upstream nodes are marked as "-".

For each of the above cases, we can infer whether a node (i) has dropped packets (called *bad for sure*), (ii) is suspected to have dropped packets (called *suspiciously bad*), (iii) has not been found to drop packets (called *temporarily good*), or (iv) must have not dropped packets (called *good for sure*):

Case 1: +{+}+. The node and its continuous immediate upstream nodes do not drop packets along the involved path, but it is unknown whether they drop packets on other forwarding paths. Therefore, the sink infers that these nodes are *temporally good.*

Case 2: +{-}+. In the case, all nodes marked as "-" must be *bad for sure.* To show the correctness of this rule, we prove it by contradiction. Without loss of generality, where node C is marked as "+", and node E, F, G are marked as "-". If our conclusion is incorrect and node E is good, E must not drop its own packets. Since E is marked as "-", there must be some upstream nodes of E dropping E's packets.

Note that the bad upstream nodes are at least one hop above E and at least two hops above C. It is impossible for it to differentiate packets from E and C, so it cannot selectively drop the packets from E while forwarding the packets from C. Even if C and the bad upstream node collude, they cannot achieve this result. This is because every packet from C must go through and be encrypted by E, and therefore the bad upstream node cannot tell the source of the packet to perform selective dropping. Note that, if a packet is forwarded to the bad upstream node without going through E, the packet cannot be correctly decrypted by the sink and thus will be dropped. Therefore, E must be bad. Similarly, we can also conclude that F and G are also bad.

Case 3: -{+}+. In this case, either the node marked as "-" or its parent marked as "+" must be bad. But it cannot be further inferred whether (i) only the node with "-" is bad, (ii) only the node with "+" is bad, or (iii) both nodes are bad. Therefore, it is concluded that both nodes are *suspiciously bad*.

Case 4: -{-}+. In this case, every node marked with "-" could be bad or good. Conservatively, they have to be considered as *suspiciously bad*. Specifically, suppose *g* is the highest-level node that is marked as "-", and *n* is its parent.

III. CONCLUSION

To alleviate the problem of packet dropping and modification, we proposed a simple yet effective scheme to identify misbehaving nodes that drop or modify packets. Extensive analysis and simulations have been conducted and verified the effectiveness of the proposed scheme in various scenarios.

IV. ACKNOWLEDGEMENTS

The authors would like to thank the teaching faculties of Mohandas College of Engineering.

REFERENCES

[1] H.Chan and A. Perrig, "*Security and Privacy in Sensor Networks," IEEE Computer,* October 2003.

[2] C. Karlof and D. Wagner, "*Secure routing in wireless sensor networks: attacks and countermeasures," the First IEEE International Workshop on Sensor Network Protocols and Applications,* pp. 113–127, May 2003.

[3] V. Bhuse, A. Gupta, and L. Lilien, "*Dpdsn: Detection of packet-dropping attacks for wireless sensor networks," In the Trusted Internet Workshop, International Conference on High Performance Computing,* December 2005.

[4] S. Marti, T. Giuli, K. Lai, and M. Baker, "*Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," ACM MobiCom,* August 2000.

[5] R. Roman, J. Zhou, and J. Lopez, "*Applying intrusion detection systems to wireless sensor networks," Third IEEE Annual Consumer Communications and Networking Conference (CCNC),* pp. 640–644, Jan. 2006.

[6] S. Lee and Y. Choi, "*A resilient packet-forwarding scheme against maliciously packet-dropping nodes in sensor networks," Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks (SASN),* pp. 59–70, 2006.

[7] F. Ye, H. Luo, S. Lu, and L. Zhang, "*Statistical En-route Filtering of Injected False Data in Sensor Networks," IEEE INFOCOM,* March 2004.

[8] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "*An Interleaved Hop-by-Hop Authentication Scheme for Filtering False Data in Sensor Networks," IEEE Symposium on Security and Privacy,* 2004.

[9] H. Yang, F. Ye, Y. Yuan, S. Lu, and W. Arbaugh, "*Toward Resilient Security in Wireless Sensor Networks," ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* May 2005.

[10] Z. Yu and Y. Guan, "*A Dynamic En-route Scheme for Filtering False Data in Wireless Sensor Networks," IEEE Infocom 2006,* April 2006.

[11] F. Ye, H. Yang, and Z. Liu, "*Catching Moles in Sensor Networks," IEEE International Conference on Distributed Computing Systems (ICDCS),* June 2007.

[12] Q. Li and D. Rus, "*Global clock synchronization in sensor networks," IEEE INFOCOM,* 2004.