Monitoring the Node due to the Effect of Packet Droppers and Modifiers in Wireless Sensor Network

D. Raman[1], V. Chandra Sekhar[2]

[1]Associate Professor in CSE Department
Vardhaman College of Engineering, Hyderabad, India
raman.vsd@gmail.com

[2]M.Tech Scholar, Computer Science and Engineering
Vardhaman College of Engineering, Hyderabad, India
chandu.shekar514@gmail.com

Abstract: ***There are many attacks in wireless sensor networks during the time of communication between them. The most common attacks are packet dropping and modification. To identify this type of attacks many schemes have been proposed, but very few of them are effectively working to identify the intruders. To address this problem, monitoring the nodes is required.***

Keywords: packet droppers, node categorization.

1. Introduction

In a wireless sensor network, sensor nodes monitor the environment, detect events of interest, produce data, and collaborate in forwarding the data toward a sink, which could be a gateway, base station, storage node, or querying user. Because of the ease of deployment, the low cost of sensor nodes and the capability of self-organization, a sensor network is often deployed in an unattended and hostile environment to perform the monitoring and data collection tasks. When it is deployed in such an environment, it 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.

2. System Model

Among wireless sensor 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. To deal with packet droppers, a widely adopted countermeasure is multipath forwarding in which each packet is forwarded along multiple redundant paths and hence packet dropping in some but not all of these paths can be tolerated. To deal with packet modifiers, most of existing countermeasures aim to filter modified messages en-route within a certain number of hops. These countermeasures can tolerate or mitigate the packet dropping and modification attacks, but the intruders are still there and can continue attacking the network without being caught. To locate and identify packet droppers and modifiers, it has been proposed that nodes continuously monitor the forwarding behaviours of their neighbours to determine if their neighbours are misbehaving, and the approach can be extended by using the reputation based mechanisms to allow nodes to infer whether a non neighbour node is trustable. This methodology may be subject to high-energy cost incurred by the promiscuous operating mode of wireless interface; moreover, the reputation mechanisms have to be exercised with cautions to avoid or mitigate bad mouth attacks and others.

3. The Proposed Scheme

Consider a typical deployment of sensor networks, where a large number of sensor nodes are randomly deployed in a two dimensional area. Each sensor node generates sensory data and all these nodes collaborate to forward packets containing the data toward a sink. The sink is located within the network. The sink is aware of the network topology, which can be achieved by requiring nodes to report their neighbouring nodes right after deployment.

There are two major components, the transmission/forwarding of packets, and the transmission/forwarding and analysis of monitoring reports.

The first component ensures that packets cannot be dropped selectively based on their sources, and the second component enables collaborative monitoring of node behaviours and identification of packet droppers. The transmission of packets can be encrypted by the source before sending to the intermediate nodes. These packets are decrypted by the intermediate nodes and adding an extra bit to them because where any changes is to be made by the third party. At the destination end that packets can be encrypted.

To overcome the packet dropping problem monitoring the node is required. For that purpose we use node categorization algorithm. This algorithm is mainly used to transfer the packet (data) from source to destination i.e., intermediate nodes based on the wireless network design. In most of the network it contains more than 2 nodes at that time it uses the one of the algorithm i.e., node categorization algorithm.

If we apply this algorithm on a tree based network then the tree topology, the sink identifies the nodes that are droppers for sure and that are possibly droppers. For this purpose, a threshold is first introduced. We assume that if a node's packets are not intentionally dropped by forwarding nodes, the dropping ratio of this node should

be lower than θ. Note that θ 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 θ, or with "-" otherwise. After then, for each path from a leaf node to the sink, the nodes' mark pattern in this path can be decomposed into any combination of the following basic patterns, which are also illustrated by Figure 1;

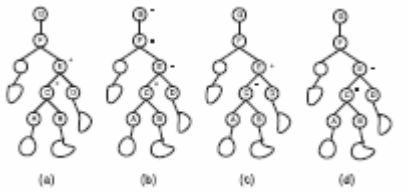

Figure 1: Basic Patterns

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

There are four different types of packets. They are;

- Bad for sure packets
- Suspiciously bad packets
- Temporarily good packets
- Good for sure packets.

Case 1: + {+}. The node and its parent node 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 temporarily good. For example, in Fig. 1a, node C and E are marked "+" and are regarded as temporarily good. A special case is, if a leaf node is marked as "+," it is safe to infer it as good since it cannot drop other's packets.

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, we examine the scenario illustrated in Fig. 1b, where node C is marked as "+," and nodes E, F, and 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, i.e., at least two hops above C.

```
1: Input: packet (0, m).
2: u = 0, m' = m;
3: hasSuccAttemp = false;
4: for each child node v of node u do
5:     P = dec(K_v, m');
6:     if decryption fails then
7:         continue;
8:     else
9:         hasSuccAttemp = true;
10:        if P starts with (R_v, v) then
11:            record the sequence number; /* v is the
               sender */
12:            break;
13:        else
14:            trim R_v from P and get m'; /*v is
               a forwarder */
15:            u = v, hasSuccAttemp = false; go to line 4;
16: if hasSuccAttemp = false then
17:     drop this packet;
```

It is impossible for them to differentiate packets from E and C, so they 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. 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 1) only the node with "-" is bad, 2) only the node with "+" is bad, or 3) both nodes are bad. Therefore, it is concluded that both nodes are suspiciously bad. The correctness of this rule can also be proved by contradiction. Without loss of generality, let us consider the scenario shown in Fig. 1c, where node C is marked as "-,"and node E is marked as "+." Now suppose both C and E is good, and hence there must exist at least one upstream node of E which is a bad node that drops the packets sent by C. However, it is impossible to find such an upstream node since nodes F and G, and other upstream nodes cannot selectively drop packets from node C while forwarding packets from node E. Hence, either node C is bad or node E is bad in this case.

Case 4: - {-}. In this case, every node marked with "-"could be bad or good. Conservatively, they have to be considered as suspiciously bad. By using this procedure the packets can identify which are dropped.

4. Conclusion

By this simple effective scheme the packets can be identify which are dropped. Monitoring the node is required which plays a crucial role during the time of packet transmission. Finally, most of the bad nodes can be identified by the node categorization algorithm with small false positive in a network. Extensive analysis, simulations, and

implementation have been conducted and verified the effectiveness of the proposed scheme.

References

[1] Chuang Wang, Taiming Feng, Jinsook Kim, and Guiling Wang "Catching Packet Droppers and Modifiers in Wireless Sensor Networks", IEEE transactions on parallel and distributed systems, vol. 23, no. 5, may 2012.
[2] H. Chan and A. Perrig, "Security and Privacy in Sensor Networks," Computer, vol. 36, no. 10, pp. 103-105, Oct. 2003.
[3] C. Karlof and D. Wagner, "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures," Proc. IEEE First Int'l Workshop Sensor Network Protocols and Applications, 2003.
[4] V. Bhuse, A. Gupta, and L. Lilien, "DPDSN: Detection of Packet- Dropping Attacks for Wireless Sensor Networks," Proc. Fourth Trusted Internet Workshop, 2005.
[5] Crossbow, "Wireless Sensor Networks," http://www.xbow.com/ Products/Wireless_Sensor_Networks.htm, 2011