Identification of Misbehaving Nodes that Can Drop or Modify the Packets in Wireless Sensor Networks

C. Ramakristanaiah[1], A. L. Sreenivasulu[2]

[1]M. Tech, Department of CSE, Intell Engineering College, Anantapur, AP, India

[2]Assistant Professor, Department of CSE, Intell Engineering College, Anantapur, AP, India

Abstract: ***In wireless sensor networks the common attacks are Packet dropping and modification made by an attacker to disturb communication. Several solutions have been proposed to detect and protect from such attacks, but very few were succeeded. In order to give one effective and efficient solution to this problem, we propose a simple scheme, which can identify misbehaving forwarders or compromised nodes that drop or modify packets and filter the modified pockets. Effectiveness and efficiency of the scheme have been verified by conducting Extensive analysis and simulations.***

Keywords: *packet dropping, packet modification, wireless sensor networks*

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 [1] 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.

To deal with packet droppers, a widely adopted countermeasure is multipath forwarding[2], in which each packet is forwarded along multiple redundant path and hence packet dropping in some but not all of these paths can be tolerated. To deal with packet modifiers, most of existing countermeasures [3], aim to filter modified messages en-route[4] within a certain number of hops.

In this paper, we propose a simple yet effective scheme to catch both packet droppers and modifiers. In this scheme, a routing tree rooted at the sink is first established. When sensor data are transmitted along the tree structure toward 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 ratio associated with every sensor node, and then runs our proposed node categorization algorithm to identify nodes that are droppers/modifiers for sure or are suspicious droppers/modifiers. As the tree structure dynamically changes every time interval, behaviors of sensor nodes can be observed in a large variety of scenarios. As the information of node behaviors has been accumulated, the sink periodically runs our proposed heuristic ranking algorithms to identify most likely bad nodes from suspiciously bad nodes. This way, most of the bad nodes can be gradually identified with small false positive.

2. DAG Establishment and Packet Transmission

All sensor nodes form a DAG and extract a routing tree from the DAG. The sink knows the DAG and the routing tree, and shares a unique key with each node.

2.1 System Initialization

The purpose of system initialization is to set up secret pair wise keys between the sink and every regular sensor node, and to establish the DAG and the routing tree to facilitate packet forwarding from every sensor node to the sink.

Each sensor node u is preloaded the following information:

- Ku: a secret key exclusively shared between the node and the sink.
- Lr: the duration of a round.
- Np: the maximum number of parent nodes that each node records during the DAG establishment procedure.
- Ns: the maximum packet sequence number. For each sensor node, its first packet has sequence number 0, the Nsth packet is numbered Ns _ 1, the ðNs þ 1Þth packet is numbered 0, and so on and so forth.

The sink broadcasts to its one-hop neighbors a 2-tuple <0; 0>. In the 2-tuple, the first field is the ID of the sender (we assume theID of sink is 0) and the second field is its distance in hop from the sender to the sink. Each of the remaining nodes, assuming its ID is u, acts as follows:

1. On receiving the first 2-tuple <v; dv>, node u sets itsown distance to the sink as du =dv + 1.
2. Node u records each node w (including node v) as it is parent on the DAG if it has received <w; dw> where dw = dv.
3. After a certain time interval, 1 node u broadcasts 2-tuple hu; dui to let its downstream one-hop neighbors to continue the process of DAG establishment.

2.2 Packet Sending and Forwarding

When a sensor node u has a data item D to report, it composes and sends the following packet to its parent node Pu:

<Pu; {Ru; u; Cp MOD Ns;D; padu;0}Ku; padu;1>;

where Cp MOD Ns is the sequence number of the packet. Ru (0<Ru<Np-1) is a random number picked by node during the system initialization phase, Paddings padu; 0 and padu; 1 are added to make all packets equal in length, such that forwarding nodes cannot tell packet sources based on packet length.

2.3 Packet receiving at Sink

We use node 0 to denote the sink. When the sink receives a packet <0;m'>, it conducts the following steps:

1. Initialization. Two temporary variables u and m are introduced. Let u=0 and m=m' initially.

Node Categorization Algorithm

In every round, for each sensor node u, the sink keeps track of the number of packets sent from u, the sequence numbers of these packets, and the number of flips in the sequence numbers of these packets
The dropping ratio in this round is calculated as follows:

du=(nu,flip*Ns +nu,max +1_ nu,rcv)/nu,flip*Ns+nu,max+1

Based on the dropping ratio of every sensor node and the tree topology, the sink identifies the nodes that are droppers for sure and that are possibly droppers

The first step of the identification is to mark each node with "+" if its dropping ratio is lower than 0, or with "-" otherwise.

Based upon the information we will categorize those nodes into

1. has dropped packets (called bad for sure),
2. is suspected to have dropped packets (called suspiciously bad),
3. has not been found to drop packets
(called temporarily good)
4. must have not dropped packets
(called good for sure):

3. Tree Reshaping and Ranking Algorithms

The tree used to forward data is dynamically changed from round to round, which enables the sink to observe the behavior of every sensor node in a large variety of routing topologies.

3.1 Tree Reshaping

The tree used for forwarding data from sensor nodes to the sink is dynamically changed from round to round. In other words, each sensor node may have a different parent node from round to round.

3.2 Identifying Most Likely Bad Nodes from Suspiciously Bad Nodes

We rank the suspiciously bad nodes based on their probabilities of being bad, and identify part of them as most likely bad nodes. Since the number of suspiciously bad nodes is potentially large, we propose how to identify most likely bad nodes from the suspiciously bad nodes

Among the above three conditions, the first one and the third one can be relatively easily implemented and verified. For the second condition, we propose several heuristics to find nodes with most-likeliness.

3.3 Global ranking-based (GR) method

The GR method is based on the heuristic that, the more times a node is identified as suspiciously bad, the more likely it is a bad node.

3.4 Stepwise ranking-based (SR) method.

It can be anticipated that the GR method will falsely accuse innocent nodes that have frequently been parents or children of bad nodes

3.5 Hybrid ranking-based (HR) method.

The GR method can detect most bad nodes with some false accusations while the SR method has fewer false accusations but may not detect as many bad nodes as the GR method

4. Handling Collusion

The packets are not distinguishable to the upstream compromised nodes as long as they have been forwarded by an innocent node. The capability of launching collusion attacks is thus limited by the scheme. However, compromised nodes that are located close with each other may collude to render the sink to accuse some innocent nodes. We discuss the possible collusion scenarios in this section and propose strategies to mitigate the effects of collusion.

There are two types of collusions
1. Horizontal collusion
2. Vertical collusion

To defeat collusion that may lead to false accusation, our scheme is extended as follows:

1. The concept of suspicious pair is extended to suspicious tuple which is a non ordered sequence of suspicious nodes.
2. All these tuples should be combined into a single tuple without duplication

5. Performance Evaluation

The effectiveness and efficiency of the proposed scheme are evaluated in the ns-2 simulator (version 2.30).

6. Conclusion

We propose a simple yet effective scheme to identify misbehaving forwarders that drop or modify packets. Each packet is encrypted and padded so as to hide the source of the packet. The packet mark, a small number of extra bits, is added in each packet such that the sink can recover the source of the packet and then figure out the dropping ratio associated with every sensor node

References

[1] V. Bhuse, A. Gupta, and L. Lilien, "DPDSN: Detection of Packet-Dropping Attacks for Wireless Sensor Networks," Proc. Fourth Trusted Internet Workshop, 2005.
[2] M. Kefayati, H.R. Rabiee, S.G. Miremadi, and A. Khonsari,"Misbehavior Resilient Multi-Path Data Transmission in Mobile Ad-Hoc Networks," Proc. Fourth ACM Workshop Security of Ad Hoc anSensor Networks (SASN '06), 2006.
[3] R. Mavropodi, P. Kotzanikolaou, and C. Douligeris, "Secmr—ASecure Multipath Routing Protocol for Ad Hoc Networks," Ad Hoc Networks, vol. 5, no. 1, pp. 87-99, 2007.
[4] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical En-Route Filtering of Injected False Data in Sensor Networks," Proc. IEEE INFOCOM, 2004.

Authors Profile

C. Ramakristanaiah passed B. Tech (CSE) in 2011 under JNT University Anantapur and pursuing M. Tech (CSE) under JNT University Anantapur.

A. L. Sreenivasulu pursuing PhD in Computer Science from JNTU Anatapur and got 14 years of teaching experience and guided several projects both UG & PG. His areas of interest include wireless sensor networks, computer networks and information security.