Advanced Node Categorized Algorithm to prevent Dropping and Modifying Packets in WSN

Mrs. Shilpa.S
Assistant Professor,
Dept of Information Science,
MVJCE.
shilpas1580@yahoo.co.in

Mrs. Bharathi.M
Associate Professor,
Dept of Computer Science,
SJCIT.
bharathigowda1@gmail.com ,

Abstract— In wireless sensor networks, an intruders may launch some attacks due to packet dropping and modifying in order to disrupt the communication. This problem is more critical if the network is deployed for some mission-critical applications such as in a tactical battlefield. Random failure of nodes is also very likely in real-life deployment scenarios. To tolerate such attacks, some of the research schemes have been proposed. But very few can effectively working. The proposed method is simple way of preventing the misbehaving forwarders that drop and modify the packets. Extensive analysis and simulations can verify the effectiveness and efficiency of the scheme.

Keywords — Packet dropping, packet modification, intrusion detection, wireless sensor networks, Directed Acyclic Graph, Sink.

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. 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. Wireless sensor networks (WSNs) consist of hundreds or even thousands of small devices each with sensing, processing, and communication capabilities to monitor the real-world environment. They are envisioned to play an important role in a wide variety of areas ranging from critical military surveillance applications to forest fire monitoring and building security monitoring in the near future [1].

In this proposed system, we propose a simple yet effective scheme to catch both packet droppers and modifiers. In this scheme, [2], [3], [4], [5], 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 [6]. 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.

Our proposed scheme has the following features:

1) Being effective in identifying both packet droppers and modifiers,

2) Low communication and energy overheads, and

3) Being compatible with existing false packet filtering schemes;

That is, it can be deployed together with the false packet filtering schemes, and therefore it cannot only identify intruders but also filter modified packets immediately after the modification is detected. Extensive simulation on general simulator has been conducted to verify the effectiveness and efficiency of the proposed scheme in various scenarios.

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management station. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization.

IDPSes typically record information related to observed events notify security administrators of important observed events and produce reports. Many IDPSes can also respond to a detected threat by attempting to prevent it from succeeding. They use several response techniques, which involve the IDPS stopping the attack itself, changing the security environment (e.g. reconfiguring a firewall) or changing the attack's content.

In the rest of the paper, Section 2 defines the system model. Section 3 describes the proposed scheme and Section 4 reports the evaluation results. Section 5 discusses the related work, and Section 6 concludes the paper.

2. SYSTEM MODEL

2.1 Network Assumptions

We 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 periodically and all these nodes collaborate to forward packets containing the data onward a sink. The sink is located within the network. We assume all sensor nodes and the sink are loosely time

synchronized [7], which is required by many applications. Attack resilient time synchronization schemes, which have been widely investigated in wireless sensor networks [8], [9], can be employed. The sink is aware of the network topology, which can be achieved by requiring nodes to report their neighboring nodes right after deployment.

2.2 Security Assumptions and Attack Model

We assume the network sink is trustworthy and free of compromise, and the adversary cannot successfully compromise regular sensor nodes during the short topology establishment phase after the network is deployed. This assumption has been widely made in existing work [8], [9]. After then, the 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:

3. THE PROPOSED SCHEME

Our proposed scheme consists of a system initialization phase and several equal-duration rounds of intruder identification phases as shown in fig. 1.

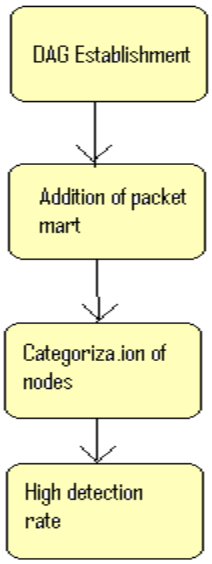


Fig. 1 Flow diagram

- In the initialization phase, sensor nodes form a topology, which is a Tree on DAG (Directed Acyclic Graph).
- In each round, data is transferred through the routing tree to the sink node. Each packet sender/forwarder adds a small number of extra bits to the packet which is called packet mart. When one round is over, based on the packet mart carried in the received packets, the sink node runs a node categorization algorithm to identify nodes that are bad for sure (i.e., packet droppers), suspiciously bad (i.e., suspected to be packet droppers) and good for sure (i.e., no packet droppers).
- The routing tree is reshaped at every round. When certain number of rounds has passed, the sink node will have collected information about node behaviors in different routing topologies. The information includes which nodes are bad for sure, suspiciously bad, and good for sure. In the following sub-sections, we

first present the algorithm for ToD establishment and packet transmission, which is followed by our proposed node categorization algorithm.

3.1 ToD Establishment and Packet Transmission:

The proposed Tree on DAG (ToD) is a semi structured approach that uses Dynamic Forwarding on an implicitly constructed structure composed of multiple shortest path trees to support network scalability. Here, Let us consider wireless tree establishment topology with 14 nodes, named from A to N respectively is viewed in fig 2.

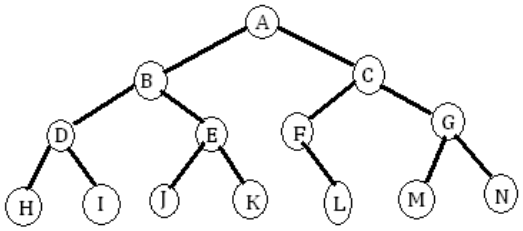


Fig. 2 Tree Establishment

The key principle behind ToD is that adjacent nodes in a graph will have low stretch in one of these trees in ToD and thus resulting in early aggregation of packets. After performing local aggregation, all sensor nodes dynamically decide the forwarding path based on the location of the sources and aggregated packets are forwarded to the sink node on ToD.

The sink node knows the ToD structure which shares a unique key with each node. When a node wants to send out a packet, it attaches to the packet a sequence number, encrypts the packet only with the key shared with the sink, and then forwards the packet to its parent on the routing tree. 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 node decrypts it, and thus finds out the original sender and the packet sequence number. The sink node tracks 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 ratio for every node. Based on the dropping ratio and the knowledge of the topology, the sink node identifies packet droppers based on the following algorithm. In detail, the scheme includes the system initialization. The purpose of system initialization is to set up secret pair wise keys between the sink node and every regular sensor node, and to establish the Tree on DAG to facilitate packet forwarding from every sensor node to the sink node.

3.2 Node Categorization Algorithm:

In every round, for each sensor node *u*, the sink node *s* keeps track of the number of packets sent from *u* and the number of packets received to *s*. In the end of each round, the sink node *s* calculates the dropping ratio for each node *u*. suppose *Nf is* the number of transmitted packets and *Nr* is the number of received packets. The dropping ratio (*du*) in this round is calculated as follows:

$$du = \frac{(Nf - Nr * Nf)}{Nf + Nr + (Nf * Nf - Nr)}$$

Based on the dropping ratio of every sensor node and the tree topology, the sink categories the nodes based upon the node categorization algorithm. This algorithm

identifies the nodes that are droppers for sure, possibly droppers and suspicious droppers. For this purpose, a threshold θ is first introduced.

Algorithm 1
Tree-Based Node Categorization Algorithm
1: **Input**: Tree *T*, with each node *u*, and its dropping ratio *du*, threshold value θ, sink node *s*.
2: **for** every sink node in *T* **do**
3: find dropping ratio *du;*
4: **if** $du < \theta$ **then**
5: Set *u* as *good for sure* or *suspiciously bad;*
6: **if** $du = 0$ **then**
7: Set *u* as *good for sure;*
8: **else if** $du > 0$
9: Set *u* as *suspiciously bad*;
10: **else**
11: *break*;
12: **else**
13: Set *u* as *bad for sure*;
14: **repeat**

We assume that if a node's packets are not intentionally dropped by forwarding nodes, then dropping ratio of this node should be lower than θ. Note that θ should be greater than 0. Here let us assume θ value is 0.5. The categorization of nodes can be taken in any one of the following cases

(i) Packet droppers for sure.
(ii) Suspicious packet droppers.
(iii) No packet droppers for sure.

Algorithm 1 specified the Tree-Based Node Categorization algorithm as for every sink node in T and the following cases exist.

Case 1 : If the dropping ratio is less than θ , then a node has not dropped packets (called *good for sure*) or the node is suspected to have dropped packets (called *suspiciously bad*).

Case 1.1**:** If the dropping ratio value is equal to zero, then the node has not dropped packets.

Case 1.2**:** If the dropping ratio is less than θ, but greater than zero means, then the node is suspected to have dropped packets.

Case 2**:** If the dropping ratio is greater than θ, then a node must have dropped packets (called *bad for sure).* The dropping packets may due to traffic, collisions, and malicious node. Based on the above cases, we develop a node categorization algorithm to find nodes that whether the node is bad for sure, suspicious bad, or good for sure. The tree used to forward data is dynamically changed from round to round and each sensor node may have a different parent node which is called tree reshaping takes place.

4. PERFORMANCE EVALUATION

Our packet dropper identification scheme is simulated in the General Simulators simulator to evaluate the effectiveness and efficiency of the proposed scheme. The objectives of this evaluation study are two-fold: firstly, testing the effectiveness and efficiency of our scheme in identifying and analyzing packet droppers; secondly, studying the impacts of various system parameters (i.e., sensor data reporting interval, average delay, round length, etc.) and testing the performance analysis of our proposed scheme. We measure the performance of our scheme with only one metric: the detection rate defined as the ratio of successfully identified bad nodes.

We run simulations in a 400×400m2 networks with randomly generated network

topology. Unless stated otherwise, we set the percentage of bad nodes to 10%, the network size to 100 sensor nodes, the per-node packet-reporting interval to 3 seconds. Also, when a bad node decides to drop packet in a round, it drops 30% of the packets. All the results are measured and averaged based on simulations over 14 random networks.

We report the packet analysis information for some of the node intervals (i.e., between each node u and sink node s) in Table 1 and the fields in Table 2 includes node interval, dropping ratio or detection rate and packet analysis information. The detection rate or the dropping ratio is calculated with the help of the dropping ratio formula. And the following figures show the Performance charts foe 2nd table values.

TABLE 1
PACKET ANALYSIS INFORMATION FOR NODE INTERVAL COMMUNICATION

Node Interval	***Detection Rate***	***Packet Analysis Information***
G - C	0.000000	No Packet Drops
C - A	0.000000	No Packet Drops
A - D	0.977423	Packet Drops
B - E	0.400044	Suspiciously Packet Drops
F - N	0.993395	Packet Drops
F - M	0.913749	Packet Drops
K - E	-NAN	No Packet Transmission

TABLE 2

ANALYSIS OF PACKET DROPPERS

Node Interval Time Interval(s)	A - C	F - N	F - M
3	0.964168	0.990674	0.963494
6	0.987295	0.995830	0.987124
9	0.992223	0.997307	0.992159
12	0.994397	0.998012	0.994364
15	0.995621	0.998424	0.995601

Fig 4.1: Performance Graph- Node interval verses time interval from F to N

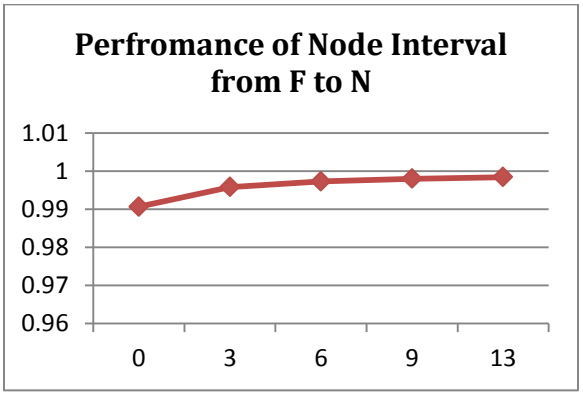


Fig 4.2: Performance Graph- Node interval verses time interval from F to M

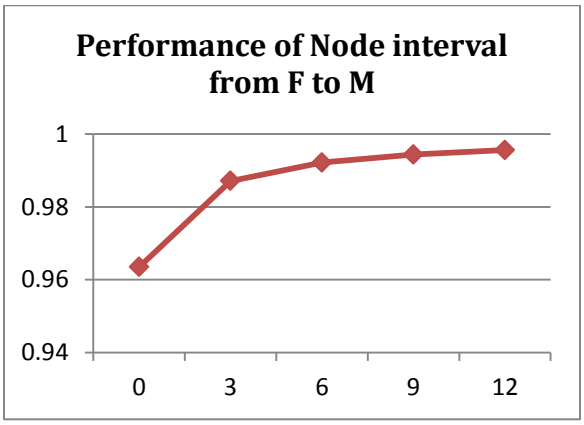
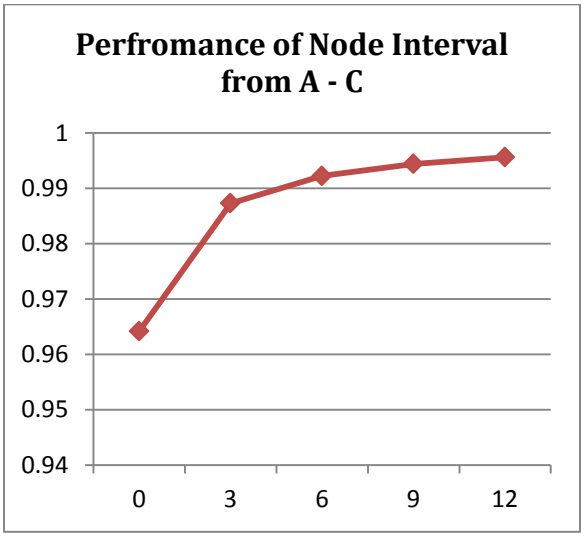


Fig 4.3: Performance Graph- Node interval verses time interval from A to C



5. CONCLUSION

Thus, this proposed scheme is a simple yet effective scheme to identify misbehaving forwarders that drop 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 node can recover the source of the packet and then figure out the dropping ratio associated with every sensor node. Finally, our tree based node categorization algorithm can identify nodes that are packet droppers for sure, suspiciously packet droppers, not packet droppers. Extensive analysis, simulations and implementation are conducted and verified for the effectiveness of the proposed scheme

References

[1] B. Barak, S. Goldberg, and D. Xiao, ―Protocols and Lower Bounds for Failure Localization in the Internet,‖ Proc. *Eurocrypt*, 2008.

[2] B. Xiao, B. Yu, and C. Gao, ―Chemas: Identify suspect nodes in selective forwarding attacks,‖ *Journal of Parallel and Distributed Computing*, vol. 67, no. 11, 2007.

[3] Chuang Wang, Taiming Feng, Jinsook Kim, Guiling Wang, Member, IEEE, and Wensheng Zhang, Member, IEEE, ―Catching Packet Droppers and Modifiers in Wireless Sensor Networks‖, *IEEE TRANSACTIONS on Parallel and Distributed System*, vol. 23, no. 5, may 2012.

[4] Chuang Wang, Taiming Feng, Jinsook Kim, Guiling Wang, Member, IEEE, and Wensheng Zhang, Member, IEEE, ―Catching Packet Droppers and Modifiers in Wireless Sensor Networks‖, *IEEE TRANSACTIONS on Parallel and Distributed System*, 2009.

[5] F. Ye, H. Luo, S. Lu, and L. Zhang, ―Statistical En-Route Filtering of Injected False Data in Sensor Networks,‖ *Proc. IEEE INFOCOM*, 2004.

[6] F. Ye, H. Yang, and Z. Liu, ―Catching Moles in Sensor Networks,‖ *Proc. 27th Int'l Conf. Distributed Computing Systems (ICDCS '07)*, 2007.

[7] H. Song, S. Zhu, and G. Cao, ―Attack-Resilient Time Synchronization for Wireless Sensor Networks,‖ *Ad Hoc Networks*, vol. 5, no. 1, pp. 112-125, 2007.

[8] I. Krontiris, T. Giannetsos, and T. Dimitriou, ―LIDeA: A Distributed Lightweight Intrusion Detection Architecture for Sensor Networks,‖ *Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08)*, 2008.

[9] J. Deng, R. Han, S. Mishra,‖ Defending against path-based DoS attacks in wireless sensor networks‖, *in: Proceedings of the Third ACM on the Security of Ad hoc and Sensor Networks (SASN 2005)*, 2005, pp. 89–96.

[10] J. Parker, A. Patwardhan, and A. Joshi, ―Cross-layer analysis for detecting wireless misbehaviour,‖ *in Proceedings of the Third IEEE Consumer Communications and Networking Conference, 2006. CCNC 2006, vol. 1. IEEE, Jan. 2006, pp. 6–9.*

[11] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, ―An

acknowledgment-based approach for the detection of routing misbehaviour in manets‖, *Mobile Computing, IEEE Transactions on*, vol. 6, no. 5, 2007.

[12] Q. Li and D. Rus, ―Global Clock Synchronization in Sensor Networks,‖ *Proc. IEEE INFOCOM*, 2004.

[13] R. Mavropodi, P. Kotzanikolaou, and C. Douligeris, ―Secmr—A Secure Multipath Routing Protocol for Ad Hoc Networks‖ *Ad Hoc Networks*, vol. 5, no. 1, pp. 87-99, 2007.

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

[15] S. Slijepsevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. Srivastava, ―On communication security in wireless ad-hoc sensor networks,‖ *in Proc. 11th IEEE Int. Workshops Enabling Technol.: Infrastructure for Collaborative Enterprises,* Jun. 2002, pp. 139–144.

[16] S. Zhu, S. Setia, S. Jajodia, and P. Ning, ―An Interleaved Hop-by- Hop Authentication Scheme for Filtering False Data in Sensor Networks,‖ *Proc. IEEE Symp. Security and Privacy,* 2004.

[17] T. H. Hai and E. N. Huh, ―Detecting selective forwarding attacks in wireless sensor networks using two-hops neighbor knowledge*," in IEEE NCA,* 2008.

[18] V. Bhuse, A. Gupta, and L. Lilien, ―DPDSN: Detection of Packet- Dropping Attacks for Wireless Sensor Networks,‖ *Proc. Fourth Trusted Internet Workshop,* 2005.

[19] W. Li, A. Joshi, and T. Finin, ―Coping with Node Misbehaviors in Ad Hoc Networks: A Multi-Dimensional Trust Management Approach‖ *Proc. 11th Int'l Conf. Mobile Data Management (MDM '10),* 2010.

[20] X. Zhang, A. Jain, and A. Perrig, ―Packet-dropping adversary identification for data plane security,‖ *in ACM CONEXT,* 2008.