Detection of Packet Dropping and Modification in Wireless Sensor Network

S.VIJAYALAKSHMI
Student of Master of Electronics and Communication Enginerring,
Sri Manakula Vinayagar Engineering College, Puducherry.
vj_lak@yahoo.com

R.KURINJIMALAR
Associate Professor, Department of Electronics and Communication Enginerring,
Sri Manakula Vinayagar Engineering College, Puducherry.
Kurunjirajmeera13.03@gmail.com

S.PRAKASH
Student of Master of Electronics and Communication Enginerring,
Sri Manakula Vinayagar Engineering College, Puducherry.
prakashsubsan@gmail.com

Abstract - Security is crucial for wireless sensor networks deployed in hostile environments. The packet droppers and packet modifiers may be random. Identifying such attacks is very difficult and sometimes impossible. In this paper the identification and filtering of packet droppers and packet modifier nodes is done using packet marks and ranking algorithms using NS2. The performance is measured using detection rate and false positive probability. The results indicate that the proposed scheme provides an effective mechanism for detecting compromised node.

Keyword*:* packet droppers and modifiers, intrusion detection. wireless sensor networks

1. INTRODUCTION

Simplicity in Wireless Sensor Network with resource constrained nodes makes them extremely vulnerable to variety of attacks. Attackers can eavesdrop on transmissions, inject bits in the channel, replay previously heard packets and many more. Securing the Wireless Sensor Network needs to make the network support all security properties: confidentiality, integrity, authenticity and availability. Attackers may deploy a few malicious nodes with similar hardware capabilities as the legitimate nodes that might collude to attack the system cooperatively. The attacker may come upon these malicious nodes by purchasing them separately, or by "turning" a few legitimate nodes by capturing them and physically overwriting their memory. Since sensor networks are deployed in unattended and hostile 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.

We expect sensor networks to consist of hundreds or thousands of sensor nodes as in Fig 1. Each node represents a potential point of attack, making it impractical to monitor and protect each individual sensor from either physical or logical attack. The networks may be dispersed over a large area, further exposing them to attackers who capture and reprogram individual sensor nodes. Attackers can also obtain their own commodity sensor nodes and induce the network to accept them as legitimate nodes, or they can claim multiple identities for an altered node. Once in control of a few nodes inside the network, the adversary can then mount a variety of attacks—for example, falsification of sensor data, extraction of private sensed information from sensor network readings, and denial of service. Addressing the problem of sensor node compromise requires technological solutions.

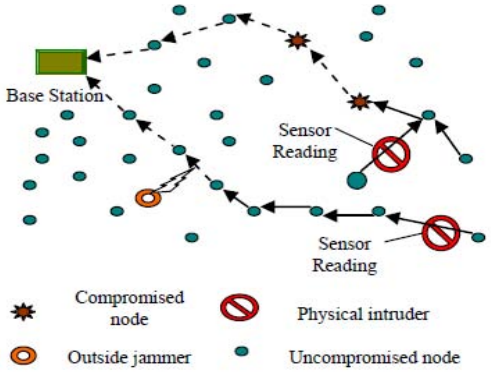

Figure 1. Sensor Network

Packet dropping is nothing but a bad node drops all or some of the packets that are supposed to be forwarded. It may also drop the data generated by itself for some malicious purpose such as blaming innocent nodes. Packet modification means a bad node modifies all or some of the packets that are supposed to be forwarded. It may also modify the data it generates to protect itself from being identified or to accuse other nodes.

This paper proposes a scheme to catch both packet droppers and modifiers. At first routing tree is established using DAG . Data is transmitted along the tree structure toward the sink. A packet sender or forwarder adds a small number of extra bits, which is called packet marks, is designed such that the sink can obtain the dropping ratio associated with every sensor node. Node categorization algorithm to identify nodes that are droppers/modifiers for sure or are suspicious droppers/ modifiers[1]. 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. The outline of this paper is as follows. In section 2, the related works are discussed in detail. In section 3 the proposed work is discussed. In section 4 and 5 , the experimental results and conclusion are given.

2. RELATED WORK

WSNs are mostly unguarded and the wireless medium is inherently broadcast in nature. This makes WSNs vulnerable to all kinds of attacks. Without proper security measures, an adversary can launch various kinds of attacks in hostile environments. These attacks can disrupt the normal working of WSNs and can even defeat the purpose of their deployment. An adversary can launch some attacks without even cracking keys used for cryptography-based solutions. DoS attacks (like packet dropping, false route request, or flooding) can deplete the network of energy without much effort on the part of an adversary[7],[8],[9]. Therefore, intrusion detection mechanisms to detect DoS attacks are needed. To be practical for implementing on WSNs.

Existing solutions for detecting packet dropping in ad hoc networks work by monitoring individual nodes. Sleep-wakeup schedules followed by nodes in a WSN make continuous monitoring impractical. Also, monitoring Individual nodes is too expensive for WSNs. DPDSN (Detection of Packet Dropping attacks for wireless Sensor Networks)[10], can also be used for multipath routing but through all paths cannot be tolerated. Marti *et al.* [2] discussed two techniques that detect compromised nodes that agree to forward packets but fail to do so. The authors use *watchdog*s that identify misbehaving nodes and a *pathrater* that helps routing protocols avoid these nodes. 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 promiscuously to the next node's broadcast transmissions. If the next node does not broadcast the packet, it is misbehaving and the watchdog detects it. Every time a node fails to forward a packet, the watchdog increments the failure-tally. If the tally exceeds a certain threshold, it is determined that the node is misbehaving; this node is then avoided with the help of the *pathrater*. The *pathrater* combines knowledge of misbehaving nodes with link reliability data to pick the route most likely to be reliable. Each node maintains a rating for every other node it knows about in the network. It calculates a path metric by averaging the node ratings in the path. The overhead of passive continuous passive listening is formidable for WSNs.

For packet modification the existing systems aim to filter modified messages en-route within a certain number of hop[3]s. 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

Ye et al. proposed a probabilistic nested marking (PNM) scheme [4]. But with the PNM scheme, modified packets should not be filtered out en route because they should be used as evidence to infer packet modifiers; hence, it cannot be used together with existing packet filtering schemes. Since these approaches have several

drawbacks like incompatibility with existing systems , high energy cost etc ,our proposed work can be used to overcome these constraints.

3. PROPOSED WORK

Our proposed scheme consists of a initialization phase and compromised nodes identification phases.

3.1 *Conditions for the Application of Algorithm* **:**

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 towards a sink. The sink is located within the network. We assume all sensor nodes and the sink are loosely time synchronized which is required by many applications. The sink is aware of the network topology, which can be achieved by requiring nodes to report their neighboring nodes right after deployment.

3.1.1 *Initialization Phase*:

In the initialization phase, sensor nodes form a topology which is a directed acyclic graph (DAG). A routing tree is extracted from the DAG. Data reports follow the routing tree structure. The purpose of system initialization is to set up secret pair wise keys between the sink and every regular sensor node. 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
- Nsth packet is numbered Ns _ 1, the Ns -1th packet is numbered 0, and so on and so forth.
- Ns: the maximum packet sequence number.

3.1.2 *Intruder Identification Phase*

In each round, data are transferred through the routing tree to the sink .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 a node categorization algorithm to identify nodes that must be bad nodes and suspiciously bad. The routing tree is reshaped every round, when a certain number of rounds have passed, sink collects enough information about node behaviors in different routing topologies. The information includes which nodes are bad for sure and which nodes are suspiciously bad. To further identify bad nodes from the potentially large number of suspiciously bad nodes, the sink runs heuristic ranking algorithms.

3.1.3 *Packet Sending*

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

$<P_u, \{ R_u, u, C_p \text{ MOD } N_s, D, pad_{u,0} \}_{Ku}, pad_{u,1}>$

Where P_u - parent node, R_u - receiving node, U- node, C_p - counter node, D - data, $pad_{u,0}$ -padding, $_{Ku}$ encryption. Paddings pad u,0 and pad u,1 are added to make all packets equal in length, such that forwarding nodes cannot tell packet sources based on packet length. Meanwhile, the sink can still decrypt the packet to find out the actual content.

3.1.4 Packet forwarding

When a sensor node v receives packet hv;mi, it composes and forwards the following packet to its parent node Pv:

$<P_v, \{ R_v, m' \} K_v>$

where m0 is obtained by trimming the rightmost log (Np) bits off m. Meanwhile, Rv, which has logNp bits, is added to the front of m'.

3.1.5 Packet receiving at the sink

The sink attempts to find a child node for every parent node by decrypting which results in a string. If the attempt fails the packet is modified and it should be dropped.If it succeeds the packet is forwarded from the respected node.

Algorithm 1. Packet Receipt at the Sink

1: Input: packet <0;m>.

2:if Success Attempt = false then decrypt

3: if decryption fails then continue, else

4: if Success Attempt = true then record sequence

5: u ←v, Success Attempt = false; go to line 4;

6: if Success Attempt = false then

7: drop this packet;

Algorithm 2. Tree-Based Node Categorization Algorithm

1: Input: Tree T, with each node u marked by "+" or "_,"and its dropping ratio du.

2: for each leaf node u in T find parent node until the sink node categorize the nodes

3: consider u as positive threshold and v as negative threshold

4: if v.mark = ''_'' then until v.mark = ''+'' or v is Sink, Set nodes from b to e as bad for sure;

5: if v is Sink then Set u as bad for sure;

6: if v.mark =''+' and if v is not bad for sure then

Set u and v as suspiciously bad else

7: if dv - du > θthen

8: Set v as bad for sure;

9: if difference du – dv > θthen Set u and v as suspiciously bad;

- Nu,max - most recently seen sequence number
- Nu,flip - the number of sequence number flips
- nu,rcv - number of received packets.

The dropping ratio in each round is calculated as follows:

$$d_u = \frac{n_{u,flip} * N_s + n_{u,max} + 1 - n_{u,rcv}}{n_{u,flip} * N_s + n_{u,max} + 1}$$

To identify most likely bad nodes from suspicious nodes:

$$S_i = \{< u_j, v_j > | < u_j, v_j > is\ a\ suspicious\ pair\ and$$

$< u_j, v_j > = < v_j, u_j >\}$

3.1.6 Ranking algorithms

I. Global ranking based approach

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. The node with the highest value is chosen as a most likely bad node and all the pairs that contain this node are removed.

2. Stepwise ranking based approach

It can be anticipated that the GR method will falsely accuse innocent nodes that have frequently been parents or children of bad nodes. Once a bad node u is identified, for any other node v that has been suspected together with node u, the value of node v's accused account is reduced by the times that u and v have been suspected together.

3. 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. After a most likely bad node has been chosen, the one with the highest accused account value among the rest is chosen only if the node has not always been accused together with the bad nodes that have been identified already.

i. Packet Modifiers

Modified packets can be detected with the afore-described scheme. Modified packets will be detected by sink and it will be dropped and hence packet modifier can be identified as packet dropper. To enable en-route detection of modifications, the afore-described procedures for packet sending and forwarding can be slightly modified as follows. when a node *u* has a data item *D* to report, it can obtain endorsement message authentication codes (MACs) from its neighbors, which are denoted as *MAC*(*D*), following existing en-route filtering schemes such as the statistical en-route filtering scheme (SEF) [5] and the interleaved hop-by-hop authentication scheme [6].

4. SIMULATION RESULTS

The objectives of this evaluation study are firstly, testing the effectiveness and efficiency of our scheme in identifying packet droppers and modifiers; secondly, studying the impacts of various system parameters. We compare the proposed global ranking (GR), stepwise ranking (SR), and hybrid ranking (HR) algorithms to provide insights on the behavior of parameters. We measure the performance of our scheme with two metrics: the detection rate defined as the ratio of successfully identified bad nodes; the false positive probability defined as the ratio of mis-accused innocent nodes over all innocent nodes

4.1 Impact Of Round Length

Considering the delay for transmitting a packet from a source node to the sink, the round length affects the number of packets received at the sink in each round, which in turn affects the detection performance. It can be seen that round length mainly affects the false positive probability.

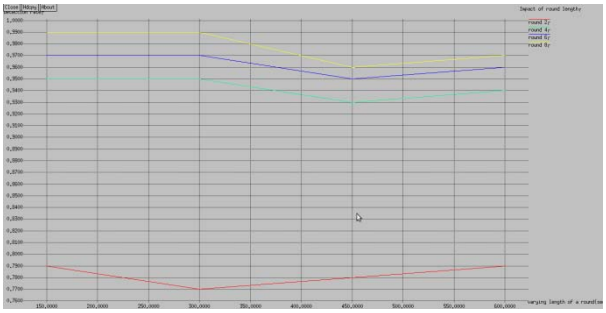

Figure 2a. Detection Rate

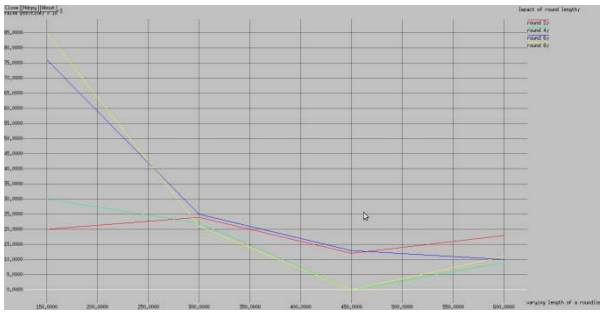

Figure 2b. False Positive

4.2 Impact Of Reporting Interval

When the sample space is small because of large reporting interval, the variance of the dropping ratio could be large, resulting in large false positive probability. This explains the phenomenon shown in fig. 3(b), the false positive probability goes up when the reporting interval increases. When the number of rounds is small, fig. 3(a) shows that the detection rate decreases as the reporting interval increases.

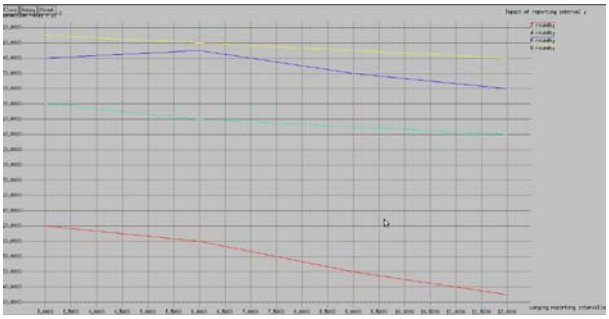

Figure 3a. Detection Rate

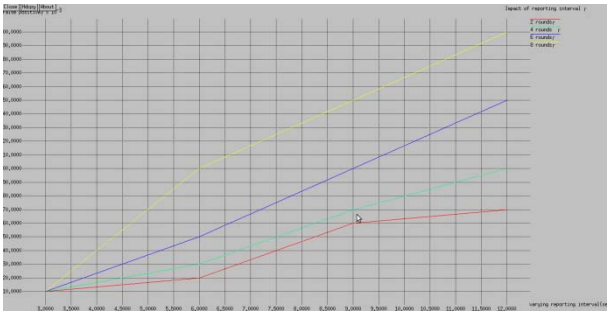

Figure 3b. False Positive

4.3 IMPACT OF DROPPING PROBABILITY

Fig. 4 shows the performance sensitivity to bad node's dropping percentage (i.e., the percentage of packets that will be dropped if a bad node decides to drop packets in a round). We vary the dropping probability between 20% and 80%. From Fig.4,we can see the all the three ranking algorithms have similar sensitivity to the dropping probability. In addition, with a high dropping probability, all the three algorithms achieve a higher detection rate in the early rounds, which means they can detect bad nodes quicker, and can achieve a lower false positive generally. This is because frequent misbehaviors can quickly distinguish bad nodes from innocent nodes..

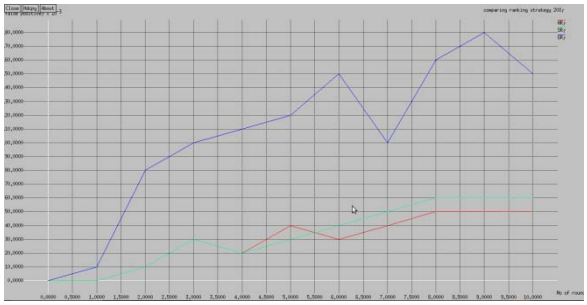

Fig 4a. Dropping Probability 20% - False Positive

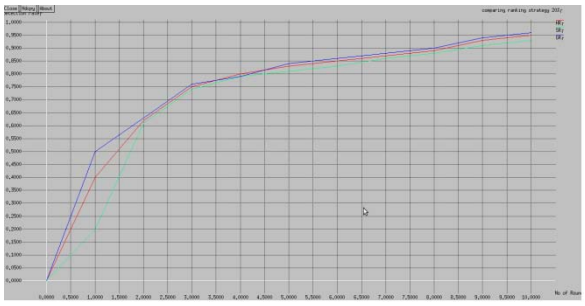

Figure 4b. Dropping Probability 20% - Detection Rate

4.4 IMPACT OF THRESHOLD

Threshold for Differentiating "+" Nodes and "-" Nodes. In order to tolerate incidental packet loss, we use a threshold θ when marking each node with "+" or "-". Fig. 4 shows the impact of this threshold on the detection performance.As depicted in Fig. 5(a), the larger is the threshold, the lower is the detection rate. This is because, fewer nodes will be marked as "-" as the threshold increases; hence, a part of bad nodes may escape from being detected.

Figure 5a. Dropping Probability 80% - False Positive

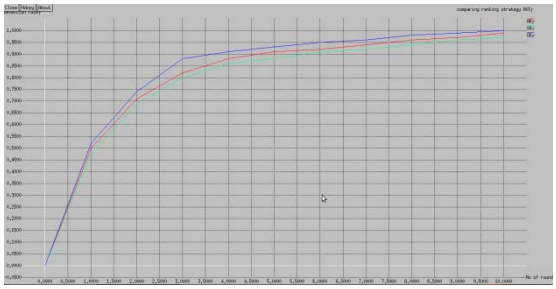

Figure 5b. Dropping Probability 80% - Detection Rate

As shown in Fig. 5(b), when the threshold increases, the false positive probability increases first and then decreases after the threshold reaches a certain value (turning point).

5.CONCLUSION

Thus the proposed scheme is effective in both detecting and filtering packet droppers and modifiers. The bad nodes can be identified from the suspiciously bad nodes. The node categorization and heuristic ranking algorithms are used for this purpose. Extensive simulations have been done to prove the effectiveness of our scheme.

REFERENCES

[1] Chuang Wang, Taiming Feng, Jinsook Kim, Guiling Wang and Wensheng Zhang, Catching Packet Droppers And Modifiers In Wireless Sensor Networks, ieee transactions on parallel and distributed systems, vol. 23, no. 5, may 2012.

[2] Marti, S., Giuli, T. J., Lai, K., and Baker, M., .Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,. Proc. 6th Annual Intl. Conf. on Mobile Computing and Networking (MobiCom.00), Boston, Massachusetts, August 2000, pp. 255-265.

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

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

[5] Z. Yu and Y. Guan, "A Dynamic En-route Scheme for Filtering False Data in Wireless Sensor Networks," Proc. IEEE INFOCOM,2006.

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

[7] Issa Khalil, Saurabh Bagchi, "MISPAR: Mitigating Stealthy Packet Dropping in Locally-Monitored Multi-hop Wireless Ad Hoc Networks".

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

[9] K. Liu, J. Deng, P.K. Varshney, and K. Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in Manets," IEEE Trans. Mobile Computing, vol. 6, no. 5, pp. 536-550, May 2007.

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