Effective Identification of Packet Droppers and Modifiers in Wireless Sensor Networks

Ranjini M G [1] | Mr.M Jayashankar [2]

[1]PG Student, Department of CS & E, PES College of engineering, Mandya, Karnataka, India
[2]Asst.professor, Department of CS & E, PES College of engineering, Mandya, Karnataka, India

ABSTRACT

In Wireless sensor network, sensor nodes are used to monitor physical or environmental condition. Sensor networks are 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 subjected 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 packet. In this paper, we propose a simple yet effective scheme to identify misbehaving forwarders that drop or modify packets. Node Categorization algorithm and Global Ranking algorithm are used to identify compromised nodes. Encryption techniques are provided to ensure reliable communication.

KEYWORDS: *Packet dropping, packet modification, wireless sensor network.*

I. Introduction

In a wireless sensor network, sensor nodes monitor the environment, detect events of interest, produce data and collaborate in forwarding the data towards a sink, which could be a gateway, base station, storage node. 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.

In this paper, we propose a simple yet effective scheme to identify 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. As the tree structure dynamically changes every certain 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 run our proposed *Global ranking algorithms* to identify most likely bad nodes from suspiciously bad nodes.

Compared with existing schemes, our scheme has the following unique characteristics:

(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

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

II. Related Works

There are several approaches made for detection of vulnerable attacks. [2][3][4][5][6] Deals with packet dropping. [2] Detection of packet dropping attacks for WSN proposes a solution to identify paths that drop packets by using alternate paths, but it succeeds only when the alternate path does not have any malicious nodes. [3] In this scheme single path data forwarding is employed and later it is convertor in multipath data forwarding. [4][5][6] are related to routing process and neighbor monitoring mechanism. [7][8][9] Deals with packet modification. [7] In this SEF detect and filters out false reports based on probabilistic key distribution. [9] Proposes Location Based Resilient security to filter out packets, but in spite of all filtering techniques intruders are able to move on and communication overhead is increased. [10] Probabilistic Nested marking is proposed to locate vulnerable nodes and it does so within the framework of packet marking, but the evidence to find packet modifiers are also filtered out. [11] In this paper extensions to Dynamic Source Routing are given such as watchdog and pathrater. Watchdog identifies misbehaving nodes and pathrater helps routing protocols avoid those nodes. Few existing system deals with selective forwarding attacks which corrupt time critical application, to overcome this factor [12] is proposed where checkpoint based multi hop acknowledgement scheme for detecting selecting forwarding attacks. Acknowledgement based identification are performed through [17][20].

III. The Proposed Scheme

A. *DAG (Direct Acyclic Graph) Establishment:*

- A routing tree is extracted from sensor node in the form of DAG ie,extracting a route without forming cycle.
- For DAG establishment it uses dynamic source routing protocol (it uses source routing instead of relying on the routing table at each intermediate device).
- Initially Base station/sink does not have a route to the reach end sensor. When it has data packets to be sent to the destination, it initiates a RouteRequest packet. This RouteRequest is flooded throughout the network. Each node, upon receiving a RouteRequest packet, rebroadcasts the packet to its neighbour. Each RouteRequest carries a sequence number generated by the source node and the path it has traversed. A node, upon receiving a RouteRequest packet, checks the sequence number on the packet before forwarding it. The packet is forwarded only if it is not a duplicate RouteRequest. The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions of the same RouteRequest by an intermediate node that receives it through multiple paths. Thus, all nodes except the destination forward a RouteRequest packet during the route construction phase. A end sensor, after receiving the first RouteRequest packet, replies to the source node through the reverse path the RouteRequest packet had traversed. Nodes can also learn about the neighboring routes traversed by data packets if operated in the promiscuous mode (the mode of operation in which a node can receive the packets that are neither broadcast nor addressed to itself).

B. *Route registration and key distribution:*

Each sensor sends report/acknowledgment on all computed path to Base station/sink .Base station on receiving route reports from all sensor, it registers all sensor node. Generates a unique secret key for each sensor and transmits key to respective sensor .

C. *Data transmission*

a. *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 u has a data item D to report, it composes and sends the following packet to its parent node Pu as shown in eq1:

$$<Pu\ .\ \{Ru\ ,\ \ u.\ Cp \text{ MOD } Ns\ ,D\}> \ldots\ldots(1)$$

Where, Cp MOD Ns is the sequence number of the packet. Ns is the maximum packet sequence number. Ru is a random number picked by node u ,its attached to the packet to enable the sink to find out the path along which the packet is forwarded. $\{X\}Y$ represents the result of encrypting X using key Y .

b. *Packet Receipt at the Sink:*

When the sink receives a packet *h*0*; mi*, it conducts the following steps:
(i) Initialization: We introduce two temporary variables *u* and *m0*. Let *u* = 0 and *m0* = *m*.
(ii) The sink attempts to find out a child of node *u*, denoted as *v*, such that *dec*(*Kv;m*) results in a string starting with *Rv*, where *dec*(*Kv;m*) means the result of decrypting *m* with key *Kv*.
(iii) If the attempt fails, the packet is identified as being modified and thus should be dropped.
(iv) If the attempt succeeds, it indicates that the packet was forwarded from node *v* to node *u*. Now, there are two cases:
– If *dec*(*Kv;m*) starts with *hRv; vi*, it indicates that node *v* is the original sender of the packet. The sequence number of the packet is recorded for further calculation and the receipt procedure completes.
– Otherwise, it indicates that node *v* is an intermediate forwarder of the packet. Then, *u* is updated to be *v*.

Algorithm 1: Packet Receipt at the Sink

```
Input: packet<0, m>.
u =0,ḿ=m;
success_attempt=false;
p= dec(k, ḿ) return success_attempt;
 if decryption fails && Success_attempt = false
Drop this packet.
Break;
else
If Success Attempt= true then
  record sequence, Success Attempt=false.
```

c. *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. To let the sink and the nodes have a consistent view of their parent nodes, the tree is reshaped as follows. At the beginning of each round *I*(*i* = 1*;* 2....), node *u* picks the [*hi*(*Ku*) MOD *np.u*] parent node as its parent node for this round, where *h* is a hash function and *hi*(*Ku*) = *h*(*hi*¡1(*Ku*)). Note that, how the parents are selected is predetermined by both the preloaded secret *Ku* and the list of parents recorded in the tree establishment phase. The selection is known by the sink. Therefore, a misbehaving node cannot arbitrarily select its parent in favor of its attacks.

D. *Node categorization*

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.
In the end of each round, the sink calculates the dropping rate for each node *u*. Suppose *nu;max* is the most recently seen sequence number, *nu;flip* is the number of sequence number flips and *nu;rcv* is the number of received packets.The dropping ratio in this round is calculated as follows:

$$du = \frac{nu;flip * Ns + nu;max + 1 - nu;rcv}{nu;flip * Ns + nu;max + 1} \quad ....(2)$$

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 *Θ* 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 *Θ*. Note that *Θ* should be greater than 0, taking into account droppings caused by incidental reasons such as collisions.

E. *Node ranking module*

a. *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. With this method, each suspicious node *u* is associated with an *accused account* which keeps track of the time that the node has been identified as suspiciously bad nodes. To find out the most likely set of suspicious nodes after *n* rounds of detection, as described in Algorithm 2 , all suspicious nodes are ranked based on the descending order of the values of their accused accounts. The node with the highest value is chosen as a most likely bad node and all the pairs that contain this node are removed from *S*1........*Sn*, resulting in new sets. The process continues on the new sets until all suspicious pairs have been removed.
Algorithm 2: The Global Ranking-Based Approach
1: Sort all suspicious nodes into queue *Q* according to the descending order of their accused account values 2: $s' \leftarrow ø$

3: *while* $\bigcup_{i=1}^{n} si \neq ø$ *do*

4: $u \leftarrow deque(Q)$

```
5: $S \leftarrow S' \wedge \{u\}$
6: remove all $< u; * >$ from $\bigcup_{i=1}^{n} si$
```

F. *Encryption and Decryption technique*

Symmetric-key encryption are algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link.

a. *Rijndael Symmetric Algorithm*

The Advanced Encryption Standard (AES), also known as Rijndael is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

For a given secret key k, a simple block cipher that does not use an initialization vector will encrypt the same input block of plain text into the same output block of cipher text. If you have duplicate blocks within your plain text stream, you will have duplicate blocks within your cipher text stream. If unauthorized users know anything about the structure of a block of your plain text, they can use that information to decipher the known cipher text block and possibly recover your key. To combat this problem, is Rijndael algorithm is used.

In Rijndael algorithm, information from the previous block is mixed into the process of encrypting the next block. Thus, the output of two identical plain text blocks is different. Because this technique uses the previous block to encrypt the next block, an initialization vector is needed to encrypt the first block of data.

- Generates complex key by adding secrete key and salt data(Salt Data helps in creating key which is harder to guess)
- Generate the two key from complex key.that is main key of 32 bytes(256 bits) and initialization vector (Iv is needed to encrypt the first block of data) of 16bytes(128 bits).
- This Encryption technique uses the previous block to encrypt the next block.
- First block of data is encrypted using 32 bytes main key and 16 bytes initialization vector . ie, Encryption key(48 bytes)=M.K(32 bytes) + I.V(16 bytes).
- Second block is encrypted by 32 bytes main key and 16 bytes of previous word .ie, Encryption key(48 bytes)=M.K(32 bytes) + previous (16 bytes)
- Third block is encrypted by 32 bytes main key and 16 bytes of previous word .ie, Encryption key(48 bytes)=M.K(32 bytes) + previous (16 bytes)

The decryption algorithm is identical with the encryption algorithm uses the same key schedule.

IV. Conclusion

This paper proposes a simple yet effective scheme to identify misbehaving forwarders that drop or modify packets. 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. The routing tree structure dynamically changes in each round so that behaviors of sensor nodes can be observed in a large variety of scenarios. Finally, most of the bad nodes can be identified by our Node categorization and global ranking algorithms.

References

[1] Chuang Wang, Taiming Feng, Jinsook Kim, Guiling Wang and Wensheng Zhang, "Catching Packet Droppers and Modifiers in WirelessSensor Networks" in IEEE Trans on Parallel Distributed Systems, vol. 36, no. 5, May 2012.

[2] Vijay Bhuse, Ajay Gupta, and Leszek Lilien, "DPDSN: Detection of packet-dropping Attacks for Wireless sensor Networks," Proc. FourthTrusted Internet Workshop, 2005.

[3] S. Lee and Y. Choi, " A Resilient Packet-Forwarding Nodes in Sensor Networks," Proc. Frouth ACM Workshop on Security of Ad Hoc andSensor Networks (SASN '06), 2006.

[4] R. Mavropodi, P. kotzanikolaou, and C. Douligeris, "Secmr-A Secure Multipath Routing Protocol for Ad Hoc Networks, vol. 5, no. 1, pp. 87- 99, 2007

[5] I. Krontiris, T. Ginneetsos, and T. Dimitriou, "LIDeA: A Distributed Lightweigth Intrusion Detection Architecture for Sensor Networks," Proc.Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), 2008.

[6] 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

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

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

[9] H.Yang, F. Ye, Y.Yuan, S. Lu, and W. Arbaugh, "Toward Resilient Security in Wireless Sensor Networks," Proc. Sixth ACM Int'l Symp. Mobile Ad "Hoc Networking and Computing (MobiHoc '05), 2005S.Zhu, S.Setia, S.Jajodia, and P.Ning, "An Interleaved Hop-by-HopAuthentication Scheme for Filtering False Data in Sensor Networks," Proc. IEEE Symp. Security and Privacy, 2004.

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

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

[12] B. Xiao, B. Yu, and C. Gao, " Chemas: Identify Suspect Nodes in Selective Forwarding Attacks," J. Parallel and Distributed Computing,Vol. 67, no. 11, pp.1218-1230, 2007.

[13] H. Chan and A. Perrig, "Security and privacy in sensor networks," IEEE Computer, Digital Object Identifier vol. 36, no. 10, pp. 103-105,Oct 2003

[14] G.Padmavathi and D. Shanmugapriya "A survey of Attacks, security mechanisms and challenges in wireless Sensor Network," Int'l Journal of Computer science and Information Security (IJCSIS), 2009.

[15] J.M. Mccune, E. Shi, A. Perrig, and M.K. Reiter, "Detection of Denial-of-Message Attacks on Sensor Network Broadcasts," Proc. IEEE Symp. Security and Policy, 2005R. Roman, J. Zhou, and J.Lopez, "Applying Intrusion Detection Systems to Wireless Sensor Networks,"Proc. IEEE Third Consumer Comm. Networking Conf. (CCNC), 2006.

[16] B. Yu and B.Xiao, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks," Proc. 20th Int'l Symp. Parallel and DistributedProcesssing (IPDPS), 2006.

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

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

[19] B. Barak, S. Goldberg, and D. Xiao, "Protocols and Lower Bounds for Failure Localization in the Internet," Proc. Eurocrypt, 2008.

[20] K. Ioannis, T. Dimitriou, and F.C. Freiling, "Towards Intrusion Detection in Wireless Sensor Networks," Proc. 13th European Wireless Conf., 2007.