Security and Reliability Enhancement in Catching Packet Droppers and Modifiers in Wireless Sensor Networks

Mrs.Sameeksha Chaudhary [1],Mr. Mukesh Goyal[2]
[1]Lecturer in ECE,[2]Lecturer in ECE
[1]Chandravati Group of Institutions,Bharatpur, India

***Abstract* - Dropping packets and modification are usual attacks that can be exposed by an opponent to interrupt the communication in wireless sensor networks. Sensor nodes are used here to monitor the environments, detect the event of interest, generate data and collaborate in forwarding the data towards a sink, which could be a gateway, base station, storage node, or querying utilize. Due to the fluent deployment, less cost of sensor node and the capability of self-organization, a sensor network is often deployed in an unattended and flightier environment to perform monitoring and data conduction tasks. When it is deployed in such an environment,it deficient the physical groin which subject to node compromise. An opponent may release assorted attacks and disrupt the network communication after compromising one or multiple sensor nodes. Due to these repetitive attacks, compromised nodes modify the packets that they are supposed to forward. A number of schemes have been proposed to diminish such attacks. Some of them can accomplished by identify the violator. To represent this state of uncertainty, Our proposed classification enables a systematical categorization of mechanisms and protocols to cope with attacks in WSNs. Since attacks can seriously disrupt the functionality of a WSN and nearly all WSNs are susceptible to insider attacks, appropriate security mechanisms and protocols are required. For a deeper understanding of the different aspects of insider attacks in WSNs, we proposed several algorithm to protect against certain types of packet modifiers and a general approach to protect against all types of modifiers.**

***IndexTerms* - Wireless Sensor Network, Multiohop, Dropping packet, Modifying packet.**

I INTRODUCTION

WSN is very useful in Disaster Relief Operations. We can drop sensor nodes from an aircraft over a wild fire. Then each node measures temperature and through we can easily derive a temperature map. We can also use sensor nodes to observe wild life. We can use WSN to construct Intelligent buildings .We can reduce energy wastage by proper humidity ventilation, air conditioning. We can also measure room occupancy, temperature and air flow. We can also monitor mechanical stress after earthquakes.WSN is also used in machine surveillance and preventive maintenance. We can also monitor tyre pressure which is not possible to measure earlier. Through WSN we can precise the use of pesticides, fertilizer in agriculture.WSN has wide applications in the field of medical and health care. The lifetime of the networks can be increased by efficiently using the energy and increasing the message transfer reliability. To make the communications efficient and simple, simple protocol architecture can be designed as their processing capabilities are low. However Wireless detector networks comprises sizable amount of little detector nodes having restricted computation capability, restricted memory area, restricted power resource, and short-range radio communication device. With a widespread readying of those devices, one will exactly monitor the surroundings. Basically, detector networks square measure application dependent and detector nodes monitor the surroundings, notice events of interest, manufacture information, and collaborate in forwarding the info toward a sink, that may well be a entry, base station, storage node, or querying user. A detector network is usually deployed in unattended and hostile surroundings to perform the observation and information assortment tasks. Once it's deployed in such surroundings, it lacks physical protection and is subject to node compromise. Once compromising one or multiple detector nodes, AN opponent could lunch varied attacks to disrupt the in-network communication. This paper deals with 2 common attacks, dropping packets and modifying packets which might be launched by compromised nodes. Existing answer for detection packet dropping in Wireless detector Networks is multi path forwarding, during which every packet is forwarded on multiple redundant methods and therefore packet dropping in some however not all methods of those methods can be tolerated. And for detection packet modifiers, most of existing step aim to filter changed message en-route with in an exceedingly bound variety of hops. These counter measures will tolerate or mitigate the packet dropping and modification attacks, however the intruders square measure still there and might continue offensive the network while not being caught.
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. 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 heuristic ranking algorithms with small false positive

I. Basic idea

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.

I. Design phase:

The approaches for detecting packet dropping attacks can be categorized as three classes: multipath forwarding approach, neighbor monitoring approach, and acknowledgment approach. Multipath forwarding is a widely adopted countermeasure to mitigate packet droppers, which is based on delivering redundant packets along multiple paths.

System architecture

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 toward 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.

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.

We also use Triple-DES and HMAC algorithm on every node to improve security.

HMAC

Two parties communicating across an insecure channel need a method by which any attempt to modify the information sent by one to the other, or fake its origin, is detected. Most commonly such a mechanism is based on a shared key between the parties, and in this setting is usually called MAC, or Message Authentication Code
Triple DES.

The DES is based on the work of IBM Corporation, and was adopted as the American National Standard (ANSI) X3.92-1981/R1987. The DES algorithm was adopted by the U.S. government in 1977, as the federal standard for the encryption of commercial and sensitive-yet-unclassified government computer data and is defined in *FIPS 46* (1977). (FIPS are Federal Information Processing Standards published by NIST).
A "block cipher" refers to a cipher that encrypts a block of data all at once, and then goes on to the next block. The DES, which is a block cipher, is the most widely known encryption algorithm. In block encryption algorithms, the plaintext is divided into blocks of fixed length which are then enciphered using the secret key. The DES is the algorithm in which a 64-bitblock of plaintext is transformed (encrypted/enciphered) into a 64-bit cipher text under the control of a 56-bit internal key, by means of permutation and substitution.

Our proposed scheme consists of a system initialization phase and several equal-duration rounds of intruder identification phases.

- 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.
- 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 (i.e., packet droppers or modifiers) and nodes that are suspiciously bad (i.e., suspected to be packet droppers and modifiers). The routing tree is reshaped every round. As a certain number of rounds have passed, the sink will have collected information about node behaviors in different routing topologies. The information includes which nodes are bad for sure, which nodes are suspiciously bad, and the nodes' topological relationship. To further identify bad nodes from the potentially large number of suspiciously bad nodes, the sink runs heuristic ranking algorithms

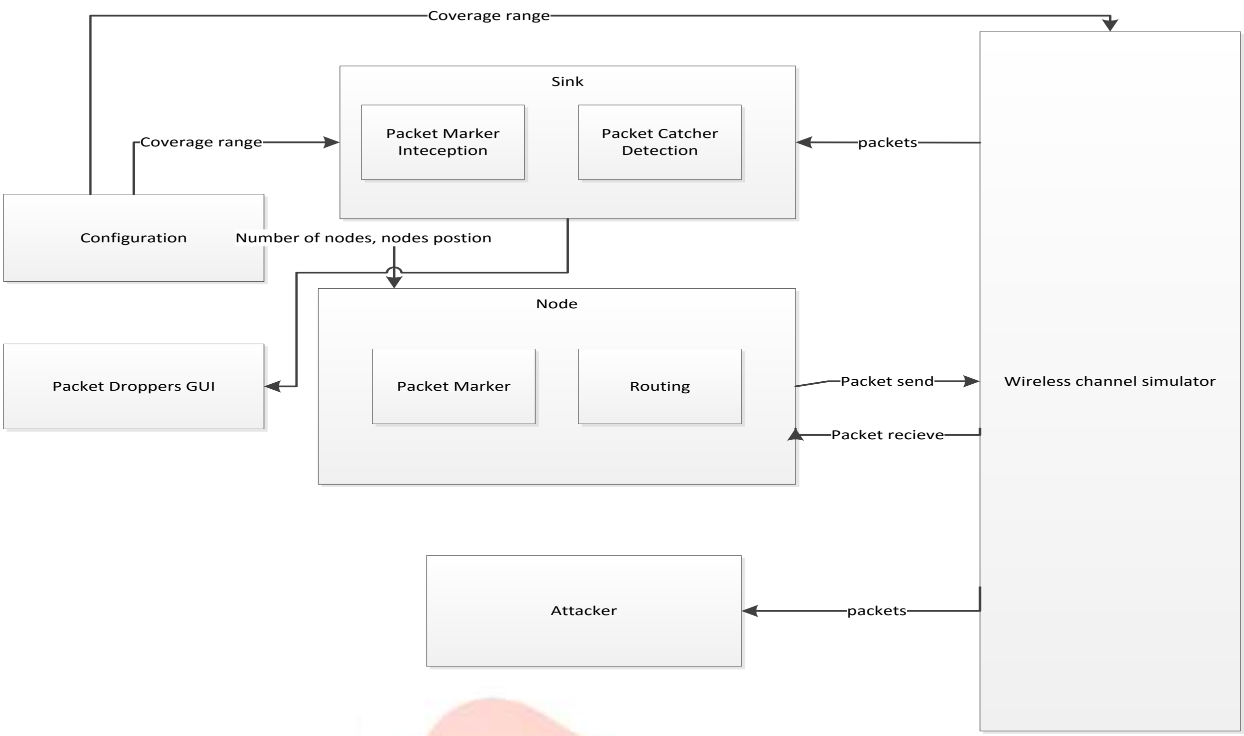


Fig1.Packet forwarding scheme

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. 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 decrypts it, and thus finds out the original sender and the packet sequence number. The sink 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 identifies packet droppers based on rules we derive. In detail, the scheme includes the following components, which are elaborated in the following.

Preloading keys and other system parameters. Each sensor node 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 (Ns+1)th packet is numbered 0, and so on and so forth.

Topology establishment:

After deployment, the sink broadcasts to its one-hop neighbors a 2-tuple. In the 2-tuple, the first field is the ID of the sender (we assume the ID 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:

On receiving the first 2-tuple (v,dv), node u sets its own distance to the sink as du=dv+1.

Node u records each node w(including node v) as its parent on the DAG if it has receive (w,dw); where dw=dv. That is, node u records as its parents on the DAG the nodes whose distance (in hops) to the sink is the same and the distance is one hop shorter than its own. If the number of such parents is greater than Np, only Np parents are recorded while others are discarded. The actual number of parents it has recorded is denoted by Np,u.

After a certain time interval, node u broadcasts 2-tuple (u, du) to let its downstream one-hop neigh-bors to continue the process of DAG establishment. Then, among the recorded parents on the DAG,Node u randomly picks one (whose ID is denoted as Pu) as its parent on the routing tree. Node u also picks a random number (which is denoted as Ru) between 0 andNp1. As to be elaborated later, random number Ru is used as a short ID of node u to be attached to each packet node u forwards, so that the sink can trace out the forwarding path. Finally, node u sends Pu,Ru and all recorded parents on the DAG to the sink.

After the above procedure completes, a DAG and a routing tree rooted at the sink is established. The routing tree is used by the nodes to forward sensory data until the tree changes later; when the tree needs to be changed, the new structure is still extracted from the DAG. The lifetime of the network is divided into rounds, and each round has a time length of Lr. After the sink has received the parent lists from all sensor nodes, it sends out a message to announce the start of the first round, and the message is forwarded hop by hop to all nodes in the network. Note that, each sensor node sends and forwards data via a routing tree which is implicitly agreed with the sink in each round, and the routing tree changes in each round via our tree reshaping algorithm presented in next section.

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.

Pu,{Ru,U,Cp MOD Ns,D,PADu,o}ku,PADu,1},……….(2.1)

Where Cp MOD Ns is the sequence number of the packet.

Ru (0<Ru<Np-1) is a random number picked by node u during the system initialization phase, and Ru is 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.

Node Categorization Algorithm

In every round, for each sensor nodeu, the sink keeps track of the number of packets sent fromu, the sequence numbers of these packets, and the number of flips in the sequence numbers of these packets, (i.e., the sequence number changes from a large number such as Ns1to a small number such as 0). In the end of each round, the sink calculates the dropping ratio 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:

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

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. For each of these scenarios, node categorization algorithm is applied to identify sensor nodes that are bad for sure or suspiciously bad. After multiple rounds, sink further identifies bad nodes from those that are suspiciously bad by applying several proposed heuristic methods.

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←0
3:while U^ni=1 Si≠0 do
4:u←deque(Q)
5:S←S∧{U}
6: remove all (U,*) from U^ni=1 Si.
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: as parents or children of bad nodes, according to previously described rules in Cases 3 and 4, the innocents can often be classified as suspiciously bad nodes. To reduce false accusation, we propose the SR method.

Algorithm:The Stepwise Ranking-Based Approach

1:S←0
2: while U^ni=1 Si≠0 do
3: u the node has the maximum times of presence in S1,...,Sn
4. S←S∧{U}
5:remove all (U,*) from U^ni=1 Si.

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. To strike a balance, we further propose the HR method, which is formally presented in Algorithm 5. According to HR, the node with the highest accused account value is still first chosen as most likely bad node. 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.

Algorithm :The Hybrid Ranking-Based Approach

1: Sort all suspicious nodes into queue Q according to the descending order of their accused account values
2: S←0
3:while U^{n}i=1 Si≠0 do
4:u←deque(Q)
5: if there exists(u,*)€ U^{n}i=1 Si.
6:S←S∧{U}
7:remove all (U,*) from U^{n}i=1 Si.

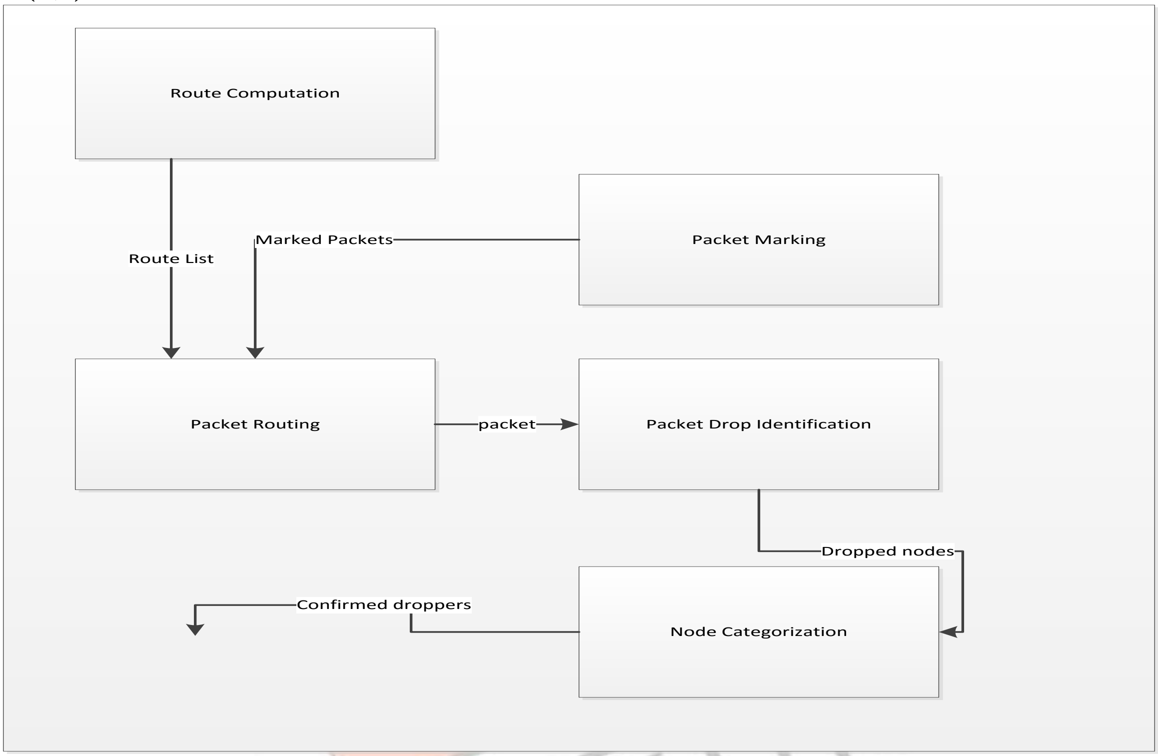


Fig2. Interaction of module

Packet Forward:

II. Implementation

1.RUN it initially.

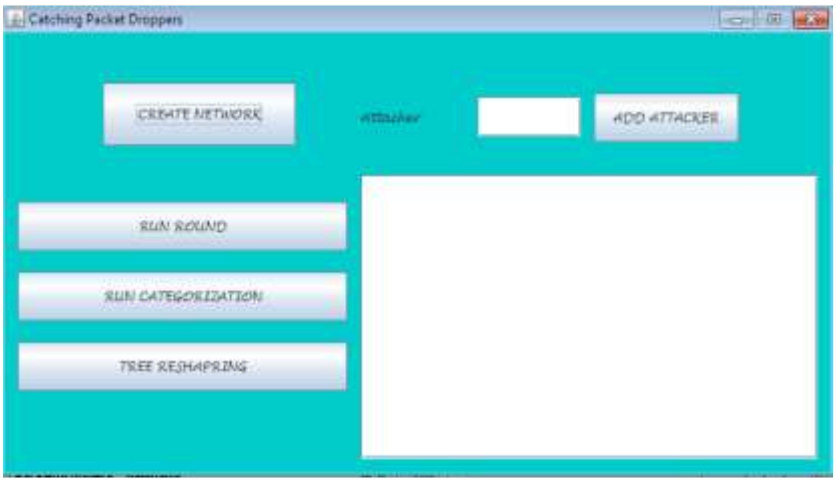
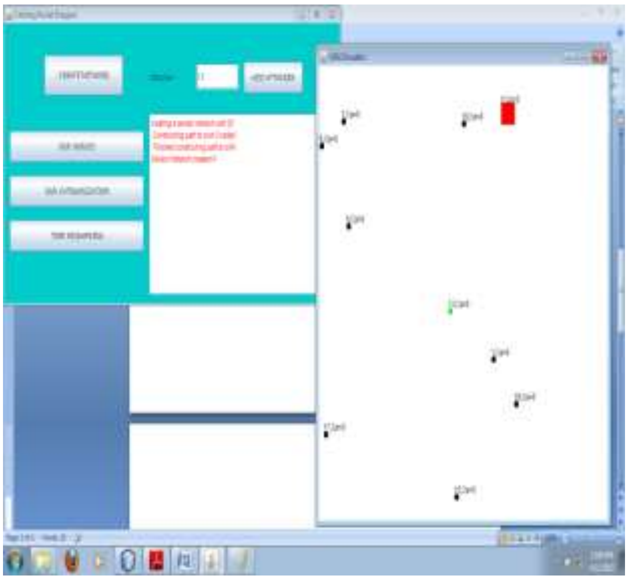
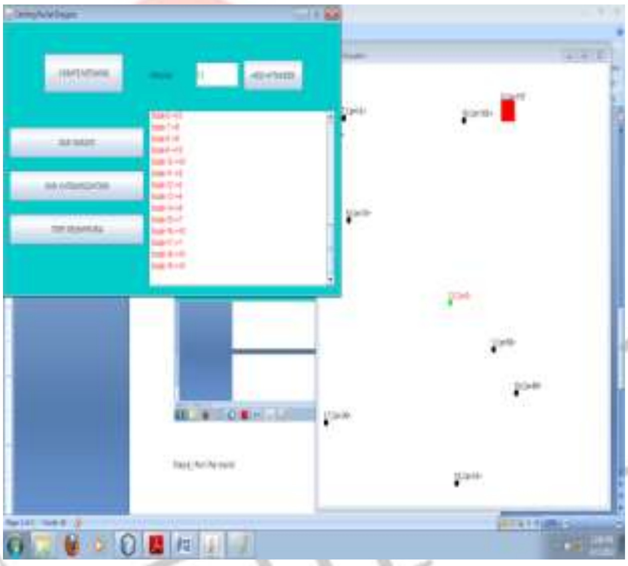


Fig.3.Create Network:

2.Nodes are generated.

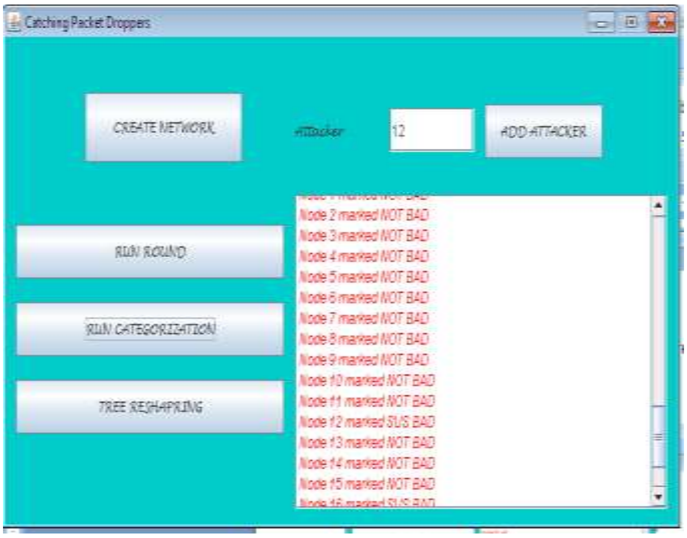


3. Mark certain nodes as attackers.

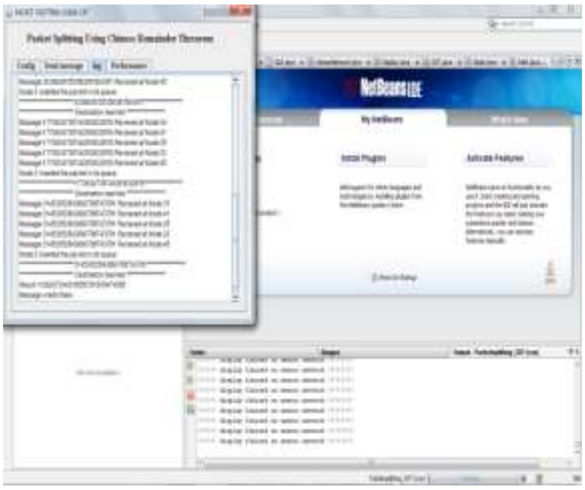


4.Run node categorization. Node as categorized into NOT BAD, SUS BAD, CONFIRMED BAD and it is shown in the log

:

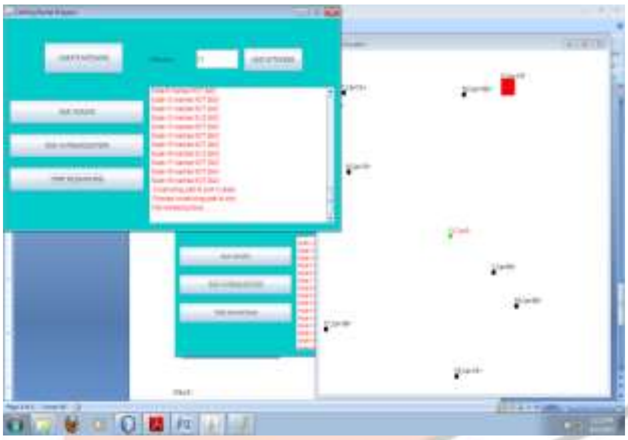


5. Log:



6. Do Tree reshaping to reconstruct route . Node marked as CONF BAD is left in tree construction.

Node marked as SU BAD is counted number of times so far they are suspected, If suspected count is more than 5 node is marked as CONF BAD.



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. 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 heuristic ranking algorithms with small false positive. Extensive analysis, simulations, and implementation have been conducted and verified the effective-ness of the proposed scheme.

References

[1] H. Chan and A. Perrig, “Security and Privacy in Sensor Networks,”Computer,vol. 36, no. 10, pp. 103-105, Oct. 2003.
[2] 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.
[3] V. Bhuse, A. Gupta, and L. Lilien, “DPDSN: Detection of Packet-Dropping Attacks for Wireless Sensor Networks,”Proc. Fourth Trusted Internet Workshop,2005.
[4] 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 and Sensor Networks (SASN ’06),2006.
[5] 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.
[6] F. Ye, H. Luo, S. Lu, and L. Zhang, “Statistical En-Route Filtering of Injected False Data in Sensor Networks,”Proc. IEEE INFOCOM, 2004.
[7] 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.
[8] H. Yang, F. Ye, Y. Yuan, S. L u, and W. Arbaugh, “Toward Resilient Security in Wireless Sensor Networks,”Proc. Sixth ACM Int’l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc ’05), 2005.

[9] Z. Yu and Y. Guan, “A Dynamic En-route Scheme for Filtering False Data in Wireless Sensor Networks,”Proc. IEEE INFOCOM, 2006.
[10] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,”Proc. ACM MobiCom, 2000.
[11] M. Just, E. Kranakis, and T. Wan, “Resisting Malicious Packet Dropping in Wireless Ad Hoc Networks,”Proc. Int’l Conf. Ad-Hoc Networks and Wireless (ADHOCNOW ’03),2003.
[12] R. Roman, J. Zhou, and J. Lopez, “Applying Intrusion Detection Systems to Wireless Sensor Networks,”Proc. IEEE Third Consumer Comm. Networking Conf. (CCNC),2006.
[13] S. Lee and Y. Choi, “A Resilient Packet-Forwarding Scheme Against Maliciously Packet-Dropping Nodes in Sensor Net-works,”Proc. Fourth ACM Workshop on Security of Ad Hoc and
Sensor Networks (SASN ’06),2006.
[14] I. Khalil and S. Bagchi, “MISPAR: Mitigating Stealthy Packet Dropping in Locally-Monitored Multi-Hop Wireless Ad Hoc Networks,”Proc. Fourth Int’l Conf. Security and Privacy in Comm. Netowrks (SecureComm ’08),2008.
[15] 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. Netowrks (SecureComm ’08),2008.
[16] S. Ganeriwal, L.K. Balzano, and M.B. Srivastava, “Reputation-Based Framework for High Integrity Sensor Networks,”ACM Trans. Sensor Networks,vol. 4, no. 3, pp. 1-37, 2008.
[17] 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.
[18] P. Michiardi and R. Molva, “Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks,”Proc. IFIP TC6/TC11 Sixth Joint Working Conf. Comm. and Multimedia Security: Advanced Comm. and Multimedia Security, 2002.
[19] S. Buchegger and J. Le Boudec, “Performance Analysis of the Confidant Protocol,”Proc. ACM MobiHoc,2002.
[20] F. Ye, H. Yang, and Z. Liu, “Catching Moles in Sensor Networks,” Proc. 27th Int’l Conf. Distributed Computing Systems (ICDCS ’07), 2007.
[21] Q. Li and D. Rus, “Global Clock Synchronization in Sensor Networks,”Proc. IEEE INFOCOM,2004.
[22] K. Sun, P. Ning, C. Wang, A. Liu, and Y. Zhou, “Tinysersync: Secure and Resilient Time Synchronization in Wireless Sensor Networks,”Proc. 13th ACM Conf. Computer and Comm. Security (CCS ’06),2006.
[23] H. Song, S. Zhu, and G. Cao, “Attack-Resilient Time Synchroniza-tion for Wireless Sensor Networks,”Ad Hoc Networks,vol. 5, no. 1, pp. 112-125, 2007.
[24] B. Xiao, B. Yu, and C. Gao, “Chemas: Identify Suspect Nodes in Selective Forwarding Attacks,”J. Parallel and Distributed Comput-ing, vol. 67, no. 11, pp. 1218-1230, 2007.
[25] X. Zhang, A. Jain, and A. Perrig, “Packet-Dropping Adversary Identification for Data Plane Security,”Proc. ACM CONEXT Conf. (CoNEXT ’08), 2008.
[26] Crossbow, “Wireless Sensor Networks,” Products/Wireless_Sensor_Networks.htm, 2011.
[27] T.H. Hai and E.N. Huh, “Detecting Selective Forwarding Attacks in Wireless Sensor Networks Using Two-Hops Neighbor Knowl-edge,” Proc. IEEE Seventh Int’l Symp. Network Computing and Applications (NCA ’08),2008.
[28] F. Liu, X. Cheng, and D. Chen, “Insider Attacker Detection in Wireless Sensor Networks,”Proc. IEEE INFOCOM,2007.
[29] K. Ioannis, T. Dimitriou, and F.C. Freiling, “Towards Intrusion Detection in Wireless Sensor Networks,”Proc. 13th European Wireless Conf.,2007.
[30] A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei, “Reputation and Trust-Based Systems for Ad Hoc and Sensor Networks,”Proc. Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks,2008.
[31] 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,2005 .
[32] B. Yu and B. Xiao, “Detecting Selective Forwarding Attacks in Wireless Sensor Networks,”Proc. 20th Int’l Symp. Parallel and Distributed Processing (IPDPS),2006.
[33] 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.
[34] B. Barak, S. Goldberg, and D. Xiao, “Protocols and Lower Bounds for Failure Localization in the Internet,”Proc. Eurocrypt,2008.