A Distributed Administration Based Approach for Detecting and Preventing Attacks on Mobile Ad Hoc Networks

Himadri Nath Saha , Prof. (Dr.) Debika Bhattacharyya , Prof.(Dr.) P. K. Banerjee

Abstract - Certain security attacks specific to Mobile Ad Hoc Networks (MANETs) such as black hole attacks, gray hole attacks and blackmail attacks and also flooding attacks are lethal in terms of hampering availability of network service. In this paper, we propose a protocol for detecting flooding, black hole, gray hole and blackmail attacks and taking measures against the nodes committing them. Our scheme is based on a concept of an underlying backbone network of administrator nodes that we assume to be trustworthy and honest throughout. These administrators have greater transmission and reception range than the general nodes in the MANET and have the power to take corrective actions on the basis of the reports sent by the other nodes. The association of these administrator nodes is dynamically increased to ensure better network coverage by upgrading certain general nodes to become administrators subject to certain constraints such as the transmission and reception range and the performance over a sufficiently large period of time. We have modeled a possible life cycle for a general node in the network and have shown how our protocol unlike the existing ones is resilient and conservative while taking actions against any node emphasizing that an honest node should not be penalized by mistake. We give an elaborate description of the procedures and how they lead to detection of the attacks.

Keywords: Black hole attack, Blackmail attack, MANET,Gray hole attack, Flooding, Watch Node, Adminstrator.

1. INTRODUCTION

THE security of communication in ad hoc wireless networks is very important and at the same time is much more challenging than it is for structured networks. Security attacks on MANETs can be broadly classified into *active* and *passive* attacks. In passive attacks, the malicious nodes attempt to obtain information form the network without disrupting the network operations. On the other hand, active attacks hamper network operations and can be carried out by nodes that are external or internal to the network. Internal attacks are harder to tackle as the nodes carrying them out are already accepted as a part of the network and are associated with other nodes in the network through already established trust relationships. We are concerned about such internal nodes that carry out active attacks like flooding, black hole, gray hole and blackmail attacks after establishing themselves in the network. Before we proceed to deal with the detection and prevention of these attacks, it is important to thoroughly understand these attacks and their characteristics.

Flooding attack: In flooding attack, a malicious node sends a huge number of junk packets to a node to keep it busy with an aim to prevent it from participating in other activities in the network. This can lead to an obvious disruption of network availability as the nodes communicating with the victim will not be attended. Apart from this threat, other complicacies can also be generated as follows:

- Two malicious nodes can cooperate to carry out an attack where one floods an honest node in their vicinity while the other carries out a packet dropping (black hole) attack thereby preventing the honest node from detecting the black hole attack being carried out by the other malicious node.

- In critical situations, where a node comes up and is waiting for receiving its identity from the other existing nodes, a malicious node can flood this node or the neighbors to delay the acceptance of the new node in the network.

These are two small examples out of many possible which justifies the need for a protocol which detects and takes action against the nodes trying to flood the network.

Black hole attack: In black hole attack, a malicious node upon receiving a route request packet from a node replies by sending a false routing reply to the sending node to misguide it to send the data to it and then it drops the data packet. This is the simplest way a black hole attack can be carried out and it is trivially easy to detect the node that is dropping all packets and consequently isolate it in the network. Let us look into a more complex scenario. In a situation where a group of nodes cooperate to create a black hole where the data packet is transferred and retransferred within the black hole until it runs out of its *time to live* (TTL) and gets eventually dropped without causing the node dropping it to be blamed anyhow. This is how cooperative black hole attack is carried out and it is increasingly challenging to detect the chain of nodes responsible and take corrective measures.

Gray hole attack: In a gray hole attack, the malicious nodes are harder to be detected as they selectively drop packets. Such a malicious node can pretend to be honest over a time in the network observing a pattern in the traffic flow. For instance, say after a node comes into the network and an existing node receives the request for identity from the new node, then it is expected that the existing node will reply with an identification data to the new node. At this moment, a malicious node can drop packets from any nodes meant for the new node thereby preventing or delaying participation of the new node in the network.

Blackmail attacks: In a blackmail attack, or more effectively a cooperative blackmail attack, malicious nodes complain against an honest node to make other nodes that need to send data to believe that routing through the victim is harmful. Such attacks can prevent senders from choosing the best route to the destination thereby hampering efficiency and throughput in the network.

Having discussed the threat areas we now introduce our approach to fight against these attacks. The crude definition of MANET calls for a cluster of mobile nodes with equal or different computing power but with equal status inter-communicating without taking the aid of any central authority whatsoever. An ideal MANET as per the basic definition incorporates only peer to peer communication.

In other words such networks are structure less.

We however have been deeply influenced by the concept of using a logical structure over on infrastructure less network as in [1].

We emphasize on deploying of MANET for a definite purpose such as military activities, fighting disaster in calamity struck areas and so on. So it is not unjust to assume that there exists a logical authority in the form of an individual or a group who sets up the communication network for a purpose and that they will always be honest. On the basis of this assumption we mark these nodes as administrator nodes that place themselves in positions so as to ensure maximum network coverage. They have large transmission and reception power than the general nodes that participate in the network activities. We make the general nodes go through four phases in their lives in the network, namely, WHITE, GRAY, BLACK and BLUE. As we have said earlier, our scheme takes special care to avoid rash decisions taken on nodes to prevent honest nodes getting misjudged as malicious. A WHITE node is one which is honest with a high probability and is the default phase of a new node in the network. A node which is under suspect is made GRAY and a GRAY node which does not improve its behavior is made a BLACK node and its isolation is effected. A WHITE node that has transmission and reception powers comparable to that of the administrators and has been honest for a sufficiently large amount of time can be upgraded to the status of BLUE nodes. We have given the general nodes the power to watch the activities of other nodes and judge independently whether other nodes in its vicinity are carrying out malicious activities. On detecting such activities, the general nodes can send out complains to the administrators who have the authority to take corrective measures on the basis of the received complains.

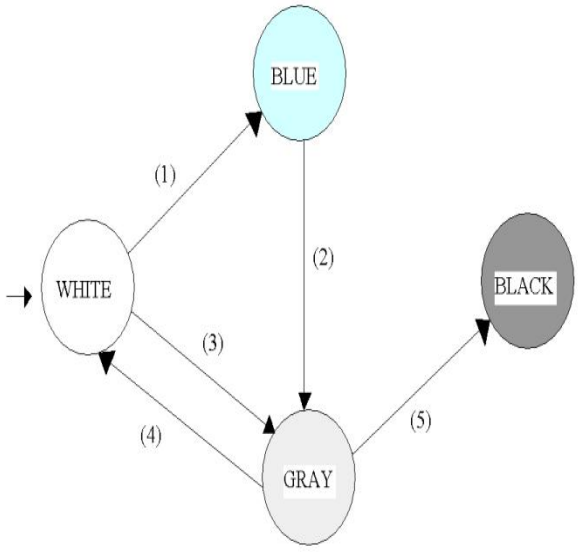

Fig 1. State diagram representing the phases in the life cycle of a general node in the network

A node is WHITE when it comes into the network. It can make transition (1) provided its hardware capabilities are as strong as the administrators and it has shown good behavior over a sufficient amount of time. However, we are very conservative about giving unlimited power to such a node. One of the existing administrators has to accept it as a BLUE node after which it can take corrective measures regarding security in the network. However any discrepancy of decision making if found out with any other initial administrators, the BLUE node is penalized by making the transition (2). A white node under suspicion is made to undergo transition (3). If a gray node's behavior continues to be suspicious then it is made to undergo transition (5). However if the GRAY node shows good behavior over a sufficiently long period of time, there is a high chance that the suspicion was erroneous and hence transition (4) is effected.

The rest of the paper proceeds as follows:

In section 2, we discuss related works in this area and the concepts which have influenced our work. Following this, we have presented the methodology of our work in section 3 and in section 4 we have concluded citing the scopes for further works in this area.

2. Related works

Agrawal et. al. [1] have proposed a protocol on the basis of the logical backbone network over infrastructure less ad hoc networks. Their protocol also assumes that the strong nodes that form the backbone are honest. They have utilized the concept of sending data in terms of small equal sized packets rather than a continuous stream. According to their scheme however, regular nodes are not capable of monitoring the activities of other nodes which according to them strengthens the chances of black hole attacks. We contradict on this issue and our scheme allows general nodes to monitor the activities of other nodes. The decisive actions however can be taken only by the administrator nodes. Thus we have distributed the monitoring work among all the nodes which adds more generality to our protocol. From [2], we get the concept of guard nodes to monitor the activities of other nodes within their range. Razak, Furnell, Clarke and Brooke [3] describes a two tier Intrusion Detection System using a friend approach which is capable of minimizing the impact of colluding blackmail attacks in the system. Another algorithm [4] is not robust with respect to cooperative malicious nodes in the network. [5] presents another compute intensive algorithm which cannot tackle gray hole attacks. In [6] we get a similar concept of backbone network that we have incorporated in our scheme.

The improvements we have sought in our scheme are efficiency and robustness related apart from our concern towards honest nodes not having to face penalty due to their behavior during abnormal traffic loads or movements in the network.

3. Methodology

For our convenience, we have visualized the MANET area on X-Y two dimensional plane where each node is aware of the coordinates (x,y) they are currently placed in. Moreover, we have assumed that the nodes have enough hardware capabilities to judge the coordinates of a node from which they are able to receive packet in a single hop. Next, we give procedures that the general nodes and the administrators execute followed by descriptions about them.

List of Terms

GEN_SEND_DATA()	A procedure used by general nodes while sending data
P	Packets
ACK	Acknowledgements
ALERT	An alert signal to caution the administrators of a possible black or gray hole attack
GEN_WATCH_NODES()	A procedure used by the general nodes to monitor the activities of other nodes in its vicinity

SND	Original sender of a packet
PREV	The node from which the packet has arrived
NEXT	The node to which the packet goes on the next hop
DST	The final destination of the packet
τ	Timestamp of the last noticed packet between a sender destination pair
f	Frequency of packet transmission between a sender destination pair
f_{max}	The threshold frequency beyond which we conclude chances of flooding attack
T_{safe}	The safe time limit beyond which packets transmitted between same sender destination pair are not subjected to suspicion of flooding attack
b	One flag bit to give suspected nodes a chance before registering complain
	against them
TTL	Time to live of a packet
TIMEOUT	A clock based time out event
DOUBT_COUNT_BLACK	A counter to store the doubts that a node is carrying out black hole attack
DOUBT_COUNT_GRAY	A counter to store the doubts that a node is carrying out gray hole attack
DOUBT_TOLERANCE	The threshold limit beyond which complain is registered against a node
L	The complain list
INIT_GEN_NODE()	A procedure which describes how a general node comes up and becomes a part of the network
PENALIZE(node)	A procedure to effect transitions as shown in Fig. 1
DETECT_BLACK_GRAY()	A procedure incorporated by the administrator nodes to detect cooperative black and gray hole attacks
ACT_ADMIN_NODE()	A procedure which describes the active

	life of an administrator node
ACT_GEN_NODE()	A procedure which describes the active life of a general node
ADMIN_SEND_DATA()	A procedure used by an administrator while sending data

Procedures for a General Node

GEN_SEND_DATA()

1. Decompose the message to be sent into small and equal sized packets, say P[1], P[2], ..., P[N].
2. Select next hop destination using routing protocol.
3. Set P[0] to a random nonce.
4. Send P[0] first, set timer and wait for ACK.
5. If ACK received within timeout, then repeat step 3 with P[1] and so on till P[N] and then go to step 7.
6. If timeout occurs, then possibility of black or gray hole attack.
7. Send ALERT in Broadcast mode.
8. Return.

In line 3 of the above procedure, the random nonce is used to verify whether at present the discovered route to the destination is proper. We believe that the initial stage of sending data is more critical in terms of becoming victims of black or gray hole attacks. However whenever we sense failure of packet delivery we alert the administrators in the neighborhood who subsequently trigger a detection procedure.

GEN_WATCH_NODES()

1. Start timer.
2. If there is packet being sensed, then
3. Capture packet P at time t.
4. Get Header information <SND, PREV, NEXT, DST>
5. If there is no record of the form <SOURCE, DESTINATION, TIMESTAMP, FREQUENCY> as <PREV, NEXT, τ, f> in the audit file, then add a record by making τ=t and f=1.
6. Else if $f<f_{max}$, then
7. $t-\tau \le T_{safe} \rightarrow f=f+1$
8. $T_{safe} < t-\tau \le 2T_{safe}$ and $f > 0 \rightarrow f=f-1$ (slow retreat)
9. $t-\tau > 2T_{safe}$ and $f > 0 \rightarrow f=[f/2]$ (fast retreat)
10. Else if $f \ge f_{max}$, then
11. If bit b=1, then append PREV into complain list L and reset b to 0.
12. Else if bit b=0 then set b=1 and refresh f to 0.
13. End if.
14. Update last entry of this packet in the audit file.

15. End if.
16. If TTL of P > 1, then
17. Start timer
18. If within TIMEOUT a packet from NEXT meant for some node, say x, does not arrive, then DOUBT_COUNT_BLACK is increased by 1 for node NEXT and packet dropping record is saved.
19. If for a node J, DOUBT_COUNT_BLACK exceeds DOUBT_TOLERANCE, then
20. If bit b=1 then, append J to complain list L, reset DOUBT_COUNT_BLACK to 0 and bit b to 0.
21. Else set b to 1 and reset DOUBT_COUNT_BLACK to 0.
22. End if.
23. If packet from NEXT meant for x had been dropped earlier then, double DOUBT_COUNT_GRAY.
24. If for a node J, DOUBT_COUNT_GRAY exceeds DOUBT_TOLERANCE, then
25. If bit b=1 then, append J to complain list L, reset DOUBT_COUNT_GRAY to 0, bit b to 0 and remove corresponding packet dropping record.
26. Else set b to 1 and reset DOUBT_COUNT_GRAY to 0 and remove corresponding packet dropping record.
27. End if.
28. Go back to step 1.

In the above procedure, lines 5 to 15 deal with the detection of a flooding attack. If the frequency of packet sending crosses a threshold between the same sender destination pair, then we start doubting a flooding attack. However, a sudden burst of data can be there at some instant. So such a behavior might not be caused by a flooding attack. Questioning the practicality of a flooding attack, we have concluded that a node with the aim of flooding the network will show such a behavior frequently if not continuously. So we give such a node a second chance with the help of the flag bit b. After the second chance, we make sure that the node makes a complain about the suspected node which will be attended to by the administrators. Lines 16 to rest deal with black and gray hole attacks. Whenever there is a packet being dropped whose TTL has not shrunk to 0, we start suspecting a black hole or gray hole somewhere. For black hole number of packets getting dropped will be much higher than for gray hole attacks. So we increase the suspicions of a black hole, that is, DOUBT_COUNT_BLACK linearly while we increase suspicions of a gray hole, that is DOUBT_COUNT_GRAY exponentially. Complains are registered once either of the doubt counts exceed a threshold DOUBT_TOLERANCE.

ACT_GEN_NODE()

1. Set timer.
2. If there is data to be sent, then create thread GEN_SEND_DATA() and execute it.
3. If there is an incoming packet to be routed, then use routing protocol to find next hop destination H and forward the packet to H.
4. Create thread GEN_WATCH_NODES() and execute it.

5. If NOT TIMEOUT, then go to step 2.
6. Create complain messages about the nodes in the complain list L.
7. Broadcast complain message.
8. Refresh timer and go back to step 1.

Here, we take care that we do not congest the network with complain messages. So we wait for a time over which we accumulate the complains in a list L and then send the complains together.

INIT_GEN_NODE()

1. Ascertain the current coordinates, say (x,y).
2. Create packet for IDENTITY_REQUEST and broadcast packet.
3. Set timer.
4. If IDENTITY_REPLY received, then go to step 7.
5. If NOT TIMEOUT, then go to step 4.
6. Move to a random (x′,y′) and repeat from step 2.
7. ACT_GEN_NODE()

(*Comment:* In this paper, we have not worked on dynamic identity allocation algorithms and we assume that all nodes possess valid identities obtained correctly and efficiently).

Procedures for an administrator node

PENALIZE(node)

1. If node is in WHITE list, then move its entry to GRAY list.
2. Else if node is in GRAY list, then move its entry to BLACK list.
3. Else if node is in BLUE list, then move its entry to GRAY list.
4. Else if node is in BLACK list, IGNORE.
5. End if
6. Broadcast updated message to other administrator nodes.

The above procedure makes use of the state diagram and its conditional transitions as presented in Fig. 1.

DETECT_BLACK_GRAY()

1. Note the SENDER and the DESTINATION of the failed packet.
2. Create more packets with the same SENDER DESTINATION pair each consisting of random nonce to aid in the detection
3. Send such a packet and store the hops.
4. If at any stage, a node is found to route the same packet twice in a circle, that node is subjected to PENALIZE(node) then and there.
5. Else if a single node is dropping packets, then a similar procedure as in GEN_WATCH_NODES() is incorporated and if criteria for attack are satisfied then PENALIZE(node) is effected.

In line 4, the idea we have presented is that for a cooperative black hole or gray hole attack,

more than one node form a closed loop in which they go on forwarding packets until TTL becomes 0. So a cycle detection not leading to packet dropping then and there is surely a malicious activity and needs to be taken care of immediately. The rest is similar to monitoring of the nodes by the general nodes.

ADMIN_SEND_DATA()

1. Decompose the message to be sent into small and equal sized packets, say P[1], P[2], ..., P[N].
2. Select next hop destination using routing protocol.
3. Set P[0] to a random nonce.
4. Send P[0] first, set timer and wait for ACK.
5. If ACK received within timeout, then repeat step 3 with P[1] and so on till P[n] and then go to step 8.
6. If timeout occurs, then possibility of black or gray hole attack.
7. Send a self ALERT to DETECT_BLACK_GRAY() thread.
8. Return.

The above procedure is similar to the one used by general nodes for sending data with the difference that it triggers its own action taking procedures rather than sending out complains.

ACT_ADMIN_NODE()

1. If there is data to be sent, then create thread ADMIN_SEND_DATA() and execute it.
2. If there is an incoming packet to be routed, then use routing protocol to find next hop destination H and forward the packet to H.
3. Probe on the forwarded packets to ensure end to end delivery. If not, then trigger DETECT_BLACK_GRAY() thread.
4. If there is an incoming complain list containing a complain about a node, say v, from u, then
5. If u is in WHITE list, then complain[v] is increased by 2.
6. Else if u is in GRAY list, then complain[v] is decreased by 1.
7. Else if u is in BLACK list, the ignore the complain.
8. Else if u is in BLUE list, then perform self detection to tally the results.
9. If Decisions match PENALIZE(v).
10. Else PENALIZE(u) keeping provisions for BLACKMAIL_DOUBT exceeding DOUBT_TOLERANCE..
11. End if
12. End if
13. If complain[v] exceeds threshold, then PENALIZE(v) and refresh complain[v].
14. End if.
15. Go back to step 1.

Lines 5 to 8 deal with giving priorities to the complaints from other nodes. If the node complaining is WHITE then higher priority is given to the complain than if the node

complaining is GRAY. We have not written other procedures that an administrator performs that are not related to the security aspects we have dealt with in our scheme. It is worth mentioning that like the general nodes, the administrators also run a similar monitoring over other nodes. The difference is that an administrator can watch over a huge area while the general nodes cannot. Then naturally a question might arise that what is then the need for the general nodes to watch other nodes' activities? The answer to this is however trivially simple. As the area of observation is large, so the number of nodes to be watched is much larger for an administrator. Consequently the process is slower. So, the general nodes' monitoring over other nodes increases the optimality of our scheme.

4. Conclusion and future work

We conclude by mentioning that we have worked upon the basic observation that the malicious nodes will continue showing malicious activities. Therefore we can differentiate between them and the nodes that are forced under situation to show behaviors which can be suspected to be malicious. Our protocol stands out among the rest in its conservative approach and avoidance to taking rash decisions. Moreover we keep provisions of nodes suspected before up to the GRAY level to be cleared of the suspicions in case improved behavior is noticed. These are extra points that we have mentioned about the protocol and which does not appear in the procedures. This is a deliberate attempt as we want to focus on the detection aspects in the procedures. We have tried to avoid clogging the procedures with supplementary features.

It is worth mentioning that there is scope of further research in this area, some of which are:

- The protocol can be further extended to incorporate the detection of worm hole attack as well
- Research on optimality analysis of our protocol is welcome.

5. References

[1] P. Agrawal, R.K Ghosh, S.K. Das, "Cooperative black and gray hole attacks in mobile ad hoc networks", ICUIMC'08.

[2] I. Khalil, S. Bagchi, N.B. Shroff, "MOBIWORP: Mitigation of the Wormhole attack in mobile multihop wireless network", Elsevier, 2007.

[3] S.A Razak, S. Furnell, N. Clarke, P. Brooke, "A Two-Tier Intrusion Detection System for Mobile Ad Hoc Networks – A Friend Approach", Springer-Verlag Berlin Heidelberg 2006.

[4] H. Deng, Wei Li, D.P Agrawal, "Routing Security in Wireless Ad Hoc Network", IEEE Communications Magazine, vol. 40, pp. 70-75, 2002.

[5] S. Ramaswamy, H. Fu, M. Sreekanaradhya, J. Dixon, K. Nygard, "Prevention of Cooperative Blackhole Attack in Wireless Ad Hoc Networks", Proceedings of 2003 International Conference on Wireless Networks, Las Vegas, Nevada, USA, pp. 570-575.

[6] I. Rubin, A. Behzad, R. Zhang, H. Luo, E. Caballero, "TBONE: A Mobile-Backbone Protocol for Ad Hoc Wireless Networks", Proceedings of IEEE Aerospace Conference, 2002, vol. 6, pp. 2727-2740.

[7] A.A Cardenas, T. Roosta, S. Sastry, "Rethinking security properties, threat models,

and the design space in sensor networks: A case study in SCADA systems", Elsevier, 2009.

[8] P.R Kumar, L.L Xie, "Ad Hoc Wireless Networks: From Theory to Protocols", Ad Hoc Wireless Networking, Kluwer Academic Publishers, 2004, pp. 175-195.

[9] W. Lou, Y. Fang, "A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available solutions", Ad Hoc Wireless Networking, Kluwer Academic Publishers, 2004, pp. 319-364.

[10] C.S.R. Murthy, B.S. Manoj, "Ad Hoc Wireless Networks", Pearson, 2004.

[11] J. Schiller, "Mobile Communications", Pearson, 2003.