Misbehaviour detection in Ad hoc Networks

T.Rajendran[1], Sreenaath.K.V[2]

[1,2]Security Engineer, Security, Technology Group,
Motorola India Electronics Pvt Limited, Bangalore, India
E-mail:rajendran.thirupugal@gmail.com, sreenaathkv@yahoo.co.in

Abstract. Ad Hoc network is a network where stations or devices communicate directly and not via an access point. It is a temporary grouping of stations to carry a specific program [1]. Since the network is ad hoc by nature, detecting and isolating misbehaving nodes is a critical task in ad hoc networks. Previous work on misbehaviour detection such as [1] depends on a central authority in detecting the misbehaviour which is an additional overhead. In this paper we propose a mechanism for misbehaviour detection in ad hoc network which defines its own protocol for both routing and data forwarding.

Keywords: Misbehaviour detection, Ad hoc networks, Next hop monitoring technique.

1 Introduction

An ad hoc network is a network in which a set of wireless stations communicate directly with one another without using an Access Point (AP) or any connection to a wired network. They have a wide array of military and commercial application. AODV is an reactive routing protocol for Ad hoc networks [5]. Ad hoc networks maximize total network throughput by using all available nodes for routing and forwarding. Therefore, more the number of nodes that participate in packet routing, greater the aggregate bandwidth, shorter the possible routing paths, and smaller the possibility of a network partition [1]. Despite all its advantages ad hoc network has the potential vulnerability by means of misbehaving nodes. A node can misbehave and fail to establish route or route the data due to its malicious nature to disrupt the network and take control of the system. Some of the previous works such as [1] suggests alternatives to overcome this problem. However they have their limitation in terms of dependency on a central authority in detecting the misbehaviour. In this paper we propose a mechanism for misbehaviour detection in ad hoc network which can be furthered and a new secure routing protocol can be developed. The paper is organized as follows. Section 2 details the assumptions the system take in to account while section 3 describes the concept behind detecting the misbehaviour. Section 4 and 5 explains detection techniques in route establishment and data forwarding. Section 6 briefs about phased detection technique while section 7 explains certification packet generation. Conclusions are drawn in section 8.

2 Assumptions

We assume the availability of a key management subsystem that makes it possible for each ad hoc node to obtain public keys of the other nodes of the network. Further, each node is capable of verifying the association between the identity of a node and the public key provided by the node. Typically this can be achieved by letting a Certification Authority (CA) generate public key, private key pairs and sign the public key and identity to create and issue public key certificates. Several distributed CA systems are proposed based on Threshold cryptography [2, 3, 4].

H() is a fast one way Hash function that generates a digest of the input it was provided with.

3. Concept

Our paper is based on the concept of next hop monitoring technique in which each node does its routing job correctly and monitors whether the immediate neighbour nodes are working according to the protocol. The key feature of the proposed scheme is its simplicity and effectiveness in identifying the malicious node.

4. Detection in route establishment phase

The main concept behind misbehaviour detection in route establishment phase is next hop monitoring technique.

- When a node receives RREQ it sends a RREQ certificate with some probability ‘p’ back to the node from which it received the RREQ. The idea of introducing a probability by which a RREQ certificate is sent back is to minimize the packet transfers among the nodes. A node would get RREQ certificates from a subset of its neighbours and these certificates will be used to prove that it has handled the RREQ packet appropriately.
- It broadcasts the RREQ to its one hop neighbours.
- Now it verifies whether all its neighbours are handling the RREQ packet correctly. A node would broadcast the RREQ packet it doesn’t have route to the destination or it isn’t the destination either. Otherwise it would reply with a RREP packet. So a node itself would be able to verify its neighbours by going to promiscuous mode and looking for RREQ broadcasts or RREP replies. To facilitate the verification step of its neighbours each node would maintain a list of its neighbours.
- If its neighbours are working as expected it replies back to the node from which it received the RREQ certifying that its neighbours are following the protocol.

- Then it will receive certificates from its neighbours saying that their respective neighbours are following the protocol.

If the node finds one of its neighbours doesn't handle the RREQ packet correctly it would request certificates to prove its handling. Thus this technique enables easy monitoring and verification of a node's activity and also enables detecting the malicious or selfish nodes which don't route RREQ packets.

5. Detection during data forwarding

The following two techniques can be used in detecting misbehaviour during data forwarding phase.

5.1. End to end route acknowledgement

In this technique whenever a node 'B' receives data packet from previous hop en route 'A', it generates a 'certificate of packet received' ('C_{AB}' representing node 'A' has sent data packet to 'B') and forwards the packet and then starts a timer. If it gets a certificate list packet 'C_{BC}','C_{CD}',..., 'C_{XDest}' from the next hop 'C' before the timer expires, 'B' would append its own certificate to the list and forward to its previous hop ('A'). If the timer expires before receiving the certificate list it would simply send the certificate 'C_{AB}' to 'A'.

When the source receives the certificate list '$C_{SourceA}$', 'C_{BC}', 'C_{CD}', ... 'C_{XDest}' it would be able to verify that all its neighbours are working correctly. This is shown in Fig. 1.

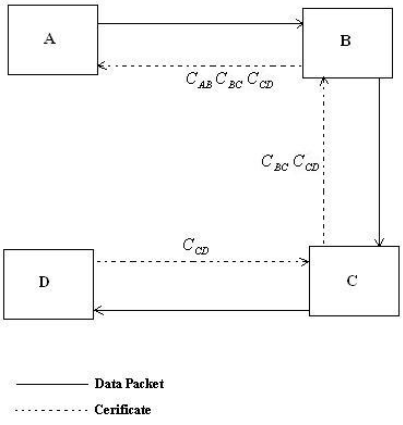

Fig. 1. End to end route acknowledgement

When source doesn't receive a particular certificate it would detect the exact link which is not available now or the link which involves malicious node. Consider a data transfer route involving the nodes 'A', 'B', 'C', 'D' as shown in Fig. 2. If the certificate 'CCD' is not available in the certificate list received by 'A', then the following scenarios are possible.

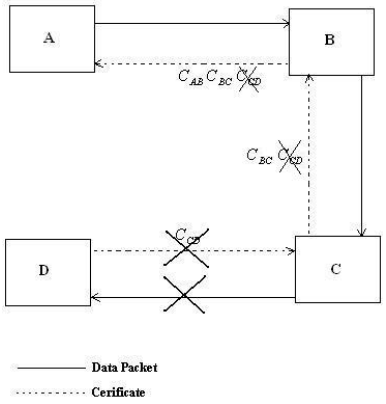

Fig. 2. Misbehaviour detection using **e**nd to end route acknowledgement technique

5.1.1. Link failure between 'C' and 'D'

In this case both 'C' and 'D' are performing correctly. 'C' would use the broadcasting method to prove that it is genuine.

- First, 'C' would alert its neighbour nodes that there was some problem with 'D'.
- The neighbour nodes would go into promiscuous mode and listen for transmission from 'C'.
- Now 'C' again transmits the packet.
- The neighbour nodes would detect this transmission and verify whether the packet is transmitted uncorrupted. Then 'C' is provided with 'certificate of packet received' along with neighbours public key certificates signed by CA.
- Now 'C' is able to prove that it actually has sent the packet. But 'D' is not able to receive it due to its misbehaviour or link failure.
- Neighbours of 'C' and 'D' would jointly analyse 'D's behaviour and decide whether it is malicious or link between 'C' and 'D' is broken.

In this case 'D' would be termed as a working node and the link would be detected as failed.

5.1.2. 'D' is malicious

In this case 'C' would use its neighbours to prove that it has transmitted the packet by the broadcasting technique. And 'D' would get detected as malicious.

5.1.3. ‘C’ is malicious

In this case ‘C’ would get detected quite easily as it won’t be able to prove that it is genuine.

5.2. Next hop monitoring technique

In this technique each node en route forwards the data packet to the next hop and monitors whether the next hop forwards the data packet appropriately. Each node en route needs to do the following

- Receive the packet from previous hop and send a certificate of packet received
- Forward the packet to the next hop according to the routing table.
- Receive certificate of packet received from next hop and forward it to previous hop.

In this technique each node (Y) would forward the data packet and get a proof (certificate of packet received from next hop (Z)) for it. Then it forwards the proof to the previous hop (X) en route (the node that is monitoring this node) for verification. It makes sure its next hop is working fine by verifying the proof provided by its own next hop. The packet transmission is shown in Fig. 3.

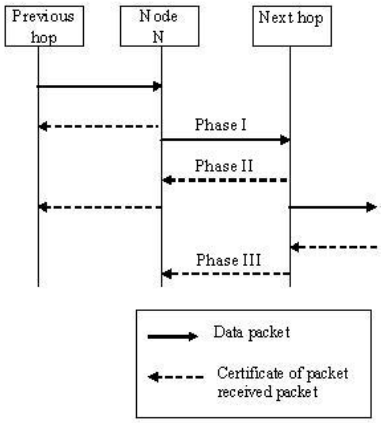

Fig. 3. Next hop monitoring technique

If there was a malicious node in the route that is not forwarding the data packets appropriately, either it won’t provide the proof of receipt to its previous hop or won’t be able to provide proof that it has forwarded appropriately. In both cases the node would get detected quite easily as follows.

Consider a data transfer between ‘A’ and ‘B’. The following is the normal operation divided into 3 phases. This is shown in Fig. 3.

Phase I

'A' forwards the packet 'p' to 'B'.

Phase II

'B' receives it and sends back a certification of received packet 'c_{ab}'.

Phase III

'B' forwards the packet according to its routing table and gets a certification 'c_{bx}' that it has forwarded correctly from the next hop (C).

'B' sends the certification 'c_{bx}' to 'A'.

The following are the different scenarios possible in the 3 phases due to misbehaviour of nodes and communication failure.

Phase I

- Link failure happens and the packet doesn't get transferred between 'A' and 'B'.
- 'A' is malicious and it either doesn't forward the packet or forwards a wrong packet.
- 'A' forwards the packet. But 'B' is malicious and it discards the packet received or says that the packet is corrupted.

Phase II

- Link failure between 'A' and 'B'.
- 'B' is malicious and either it doesn't send certificate 'c_{ab}' or sends a wrong certificate.
- 'A' is malicious and it either discards the received packet or says that the certificate is invalid.

Phase III

- Link failure between 'A' and 'B'.
- 'B' doesn't receive a certificate from next hop on the route due to some problem.
- 'B' is malicious and either it doesn't send certificate 'c_{bx}' or sends a wrong certificate.
- 'A' is malicious and it either discards the received packet or says that the certificate is invalid.

Now let's look at how these scenarios can be distinguished to identify if there is any link failure or any node is misbehaving using the next hop monitoring technique.

If a node 'X' wants to prove that it is sending the correct packet or valid certificate and 'Y' is misbehaving, then it employs the following steps.

- First, 'X' would alert its neighbour nodes that there was some problem with 'Y'.
- The neighbour nodes would go into promiscuous mode and listen for transmission from 'X'.
- Now 'X' again transmits the packet.

- The neighbour nodes would detect this transmission and verify whether the packet is uncorrupted or if it was a certificate whether it was valid. Then ‘X’ is provided with ‘certificate of packet received’ along with neighbours public key certificates signed by CA.
- Now ‘X’ is able to prove that it actually has sent the packet. But ‘Y’ is not able to receive it due to its misbehaviour or link failure.
- Neighbours of ‘X’ and ‘Y’ would jointly analyse ‘Y’s behaviour and decide whether it is malicious or link between ‘X’ and ‘Y’ is broken.

This technique can be used to identify the different scenarios listed above.

6. Phased detection techniques

Both the above mentioned misbehaviour detection techniques can be employed in a phased manner, in which a small number of consecutive nodes in the route employ this to detect and remove malicious nodes within themselves. This would reduce the no of packets transferred among nodes en route and the computing power needed to process the detection technique but still would detect the misbehaving nodes effectively.

7. ‘Certificate of packet received’ packet generation

Digital signature techniques can be employed to prove a node that a packet has been received by a node. In our paper we assume the availability of a Certification Authority to assign public key, private key pairs to nodes in the ad hoc network. Also we have ‘H()’ a fast one way hash function. In this case a node (B) receiving a packet (p) from a node (A) could generate a ‘Certificate of packet received’ packet as follows.

B hashes the packet to create digest of the message ‘H(p)’.

Then B uses its private key to sign the [digest, time ‘t’] packet and sends [signature, t] to A.

Now A could use this digital signature to prove anybody that B has received the respective packet at the time ‘t’.

8. Conclusion

In this paper we have proposed the mechanism for misbehaviour detection in Ad Hoc network. This method proves to be much more efficient than the existing mechanism and has less dependency with central authorities (nodes that detect the misbehaviour)

in detecting the misbehaving node. Further work needs to be carried out in terms of protocol definition, flow and implementation of the same.

References

1. Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", Proc. 6th annual International Conference on Mobile computing and Networking, U.S.A, 2006.
2. L. Zhou and Z. J. Haas, "Securing ad hoc networks," IEEE Network, vol. 13, no. 6, pp. 24–30, 1999.
3. M. Bechler, H.-J. Hof, D. Kraft, F. Pählke, L. Wolf, "A Cluster-Based Security Architecture for Ad Hoc Networks" IEEE INFOCOM 2004.
4. T. Pedersen, "A threshold cryptosystem without a trusted party," in Advances in Cryptology, Proc. Eurocrypt'91, ser. LNCS, vol. 547. Springer-Verlag, 1991.
5. C.E. Perkins, E.M. Royer, S.R. Das, "Ad hoc On-Demand Distance Vector Routing,", draft-ietf-manet-aodv-08.txt, IETF MANET Working Group, June 1st, 2001.