Black Hole Prevention using Digital Signatures and Hash Function in AODV Protocol

Adarsh D.V
4th sem M.tech CSE
SVIT, Bangalore

Mrs. Chandrika C N
Asst.prof Dept of CSE
SVIT, Bangalore

***Abstract:*- In this paper the security mechanism is shown to enhance performance & security of AODV (Adhoc On-demand Distance Vector) under black hole attack for MANET (Mobile Ad-hoc Networks). so, the use of security mechanisms helps to improve the security of AODV. The digital signature and hash chain to enhance the security and performance of the AODV from malicious and unauthenticated nodes.**

Keywords— Security, performance, AODV, black hole attack

I. INTRODUCTION

Adhoc networks is a collection of independent nodes that does not have a base station. Each node acts as a dual role of transmitter and receiver[1]. Nodes have a limited range of transmission so it takes the help of neighbouring nodes to forward the data[1],[2]. These networksdoesnt have the base statons. In some areas the base station can not be situated[2]. so the adhoc networks can come into picture. In the battle field ad hoc network, the centralized base station can not be situated so the adhoc networks play an important role in providing the communication[2],[3]. These kind of scenarios motivate us to make use of hash chain mechanism and digital signature in providing high security[5].

II. EXISTING SYSTEM

AODV is a distance vector routing protocol that has been naturally build for MANETs[3]. It is an on demand protocol and reactive in nature as it searching the routes only when required. AODV makes widespread use of sequence numbers in control packets to avoid the problem of generation of routing loops[4]. When a source node is interested to communicate with a destination node whose route is unknown, it broadcasts a RREQ (Route Request) packet[4],[5].

Each RREQ packet contains a Request ID, source and the destination node IP addresses and sequence numbers along with a hop count and flags. The Request ID field uniquely identifies the RREQ packet; the sequence numbers gives information regarding the freshness of control packets and the hop-count maintains the number of nodes between the source and the destination[6]. Recipient node of the RREQ packet that has not find the Source IP and ID
pair or doesn't maintain a fresher (larger sequence number) route to the destination rebroadcasts the same packet after incrementing the hop-count. Such intermediate nodes also create and preserve a REVERSE ROUTE to the source node for a certain time[7].

AODV message types

As in the FIG1 and FIG2 discusses about how the routing takes place prior to the data send and receive.

Route Request-RREQ

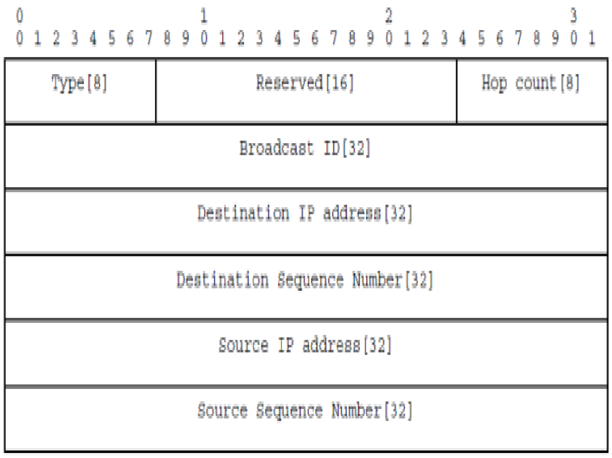

FIG 1 structure of RREQ packet

- Type: Type of message.
- Reserved :Reserved for future use
- Hop count: Numbers of hops from the source IP address to the node handling the request.
- Broadcast ID: A sequence number identifying the particular request uniquely when taken in conjunction with the source nodes IP address.
- Destination IP address: IP address of the destination for which a route is required.
- Destination sequence number: The last sequence number received in the past by the source for any route towards the destination.
- Source IP address: IP address of the node that originated the request.
- Source sequence number: current sequence number for route information generated by the source of the route request.

Route Reply-RREP

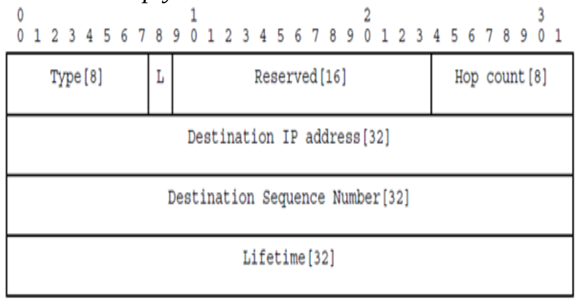

FIG 2 structure of RREP packet

- Type: type of message.
- L:If the L-bit is set the message is a hello message and contains a list neighbors.
- Reserved: Reserved for future use. Currently sent as 0 and ignored on reception.
- Hop count: Number of hops
- Destination IP address: IP address of the destination
- Destination sequence number: the destination sequence number associated to the particular route.
- Lifetime: Till what time it should exists in the network.

Hello

Hello messages are a special case of Route reply messages. The difference is that a hello messages always supplies the route to itself. This means that the hop-count field is set to 0, the destination address set to the IP address and the destination sequence number set to the nodes latest sequence number.

Link Failure

Link failure messages are also special route reply messages, but in this case the destination reflects the route that has broken. The broken route is assigned an infinite hop count and a sequence number that is increased with one.

When the RREQ packet arrived at the destination node or any intermediate node that has a fresher route to the destination a RREP (Route Reply) packet is generated and sent back to the source. RREP packet contains the destination node sequence number, the source and the destination IP addresses, route lifetime along with a hop count and flags.Intermediate node that receives the RREP packet, increments the hop count, establishes a Forward Route to the source of the packet and transmits the packet on the Reverse Route.

When a link failure is detected for a next hop of an active route a RERR (Route Error) message is sent to neighbors that are in the acive rote. The key vulnerabilities present in the basic AODV routing protocol are:

1. Deceptive incrementing of Sequence Numbers
2. Deceptive decrementing of Hop Count

AODV message pseudo codes

Sending RREQ

Whenever a source wants to send the data it broadcasts or floods the RREQ

```
1.  If ( route doesn't exists ){
2.  Check cache for already existence request that is sent for destination
3.   If ( the request has not been sent already ){
4.  Create a RREQ packet
5.  Add (dest addr, broadcast ID ) to cache
6.  broadcast RREQ locally
7.  the timer is set for RREP_WAIT_TIME for rebroadcasting RREQ
8.  Increment broadcast ID
9.  }
10. }
```

Receiving RREQ

When an intermediate node receives the RREQ packet, it should check that it has processed the RREQ already. If it is processed already then it discards the RREQ. Else the RREQ is cached such as broadcast ID and source address.

```
1.  If ( (broadcast ID, source addr ) are present in cache ) {
2.  Discard the request as it has been already processed
3.  } else {
4.  Add(broadcast ID, source addr) to cache
5.  }
```

The next step is to update or create the route entry in the routing table.

```
1.  If ( the route doesn't exists to source ) {
2.   Route entry is created for the source address
3.  } elseif (source seqno in route entry < source seqno in RREQ) {
4.  route entry is updated for the source address
5.  }else if ((source seqno in route entry = source seqno in RREQ ) AND
6.   (hop count in route entry < hop count in RREQ )) {
7.  source address is updated in the route entry
8.  }
```

Forwarding RREP

The receiving RREP is checked that it has been processed. If it is not been processed then it will forward the RREP

```
1.  If ( there is no route to the destination ) {
2.  route entry is created for the particular destination
3.  } elseif(Destination seqno in route entry < destination seqno in RREP){
4.  The route entry is updated for the destination
5.  } elseif (( Destination seqno in route entry = destination seqno in RREP )
6.  AND (hop count in entry > hop count in RREP))
7.  The route entry is updated for the destination
8.  }
```

```
9.  If ( there is a route exists already ) {
10. Forward the RREP to source
11. }
```

Receiving RREP

When the source receives the RREP the routing tabble is updated

```
1.  If(there is no route to the destination){
2.  A route entry for destination is created
3.  } elseif ((Destination seqno in route entry < destination seqno in RREP){
4.  route entry is updated for the destination
5.  }elseif((Destination seqno in route entry = destination seqno in RREP = ) AND
6.  (hop count in entry > hop count in RREP )) {
7.  Route entry is updated in the destination
8.  } else {
9.  RREP is discarded
10. }
```

The main disadvantage of the AODV is black hole attack. When the source requests for the route the attacker replies(RREP) to the source as pretending as the original destination. The black hole buffers the RREQ packet and this information is used to send the RREP to the destination.

As there is no security mechanism is implemented in AODV to with stand this black hole. When ever the destination recieves the RREP it begins to transmit the data packets that are to be sent.

III. PROPOSED SYSTEM:

In order to protect from the blackhole attack two data structures are added to the AODV and this routing protocol is called SAODV.

SAODV (Secure AODV) uses two Mechanisms

Digital Signature:- the digital signatures are used to maintain the identity of a particular user. i.e., no two users have the have digital signatures.

It is used to protect the integrity of the non-modifiable data in RREQ/RREP messages. Such as SourceIp, DestinationIP, FLAGS.

Hash Chain:-It is used to Authenticate the hop count of RREQ/RREP messages. This technique is used to authenticate each and every hop.

Steps for generating Hash Chain:

1. Generates a random number (seed).
2. Sets the Hopcount_Limit field to the TimeToLive (TTL) value.
3. Sets the Hash field to the seed value.
4. Sets the Hash Function field to the identifier of the hash function that it is going to use.
5. Calculates Top Hash by hashing seed Hopcount_Limit times.

IV. IMPLEMENTATION:

In this section we will briefly discuss about how to implement the black hole patch for the AODV and the implementation of SAODV.

Blackhole Implementation:

As in this blackhole first we need to add an agent for the attack in the sendReply function that exists in the AODV.CC and we need to create an keyword for the attacker and and in this function whenever an route request occurs it should cache it and it needs to send route reply as if it is destination.

And in the tcl script we need to call it by that keyword and also we need to specify the the time at which it needs to start its function.

SAODV Implementation:

The SAODV needs minor changes in the AODV protocol. First step is to add the data structure to the RREQ and RREP message. The data structures added to the protocol are digital signature, hop count and hash value in the packet.cc file. Second step is to add a random function(built in function in Ubuntu as rand as keyword). Third step is to add the digital signature by adding 5 to the index in the sendRequest.

Digital_signature=index+5;

Fouth step is to compute the hash value to the hop count using the hash value algorithm. Add these values to the generating the generated RREQ packet

Fifth step is to make changes in the recvRequest. The received message is taken and the digital signature, hash value is calculated and it is verified in each and every hop.

Sixth step the necessary data structure such as digital signature, hopcount and hash value is added to the RREP packet. These data structures are added in the sendReply function that exists in AODV.CC file.

Seventh step is to verify the digital signature and hash value that is recived by sendReply function compared with recvReply.

If all the parameters such as digital signature and hash values are verified in all the necessary nodes then the source sends the data otherwise it drops the data packets..

As the digital signature and hash mechanisms are used the black hole is identified in the next hop as the digital signature does not matches so the the RREP packet is dropped in the next node to the black hole node.

V. SIMULATION:

All simulation experiments are developed and simulated on an Intel I3 machine using Ubuntu 12.04 LTS with 4 GB RAM and the network simulator NS2 version NS- 2.35.

CONCLUSION:

This scheme fulfills almost all security requirements with using central certification authority and key management scheme which makes it more easily expandable and less complex in computation. According to the simulations that were performed in NS2, the newly proposed security scheme, built on top of normal AODV routing protocol, achieves an overall good results. Thus, the proposed scheme proves to be more efficient securing AODV routing protocol in defending against both malicious and unauthenticated nodes.

REFERENCES:

[1] Pirzada, McDonald, “Secure Routing with the AODV Protocol”, 2005 IEEE pp.57-61
[2] Stephan Eichler and Christian Roman, “Challenges of Secure Routing in MANETs: A Simulative Approach using AODV-SEC”, Aug 2006.
[3] Lu Jin, Zhongwei Zhang, Hong Zhou, “Performance Comparison of the AODV, SAODV and FLSL Routing Protocols in Mobile Adhoc Networks”, 2007
[4] Alaa S. Dalghan, Mohamad M. Gamloush, Raji M. Zeitouny, and Yasser M. Shaer, “Securing Mobile Adhoc Networks”
[5] Yih-Chun Hu, “A Survey of Secure Wireless Adhoc SRouting”, IEEE, 2004
[6] Monis Akhlaq, Nomal Jafri et al. “Data Security Key Establishment in AODV”, WSEAS, 2007
[7] Tuulia Kullberg, “Performance of the Ad-hoc On-Demand Distance Vector Routing Protocol”, 2004