A MODIFIED ADAPTIVE-SAODV PROTOTYPE FOR PERFORMANCE ENHANCEMENT IN MANET

[1]Alekha Kumar Mishra, [2]Bibhu Dutta Sahoo

[1,2]Department of Computer Science and Engg., National Institute of Technology, Rourkela, Orissa
[1]alekha@gmail.com, [2]bdsahu@nitrkl.ac.in

ABSTRACT:
Mobile ad hoc networks are vulnerable to various security threats because of its dynamic topology and self-configurable nature. SAODV (secured Ad hoc On Demand Vector routing) is one of the popular existing secured mechanism which takes help of digital signature and hash chain techniques to secured AODV packets. Since, digital signature technique consumes heavy computational time, the degradation of SAODV performance can be a major issue. In a recent work called A-SAODV(Adaptive SAODV), an adaptive mechanism that tunes the behaviour of SAODV t improve its performance. In this paper we have proposed an extension to Adaptive-SAODV of the secure AODV protocol extension, which includes further filtering strategies aimed at improving its performance. Moreover, we analyze how our proposed algorithm can help to further improve the performance of adaptive SAODV.

Keywords: AODV, SAODV, A-SAODV, TTL threshold

1. INTRODUCTION

Wireless ad hoc network [1] is gaining its popularity day by day because the devices communicate with each other using a wireless physical medium without relying on pre-existing wired infrastructure. Moreover, each node in an ad hoc network are self-configurable in nature and takes help of “multi-hop routing” technique to communicate with those nodes which are beyond communication range. But, these features give additional vulnerabilities along with those existing in the traditional wired network.

Since the advent of Defense Advanced Research Project Agency (DARPA) packet radio network in the early 1970s, a number of protocols have been developed for ad hoc mobile networks. The existing protocols can be broadly categorized into 2 types; Table-driven (proactive) and Demand-driven (reactive). Some examples of table-driven protocols are DSDV (Destination-Sequenced Distance-Vector Routing), CGSR (Cluster-head Gateway Switch Routing), WRP (Wireless routing protocol).Two most popular demand-driven routing protocols of this type are DSR (Dynamic Source Routing) and AODV (Ad Hoc On-demand Distance Vector) protocols. None of these protocols has any security mechanism for protecting an attacker to include himself in the routing operation. However, many proposals can be found to add security features to the existing protocol which are aimed either guaranteeing authenticity and integrity or monitoring the behaviour of other nodes. Still most of them fail to find a proper trade-off between security and performance with respect to limited resources of a participating node.

In this article first we briefly discuss the AODV protocol in section-2. Section-3 will explain various attacks on AODV. Then section 4 and 5 will describes the well know security extension of AODV, Secured AODV protocol and the adaptive mechanism for tuning SAODV respectively. Then, we give our proposed variation of adaptive-SAODV algorithm along with its analysis.

2. AD HOC ON-DEMAND DISTANCE VECTOR ROUTING (AODV)

AODV [2, 3] is the most popular reactive routing protocol in MANET. The reactive implies that a node exchange routing information only when it need to transfer some data and keep the routing information updated as long as the communication with the node exists. When a source node need to send some data to another node and it doesn’t have or have invalid path to the same, then it starts a route discovery process in order to establish a route towards destination node by sending route request message (RREQ) to all its neighbours. Neighbouring nodes receive the request, increment the hop count and forward the message to their neighbours. This broadcasting of RREQ message is known as flooding. The objective of RREQ message is not only to find a path to destination but also making other nodes learn about a route toward source node (reverse route). When an intermediate node receives a RREQ message from a node A for S, then it has a reverse route to node S through A with path length equals to hop count field of RREQ. Finally, when RREQ message reaches destination node, it response by initiating a route reply message (RREP). The RREP is sent as a unicast, using the path towards the source node established by the RREQ. Similarly, the RREP message allows intermediate nodwes to learn a route towards the destination node. Hence, the end of the route discovery process, packets can be delivered from the source to the destination nod eand vice versa. A third kind of routing message, called route error (RERR), allows nodes to notify breakage of link between any two nodes or information about those nodes which are unreachable at present.

In AODV it is not necessary that always a RREQ should reach the destination node. Any intermediate node already has a valid route towards destination, can generate a RREP message and does not forward the RREQ any further. This enables quicker replies and limits the flooding of RREQS. AODV uses a sequence number to identify the freshness of routing information. Each node maintains its own sequence number and increments it before sending any new RREQ or RREP message. These sequence numbers are included in the routing messages and also stored in routing tables. AODV always give preferences to fresh or new information, thus node updates its routing table if they receive a message with a sequence number higher than the last recorded one for the destination. Reader can go through AODV links for more detailed information.

3. SECURITY ATTACKS ON AODV

Since AODV has no security mechanisms, malicious nodes can perform many attacks just by not behaving according to the AODV rules. Thus, to ensure the overall security of the network, it is important to develop security mechanisms that can survive malicious attacks from “insiders” who have full control of some nodes. In order to protect against insider attacks, it is necessary to understand how an insider can attack a wireless ad-hoc network. Several attacks have been discussed in different literatures. However, the two papers [4] has adopted a systematic way to study the insider attacks against mobile ad-hoc routing protocols. Based on the composition of operations for performing attack as mentioned in above article, misuses of AODV have been classified into two categories: atomic misuses and compound misuses. Intuitively, atomic misuses are performed by manipulating a single routing message, which cannot be further divided. In contrast, compound misuses are composed of multiple atomic misuses, and possibly normal uses of the routing protocol.

First, it is necessary to identify a number of misuse goals that an inside attacker may want to achieve and are listed as follows.

Route Disruption (RD):- Route Disruption means either breaking down an existing route or preventing a new route from being established.

Route Invasion (RI):- Route invasion means that an inside attacker adds itself into a route between two endpoints of a communication channel.

Node Isolation (NI):- Node isolation refers to preventing a given node from communicating with any other node in the network. It differs from Route Disruption in that Route Disruption is targeting at a route with two given endpoints, while node isolation is aiming at all possible routes.

Resource Consumption (RC):- Resource consumption refers to consuming the communication bandwidth in the network or storage space at individual nodes. For example, an inside attacker may consume the network bandwidth by either forming a loop in the network.

As an example, route disruption, route invasion and node isolation has been shown diagrammatically using figure 1, 2 and 3 respectively.

Analysis of atomic misuses can be done in an effective way through understanding the effects of possible atomic misuse actions. Each atomic misuse action is an indivisible manipulation of one routing message. Specifically, the atomic misuse actions in AODV have been divided into the following four categories:

Drop (DR): Here, the attacker simply drops the received routing message.

Modify and Forward (MF): After receiving a routing message, the attacker modifies one or several fields in the message and then forwards the message to its neighbor(s) (via unicast or broadcast).

Forge Reply (FR): The attacker sends a faked message in response to the received routing message. Forge Reply is mainly related to the misuse of RREP messages, which are in response of RREQ messages.

Active Forge (AF): The attacker sends a faked routing message without receiving any related message.

The more interesting and complex one is that an attacker can combine several atomic misuses in a planned way and launch them. We have not discussed these attacks in detail here; interested reader can refer the literature [4].

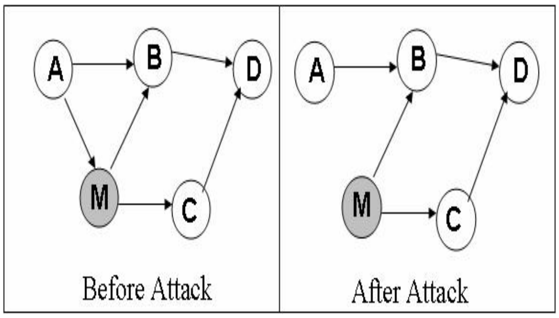

Figure 1: Node M performing Route Disruption for path A-C.

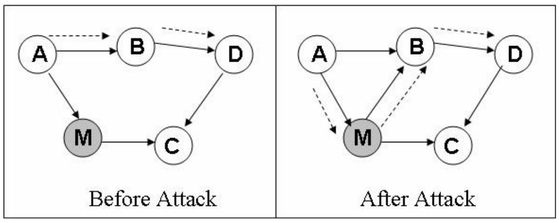

Figure 2: Route invasion

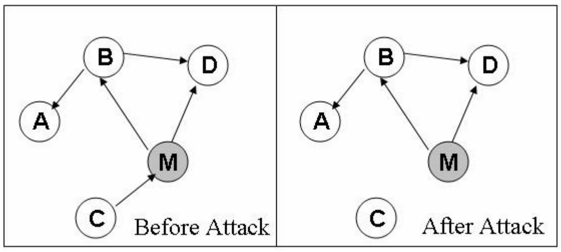

Figure 3: Node isolation

4. SECURED AD HOC ON DEMAND VECTOR ROUTING (SAODV)

One of the most popular existing security mechanisms for AODV is secured AODV [5, 6], which has been proposed by Zapata and Asokan in 2002. Secured AODV (SAODV) extends the AODV message format to include security parameter for security the routing messages. Considering RREQ and RREP message in SAODV protocol there are two alternatives for ensuring secured route discovery; first, only destination is allowed to reply a RREP and the second, any intermediate node which has valid routing information allowed to reply a RREP. Two mechanisms are used to secure the message. Digital Signature is used to authenticate non-mutable fields and Hash chain to secure mutable field like hop count information. For non-mutable field the authentication is done in an end-to-end manner. The hash chain mechanism helps any intermediate node to verify that the hop count has not been decreased by any malicious node. A hash chain is formed by applying a one-way hash function repeatedly to a seed (random number). Since, SAODV uses two way for performing verifying authentication of message, signing and verifying mechanism by sender and receiver also differs up to some extent

In the first one, where only destination is allowed to reply, every time a RREQ is sent, the sender signs the message with its private key.

An intermediate node verifies the signature before creating or updating the reverse path to the source and stores it only if verification is successful. For RREP message the final destination node sign the message using its private key. Intermediate and final node again verifies the signature before creating a route to that host.

In the second one the signing and verifying process is almost similar to the first one. But the difference is that the RREQ message also has a second signature that is always stored with the reverse path route. The second signature is needed to be added in the gratuitous reply (see AODV message format) of that RREQ and in regular RREPs to future RREQs that node might reply as an intermediate node. An intermediate node that wants to reply a RREP needs not only the correct route, but also the signature corresponding to that route to add in the RREP and the lifetime and the originator IP address fields that work with that signature. All the nodes that receive the RREP and that update the route; store the signature, the lifetime and originator IP address with that route. SAODV does not take help of any extra message for security operations. However, SAODV messages are significantly larger and require heavy computation time because of digital signatures especially for double signature mechanism. The route discovery mechanism of SAODV has been concisely discussed in algorithm 1.

Algorithm 1: SAODV Route Discovery algorithm

```
1) Sender Generates RREQ packet;
2) Sender signs all non-mutable fields (except hop count and hash chain fields) with its private key;
Apply Hash to a seed to generate hash chain field;
if (intermediate node can reply){
        Clear destination only tag;
        Include second signature in the signature extension;
}
Append signature extension to RREQ packet;
3) Broadcast RREQ to all neighbour nodes;
4) Intermediate node receives RREQ packet;
5) Node Verifies signature with public key of source (from RREQ packet);
        If (valid packet)
          then update routing information of
               source in any (establishment of
               reverse path);
6) if (destination I.P == node I.P){
        Generate RREP;
        Sign all the signs all non-mutable fields (except hop count and hash chain fields) with its private key;
        Apply Hash to a seed to generate hash chain field;
        Append signature extension to RREP packet;
        Unicast RREP to the neighbor which is in the reverse path for the source node;
}
else if ( Node has valid route for destination
                 && !(Destination only tag)){
        Generate RREP;
        Copy the signature and other necessary field of source to the signature extension;
        Sign all the signs all non-mutable fields (except hop count and hash chain fields) with its private key;
```

```
            Apply Hash to a seed to generate hash
            chain field;
            Append signature extension to RREP
            packet;
            Unicast RREP to the neighbor which is
            in the reverse path for the source node;
}
else
            Forward RREQ to all its neighbouring
            node;
```

5. ADAPTIVE-SAODV

In a recent work, Cerri and Ghioni [7] proposed an adaptive mechanism that tunes its behaviour for optimizing the performance of routing operation. They developed a prototype called Adaptive SAODV (A-SAODV) which is a multithreaded application. Cryptographic operations are performed by a dedicated thread to avoid blocking the processing of other message and other thread to all other functions. As they have suggested, each node has queue of routing message to be signed or verify and the length of the queue implies the load state of the routing thread. Whenever a node processes a route request and has enough information to generate a RREP on behalf of destination, it first checks its routing message queue length. If the length of the queue is below a threshold then it reply otherwise, it forwards the RREQ without replying. Algorithm 2 shows all above operations performed by an intermediated node systematically.

Algorithm 2: A-SAODV algorithm

```
1) Intermediate node receives RREQ packet;
2) if ( Node has valid route for destination
                  && !(Destination only tag)){
            L=length(routing packet queue to be
            signed or verified);
            if( L >= queue_threshold )
                simply forward the packet to its
                neighbouring nodes;
            else
                reply to source node using the
                procedure involved in SAODV
                protocol;
```

The threshold value can be changed during execution. The prototype also maintains a cache of latest signed and verified message in order to avoid signing and verifying the same message twice. This adaptive reply decision has a significant improvement on the performance of SAODV.

6. PROPOSED WORK

We too have proposed a bit of modification to Adaptive SAODV so that the overloading of a node with heavy cryptographic computations like signing signing and verifying routing packet can be relaxed up to a possible extent. For this each node has to maintain a queue length field for all neighbouring node along with the list of neibourhood nodes which they may update and exchange with the help of hello message periodically. This shows that our modification does not need any additional packet to be transmitted over the network. As per our method, when an intermediate node receives a RREQ and finds that it has the valid route to the destination, it check its time to leave field(TTL), if its below some predefined threshold then simply forward it. If the above condition is not true then it looks for its routing packet queue size; if it is higher that the predefined threshold then the node finds the next hop node on the path to destination. If it finds that the next hop neighbour node's routing packet queue has length less than the threshold value then it simply forward RREQ only to this neighbouring node, otherwise, it reply to the source using method involved in SAODV. We assume that this mechanism will have positive impact on the performance parameters like Packet delivery ratio, routing overhead etc. The algorithm for the given mechanism is shown in algorithm 3.

Algorithm 3: Extension to A-SAODV

```
/*Each node exchange their routing packet queue
size (route load) periodically with the help of
Hello message.*/
1) Intermediate node receives RREQ packet;
2) if ( Node has valid route for destination
                  && !(Destination only tag)){
      node_L = length(routing packet queue to be
      signed or verified);
      if(RREQ.TTL <=TTL_threshold)
          forward the packet to all neighbours;
      else if( node_L >= queue_threshold ){
          nbd_to_dest = the neighbour node which
          is equal to the next hop in the route entry
          to the destination;
          nbd_L= length(routing packet queue of
          the nbd_to_dest);
          if ( nbd_L < queue_threshold )
                simply forward the packet to
                nbd_to_dest;
          else
                forward the packet to all the
                neighbouring nodes;
      }
      else
          reply to source node using the procedure
          involved in SAODV protocol;
```

7. ANALYSIS OF PROPOSED WORK

As we know that the time to leave (TTL) field is the number of hops to be traveled by the packet before being discarded by a router. A small value of TTL say t, implies that either the packet going to reach its destination within t hops or going to be discarded after t hops. So, choosing a sufficiently small TTL value as TTL_threshold field, any intermediate node is allow to reply a route request only if TTL field of the RREQ packet is larger than the TTL_threshold value. Otherwise, the request packet is simply forwarded to all neibouring nodes assuming that either destination is within TTL_threshold hop neibourhood of it or packet is to be dropped after TTL hops. This may significantly reduce the queue length of any intermediate node in the path to destination.

Secondly, in A-SAODV an intermediate node having a route to destination simply forward a route request for same without sending reply if it founds that its current routing message queue length is more than threshold queue length. If an intermediate node has a valid path to destination then among all the copy of forwarded packets to all neighhouring nodes, the packet which has been forwarded to the next hop node of route entry for destination will follow the optimal path to destination. Our proposed modification is an additional checking to see that the whether next hop to the destination's load factor is less than the threshold level. If yes, then the request packet is simply forwarded to next hop node instead of forwarding to all neibouring nodes. This may in turn relax the load of all neighbouring nodes which are not an active member of the optimal path to the destination. However, this is yet to be proved though simulation.

8. CONCLUSION

Securing AODV is still an open area for research work. The work is also open for a way to provide intermediate hop authenticity verification which still lacks in existing literatures. The existing mechanisms like SAODV able to secured the protocol with its signature extensions. But the overhead of cryptographic calculation still persist in the proposed mechanisms. A-SAODV is one of the steps towards optimizing the routing performance of secured protocols with help of a threshold mechanism. Hence, the strength of a secured protocol for AODV not only depends on the strength of the cryptographic mechanism but also on the routing performance metrics.

.

9. REFERENCES

1. *D. Remondo , "Tutorial of Wireless Ad Hoc Networks", HET-NETs 2004.*
2. *C. E. Perkins and E. M. Royer, "Ad Hoc On-Demand Distance Vector Routing", Proc. 2nd IEEE Wksp. Mobile comp. Sys. And Apps. Feb, 1999, page 90-100*
3. *C.E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad Hoc On-Demand Distance Vector Routing", IETF RFC 3561, July 2003*
4. *P.Ning, K. Sun, "How to misuse AODV: A Case Study of Insider Attacks Against Mobile Adhoc Routing Protocols", Info Assurance Wksp, IEEE sys, Man and Cybernetics Soc, june 2003, page 60-67*
5. *M. Gurrero Zapata and N. Asokan, "Securing Adhoc Routing Protocols", Proceeding 1st ACM Workshop. Wireless Sec., 2002, page 1-10.*
6. *M. Gurrero Zapata, "Secure Ad hoc On-Demand Distance Vector (SAODV) Routing", Mobile Ad Hoc Networking Working Group INTERNET DRAFT,5th September 2006*
7. *Davide Cerri, Alessandro Ghioni, "Securing AODV: The A-SAODV Secure Routing Prototype", IEEE Communication Magazine, Feb. 2008, page 120-125*