**SOAP AODV**

**(Secure Optimized Adaptive AODV with Priority Based Packet Forwarding)**

Sonika Malik

Information Technology

MSIT

Delhi, India

Sahil Kapoor

Information Technology

MSIT

Delhi, India

Chirag Chopra

Information Technology

MSIT

Delhi, India

Vivek Mishra

Information Technology

MSIT

Delhi, India

vivekmishra1211@gmail.com

Rashmi Chaudhary

Information Technology

MSIT

Delhi, India

Rashmii0525@gmail.com

***ABSTRACT* - Mobile Adhoc Network (MANET) is a multi-hop wireless network of mobile terminals, forming a network without the help of any established infrastructure or centralized servers. Since, any dedicated routers are absent, every node requires adding towards the configuration and maintenance of the routing framework. Open peer-to-peer architecture, dynamic topology, shared wireless medium are major the challenges in the security design of MANETs. Adhoc On-demand Distance Vector (AODV) is one of the widely used protocols for routing. This paper aims to extend my previous paper “Avant-garde security in AODV” with optimization, adaptability and priority mechanism for forwarding packets. Security is implemented using data encryption methods like DES and RSA with hash chain. Optimization is done by selective forwarding of request packets rather than broadcasting based on different metrics. A priority mechanism is also introduced which forwards important packets from buffer before others. The protocol adapts itself to select security modules based on urgency level and security level of data packets being forwarded. This paper is backed up with results acquired through AODV simulator in C++.**

***Keywords***

***Wireless Networks, Adaptive, AODV, MANET, Optimization, Security***

1. **INRODUCTION**

Mobile Ad-Hoc network focuses on wireless communication without any fixed infrastructure. All Ad-Hoc routing protocol have different routing strategies so factors such as End to End Delay, Traffic Overhead and packet delivery ratio and power consumption. Routing generally deals with the route discovery between source and destination. Nodes in

network change the position as per requirement of system so topology varies time to time.

The routing protocols are mainly divided in to Proactive and Reactive Protocol. Proactive routing protocols are table-driven. The reactive routing protocols (e.g. AODV) create and keep routes only if these are needed, on demand and typically use distance-vector routing algorithms that have info about next hops to adjacent neighbors and costs for paths to all destinations. So, link-state routing algorithms are more dependable, less bandwidth demanding, but also more complex and CPU hungry. AODV is a reactive routing protocol. It uses an on demand algorithm; it builds a route between nodes only when it is requisite by source node. To ensure that freshness of routes, sequence numbers are used. [2]

When a source node requires a route to a destination, it broadcasts a Route Request (RREQ) packet through the network. RREQ packet has hop count, source and destination IP address and Destination sequence number (DSN). Nodes receiving this packet update their info for the source node and create backwards pointers to the source node in the routing tables. If any node has a sequence number more than or equal to that contained in the RREQ it will send a route reply packet. A node receiving the RREQ can send a Route Reply (RREP) if it is either the destination, or if it has a route to the destination with a matching sequence number more than or equal to that contained in the RREQ [6]. In this case, it unicasts a RREP to the source. Else, it rebroadcasts the RREQ if there was route link failure. When the source node receives the RREP, it starts forwarding data packets to the destination. If the source gets an RREP with a greater sequence number with the less hop count, it considers this route as the best and begins forwarding data packets through it. Active route is one, which has data transmission from the source to the destination, as long as there are data packets travelling from the source to the destination along that route.

If the source breaks transmission of the data packets, the link will timeout and deleted from the source routing table. If a route link error occurs while the route is active, the node that discovered the link failure will propagate a Route Error (RERR) message to the source node. If the sources node receives the RERR, still demands the route, sources will initiate a route discovery again.

Major problem with AODV protocol arrives that malicious nodes may theft or tamper with the data forwarded through them. For this, packets need to be encrypted to ensure security. Further, broadcasting packets to all nearby nodes leads to wastage of bandwidth and space. Our protocol tackles both problems along with introducing new priority-based packet forwarding algorithm.

1. **PROTOCOL DESCRIPTION**

Our protocol basically implements four separate modules: Security of data packets, Optimization of network, Adaptive selection of security modules and Priority based mechanism for packet forwarding- together forming complete package of SOAP-AODV protocol. The original AODV packets are extended for two fields – urgency level and privacy level (which may be high, medium or low). Based on these two levels, optional security modules are selected to ensure required security needs are satisfied without too much overhead. All four modules are explained one by one.

1. *Security of data packets*

Adhoc networks don’t depend on superfluous hardware, which makes them an ideal candidate for military services and operations. For example, war zone ad hoc network, we would surely be first afraid with the efficient and in time delivery of the message but with this, we will have to be more concerned about the strong privacy or secrecy of the information also. AODV doesn’t provide any security for data it is sending. This is major drawback in AODV, which has been addressed and countered in various researches. Our paper also uses two techniques, one implemented for MANETs where symmetric keys can be provided and other where asymmetric keys are used.

1. *i-AODV*

We concern the security problem introduced by the instability of physical layer or link layer. There are two main motivations, which encourage nodes to misbehave: selfishness and malice. When dealing with packet forwarding, there are several kinds of availability and integrity attacks: dropping (complete or partial), misrouting, modification and fabrication. Malicious cooperation (such as a wormhole attack) and identity changes are also challenges attacks.

The nodes in network are initialized with empty routing tables. Every node has a Unique ID (which may be its IP Address). The network distributes UNIQUE KEY pairs for encryption and decryption of messages to all nodes. For N nodes, total key pairs are n\*(n-1). No two nodes know the secret key pair of any other node pair. Every node has an encoding key for each node in network, which it uses to encode the data packet, and decoding key for each node, which is used to decode the data. NCI index (Non Credibility Index) is stored in trust table. It is updated whenever a packet is received. If Packet is corrupted it increases the NCI value of nodes in route path, depicting probable malicious nodes. If the packet is received successfully, then the good behavior of nodes is path is rewarded with decrease in the NCI index. NCI index is only increased at the

The data packet is encrypted using symmetric key encrypting algorithm using the secret pair of keys between source and destination node. Then the message goes under RSA encryption and signature is generated. The public key and digital signature are sent along with the packet (can be secured via hash chain). Destination node maintains a special table; let us call it trust-table, of all nodes in network. This trust-table is supposed to store non-credibility-index (NCI) along with IP address of possible malicious nodes.

Whenever destination receives a tampered packet (which is detected by the RSA algorithm using public signature and key provided as shown in SAODV in previous chapter), it increases NCI Value of all nodes in route path, depicting one or more than one nodes in the path is malicious. If the packet is delivered with successful signature verification, the NCI Value of all nodes in route path is decreased. This ensures that no ‘good’ node is getting killed in network because it appears too frequently in bad paths. It acts like Artificial Intelligence to filter good and bad nodes through the trust table it forms during course of time. There is a threshold value set for NCI value for nodes, after which the nodes will be avoided in setting up path. Reducing values to lower levels to avoid rejection of too many paths normalizes the NCI index, if higher than threshold value for more than half nodes in the network. This ensures that network is always up and running.

1. *f-AODV*

The main problem in distributing secret keys is that they must be connected to each other or some common server at least once. This is not feasible in cases where few nodes may or may not be present be at the time of feeding of keys to nodes. Further if database of keys were leaked, the entire security structure entire would collapse. Or if the encryption/decryption algorithm were cracked, it would be easy to crack the message using brute force attack. To avoid these problems, we can make keys to be delivered dynamically from Source to Destination through a ‘super secure’ channel before the data packet is sent. fAODV is solution for such kind of needs when we have information about our ‘friend’ nodes, which may or may not be present at beginning of set up of MANET. When someone node wants to establish a connection, the path set up initially is through only these friend-nodes and the key for encryption/decryption of data packet sent are delivered through this ‘super secure’ channel only.

The fAODV can use a bit of time in setting up the super secure channel initially, which is directly proportional to number of friend nodes in network for obvious reasons. This type of network can fail if the source gets isolated from other trusted nodes and is never able to set up the ‘super secure’ channel

The private key and transpose matrix are only sent until acknowledgement from destination is received. Until then, RREQ is forwarded only through ‘super secure’ path. After that the whole system switches to normal AODV and sends packet to all in range legitimate nodes.

1. *Optimization of network*

AODV broadcasts it RREQ packet to all in-range nodes. These in-range nodes then broadcast the request to nodes in their affinity. There is a high probability that nodes near the current sender include nodes, which have already received RREQ packet through source (or previous sender) node. The broadcasting happening at each node causes lot of unnecessary network congestion since same packet is forwarded and received again and again. The node rejects similar packets using their sequence number. The flooded broadcasting also wastes buffer space at each node and processing time taken to analyze and compare packets. To tackle this problem, instead of broadcasting to all in-range nodes, we selectively chose few nodes to forward packets based on metrics like their distance between sender and receiver nodes, bandwidth available and battery level. These values are updated periodically in each node for nodes in its affinity and can be retrieved by RREP packets or by sending special request packet for them. These values are stored in routing table. Each node in affinity of sender is scored using following formulae:

Si = (x \* batteryLeft) + (y \* bandwidthAvailable) + (z \* distanceFromNode) .

where, Si is score of ith node; x,y and z are the weight provided by network to each metric which may vary according to user needs.

The metrics can be weighed differently in different situations. For wireless sensor network, battery may have more weightage than bandwidth or for a crowded mesh network, distance can have more priority.

Since scores are scored in routing table, whenever a packet is to be sent, the sender checks scores against IP address of in-range nodes and selects top n nodes(n being maximum number of node to which packet is multicast) to forward the packets to them. This selective multicast has great advantage over conventional way. The two selected nodes have great probability to be out of range, (since decision included distance metric), which reduces their chances of interacting and thus, reducing network and processing overhead.

1. *Adaptive selection of security modules*

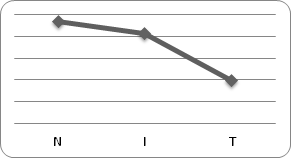
We know that implementing new security modules add to processing overhead and hence, some delay in sending of packets. In other words, we gain security at the cost of processing time. It may be noted that, security requirement of each data packet may vary, some may be important transaction requiring encryption and other may be just an acknowledgement, we use multiple security modules which are selected based on privacy and urgency parameters. The three modules- no security, i-AODV and f-AODV- are selected as shown.  
The adaptive nature allows significant gain over delay caused due to processing time wasted on encryption and decryption of data. Each packet can be individually catered for its security needs, which allows great control over network overhead. When no security is required, we may use original AODV. With low urgency and high security, we use the slowest fAODV. In other cases, iAODV is most optimum choice. SOAP AODV package is provided with an algorithm to dynamically select either of three modules among the nine possible combination of security and priority.

Table I. Adaptive Selection Parameters

|  |  |  |
| --- | --- | --- |
| *Urgency* | *Security* | *Module Selected* |
| High | Low | No security |
| High | Medium | i-AODV |
| High | High | i-AODV |
| Medium | Low | No security |
| Medium | Medium | i-AODV |
| Medium | High | f-AODV |
| Low | Low | i-AODV |
| Low | Medium | f-AODV |
| Low | High | f-AODV |

1. *Priority Mechanism*

When packets arrive at the node, they are stored in buffer and are broadcasted one by one in queue to all nodes in its affinity. Suppose a network has two kinds of packets, high number of acknowledgements and some urgent data packets. The delivery of acknowledgements doesn’t matter much but data packets should be delivered as soon as possible. If a node has lot of acknowledgements waiting in buffer and data packet arrives, it will have to wait till all previous acknowledgements are served. In such kind of networks, we can use the urgency field of SOAP AODV for informing nodes which packet requires urgent attention. The packets with high urgency level are put on top of queue and are served before others. To avoid common of problem starvation or waiting forever, we increase urgency level for waiting packets after certain time interval, so that they are served accordingly.



1. **RESULTS**

The proposed AODV system was implemented using simulator created in C++ by modifying original AODV code. The structure of AODV RREQ, RREP and Data packets were altered for including new fields. New columns were added to the routing table to store the battery and bandwidth status of nodes. The effect of each security module was measured on basis of RREQ requests, RREP replies and packet delivery ratio (packets sent/packet ratio) and Optimization was measured for change RREQ and RREP packets.

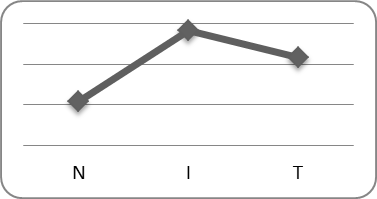
*Fig. 2. Throughput of network*

#### *Network Congestion*

i. AODV (Low): Since there is no security or path correction, the RREQ packets are less.

ii. iAODV (High): Due to path rejection, amount of RREQ packets increases as compared to fAODV or AODV, causing network congestion. Intelligent normalises the trust table to make sure not many nodes are facing rejections.

iii. fAODV (Low): The RREQ packets sent in fAODV are less than in beginning since they only travel through trusted nodes. Since there is no path rejection in fAODV, network congestion is very less than fAODV and iAODV.



*Fig. 1. Network Congestion*

1. Throughput (*Packet received per Unit time*)

i. AODV (High): The throughput of entire network is highest.

ii. iAODV (Higher): Very high throughput, if we ignore the loss of packets.

iii. fAODV (Average): Lower throughput because of time wasted in finding super secure path for delivering the key.

1. *Delivery Ratio (Packets Successfully Received/ Packet Sent)*

i. AODV (High): The delivery ratio is very high but no security is implemented.

ii. iAODV (Average): Number of packets delivered is almost half than AODV but all packets delivered are safe and secure.

iii. fAODV (Average): Packet delivery ratio is slightly poorer than iAODV, but it requires no key distribution mechanism. Further considering the fact that time is wasted in setting up first time super secure channel, performance will be better for fAODV for large data.

*Fig. 3. Through Delivery Ratio*

1. *Optimization*

A significant 30% reduction in Route Requests for all the security modules with negligible effect in packet received ratio shows that multicasting packet based on metrics greatly reduces network usage.

Though network is also used for sharing battery and bandwidth details, the drop in Route Requests is significantly higher and hence, overcomes the former disadvantage easily.

*Fig. 4. Comparison of AODV and Optimized AODV*

1. **CONCLUSIONS**

iAODV was able to identify malicious paths over the time and avoid them. Rejecting malicious paths caused network overhead because new paths have to be found every time there is malfunction. fAODV had lesser requests but takes time to find destination for first time but then speeds up like normal AODV after keys are delivered. Their implementation is network environment dependent, though they can also be used simultaneously in networks using complete SOAP AODV.

The idea of selective multicast gave significant advantage over flooding. The amount of RREQ sent in SOAP AODV were almost two-thirdof original one without really compromising with amount of data packets sent.

Adaptability and priority packet forwarding does not affect performance of AODV since they use unused reserved bits of the AODV original packet, but act as additional features of SOAP AODV protocol.

1. **FUTURE WORK**

The scoring is currently done based on distance, battery and bandwidth. If nodes can determine coordinate-location of in-range nodes, we can improve our scores based on directions the node lie, and select top node in either direction. This will help sending packets in each direction and finding the destination would be faster.

1. **REFERENCES**

[1.] Avant-Garde Security in AODV, IJERMCA Nov 2013 by Sonika Malik, Sahil Kapoor, Chirag Chopra, Rashmi Chaudhary and Vivek Mishra

[2.] Securing and Enhancing Routing Protocols for Mobile Ad hoc Networks by Manel Guerrero Zapata

[3.] Security Engineering: A Guide to Building Dependable Distributed Systems- Cryptography by JOHN F. KENNEDY

[4.] Lecture Notes on “Computer and Network Security” by Avi Kak

[5.] SFWR 4C03: Computer Networks and Computer Security by Lecturer: Kartik Krishnan