AODV: Security Consideration

Serin V Simpson
PG Student, Computer Science
Thejus Engineering College
Thrissur, India

Saju P John
HOD, Computer Science Department
Thejus Engineering College
Thrissur, India

ABSTRACT
AODV (Ad hoc On Demand Vector) is a reactive routing protocol in wireless mobile ad hoc network (MANET). AODV is accepted because of its capacity to adjust quickly in dynamic system environment with least overhead and small management packet size. The adhoc networks are all that much helpless against Dos attacks on the network layer. Blackhole and Grayhole attacks are the far reaching attacks on adhoc networks. Here the malicious nodes intrude on information transmission in the system by transmitting false routing data. The AODV routing protocol was at first created without considering security as a top priority. So it is not ready to shield against any sort of security attack. However there are numerous security mechanisms available that make AODV secure.

General Terms
AODV, R- AODV, MR- AODV, RSA key exchange.

Keywords
AODV, MANET, ERDA, Digital Signature, R- AODV, MR-AODV, RSA key exchange.

1. INTRODUCTION
Mobile ad hoc network (MANET) is a dynamic network comprises of mobile devices associated together by wireless connection. More often than not, the nodes in MANET are mobile and can demand to unite or leave the network. Therefore, network topology will often change. In circumstance where mobile nodes are in the same wireless range, they can communicate specifically however, in the event that wireless is not in the same range, communication will be lost. To make communication accessible albeit wireless which is out of range, collaboration from different nodes is obliged to hand-off essential messages. In networking terms it is called multi-hop network. As there is no base or unified administration in MANET, every node needs to assume two parts, i.e., as a host and as a router. As a router in a multi-hop network, every node needs to control and deal with the routing path. So as to do that, they oblige a standard routing protocol to encourage the communication participation. Routing protocol likewise makes MANET get to be alluring to network clients whereby making any network will be quick and straightforward. Lamentably, MANET is likewise helpless against attack like some other networks. Actually it is more helpless than wired network. Among well known malicious attacks in MANET are eavesdropping, spoofing, control packet modification and denial of service. AODV is a distance vector routing protocol that has been characteristically construct for MANETs.

2. AODV
AODV makes widespread utilization of sequence numbers in control packets to keep away from the issue of era of routing loops. At the point when a source node is intrigued to communicate with a destination node whose route is obscure, it broadcasts a RREQ (Route Request) packet. Every RREQ packet contains a Request ID, source and the destination node IP addresses and sequence numbers alongside a hop count and flags. The Request ID field particularly recognizes the RREQ packet; the sequence numbers gives data with respect to the freshness of control packets and the hop-count keeps up the number of nodes between the source and the destination. Beneficiary node of the RREQ packet that has not discover the Source IP and ID pair or doesn't keep up a fresher (larger sequence number) route to the destination rebroadcasts the same packet in the wake of increasing the hop-count. Such intermediate nodes additionally make and safeguard a REVERSE ROUTE to the source node for a certain time. At the point when the RREQ packet landed at the destination node or any intermediate node that has a fresher route to the destination a RREP (Route Reply) packet is created and sent over to the source. RREP packet contains the destination node sequence number, the source and the destination IP addresses, route lifetime alongside a hop count and flags. Intermediate node that receives the RREP packet, augments the hop count, makes a Forward Route to the source of the packet and transmits the packet on the Reverse Route. At the point when a connection disappointment is identified for a next hop of a dynamic route a RERR (Route Error) message is sent to its dynamic neighbors that were utilizing that specific route. The AODV is ordered as a dynamic reactive routing protocol. In a reactive routing protocol, route will be built focused around interest (upon request by source node). The methodology to find [3] routing path to destination node is delineated in Figure 01.

2.1 Route Discovery Process
In this delineation, source node S will broadcast control packets, RREQ message to its neighbors A, B and C keeping in mind the end goal to discover the best conceivable path to destination node D. After getting RREQ message, the received node either: [7]

a) Reply to the source node with a RREP message if received node is the destination node or an intermediate node with a 'fresh enough' route data to the destination, or

b) Update the routing table passage which will be utilized as a part of the reverse path and rebroadcasting of RREQ message until destination node or intermediate node with 'fresh enough route' is arrived.

An intermediate node is accepted to have a 'fresh enough routes' to destination node if destination sequence number in its routing table is more prominent than or equivalent (with less hop count) to destination sequence number in RREQ message.

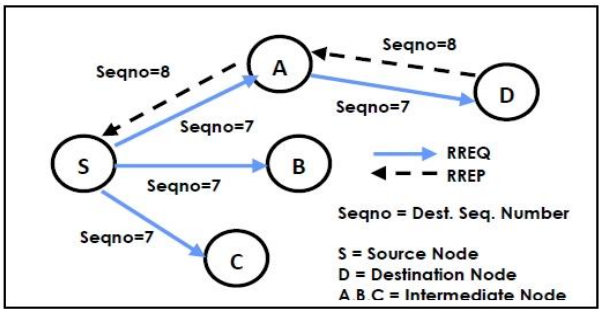

Fig 01: AODV route discovery process [3]

As specified in above area, for part a) above, after accepting RREQ message from node A, destination node D will reply with RREP message to node S by sending the message to node A. Thus, node A will forward the message to source node S. When source node S receives RREP message, it will prepare the message by calling AODV recvReply() function. This capacity will upgrade the route passage for destination D if both of this condition is met.

a) The destination sequence number in routing table is short of what destination sequence in RREP message or

b) The destination sequence number in routing table is equivalent with destination sequence number in RREQ message however with hop count is not exactly the one in routing table.

On the off chance that where node S receives different RREP messages, this capacity will choose the RREP message with the most elevated destination sequence number esteem [8].

3. SECURITY ISSUES IN AODV

3.1 Black Hole Attack

A Black hole attack is a denial of service attack where a malicious node can dishonestly asserts it has 'fresh enough route' data to the destination. The modus operandi of a Black hole attack in AODV is by attacking control message sent amid route disclosure process whereby a manufactured RREP message is sent out to catch the consideration of different nodes. Deceivingly, the malicious node will assert that it has the, fresh enough route data to the destination. In the event that alternate nodes fall into this trap, they will send their information packets through the malicious node [11].

3.2 Gray Hole Attack

Grayhole attack is said to be an extension of Blackhole attack because we cannot predict the nature of the malicious nodes. Sometimes it may behave like malicious nodes and sometimes like normal nodes [9].

4. SECURITY BASED ENHANCEMENTS ON AODV

4.1 Enhance Route Discovery for AODV (ERDA)

Jalil et al. [3] [12] proposed ERDA (Enhanced Route Discovery AODV) to improve previous methods in terms of reducing the overhead incurred during route discovery. The proposed solution will utilize least change to existing AODV algorithm. There are three new components acquainted with enhance the current AODV in recvReply() function in particular are 1) the rrep_table to store approaching RREP packet, 2) mali_list to keep the identified malicious nodes personality and 3) the rt_upd, parameter to control the routing table update. For the most part, the proposed technique is partitioned into two sections; 1) securing routing table update, 2) recognizing and segregating malicious node.

4.1.1 Securing the Routing Table Update

In ordinary AODV as depicted over, the forward path routes in the node's routing table will be updated focused around a) destination sequence number in the routing table is lower than the one in RREP's message or b) destination sequence number in the routing table and in RREP message is equivalent however number of hop in RREP packet is lower. In this proposed system, ERDA forces an additional condition by presenting a third parameter called rt_upd. This parameter can receive either genuine or false esteem. Of course, the quality is situated to genuine which implies the routing table is permitted to be updated and it is redundant from the first RREP message received by the node.

Figure 02 shows how ERDA functions amid route revelation stage and how it updates routing table. In AODV route disclosure methodology, route request (RREQ) message is sent out by a source node S to discover a fresh route to destination node D. All neighboring nodes that have received this request and have "fresh enough route data" in their routing will react to node S including the destination node D as showed in Figure 02 (a). Rreps received by node S are put away in rrep_tab table. Figure 02 (b) demonstrates the data contained in the rrep_tab table for node S. The data put away in the rrep_tab table incorporates node_id and destination sequence number. Accepting that the network is under Black hole attack, malicious node M would be the first node to react to node S, the routing table of node S is updated with the data gave by node M as delineated in Figure 02 (c).

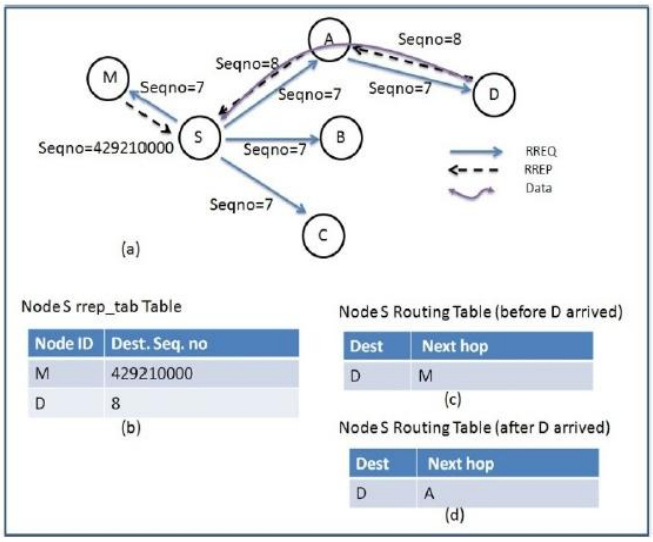

Fig 02: Routing update in ERDA [3]

On the other hand, since the estimation of the rt_upd parameter in the ERDA is situated to 'true', the routing update does not stop yet permits next RREP message to update too. Consequently, when node S receives the RREP message from node A, the message will be acknowledged in spite of the fact that the destination sequence number is littler than the one in the routing table. Thus, in Figure 02 (d) the previous route passage is overwritten by the later RREP originating from node A. When the redesigning receives the RREP message from the destination node D, the rt_upd parameter worth is then situated to false. Any RREP message that comes after this point will be denied from upgrading the routing table until the procedure of recognizing malicious node is finished [3].

• *Algorithm for Securing the Routing Table Update*

Step 01: route request (RREQ) message is sent out by a source node S to discover a fresh route to destination node D.

Step 02: all neighboring nodes that have received this request and have "fresh enough route data" in their routing will respond to node S including the destination node D.

Step 03: rreps received by node S are put away in rrep_tab table. (The data put away in the rrep_tab table incorporates node_id and destination sequence number.)

-- Assuming that the network is under Black hole attack, malicious node M would be the first node to respond to node S with a high sequence number. --

Step 04: the routing table of node S is updated with the data gave by node M.

Step 05: since the estimation of the rt_upd parameter in the ERDA is situated to 'true', the routing update does not stop yet permits next RREP message to update also.

Step 06: when node S receives the RREP message from node A, the message will be acknowledged in spite of the fact that the destination sequence number is littler than the one in the routing table.

Step 07: as a result, the previous route entrance is overwritten by the later RREP originating from node A.

Step 08: once the updating receives the RREP message from the destination node D, the rt_upd parameter worth is then situated to false.

Step 09: any RREP message that comes after this point will be denied from updating the routing table until the methodology of locating malicious node is finished.

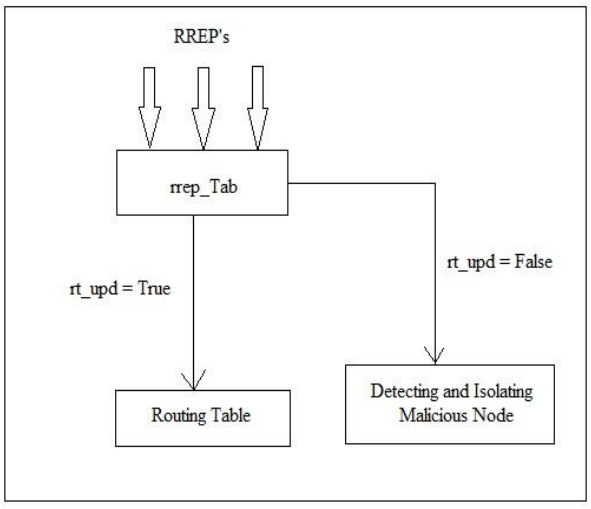

Fig 03: Usage of rt_upd parameter in ERDA.

4.1.2 Detecting and Isolating the Malicious Node

ERDA is flawlessly coordinated into AODV with no change to the current conduct of AODV protocol. ERDA has the capacity spare the RREP's data like node id and destination sequence number into the rrep_tab table without bringing about any handling overhead. In ERDA, the methodology of updating the route passage will be proceeded until the estimation of rt_upd parameter is situated to 'false'. Amid the rt_upd in a "false" express, the data in the rrep_tab table will be broke down utilizing the heuristic technique whereby the node id which has particularly high destination sequence number will be separated as a malicious node and the character of those suspected nodes will be kept in mali_list list. The reason for mali_list is to illuminate the node to separate those recorded nodes from partaking in the route revelation updates. Subsequently, any control messages (e.g. RREP or RREQ) that originate from those recorded nodes will be tossed by the node. Keeping in mind the end goal to guarantee that this methodology does not expend memory, the rrep_tab table will be flushed once the procedure of recognizing malicious node is finished and the rt_upd parameter esteem again is situated over to "true" [3].

• *Algorithm for Detecting and Isolating the Malicious Node*

Step 01: the procedure of updating the route section will be proceeded until the estimation of rt_upd parameter is situated to 'false'.

Step 02: during the rt_upd in a "false" express, the data in the rrep_tab table will be examined utilizing the heuristic system.

Step 03: node id which has astoundingly high destination sequence number will be disengaged as a malicious node.

Step 04: the character of those suspected nodes will be kept in mali_list.

Step 05: in request to guarantee that this procedure does not devour memory, the rrep_tab table will be flushed once the methodology of recognizing malicious node is finished.

Step 06: the rt_upd parameter esteem again is situated over to 'true'.

• *Disadvantage:*

On the off chance that attacker produces a RREP with a little destination sequence number, (which is practically equivalent to the destination sequence number in the RREP produced by the real destination) then the comparison cannot be executed.

4.2 Using Digital Signature and Hash Chain

This method is proposed by Soni and Nayak [2]. They incorporated two mechanisms to secure the AODV messages: digital signatures to confirm the non-modifiable fields of the messages, and hash chains to secure the hop count data (the main modifiable data in the messages). For the non-modifiable data, confirmation is performed in an end-to-end way, yet the same sort of strategies can't be connected to the modifiable data. The data with respect to the digital signature and hash chains is transmitted with the fundamental AODV message as an expansion message. Proposed security system uses hash chains to verify the hop count of RREQ/RREP messages in such a route, to the point that permits each node that receives the message (either an intermediate node or the last destination) to check that the hop count has not been decremented by an attacker. Digital signatures are utilized to secure the respectability of the non-modifiable information in RREQ/RREP messages. That implies that they sign everything except for the hop count of the AODV message and the hash from the adjusted AODV augmentation. At the point when a RREQ is received by the destination itself, it will reply with a RREP just in the event that it satisfies the AODV's prerequisites to do so. This RREP will be sent with a RREP Signature Extension. At the point when a node receives a RREP, it first confirms the signature before making or updating a route to that host. Just if the signature is checked, it will store the route with the signature of the RREP and the lifetime. Given us a chance to expect that there is a key

management sub-system that makes it feasible for every ad hoc node to acquire open keys from alternate nodes of the network. Further, every ad hoc node is equipped for safely confirming the relationship between the character of a given ad hoc node and people in general key of that node. How this is attained to relies on upon the key management scheme [2].

• Hash Chain Mechanism to secure modifiable field (hop count) of AODV message:

A hash chain is structured by applying a restricted hash function more than once to a seed. Each time a node begins a RREQ or a RREP message, it performs the accompanying operations:

• Generates a random number (seed).

• Sets the Hopcount_Limit field to the TimeToLive (TTL) value. Hopcount_Limit = TTL

• Sets the Hash field to the seed value. H_field = seed

• Sets the Hash Function field to the identifier of the hash function that it is going to use. H_function = h

• Calculates Top Hash by hashing seed Hopcount_Limit times. T_Hash = h(seed) _Hopcount_Limit times (Where, h is a hash function and hi(x) is the result of applying the function h to x i times.)

In addition, each time a node receives a RREQ or a RREP message, it performs the accompanying operations so as to confirm the hop count:

• Applies the H_function, Hopcount_Limit minus Hop Count times to the value in the H_field, and verifies that the resultant value is equal to the value contained in the T_Hash field.

T_Hash = = h (H_field) _Hopcount_Limit times (Where, a = = b reads: to verify that a and b are equal.)

• Before rebroadcasting a RREQ or forwarding a RREP, a node applies the hash function to the Hash value in the Signature Extension to account for the new hop. H_field = h(H_field)

The H_function field indicates which hash function has to beused to compute the hash. Trying to use a different hash function will just create a wrong hash without giving any advantage to a malicious node. H_function, Hopcount_Limit, T_Hash, and H_field fields are transmitted along with the AODV message. .

• *Disadvantage:*

Here the mechanism provides security only to the data which we are sending. An attacker can attack a node directly by manipulating its sequence number. Thus this mechanism fails to prevent the direct attack on nodes.

4.3 Reliable AODV (R- AODV)

As portrayed in [5] and [6], R-AODV enhances route revelation procedure of AODV by acquiring security into AODV protocol and keeps Blackhole and Grayhole nodes from participating in information transmission stage. It uses number of sent out RREQs, number of received RREPs and routing table sequence number to dynamically figure a PEAK value after every received RREPs; the PEAK value is figured by adding these three parameters to the past PEAK value. Destination sequence number of received RREP is contrasted with this PEAK value with discover presence of a malicious node. R-AODV uses default routing packets viz. RREQ and RREP to advise different nodes in the network about presence of malicious nodes as opposed to utilizing additional control packets that endeavors to diminish climb in routing overhead. It adjusts the functionalities of node sending RREQ, node getting RREQ and node accepting RREP. Source node sending RREQ engenders data about adversaries to different nodes in the network; nodes accepting RREQ disengage the malicious nodes; nodes getting RREP identify the presence of malicious nodes. Hence, R-AODV recognizes and detaches various malicious nodes amid route determination stage which helps setting up a short and secure route towards destination [4].

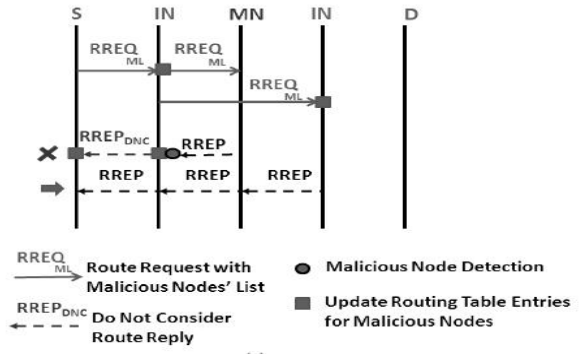

Fig 04: Working of R-AODV [4]

Fig. 04 represents the route disclosure methods of R-AODV in vicinity of a malicious node. As demonstrated in figure when a malicious node is identified by an intermediate node in the wake of getting RREP, R-AODV marks the RREP as DO_NOT_CONSIDER and imprints the node sending RREP as MALICIOUS_NODE in the routing table; the RREP is then sent on the reverse path to the source which updates routing tables of every last one of nodes on the reverse path with malicious node section; a route towards destination is picked by selecting unmarked RREPs.

• *Algorithm Steps:*

- At whatever point an intermediate node receives a RREP having sequence number more prominent than the figured Peak value, it is stamped as Do_not_consider.
- The node sending that RREP is checked as a malicious node in the routing table.
- RREP is then sending to the source node by means of reverse path.
- Each node getting this RREP updates route passage for the malicious node.
- The source node rebroadcasts RREQ alongside a rundown of malicious nodes to illuminate alternate nodes about the vicinity of malicious nodes in the network.
- Thus, malicious nodes can be isolated

• *Disadvantage:*

In the event that attacker produces destination sequence number which is short of what or equivalent to PEAK value, the node can't be discovered as a malicious node.

4.4 Modified Reliable AODV (MR- AODV)

MR AODV is a modified version of R-AODV. This work is done by Jhaveri [4]. In MR-AODV, when a node detects a malicious node, it updates the routing table with malicious node entry and discards the RREP as shown in Fig. 05.

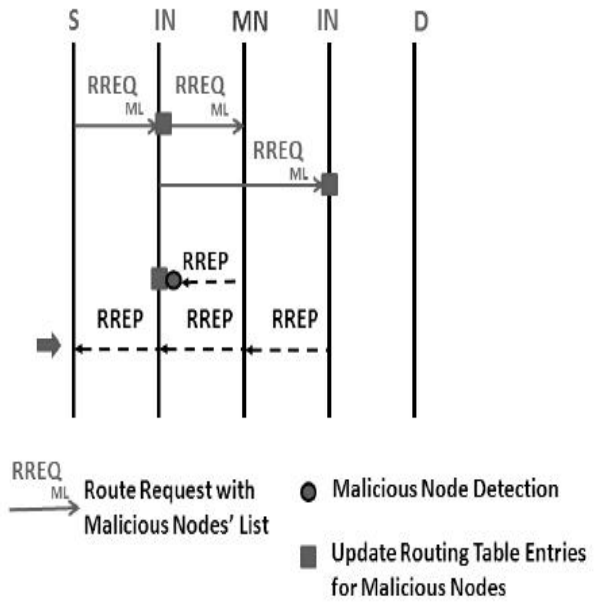

Fig 05: Working of MR-AODV [4]

It is not one or the other sent on the reverse path nor obliges a DO_NOT_CONSIDER flag; subsequently, all RREPs arriving at to the source node will be sent by genuine nodes just; the RREP demonstrating most limited fresher path will be picked for information transmission by the source node. Consequently, MR-AODV endeavors to decrease routing overhead by not sending RREP after discovery of misbehavior [4].

• *Design of Algorithm*

The paper presents the functionalities of node sending RREQ, node receiving RREQ and node receiving RREP in form of flow-charts as follows:

4.4.1 Node broadcasting RREQ

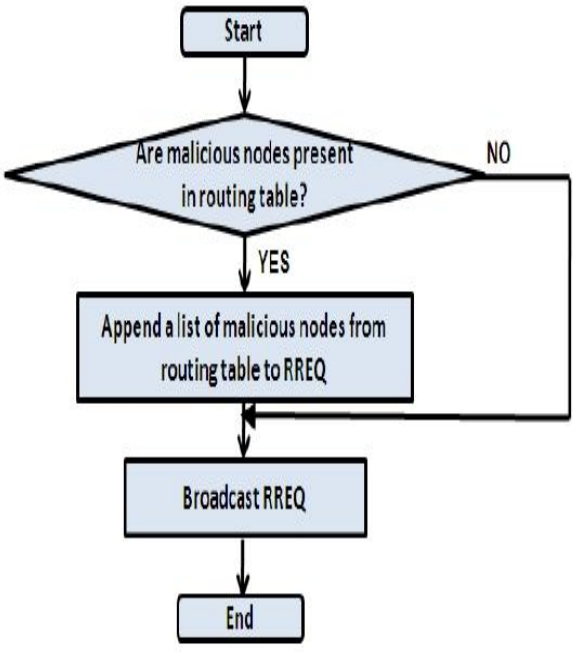

Fig 06: The functionaliy of node broadcasting RREQ [4]

4.4.2 Node receiving RREQ

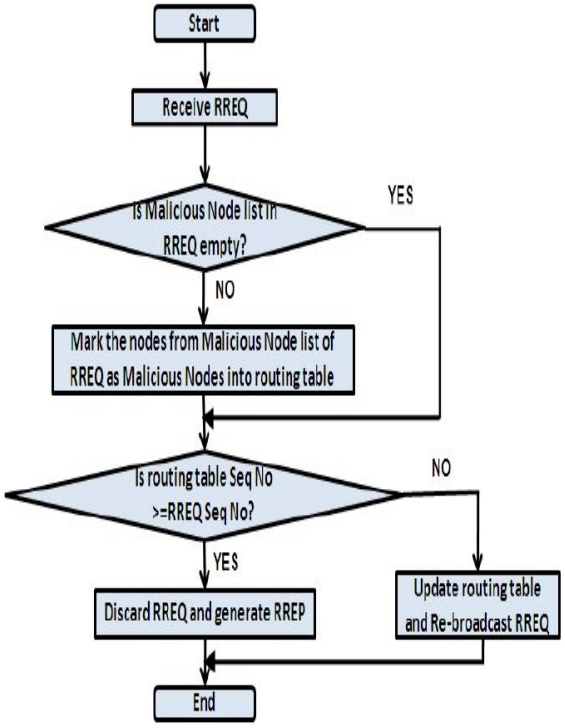

Fig 07: The functionality of node receiving the broadcasted RREQ [4]

4.4.3 Node receiving RREP

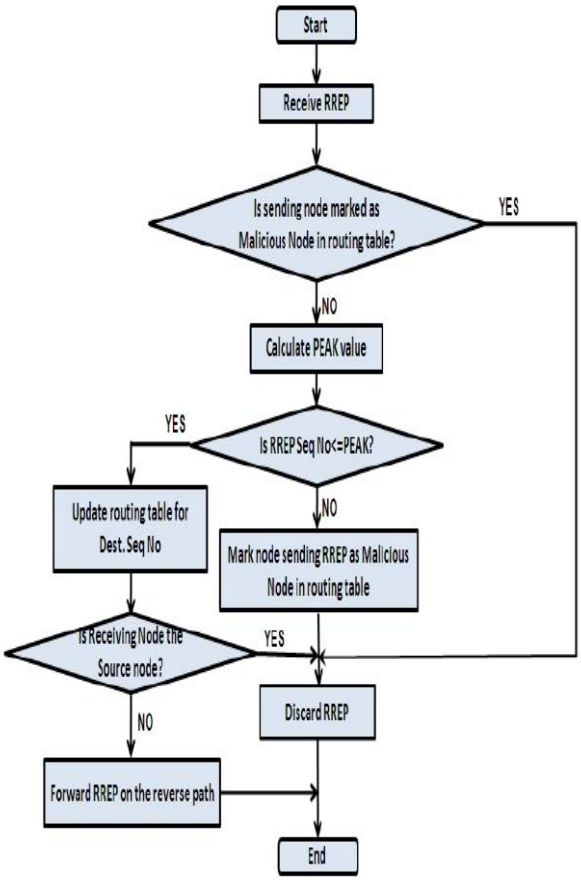

Fig 08: The functionality of node receiving RREP [4]

• *Algorithm Analysis*

MR-AODV has following similarities with R-AODV:

- It adds processing overhead in the structure calculation of PEAK value.
- 4 Bytes of PEAK value are apportioned in memory.
- Two variables utilized for computation of PEAK value viz. NO_OF_SENT_RREQ and NO_OF_RECEIVED_RREP assign 2 Bytes each.
- MALICIOUS_NODE_LIST in RREQ contains 2 Bytes for every section of blacklisted node. The overhead in time is as far as era of Malicious_node_list.
- Routing table is adjusted by addition of a MALICIOUS_NODE field which obliges 2 Bytes for every node section.
- It does not require any additional table to keep up.
- It reductions standardized routing overhead as it doesn't present any additional control packets.
- if attacker produces destination sequence number which is short of what or equivalent to PEAK value, the node is not located as a malicious node [4].

However, unlike R-AODV, MR-AODV:

- does not apportion 2 Bytes for DO_NOT_CONSIDER flag.
- Only RREQ is utilized to advise different nodes in the network about presence of malicious nodes; RREP is not utilized for this reason. As RREP is unicasted while RREQ is broadcasted, this ends up being a superior decision which serves to decrease routing overhead.

Both R-AODV and MR-AODV are just as fit for confining numerous malicious nodes and giving equivalent climb in PDR, yet MR-AODV has an edge over R-AODV as it tosses RREP sent by malicious nodes instead of sending it on the reverse path.

• *Disadvantage:*
If attacker generates destination sequence number which is less than or equal to PEAK value, the node cannot be detected as a malicious node.

4.5 Using RSA key exchange and Encryption

Sibichen and Seedhar, added the RSA key exchange mechanism into AODV [1] [10]. In this mechanism, when the network comprising of various nodes is made, it first checks whether there is any malicious nodes existing in the network. To evacuate these malicious nodes, an advanced AODV protocol component is utilized. Along these lines the malicious nodes are segregated. On the off chance that any intermediate node receives false routing data from its neighbor node, then that node is to be considered as a malicious node.

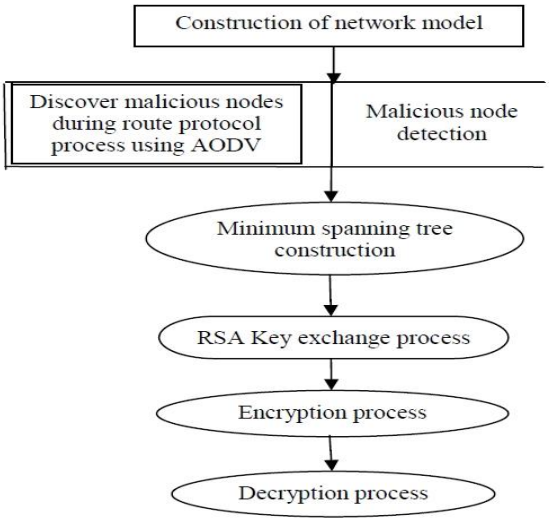

Fig 09: Work Flow [1]

The intermediate node advises alternate nodes about the malicious node through the route reply packet and each node getting the data updates its routing table to stamp the node as malicious. At the point when a RREQ is sent, a rundown of malicious nodes is affixed and alternate nodes update its routing table. Along these lines, the nodes can recognize the rundown of malicious nodes by recognizing off base routing data or by checking routing table with the goal that they can advise different nodes not to consider the routing data from the malicious nodes. The network comprises of a few nodes which are associated by connections. Each node has a remarkable id and each packet is stamped by the id of its source node. This fundamental data is kept up at every machine node in the network. The nodes are arranged in spanning tree topology. The spanning tree keeps up security affiliations just with neighboring nodes [1] [10].

The Neighborhood Key technique, in which every node offers mystery key just with the verified neighbors in the adhoc network, is utilized. This evades bunch re-keying. The time taken to trade the key is less furthermore there is an increment in verification moreover. At whatever point there is a change in the set of confirmed neighbors, a node must register another key and send this new key to all its validated neighbors. This technique guarantees security objectives like trustworthiness and privacy in adhoc networks. After the key trade, the message is encoded twice, by utilizing neighborhood key and message particular key. Hence the security is expanded. This model uses R-AODV for discovering malicious node [1].

4.5.1 Construction of spanning tree

A spanning tree is developed by computing the base distance in the middle of every single nodes which can cover all the nodes without framing a cycle. Spanning trees are not difficult to build and modest. As the spanning tree holds security tie-ups just with neighbors, security can be impressively expanded. Spanning Tree Protocol is a network protocol which secures and keeps up a spanning tree interfacing a gathering of mobile gadget in the wireless ad hoc network and no association with the nodes that are not piece of the tree. The distance between every node in the network is figured by utilizing the equation [1],

$$\text{Distance } (i, j) = \sqrt{[(xj-xi)2 + (yj-yi)2]}$$

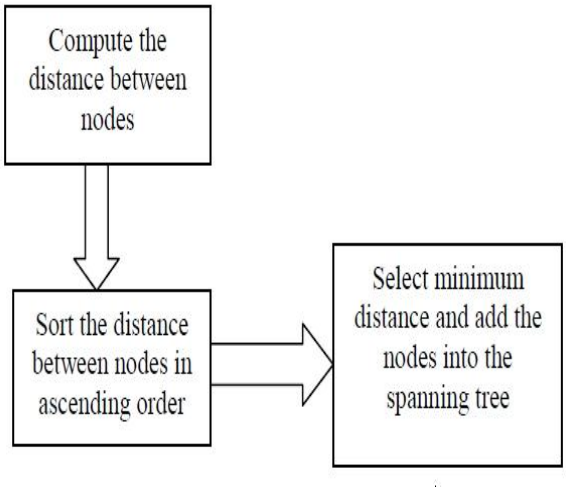

Fig 10: Construction of spanning tree [1]

4.5.2 Authentication among nodes

After development of spanning tree, authentication among the nodes ought to be performed utilizing public key certificates. Mystery keys can be traded just with the authenticated neighbors in the network. Every node has its own certificate which has been marked by a trusted outsider. Shockingly when a node receives protocol message, it requests marked certificate from the other node by sending a certificate request message which incorporates the node's own certificate. In the wake of getting this request, it confirms the signature of the certificate and in the event that it is legitimate, it stores the certificate and sends certificate reply message. Confirmation of certificate is carried out by the accepting node and once they are ended up being authenticated the nodes can begin communication. At the point when node B receives a protocol message from node An and if the certificate of An is obscure, node B tosses the message, and sends Certification request message to A which incorporates B's certificate. At the point when A receives a request, it checks the signature of B's certificate and on the off chance that it is legitimate, A stores the certificate. Node A sends Certification reply message to B that incorporates A's certificate. After getting the message, node B checks the signature of A's certificate and on the off chance that it is substantial, B stores the certificate. When certificates are traded, the nodes begin trading mystery keys. They are utilized for encryption or marking the messages. Every last node acknowledges messages just from authenticated neighbors [1] [10].

4.5.3 RSA KEY exchange

The system utilizes RSA key exchange mechanism to guarantee security. Every last node has its own particular symmetric key called the Neighborhood Key. For encryption and unscrambling every node must have entry to the next node's neighborhood key. At the source node, neighborhood key is encoded with the public key of the receiver and transmitted to the destination node. At the destination node, neighborhood key is decoded with its own private key. It diminishes communication overhead with the capacity to have static, constant keys [1].

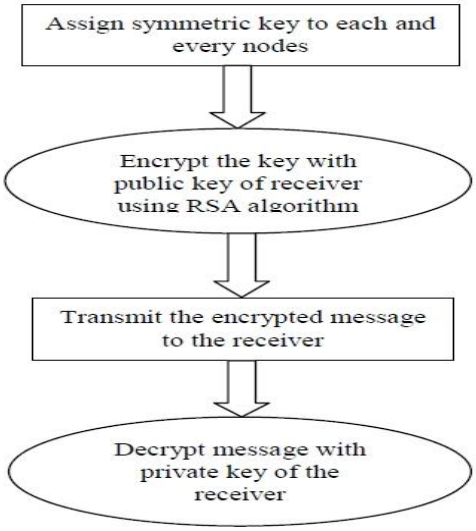

Fig 11: RSA Key exchange [1]

4.5.4 Encryption of message

Each node has a symmetric key called neighborhood key. The message is initially scrambled with the message particular key which is the MAC address. At that point, the message particular key is encoded with neighborhood key. At that point, the sender annexes the destination nodes ID and sends to its authenticated neighbors [1].

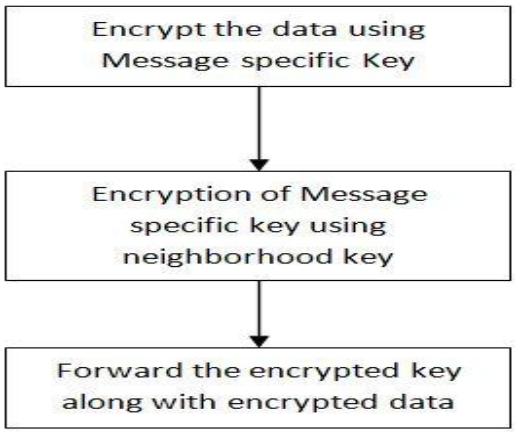

Fig 12: Encryption of message [1]

For scrambling the message, two symmetric encryption algorithms are utilized i.e. the area key and the message particular key. This aides in enhancing the security against Grayhole and Blackhole attack.

4.5.5 Decryption of message

At the receiver, it first checks whether the ID matches or not. On the off chance that the affixed ID matches with the node's ID, then it is the expected beneficiary and decoding is initially performed with neighborhood key of sending node and the plain instant message is gotten. Further decoding is finished with the message particular key and the first message is gotten. In the event that the ID does not match, that node is not the planned beneficiary. So it re-scrambles the message with the area key and sends to its authenticated neighbor nodes. The system is rehashed until destination node is discovered and the first message is unscrambled at the destination node [1].

• *Disadvantage:*
Here it uses R-AODV as well as RSA encryption mechanism. Thus the delay time will be higher. Also it requires an encryption as well as decryption in every intermediate node to change the neighbor key.

5. CONCLUSION

Among different others, the Grayhole and Blackhole attacks are considered as the most perilous attacks towards adhoc network. Despite the fact that, there exist a few mechanisms for securing adhoc networks from such attacks, traditional preventive methodologies in this respect have genuine restrictions and a few disadvantages. Nodes impart a solitary symmetric key for encryption and decoding of messages. Likewise, there is an issue of gathering re-keying which is intricate and lengthy undertaking. Likewise AODV neglects to evacuate malicious nodes amid the route disclosure process and accordingly does not succeed to exchange all information packets to the destination under Blackhole and Grayhole attacks. A large portion of the traditional routines need dependability. Likewise, under these attacks, the Packet Delivery Ratio (PDR) and throughput, may diminish, as the number of malicious nodes increments. The improvements just include least change to existing AODV protocol streams. Discovery and the confinement of malicious nodes are the significant objectives of these improvements.

6. REFERENCES

[1] Sibichen, S and S. Sreedhar. 2013. An Efficient AODV Protocol and Encryption Mechanism for Security Issues in Adhoc Networks, International Conference on Microelectronics, Communication and Renewable Energy, 184-194.

[2] Sony, S. J. and S. D. Nayak. 2013. Enhancing Security Features & Performance of AODV Protocol under Attack for MANET, IEEE Transactions on mobile computing, 325-328.

[3] Jalil, K. A., Z. Ahmad and J. L. A. Manan. 2011. Securing Routing Table Update in AODV Routing Protocol, IEEE Conference on Open Systems, 116-121.

[4] Jhaveri, R. H. 2013. MR-AODV: A Solution to Mitigate Blackhole and Grayhole Attacks in AODV Based MANETs, Third International Conference on Advanced Computing & Communication Technologies, 254-260.

[5] Jhaveri, R. H., S. J. Patel and D. C. Jinwala. 2012. A Novel Solution for Grayhole Attack in AODV Based MANETs, In Proc. of Third International Conference on Advances in Communication, Network and Computing: Springer, 60-67.

[6] Jhaveri, R. H., S. J. Patel and D. C. Jinwala. 2012. Improving Route Discovery for AODV to Prevent Blackhole and Grayhole Attacks in MANETs, INFOCOMP Journal of Computer Science, 1-12.

[7] Zain, J. M., W. M. B. W. Mohd and E. E. Qawasmeh. 2011. Software Engineering and Computer Systems, Second International Conference, ICSECS, 102-115.

[8] Lakhtaria, K., B. N. Patel, S. G. Prajapati and N. N. Jani. 2009. Securing AODV for MANETs using Message Digest with Secret Key, International Journal of Network Security & Its Applications, 111-116.

[9] Patil, S. and J. Raghatwan2. 2014. Survey on Security in Wireless Ad-hoc Network, International Journal of Science and Research, 2041-2044.

[10] Sibichen, S and S. Sreedhar. 2013. An Efficient AODV Protocol and Encryption Mechanism for group Communication and Preventing Attacks in Adhoc Networks, International Journal of Scientific & Engineering Research, 01-06.

[11] Khachar, K. N. and J. B. Shah. 2014. Detection and Prevention of Black Hole Attack in Mobile Ad-Hoc Networks: A Survey, IOSR Journal of Computer Engineering, 108-112.

[12] Jalil, K. A., Z. Ahmad and J. L. A. Manan. 2011. ERDA: Enhanced Route Discovery Mechanism for AODV Routing Protocol against Black Hole Attacks, IEEE Conference on Open Systems, 116-121.