http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (5): 656-663, 2007
ISSN 1812-5638

Providing Routing Security Using ROS Protocol in MANET and Performance Comparison with AODV

J. Martin Leo Manickam and S. Shanmugavel
Department of ECE, Anna University, Chennai, India

Abstract: Many of the proposed routing security solutions for the Mobile Ad hoc Networks (MANET) requires cryptographic techniques or a secret association between the communicating entities. These security solutions work well for the anticipated attacks but fails for unanticipated attacks. In this study, a Resiliency Oriented Secure (ROS) routing protocol for MANET is proposed. Present proposed protocol is a secure extension of AODV, in which each node suspects routing packet based on update time interval in its route table entry and confirms it with local neighbors. The evaluation results show that present protocol has better performance than Ad hoc On demand Distance Vector (AODV) routing protocol under attack in terms of delivery ratio, routing overhead ratio, average end-to-end delay and average route acquisition latency.

Key words: MANET, routing protocol, security, AODV, resiliency, threats

INTRODUCTION

An ad hoc network consists of low power wireless nodes, capable of communicating with each other by co-operating in the establishment of routes in forwarding packets to the destination. This network does not require any centralized infrastructure. The multi hop routes are established instantaneously and exist as long as there is a communication between the entities. As there is no centralized infrastructure, the network is easy to deploy and provides unlimited mobility, whereas more vulnerable to security threats. Many efficient routing protocols as proposed by Perkins *et al*. (2003) and David and David (1996) have better network performance whereas more vulnerable to security threats mentioned in Ning and Sun (2001). Several security solutions as proposed by Deng *et al*. (2002), Yang *et al*. (2004), Wang *et al*. (2003) and Hu *et al*. (2003) requires a centralized server for key distribution or a secret understanding between communicating entities.

Several security schemes based on AODV and Dynamic Source Routing (DSR) protocols have been proposed. Zhou and Haas (1999) proposed a secure routing protocol uses threshold cryptography and depends on secret sharing servers to protect the routing information. Papadimitratos and Haas (2003) proposed SRP, which requires for every route discovery, source and destination must have security association between them. Johnson and Perig (2002) and Rifa-Pous and Herrera-Joancomarti (2007) proposed a secure routing protocol in which hash chains are used to secure the hop counts. But, finding a suitable key distribution technique and authentication scheme is not a straight forward approach. Hu *et al*. (2002) proposed ARIDANE requires clock synchronization, which is not feasible in realistic ad hoc networks. Levine *et al*. (2002) proposed ARAN, uses a trusted certificate server to provide authentication. Every node along the path should sign the route discovery/route reply message. Yang *et al*. (2006) described a self organized approach, in which each node requires a token in order to participate in the network operations. A node is said to be malicious if it does not have any tokens. The neighbor verification mechanism is based on tokens and employs the asymmetric cryptographic primitives, RSA standard. The RSA algorithm involves more computation overhead in signing/decrypting and verifying/encrypting operations. Moreover, control overhead will be high as each routing message is verified with its next hop. Hence, this self organized approach is achieved at the cost of increased overhead, due to asymmetric cryptography primitives. Several trust based routing protocols proposed by Theodorakopoulos and Baras (2006) and Pirzada *et al*. (2006) has high routing overhead and the average end-to-end delay of data packets increases.

Many secure routing protocols is found to have better network performance in the presence of anticipated attacks but fails to unanticipated attacks. Hence, our main goal is to design a resiliency oriented secure routing protocol to meet the following objectives:

Corresponding Author: J. Martin Leo Manickam, Department of ECE, Anna University, Chennai, India Tel: 91-44-22781492

- **Minimal cryptographic techniques:** Secure routing protocol should employ minimum cryptographic techniques. Because, the extensive use of cryptographic techniques may degrade the performance of the network in terms of the overhead ratio, route acquisition time, power consumption etc. Present proposed protocol does not use cryptographic techniques in route discovery process. This is achieved by preventing any participating nodes to modify the routing fields in RREQ packet.
- **Intrusion tolerant:** As malicious attacks are unavoidable in MANET environment, the routing protocol should be intrusion tolerant. The protocol should be able to work well with acceptable performance degradation.
- **Multi fence security:** Security should be incorporated in each and every parameter defined in the routing protocol. These parameters should be strictly monitored for each update in the routing table to thwart the known and unknown security threats. In our proposed protocol, a new parameter called Update Time Interval (T_{UTI}) is defined to determine a malicious routing packets. T_{UTI} is the time interval between successive updates in the route table for the same destination.
- **Collaborative security approach:** Protocol should enable the participating nodes to make a decision whether to accept or discard a suspected routing message based on its own available data as well as from neighbor data. Moreover, a collection of data from its neighbors should be considered to influence its own data.

In this research, the ROS routing protocol which is able to perform well with acceptable performance degradation even in the presence of malicious attacks is proposed. Present proposed protocol does not require any cryptographic techniques for route discovery. The protocol detects a malicious packet based on update time interval in its route table entry. It checks the suspected packet with its previous hop or local neighbors instead of all the received routing packets, thereby minimizing the routing overhead ratio to enhance the performance of the network.

AODV routing protocol: When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a path discovery process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors and so on, until either the destination or an intermediate node with a fresh enough route to the destination is located. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates and together with the node's IP address, uniquely identifies a RREQ. Along with its own sequence number and the broadcast ID, the source node includes in the RREQ the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ. During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received to establish a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destination or intermediate node responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ. As the RREP is routed back along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the RREP came. The forward route entries indicate the active forward route. Associated with each route entry is a route timer which will cause the deletion of the entry if it is not used within the specified lifetime. If a source node moves, it is able to reinitiate the route discovery process to find a new route to the destination. If a node along the route moves, its upstream neighbor notices the move and propagates a link failure notification message (an RREP with infinite metric) to each of its active upstream neighbors to inform them of the erasure of that part of the route. These nodes in turn propagate the link failure notification to their upstream neighbors and so on until the source node is reached. The source node may then choose to reinitiate route discovery for that destination if a route is still desired.

Resiliency Oriented Secure (ROS) Routing Protocol

Assumptions: Present protocol is based on the following assumptions.

- Any two nodes within the wireless communication range can interact with each other over the shared wireless channel. Each wireless interface may operate in the promiscuous mode, i.e., if node A is within the communication range of node B, then node A can overhear all the communications going on at node B.
- All the overheard packets are stored in the route cache.

ROS routing protocol: Path discovery process is initiated whenever source node needs to communicate with another node for which it has no routing information in its table. Source node initiates path discovery by broadcasting a route request (RREQ) packet to its neighbors. The RREQ contains the following fields source address, source sequence number, broadcast id, destination address, destination sequence number, upstream neighbor address. Whenever a node issues or receives a RREQ packet, it will update its route table along with T_{UTI} for the entry. On receiving the RREQ packet, neighbors will process it as mentioned below:

- If an identical RREQ packet is processed earlier, it will drop the received RREQ packet.
- If the received RREQ packet is new and produces an update in the route table less than T_{UTI} sec, then it will initiate a misbehavior detection mechanism.
- If the received packet is new and produces an update in the route table above T_{UTI} sec, it will update the route table and rebroadcasts the same RREQ packet by replacing the upstream neighbor address by its node address.

If a route is found in the route table, then the intermediate node unicasts a RREP packet to the source. As the RREP travels back to the source, each node along the path sets up a forward pointer to the node from which the RREP came, updates its timeout information for route entries to the source and destination and records the latest destination sequence number for the requested destination. Moreover, all the overheard RREP packets are stored in the route cache. Any modification done on the unicasted RREP packet can be detected. It propagates the first RREP for a given source node towards that source. If it receives further RREP's, it updates and propagates the RREP packets, only if the RREP contains a greater destination sequence number. When link failure occurs, then a special RREP packet is unicasted to its upstream neighbors.

Misbehavior detection

Detection of routing update misbehavior on broadcasted routing packet: Consider the network topology shown in the Fig. 1. Source S broadcast a RREQ message (S1) to its neighbors A and M. They will rebroadcast the packet without doing any modification in source and destination sequence number if no route is found in its route table to the destination. As source S is in the transmission range of A and M, any modifications in the routing message by A and M can be identified by S and it will discard any routing packets from node M. Consider M as a malicious node that modifies (increases the source sequence number) in the RREQ packet S1 and rebroadcasts the routing message M1 to its neighbors (S, A and B). Assume that due to channel characteristics, S is unable to overhear the transmission of M. In this scenario, node A being a neighbor of both S and M is able to receive two different RREQ packets S1 and M1 broadcasted by S and M, respectively. The RREQ packet broadcasted by node M will produce an update in the route table of node A in less than T_{UTI} sec. Then, A will unicast a check request packet (A2) to S as shown in Fig. 2. Node S may respond it by unicasting (S2) a positive check acknowledgment if it has broadcasted the packet or a negative check acknowledgment if it has not broadcasted the packet as shown in Fig. 2. On receiving the positive check acknowledgment packet, A will update its routing table. If A receives a negative check acknowledgment packet or no packet within a time interval, node A will discard the packet broadcasted by M and keeps it in the malicious list. Any further routing packets from it will not be considered for a short time interval. The flow chart describing the misbehavior detection mechanism on broadcasted routing packet in ROS protocol is shown in Fig. 3.

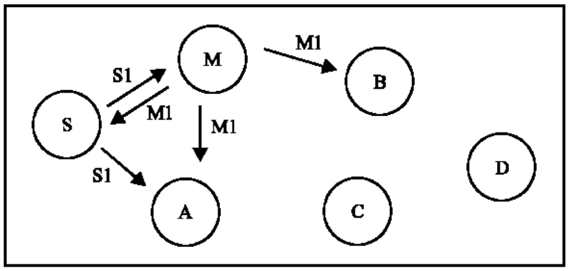

Fig. 1: Node S broadcasts a routing message (S1) to neighbors A and M. Malicious node M modifies the contents of the routing message (S1) and rebroadcasts (M1) to its neighbors S, A and B

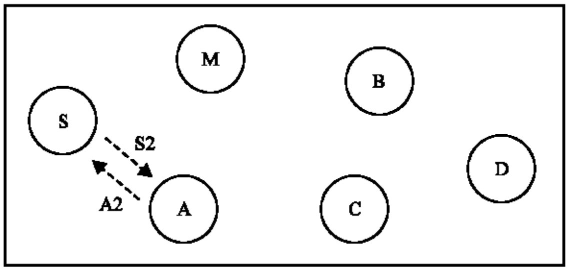

Fig. 2: Node A unicasts a check request packet (A2) to node S. Node S unicasts a positive/negative check acknowledgment (S2) to nodeA

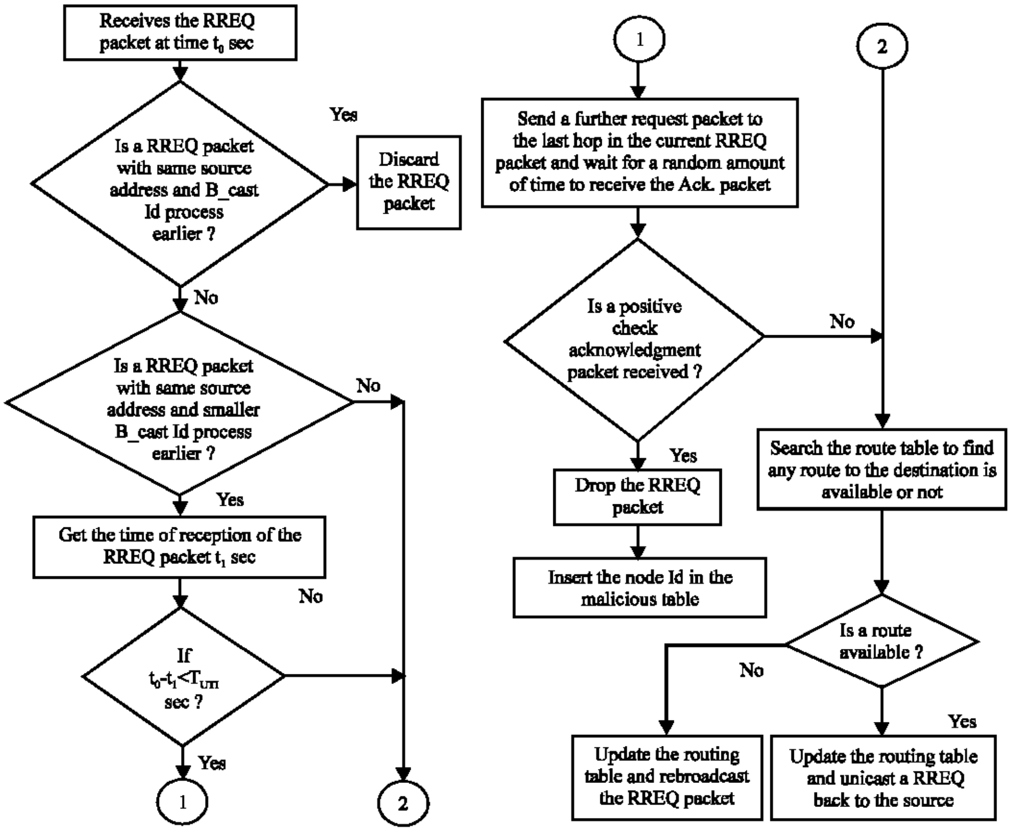
Receives the RREQ packet at time t_0 sec
Is a RREQ packet with same source address and B_cast Id process earlier ?
Yes
Discard the RREQ packet
No
Is a RREQ packet with same source address and smaller B_cast Id process earlier ?
No
Yes
Get the time of reception of the RREQ packet t_1 sec
No
If t_0-t_1<T_{UTI} sec ?
Yes
1
2
1
Send a further request packet to the last hop in the current RREQ packet and wait for a random amount of time to receive the Ack. packet
Is a positive check acknowledgment packet received ?
No
Yes
Drop the RREQ packet
Insert the node Id in the malicious table
2
Search the route table to find any route to the destination is available or not
Is a route available ?
No
Yes
Update the routing table and rebroadcast the RREQ packet
Update the routing table and unicast a RREQ back to the source

Fig. 3: Flowchart describing the misbehavior detection on broadcasted routing packet

Detection of routing update misbehavior on unicasted routing packet: Consider the node D unicast a RREP packet (D1) to its next hop B as shown in Fig. 4. Node C, being a neighbor of D and B, can overhear their transmission and stores it in its route cache.

On receiving the routing message from node D, node B updates its routing table and transmits the routing message (B1) without doing any modification in routing fields as shown in the Fig. 4. This transmission can be verified or checked by overhearing the transmissions of B by D. If D is unable to overhear the transmission of B, it will broadcast a check request packet (D2) to its neighbors as shown in the Fig. 5. As node C has overheard the transmission of B, it will check its route cache and unicasts a positive check acknowledgment (C2) to D if node B has unicasted the RREP packet correctly as shown in the Fig. 5. If node D has not received any reply from its neighbors for a short time interval, it will choose an alternate path to the source. The flow chart describing the misbehavior detection mechanism on unicasted routing packet is shown in Fig. 6.

Update time interval: Consider the network topology shown in the Fig. 7. Let T_{prog} and T_{pros} be the average propagation and processing time for a control packet. Assume that, the node A broadcasts the RREQ packet A1 at time t_1 sec. Then, node B receives the packet after t_1 $+T_{prog}$ sec, update its route table with the recent source sequence number and rebroadcasts the packet. Simultaneously, the node C being malicious increases the source sequence number and rebroadcasts the packet C1. Node B receives the packet C1 after $t_1 + 2T_{prog} + T_{pros}$ sec and falsely updates its route table as the packet contains higher sequence number so that the path from D to S will be D-B-C-A-S instead of D-B-A-S. We have defined the update time interval (T_{UTI}) as the time interval between the successive updates for the same entry in the route table. Then, the minimum T_{UTI} is given by $T_{prog} + T_{pros}$ secs. The upper bound for T_{UTI} will be the route reply wait time (T_{RREP_Wait}). Then, the update time interval is bounded as follows:

$$T_{prog} + T_{pros} < T_{UTI} = T_{RREP\_Wait}$$

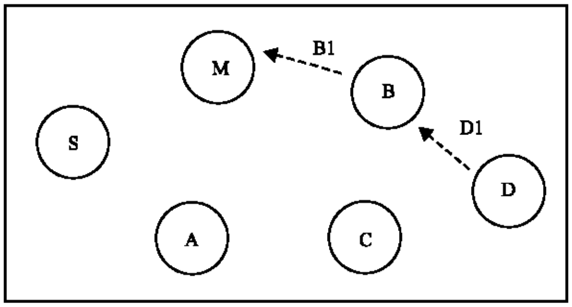

Fig. 4: Node D unicasts a routing message (D1) to node B. Node B unicasts (B1) the packet to node M. But node D can't overhear the transmission

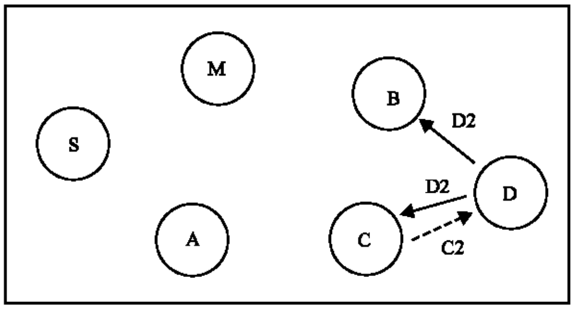

Fig. 5: Node D broadcasts check request packet (D2) to its neighbors (B and C). Node C unicasts positive check acknowledgment (C2) to node D

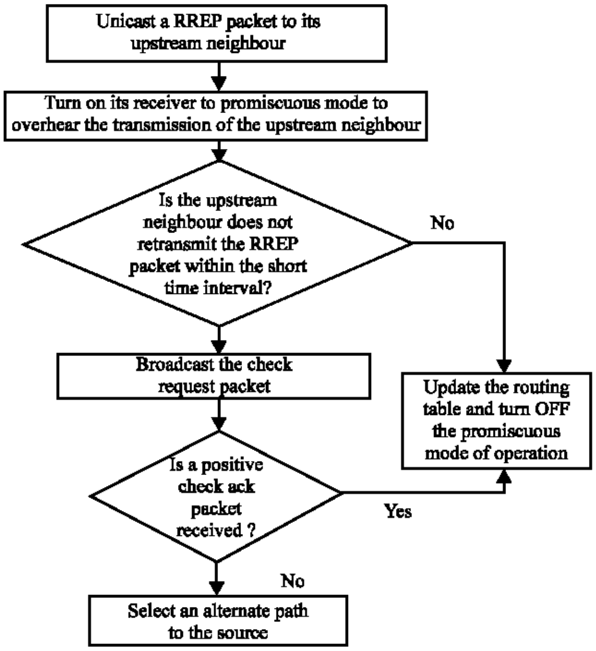
Unicast a RREP packet to its upstream neighbour
Turn on its receiver to promiscuous mode to overhear the transmission of the upstream neighbour
Is the upstream neighbour does not retransmit the RREP packet within the short time interval?
No
Broadcast the check request packet
Update the routing table and turn OFF the promiscuous mode of operation
Is a positive check ack packet received ?
Yes
No
Select an alternate path to the source

Fig. 6: Flowchart describing the misbehavior detection on unicasted routing packet

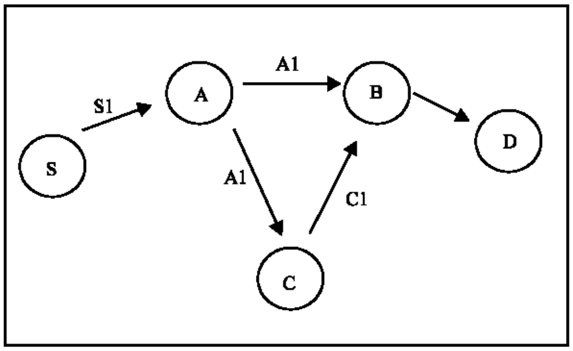

Fig. 7: The update time interval

In our proposed protocol, the maximum bound for T_{UTI} is chosen, so that no node need not update its route table for the same entry in less than T_{UTI} sec, if there is no malicious neighbor. Whenever a node receives a routing packet that produces update in the route table in less than T_{UTI} secs, then the misbehavior detection mechanism will be initiated.

Performance evaluation: The proposed ROS protocol is tested using the public domain simulator, GloMoSim. The algorithm is evaluated in terms of packet delivery ratio, routing overhead ratio, route acquisition time and end-to-end delay for the data packets and the results are presented.

The simulation for evaluating the proposed ROS was implemented within the GloMoSim Library. The GloMoSim (GLObal Mobile information system SIMulator) provides a scalable simulation environment for wireless network systems. It is designed using the parallel discrete-event simulation capability provided by PARSEC (PARallel Simulation Environment for Complex Systems) Meyer *et al.* (1998). It is a C-based simulation language developed by the Parallel Computing Laboratory at UCLA, for sequential and parallel execution of discrete event simulation model.

Simulation environment and methodology: In the simulation, a network of mobile nodes placed randomly with in a 1000×1000 m area is modeled. Radio propagation range for each node was 250 m (Gomathy and Shanmugavel, 2005) and a channel capacity of 2 Mbps is chosen. There were no network partitions through out the simulation. Table 1 indicates the simulation environment for analyzing the performance of the proposed ROS protocol. Table 2 show the simulation parameters, which are used as default values unless otherwise specified. The size of the data payload is 512 bytes. Data sessions with randomly selected sources and destinations were simulated. Each source transmits data packets at rate of 2 packets sec^{-1}. The node 9 was chosen as malicious

Table 1: Simulation environment

Processor	2.93 Ghz. PIV
Harddisk	80 GB
RAM	256 MB SDRAM
Operating system	Windows 2000

Table 2: Simulation parameters

Simulator	GloMoSim
Simulation area	1000* 1000 m
No. of mobile hosts	30
Transmission range	250 m
Movement model	Random waypoint
Maximnm speed	5-20 m sec^{-1}
Traffic type	CBR (UDP)
Data payload	512 bytes
Packet rate	2 packet sec^{-1}
No. of malicious nodes	1

node that modifies the broadcast id and source sequence number in the incoming RREQ packet and drops the RREP and data packets.

Performance metrics: The following metrics were chosen to evaluate the impact of the sequence number attack:

- **Packet delivery ratio:** The fraction of application level data packets sent that are actually received at the destination.
- **Routing overhead ratio:** The fraction of routing packets transmitted for every dada packet sent. Each hop of the routing packet is treated as a packet.
- **Average route acquisition latency:** This is the average delay between the sending of a RREQ packet by a source for discovering a route to destination and the receipt of the first corresponding RREP packet. If a route request is timed out and needed to be retransmitted, the sending time of the first transmission was used to calculate the latency.
- **Average end-to-end delay of data packets:** This is the average delay between the sending of the data packet by the CBR source and its receipt at the corresponding CBR receiver. This includes the route discovery time, the queuing delay at node, the retransmission delay at the MAC layer and the propagation and transfer time in wireless channel

Performance comparison: The performance of the proposed ROS protocol is compared with AODV with and without malicious attack under various number of connections and mobility. Figure 8 shows the delivery ratio versus the number of connections in the network under three scenarios: (I) AODV without a malicious attack (ii) AODV with an attack and (iii) ROS protocol with an attack. The malicious attack performed against the AODV protocol has a very big impact on delivery ratio, decreasing it to lower than the half compared to the normal AODV. This is because the malicious node attracts the data traffic to itself by falsely modifying the source sequence number in the RREQ packets. It can be seen that the delivery ratio increases with the number of connections in the network and it reaches its maximum when the number of connections is 20. When the number of connections increased beyond 20, the delivery ratio starts decreasing. It is obvious that when the number of connections increases, the control packets originating from the malicious host also increases causing additional processing delay in the proposed ROS protocol and hence the delivery ratio decreases.

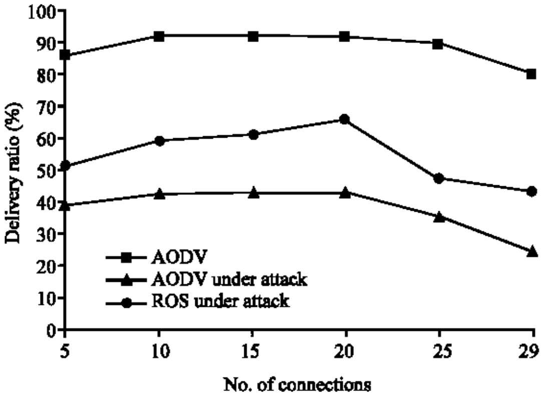

Fig. 8: Delivery ratio vs number of connections

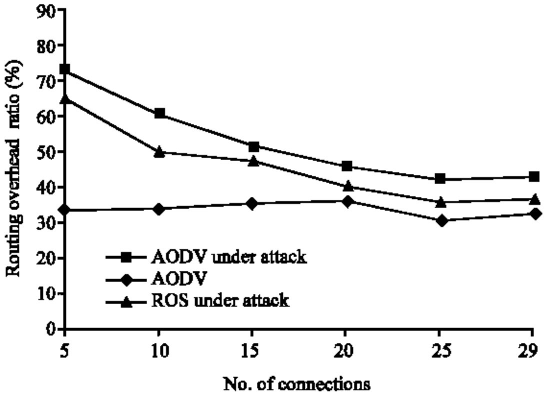

Fig. 9: Routing overhead ratio versus number of active connections

Figure 9 shows the routing overhead ratio versus number of connections. It was found that the additional routing overhead introduced by the attacking node reaches 78% when the network size is small and decreases as the number of connections increases. This behavior is normal since when there are only five active connections in the network, the fresh route to the destination nodes are limited and therefore, it is essential that more number of nodes has to participate in the route discovery process.

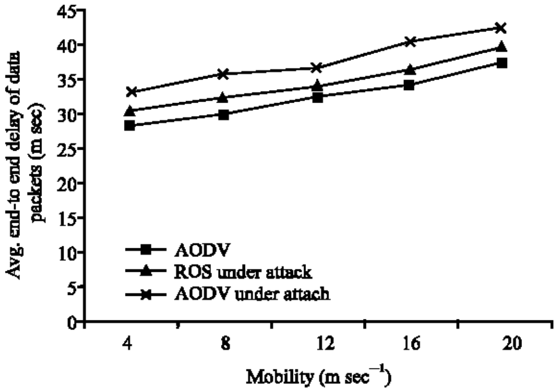

Fig. 10: Average end-to-end delay versus mobility

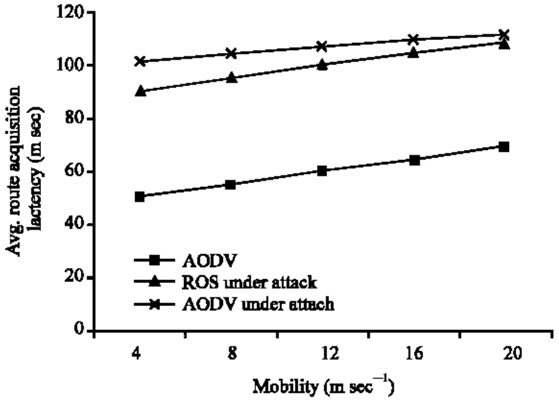

Fig. 11: Average route acquisition latency versus mobility

The simulation results show that the routing overhead introduced by a malicious node reaches 50.3% while AODV has 38.7% average of routing overhead. Figure 10 shows the average end-to-end delay for the data packets is almost similar to that of AODV (Fig. 10). This is because the processing of data packets is identical in either protocol and so the average latency remains the same. Figure 11 shows the average route acquisition Latency for ROS protocol is almost double that for AODV. While processing ROS routing protocol packet, the nodes has to send a check request packet to the last hop and waits for a random time to receive the acknowledgment. This intermediate check and verification causes additional delays at each hop and so the route acquisition latency increases.

CONCLUSIONS

Present proposed ROS routing protocol does not require any cryptographic schemes and hence eliminates the requirement of a centralized agent for key distribution. Evaluation results show that the performance of the protocol is better than the Ad hoc On Demand Distance Vector routing protocol (AODV) under malicious attack. In our proposed protocol, the malicious nodes are suspected/identified based on a single time parameter. Now, we plan to extend this work to design a comprehensive resiliency oriented secure routing protocol by including more parameters for misbehavior detection on control packets.

REFERENCES

David, B. and A. David, 1996. Dynamic source routing in ad hoc wireless networks. Mobile Computing, Chapter 5, pp: 153-81.

Deng, H., W. Li and D.P. Agrawal, 2002. Routing security in wireless ad hoc networks. IEEE Commun. Mag., 40: 70-75.

Gomathy and S. Shanmugavel, 2005. Supporting QoS in MANET by a fuzzy priority scheduler and performance analysis with multicast routing protocols. EURASIP J. Wireless Commun. Networking, 3: 426-436.

Hu, Y.C., A. Perrig and D.B. Johnson, 2002. Ariadne: A secure on-demand routing protocol for ad hoc networks. Proceedings Annual International Conference of Mobile Computing and Networking (MobiCom'02), pp: 12-23.

Hu, Y.C., A. Perrig and D.B. Johnson, 2003. Rushing Attacks and Defense in Wireless ad Hoc Network Routing Protocols. Proceedings of the 2003 ACM Workshop on Wireless Security, September, San Diego, CA, USA.

Johnson, H.D. and A. Perrig, 2002. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. Proceeding IEEE Workshop on Mobile Computing Systems and Applications, pp: 3-13.

Levine, D.B.N., E. Royer and C. Shields, 2002. A Secure protocol for ad hoc networks. Proceedings of International Conference on Network Protocols. (ICNP), pp: 78-87.

Meyer, B.R. and M. Takai *et al.*, 1998. Parsec: A Parallel simulation environment for complex systems. IEEE Comput., 31: 77-85.

Ning, P., K. Sun, 2003. How to misuse AODV: A case study of insider attacks against mobile ad hoc routing protocols. 4th Annual IEEE Information Assurance Workshop, pp: 60-67.

Papadimitratos and Z. Haas, 2003. Secure link state routing for mobile ad hoc networks. IEEE Workshop. Security and Assurance in Ad Hoc Networks.

Perkins, C.E., E.M. Belding-Royer and S. Das, 2002. Ad hoc on Demand Distance Vector (AODV) routing. IETF Internet Draft, draft-ietf-manet-aodv-10.txt, March.

Pirzada, A.A., C. McDonald and A. Dattal, 2006. Performance comparison of trust-based reactive routing protocols. IEEE Trans. Mobile Comput., 5: 695-710.

Rifa-Pous, H. and J. Herrera-Joancomarti, 2007. Secure dynamic MANET On-demand (SEDYMO) routing protocol. 5th Annual Conf. Communi Networks and Services Research (CNSR '07), pp: 372-380.

Theodorakopoulos, G. and J.S. Baras, 2006. On trust models and trust evaluation metrics for ad hoc networks. IEEE J. Selected Areas Commun., 24: 318-328.

Wang, W., Y. Lu and B. Bhargava, 2003. On vulnerability and protection of ad hoc on-demand distance vector protocol. Proceedings of International Conference on Telecommunications (ICT'03), February.

Yang, H., H. Luo, S. Lu and L. Zhang, 2004. Security in mobile ad hoc networks: Challenges and solutions. IEEE Wireless Commun., 11: 38-47.

Yang, H., S.J.X. Meng and S. Lu, 2006. SCAN: Self-organized network-layer security in mobile ad hoc networks. IEEE J. Selected Areas Commun., 24, pp: 261-273.

Zhou and Z.J. Haas, 1999. Securing ad hoc networks. IEEE Network Mag., 13: 24-30.