FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 40 (2015) No. 4
DE GRUYTER OPEN
DOI: 10.1515/fcds-2015-0016
ISSN 0867-6356
e-ISSN 2300-3405

MODIFIED FIDELITY BASED ON-DEMAND SECURE (MFBOD) ROUTING PROTOCOL IN MOBILE AD-HOC NETWORK

Himadri Nath SAHA[1],Rohit SINGH[2],
Debika BHATTACHARYYA[3],P.K.BANERJEE[4]

Abstract: With advent of technology MANET is becoming more and more ubiquitous, and so is the vulnerability of such networks to attacks. In this paper, we propose a secure, lightweight, on-demand routing protocol for MANETs. It uses the concept of fidelity to allocate trust to a neighbor, thereby taking the decision whether to send data via that neighbor or not. To combat attacks efficiently new packets like report and recommendation are used. After receiving a few of these packets a node can conclude about the behavior of a node, thereby identifying and blacklisting the malicious nodes. We try to impose bounds for the fidelity with reference to the battery of the node, which restricts a node to increase its fidelity to infinity and become dominant in the network. This protocol not only finds a secure route to transmit data, but also identifies the malicious nodes in the network. Our protocol exhibits high packet delivery fraction, with low normalized routing load and low end to end delay; which has been observed while simulating in GloMoSim platform. We have observed that our protocol performs not only better than other existing secure routing protocol in a malicious environment, but also combats, many attacks which have not been dealt with these protocols.

Keywords: Fidelity, Recommendation, Blacklist, Secure Protocol, Glomosim.

[1] Assistant Professor, Department of Computer Science and Engineering, Institute of Engineering & Management, Kolkata, India, email- him_shree_2004@yahoo.com
[2] Undergraduate Student, Department of Computer Science and Engineering, Institute of Engineering & Management, Kolkata, India, email- roh9singh@yahoo.in
[3] Head of Department, Department of Computer Science and Engineering, Institute of Engineering & Management, Kolkata, India, email-bdebika@yahoo.com
[4] Professor, Department of Electronics And Communication Engineering, Jadavpur University, Kolkata, India, email-pkbju10@yahoo.com

1. Introduction

Mobile ad hoc networks (MANETs) are an independent association of mobile nodes which communicate over a constrained wireless links. MANETs highlighting characteristics of autonomously communicating with nodes directly, without centralized infrastructure, makes it stand apart from other conventional wireless networks, like as cellular networks and IEEE 802.11 networks. Moreover, MANETs are highly adaptive, for networks with high mobility, or which demand constant formation and deformation without the need for any system administration. Initially, it was conceptualized mainly for crisis situations like battlefields or rescue operations. Nodes can be any wireless device like personal computers (laptops), mobile phones, etc. Figure 1 illustrates what MANET is, along with its application. In general, a wireless node can be any computing equipment that employs the air as the transmission medium. As shown in Figure 1, the nodes wirelessly communication among them, used by army, emulated as mobile nodes.

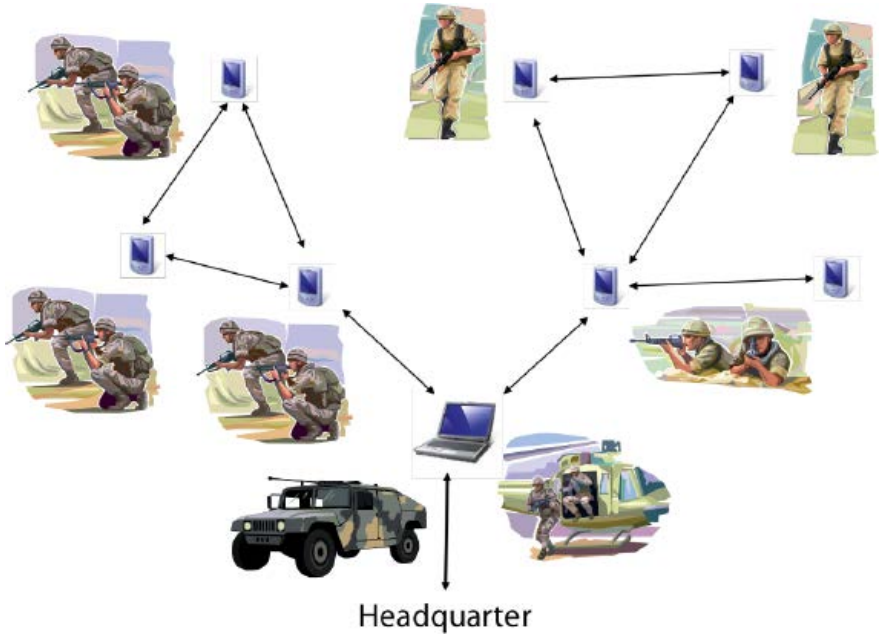

Figure 1: MANET Overview

It allows the devices to maintain connections to the network as well as add and remove devices to and from the network. User can design such networks at cheaper costs and minimize time. MANET has the following characteristics, such as [4]:

- Weaker in security
- Device size Limitation
- Battery life
- Dynamic topology
- Bandwidth and slower data transfer rate

Moreover, there are many complicated attacks which MANETs have to face such as Blackhole, Greyhole, Jellyfish, DoS, Blackmail, Rushing, Sybil, and others [12]. To overcome these weaknesses and attacks, we developed a robust and secure routing protocol.

In Section 2, we have presented a review of the existing secure routing protocols. In Section 3, we introduce our protocol's assumption, data structure, packet formats and the definitions associated with the protocol. In Section 4, we explain the model with an example, followed by the algorithm and flowchart in Section 5. In Section 6 we explain the software simulation environment along with the results and comparison of the same with other mentioned protocol.

2. Related Work

In this section we have reviewed some of the existing secure routing protocols for MANETs, and present their shortcomings.

Sanzgiri et.al [14] have proposed Authenticated Routing for Ad hoc Networks (ARAN), which uses asymmetric cryptography. It is secure as long as CA (certification authority) is not compromised. Since it uses public key encryption confidentiality is guaranteed; network structure is not exposed, and gives resistance to most of the attacks. Though the protocol maintains a high PDF, it requires extra memory, along with high processing overhead for encryption. Moreover, it does not use hop count, so the discovered path may not be optimal. In every hop the RREQs and RREPs are authenticated, thereby preventing impersonation easily. Still ARAN is affected by attacks like black hole, wormhole and rushing attacks.

Zapta et.al [15] have proposed Secure-AODV (SAODV**)**, which uses digital signatures to authenticate non-mutable fields of the routing control messages and one-way hash chains, thereby securing hop count information. The protocol is resilient against attacks like Dos and Black-hole. However, there are some disadvantages which include availability of nodes private keys to other nodes, possibility of MIM attacks by invader nodes, possibility of a simulation adjacency feature by the invader node while sending reply message.

Li et.al [9] have proposed A Trusted AODV (TAODV) routing protocol. It is a trust based model which uses trust recommendation and later on combining these to derive a logical conclusion. It exchanges, trust via two packets called TREQ and TREP. On the basis of the opinions coming in from these packets the judging is done. A trust judgment has three categories, belief, disbelief and uncertain, upon which the next decision is taken. It uses flexible security levels, which help the protocol to respond well when it is being attacked. The computational overhead of each authentication operation is high, and it may even lead to high traffic when there are many nodes with uncertain category.

Saha et.al [13] have proposed Fidelity Based On Demand (FBOD) routing protocol, which is based on the concept of fidelity. Fidelity is an integer number that is associated with each node. This fidelity of a node denotes many things about the node itself and also deciphers other information regarding the topology of the entire network. The advantage is that the approach reduces the computational overhead to a lot extent. It has reduced the amount of network activity for each node required to route a data packet and how this scheme prevents various attacks which may jeopardize any MANET. However, the protocol cannot deal with blackmail attacks, nor can it deal with greyhole attack effectively. It takes time to detect and eliminate a malicious node from the network.

Our contribution is to provide a secure and robust mechanism for establishing communication between network entities under the assumptions listed in Section 3.1 and to mitigate the effects of the malicious entities as presented in [12]. However, it may be noted that it is not possible to eliminate the malicious entities completely from the network, but only to eliminate some of their adversarial effect; hence, the appropriate trade-off is done. The motivation of our work is to:

- Develop a co-operative routing algorithm that gives weight to past activities of a Neighboring Node in context to the successful communication, i.e., successful sending and receiving of data packets; in the form of fidelity.
- Detect and eliminate the malicious entities from the network, through blacklisting and recommendation.
- Use schemes like culprit array and report to decrease the flooding of the same packets to the same node.
- Use busy wait and prevent attacks like rushing and wormhole; and prevent formation of loops.
- Use a single row Routing table, to save the most recently used path; and prevent frequent route searching, when a huge number of packets need to be sent to a destination.
- Have a high packet delivery fraction and end-to-end delay; and low normalized routing load.
- Decrease the routing overhead by developing new packets.
- Use a light self-organized key management scheme.

3. Proposed Secure Routing Protocol

3.1. Assumption

In this section, we have discussed about the network assumptions, which we have made while implementing this protocol:

- Each node has a unique identifier (IP address, MAC address or certificate serial number).
- The wireless communication links between the nodes are symmetric; that is, if node N_i is in the transmission range of node N_j, then N_j is also in the transmission range of N_i. This is typically the case with most 802.11 [2] compliant network interfaces.
- The link-layer of the MANET nodes provide transmission error detection service. This is a common feature of most 802.11 wireless interfaces.
- Any intermediate node on a path from a source to a destination may be malicious and therefore cannot be fully trusted. The source node only trusts a destination node, and vice versa.
- The ideal power of a node is not considered while calculating.

3.2. Data structure

In this section we have discussed about the 4 different kinds of Data structures any node in a network can use, as shown in Table 1.

Data Structure	Description	Fields
Neighbor Table (NBRT)	It is a table with N (the number of neighbors) number of rows.	Node Address, Fidelity, Culprit Record[30], Active Flag.
Routing Table	It is a single entry table, which has the detail of the last successful communication.	Destination Address, Next Hop, Hop Count.
Data Buffer	It is a single entry with the encrypted data for the present destination.	Destination Address, Encrypted Message.
Blacklist	It is a table with rows having details of the nodes, which have been recommended by other neighbor nodes.	Destination Address, Blacklisted Node Address, Times of Report.

Table 1: Data Structure Formats

3.3. Packet Format

In this section we define the various packets which we have used in our protocol and explain the structure for the same as shown in Table 2.

Packet Name	Structure
NREQ	{Source Address}
NREP	{Source Address, Destination Address, Battery Power}
RREQ	{Hop Count, Source Address, Destination Address, Current address, Next Hop Address, Fail Array[]}
RREP	{Hop Count, Message Count, Source Address, Destination Address, Current address, Last Hop Address, Digital Signature}
Fail Message	{Source Address, Destination Address, Fail Array[],Digital signature}
Data	{ Hop Count, Source Address, Destination Address, Current address, Encrypted Message[]}
ACK	{ Hop Count, Source Address, Destination Address, Next Hop Address, Last Hop Address, Digital Signature}
Report	{Source Address, Destination Address, Next Hop Address, Last Hop Address, Culprit, Digital Signature}
Recommendation	{Source Address, Culprit, Digital Signature}

Table 2: Packet Format

3.4. Definitions

3.4.1. Fidelity:

The fidelity of a node is a measure of how much a node (say) A trusts a neighboring node (say) B over another neighboring node (say) C, while transmitting a data packet to its destination. Thus, it is not an absolute concept, but it varies from node to node. Let us assume B has a fidelity value $\boldsymbol{\varphi}_{BA}$ with respect to A, while C has a value $\boldsymbol{\varphi}_{CA}$ with respect to A. If $\boldsymbol{\varphi}_{BA} > \boldsymbol{\varphi}_{CA}$,then the data packet would obviously be forwarded through node B to the destination. This process is repeated in the case of each intermediate node within the routing path until it reaches the destination. Fidelity can be represented by Equation 1.

$$\varphi = f(Acknowledgement, Recommendation, Report) \qquad (1)$$

- Whenever a neighbor, Q, of a node, P, transmits a data packet, successfully that it received from P, the fidelity value, φ, of Q is increased by P; as shown in Equation 2.

$$\varphi \textbf{ := } \varphi \textbf{ + } \textbf{\textit{1}} \text{ on receiving } \textbf{\textit{ACK}}\textbf{(P,Q)} \quad (2)$$

- If a node P, does not receive any acknowledgement for a data packet that it sent to its neighbor Q, the fidelity value, φ, of that neighbor is decreased by 1; as shown in Equation 3.

$$\varphi \textbf{ := } \varphi \textbf{ – } \textbf{\textit{1}} \text{ on not receiving } \textbf{\textit{ACK}}\textbf{(P,Q)} \qquad (3)$$

- If a node, P, receives Report about some other node R in the data path, from its neighbor, Q, the fidelity value, φ, of its neighbor R, is decreased by 1; as shown in Equation 4.

φ ***:=*** φ -1 on receiving ***Report*(P, Q, R)** (4)

- Whenever a node, P, receives a recommendation, (bad report) from its neighbors says, A, about another neighbor, Q, the node decreases the fidelity, φ, of the recommended neighbor by 1; as shown in Equation 5.

  φ ***:=*** φ - ***1*** on receiving ***Recommendation(P, A, Q)*** (5)

  If Node P receives recommendations from *I* different neighbor nodes, about node Q, then node Q is blacklisted; where *I* is the Network Improvement Factor, as defined in Section 3.4.3.

Relying on a node too much for transmission of the same packet can sometimes prove to be futile, since the battery power of every node is finite. This continuous transmission through a single node may drain energy of a node such that it will be unable to send any more packets in the future. Moreover, a node with high fidelity, if suddenly behaves maliciously will lead to wastage of many packets; since this node will be selected over other non-malicious nodes due to its high fidelity value for a number of iterations. Thus fidelity needs to have a range as shown in Equation 6.

$$min\left\{-I, -\left\lfloor\frac{X}{D+Y}\right\rfloor\right\} \leq \boldsymbol{\varphi} \leq \left\lfloor\frac{X-Y}{D+A}\right\rfloor \quad (6)$$

Where,

X: The initial battery power of a node in mAh.

Y: The total power consumed for receiving & forwarding protocol packets, in mAh, i.e., packets like NREQ, NREP, RREQ, RREP; as shown in Equation 7.

P_1= Total power consumed for receiving & forwarding NREQ.
P_2= Total power consumed for receiving & forwarding NREP.
P_3= Total power consumed for receiving & forwarding RREQ.
P_4= Total power consumed for receiving & forwarding RREP.

$$Y = P_1 + P_2 + P_3 + P_4 \quad (7)$$

D: The total power consumed for each data packet in mAh; as shown in Equation 8.

P_5= Total power consumed for forwarding Data packet.
P_6= Total power consumed for encrypting Data packet.

$$D = P_5 + P_6 \quad (8)$$

A: The total power for acknowledgement packet in mAh; as shown in Equation 9.

P_7= Total power consumed for receiving ACK packet.
P_8= Total power consumed for decrypting ACK packet.

$$A = P_7 + P_8 \quad (9)$$

Theorem 1: The fidelity will not be decreased further if the fidelity is $\varphi_{min} = min\{-I, -\left\lfloor\frac{X}{D+Y}\right\rfloor\}$

Proof 1: Let a node say, *A*, with only one neighbor say, *B*, which is a malicious node. The node B drops all the ACK packets coming in from the destination node. Initially, $\varphi_{BA} = 0$. With each drop in ACK packet the fidelity will decrease by one. Hence, the number of failed transmissions will signal a drop in fidelity. Node A can decrease the fidelity of node B, as long as it has battery power to transmit data.

$$\varphi_{min} = \left\lfloor \frac{X_A}{D + Y} \right\rfloor \quad (10)$$

Moreover, if node *A* has *I* new neighbors, say $\{N_1, N_2 \ldots . N_I\}$, then node A can receive *I* unique recommendations about node B; hence decreasing the fidelity by *I* time, and later blacklisting node B. Combining, both the scenarios, the minimum value is considered which is represented in Equation 6.

Theorem 2: The fidelity will not be increased further if the fidelity is $\varphi_{max} = \left\lfloor \frac{X-Y}{D+A} \right\rfloor$

Proof 2: Let there be a node say, *A*, which is the only intermediate node between the source say, *S*, and destination say, D, which receives and forwards the packets, and there are no malicious nodes.

The battery power for node *A*, will drain for each data reception & transmission till it is exhausted. After, the initial connection establishment, i.e., the route discovery process; the remaining battery power will be X-Y. The battery power consumed for each subsequent data transfer, will be D+A. Thus the maximum no. of times the data transfer will occur, safely without loss in packets will be:

$$\varphi_{max} = \left\lfloor \frac{X_A - Y}{D + A} \right\rfloor \quad (11)$$

Hence, there is a direct co-relation between the battery power and the maximum fidelity.

3.4.2. Busy Wait:

Busy wait is the period through which a node will wait for the reply of a protocol packet from another neighbor node. During this period, a node will not communicate with any other node, unless there is a reply from that neighbor node or the time has elapsed. This busy time varies for packet types. The Hop count and Message Count are used to calculate this busy time.

Hop Count is the measure of the number of hops a packet has gone through from the originator node. All packets measure the hop count. It is incremented by one for all forward hops and decremented by one for all backward hops. Hence, packets like NREQ, RREQ and Data increment the hop count and packets like NREP, RREP and ACK decrement the hop count.

Message Count is the measure of the total hops required for the source node to send data to the destination node. All the nodes in the data path get this value through the RREP packet.

$$Busy\ Wait = f(\boldsymbol{\tau_1}, \boldsymbol{\tau_2}, \boldsymbol{\tau_3}, \boldsymbol{\tau_4}) \quad (12)$$

Theorem 3: A node will broadcast a NREQ packet, and wait for $\boldsymbol{\tau_1}$ for NREP packet to arrive.

$$\boldsymbol{\tau_1} = 2 * (Average\ Delay) \quad (13)$$

Proof: A node N_1 sends a neighbor request packet to node N_2 (neighbor node). The packet will propagate from node N_1 to node N_2 and a NREP will come from node N_2 to node N_1 in reply. The propagation time is the Average Delay. Hence, Equation 13.

Theorem 4: A node will send a RREQ packet, and wait for $\boldsymbol{\tau_2}$ for RREP packet to arrive.

$$\boldsymbol{\tau_2} = 2 * (Average\ delay) * (Network\ Diameter) \quad (14)$$

Proof: A node N_1 sends a route request packet to node N_2 (neighbor node). If node N_2 is the destination node, it will reply; else, it will forward the RREQ packet to another neighbor node is search of the destination node. The packet will propagate from node N_1 to node N_2 and continue up till N_i. Hence, the Network Diameter will be the maximum number of hops possible in the network. Let, N be the total number of nodes in the network.

$$Network\ Diameter = (N\text{-}1) \quad (15)$$

The node N_1 will maximum wait for the RREP packet to arrive. Hence, Equation 14.

Theorem 5: A node will send back a RREP packet to the last hop and wait for $\boldsymbol{\tau_3}$ for Data packet to arrive.

$$\boldsymbol{\tau_3} = 2 * (Average\ delay) * (Hop\ Count) \quad (16)$$

Proof: A node *B* sends the RREP packet to node *A* (last seen hop). The packet will propagate from node B to node A trailing the same path through which the RREQ packet came. Node A after verifying the RREP packet will encrypt the data packet and forward it through the same path. The time to encrypt the data packet considered negligible. The propagation time is the Average Delay

Theorem 6: A node will send Data packet to the next hop, and wait for $\boldsymbol{\tau_4}$ for the ACK packet to arrive.

$\boldsymbol{\tau_4}$=2*(Average delay)*(Message Count-Hop Count) (17)

Proof: A node *A* sends data packet to node *B* (next hop). Message count is the total number of hops from source to destination, while hop count is the number of hops the particular packet has gone through. If Node B is the destination node, it will reply with the ACK packet, i.e, Message Count- Hop Count is 1; else it will forward the data to the next hop which replied to the RREP for that Source-Destination pair. The propagation time is the Average Delay.

If a node does not receive an ACK packet from the node, it will reply with a Report packet, which is a negative acknowledgement.

This busy wait allows mitigating attacks like rushing and wormhole. Moreover, this helps in finding out the maximum time required to detect a malicious node in the path.

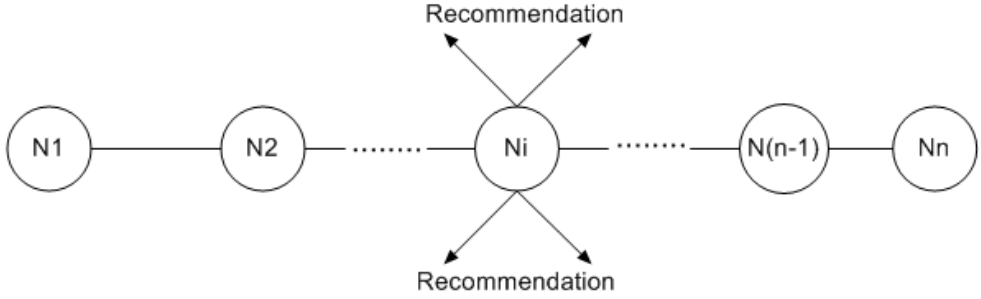

Figure 2: A linear Node placement

Let, there are N nodes placed in a pattern as shown in Figure 2. Let (i+1)th node be a malicious node. The maximum time required by ith node to detect (i+1)th node is likely a malicious node is shown in Equation 18-20. Where, hop count at any instance is i; message count is m; and the average delay is δ. Assuming the Neighbor searching is periodical and has already been done; and the drop in packets is only due to malicious nodes within the path. After the node i has detected a malicious node, it broadcasts a recommendation packet to its neighbor and a report to the last seen address.

$$\theta_i = \boldsymbol{\tau_2} + \boldsymbol{\tau_3} + \boldsymbol{\tau_4} \quad (18)$$

$$\theta_i = 2 * \delta * (N-1) + 2 * \delta * i + 2 * \delta * (\mathrm{m} - \mathrm{i}) \quad (19)$$

$$\theta_i = 2 * \delta * (N + m - 1) \quad (20)$$

In worst case the nth node will be the destination node; hence, the message count will be at most (N-1). Hence, the Equation 21.

$$\theta_i = 4 * \delta * (N-1) \quad (21)$$

3.4.3. Network Improvement Factor

This secure routing protocol not only aims at detecting malicious nodes, but also eliminating these nodes from the network as soon as possible. The Network Improvement Factor (I) is the factor used to calculate the total time required to eliminate a malicious node from the network, shown in Equation 22.

To eliminate a node, i.e., to blacklist a node, it needs to get unique recommendations from I different neighbors. A neighbor node generates a recommendation only after it detects a malicious node in its path, as shown in Equation 18. The more/lesser the value I, the more/lesser time will it take to eliminate a malicious node. Therefore, the optimal value of I needs to be calculated, which is dependent on the

network. A network with low packet delivery fraction will try to increase the probability of successful packet transfer, by choosing a larger I. While, a network with end-to-end delay as a concern, will choose a lower I.

$$\Upsilon = \sum_{1}^{I} \theta_i \quad (22)$$

The protocol maintains a count value to blacklist a node. We took several snapshots of the network, to deduce an average case. The count increases only after a node receives a recommendation and the recommended node is its neighbor. As the count increases, the fidelity decreases, thereby decreasing the chance of that node to be reselected. Table 3 defines the time required by the network to eliminate a malicious node after it commences with its malicious behavior.

I Nodes	1	3	5	7	9
10	125	425	650	875	1025
20	500	2125	3300	4600	5150
30	700	2750	4050	5800	6300

Table 3: Time required (in ms) for different values of I and number of nodes

Since our protocol's aim is security, and we would like to isolate the malicious node as fast as possible, we consider the minimum I value, since 1 and 2 will be too fast and harsh for the network, we choose I=3.

3.4.4. Choice of Cryptographic Tools:

The protocol uses public-key cryptography and digital signature in order to counter the malicious effects of the network nodes. In public key cryptography, the public key of the destination node is used to encrypt the data [8,10]; and the private key of the destination node to decrypt it [6]. This ensures that no intermediate node can tamper with the data in its path from source to destination. The protocol uses RSA public-key encryption or decryption algorithm.

In digital signature [10], the sending and receiving entities follow a two-fold process, by signing the digest (hash value). First, the hash of the data to be sent is calculated using a hashing algorithm. Next, this hashed message is signed (encrypted) using the private key of the sender, and attached to the message as the digital signature. The receiver calculates the hash of the original content of the packet and decrypts the encrypted hash of the message using the public key of the sender. If the hash of the received message and the decrypted hash from the digital signature are the same, then the contents of the message are correct. This provides a means of non-repudiation of the message, i.e. the sender cannot deny sending the data, and message integrity. This scheme also ensures message authentication. The protocol uses the RSA algorithm for asymmetric key cryptography and the SHA-1[5] hashing algorithm for calculating hash of the data. Some typical functions used in the algorithm are shown in Table 4.

NAME OF FUNCTION	RETURN TYPE	DESCRIPTION
loadKey(GlomoNode *node, const GlomoNodeInput * nodeInput)	Bool	Loads the public key of the sender from the key file, returns true if successfully loaded, false otherwise.
executeAttackVector(GlomoNode *node, Message *msg)	Int	Executes the malicious node activity as defined in the attack vector file. Returns 0 on successful completion, 1 otherwise.
RSA_public_encrypt(strlen(msg),(unsigned char*)msg,encrypted,rsa, RSA_PCKS1_PADDING)	Int	Encrypts the data using the public key
VerifySignature(&(pkt),pkt->sign)	Int	Verifies the digital Signature and hash of data.

Table 4: Some typical cryptographic functions used in the algorithm.

We have used self-organized key management to make the key management lighter. Although, there are several self-organized key management protocols [3,7], most of them take up huge time and space to collect and maintain the certificates. To circumvent this problem, we have proposed a key management scheme named Self-Organized key management based on fidelity relationship list and dynamic path which is efficient and takes up much lesser space to store the collected certificates. In this scheme we, as explained in [11], we use a fidelity relationship list based on which certificates are collected by source nodes for a path (selected previously on the basis of a fidelity parameter in the network). This scheme creates fidelity lists only consisting of paths for as long as they are active, hence the memory usage is reduced. A list containing pairs of nodes in tuples of (verified->verifier) form is build based on which node directly verifies the identity of another node.

4. MFBOD Model

In this section we have discussed our fidelity model along with the routing details of our protocol, referring to Figure 3.

Initially, a node before moving onto the Fidelity Judgment stage has to search for its neighbors by broadcasting the neighbor requests (NREQ). It then waits for the neighbor reply (NREP) to arrive, as shown in Figure 4 (a & b). If the destination node is its immediate neighbor, then the route request is sent directly, otherwise the Fidelity Judgment starts.

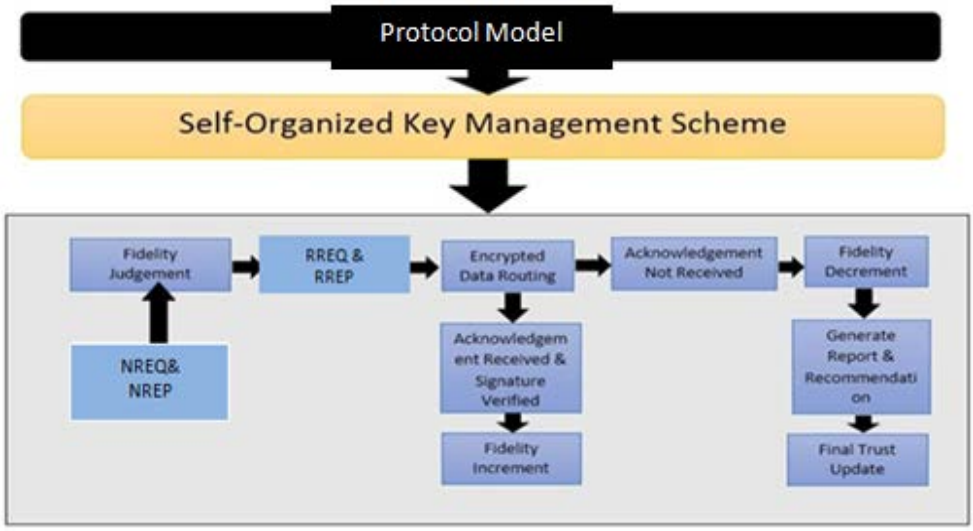

Figure 3: Modified Fidelity Based On-Demand Model

In the next stage, the most trustworthy node is selected from the neighbor table, which is concluded from the fidelity values. Since, a node with higher fidelity indicates that it is a reliable node in the network and it has transmitted packets more dutifully than other nodes. In Figure 4 (c), with respect to node A, the fidelity of B is more than that of C, and hence B is more trustworthy than C.

After the trustworthy node is selected, the corresponding node is sent the route request (RREQ) and the originator node waits for a time interval $\boldsymbol{\tau_1}$, if the Destination is just next to the Source ,i.e, the hop count between source and destination is 1 or $\boldsymbol{\tau_2}$ otherwise, for the route reply (RREP) to arrive. After timeout it will move onto selecting the next available node from the neighbor table. This node in turn repeats this process till it gets the destination node in its neighbor table or the time to live is over. In Figure 4 (d), Node A waits for RREP to arrive, and in the meanwhile node B sends NREQ to its neighbors, and the process is repeated, till destination is reached.

After the destination is discovered, the destination sends the route reply back to the last hop, waits for $\boldsymbol{\tau_3}$ for data to arrive. Once the Source Node gets the route reply and verifies it; it prepares the data to be sent through the same path through which the route reply came back. The Source node encrypts the data with public key of the destination node and forwards it to the Next Hop. The Intermediate node just keeps on forwarding the data packet to the next till it reaches the destination. Each node in this communication waits for a time interval $\boldsymbol{\tau_4}$ for the ACK packet to arrive.

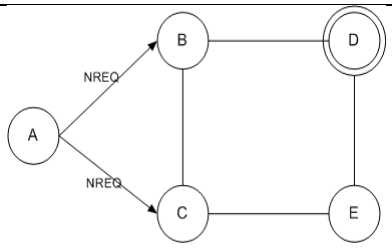

Figure 4(a)

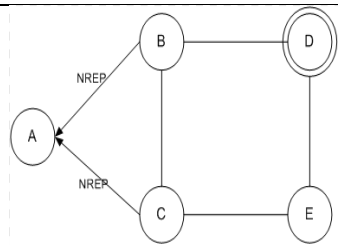

Figure 4(b)

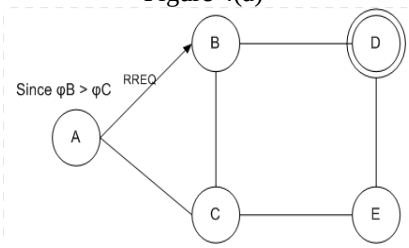

Figure 4(c)

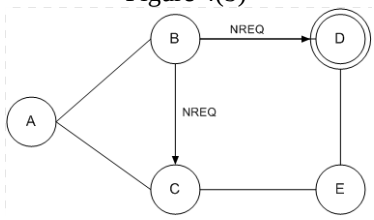

Figure 4(d)

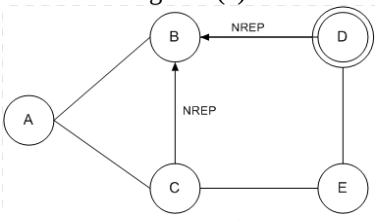

Figure 4(e)

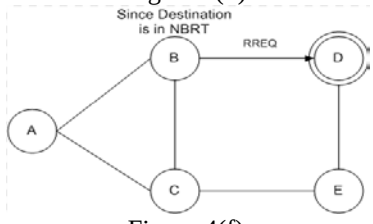

Figure 4(f)

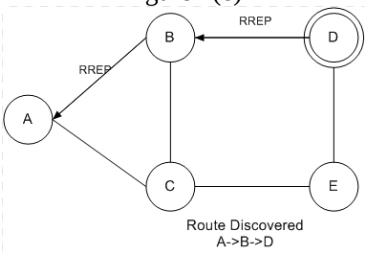

Figure 4(g)

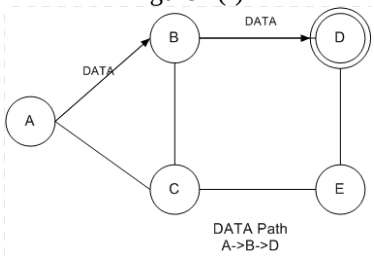

Figure 4(h)

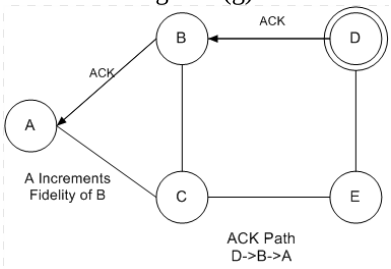

Figure 4(i)

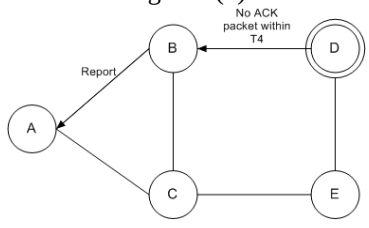

Figure 4(j)

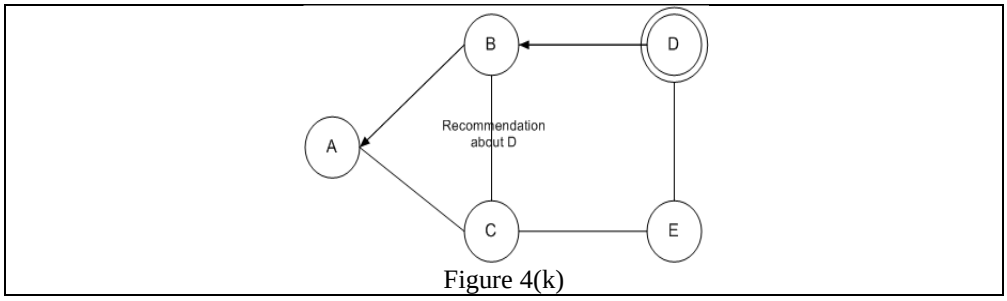

Figure 4(k)

Figure 4 (a) to (k): Route Discovery & Data Flow for Sender A to Destination D

The Acknowledgement(ACK) is generated only by the Destination Node, as shown in Figure 4 (i), which is signed, so that the source node can be sure that the destination has received the data packet, through the same Routing Path in a secured manner. The Intermediate nodes keeps on forwarding the acknowledgement packet, and along with it increments the Fidelity. If this packet is not received by the Intermediate nodes within time τ_3 it will assume that the communication is unsuccessful and decrement its Fidelity, then sends back a Report packet instead and broadcasts a Recommendation packet.

An intermediate on receiving a Report decreases the fidelity of the node sending the Report; then it generates a Report and sends it back to the last seen address, which continues till the source node, as shown in Figure 4 (j). Only the node generating the first Report will broadcast the recommendation packet, as shown in Figure 4 (k). On receiving Recommendation any node will first decrement the fidelity of the recommended node by 1, and when 3 such recommendations arrive against the same node from 3 different nodes then that node is Blacklisted and not used for further communication.

5. Algorithm and Flowcharts

In this section we present the Algorithm along with its Flowcharts. We divide the algorithm into different sections with respect to the Sender, Intermediate and Destination node.

5.1. Sender Node

The sender node selects a node and routes data as shown in Figure 5 and explained as follows:

Step 1: Start.
Step 2: Send NREQ and Wait for τ_1.
Step 3: Check whether NREP is received or not. If yes then repeat Step 3.1 & 3.2 for time T
Step 3.1 If a node is an existing Neighbor in the Neighbor Table then

Step 3.1.1:Update the entries in the neighbor table
Step 3.2:Else
Step 3.2.1 Enter the new neighbor in the table with Fidelity initialized to 0.
Step 3.3 : Check If the destination node is in the neighbor table(NBRT) then
Step 3.3.1: Send RREQ to that Node and wait for time τ_1
Step 3.3.2: If RREP is received and verified, then Goto Step 5
Step 3.3.3: Else, Goto Step 9
Step 3.4: Else
Step 3.4.1: Find the Maximum_Fidelity φ from the remaining nodes.(If there are no nodes in the neighbor table then Maximum_Fidility=NULL).
Step 3.4.2: If no node selected,i.e, φ=NULL, then Go to Step 2
Step 3.4.3: Select the nodes with Fidelity equal to the Maximum_Fidelity.
Step 3.4.4: If there are more than one nodes
Step 3.4.4.1: Select the node with maximum battery power.
Step 3.4.4.2: In case of tie select Random Node.
Step 3.4.5: Send the RREQ to the selected Node and wait for τ_2.
Step 3.4.6: If RREP not received then
Step 3.4.6.1: Add the node in the fail Array.
Step 3.4.6.2: Update the Fail Array from the Fail message if received and verified. These Node/s will not be considered.
Step 3.4.6.4: Goto Step 3.4.1
Step 3.4.7: Else, Goto Step 5
Step 4: Else
Step 4.1: Go to Step 2.
Step 5: Encrypt Data and wait For ACK for time τ_4 .
Step 6: Check whether ACK or Report received or not. If yes,
Step 6.1: Increment the fidelity by 1,only if the Maximum fidelity is yet to be reached and the node isn't destination node
Step 7: Else
Step 7.1: Decrement the fidelity by 1,only if the Minimum fidelity is yet to be reached and the node isn't the destination node.
Step 7.2: If last_seen_address was destination node, then Goto 9
Step 7.3:Else, Goto 3.4.1
Step 8: If Source wants to send more data, then
Step 8.1: If Same Node, then Goto Step 5
Step 8.2: Else, Goto 2
Step 9: Stop

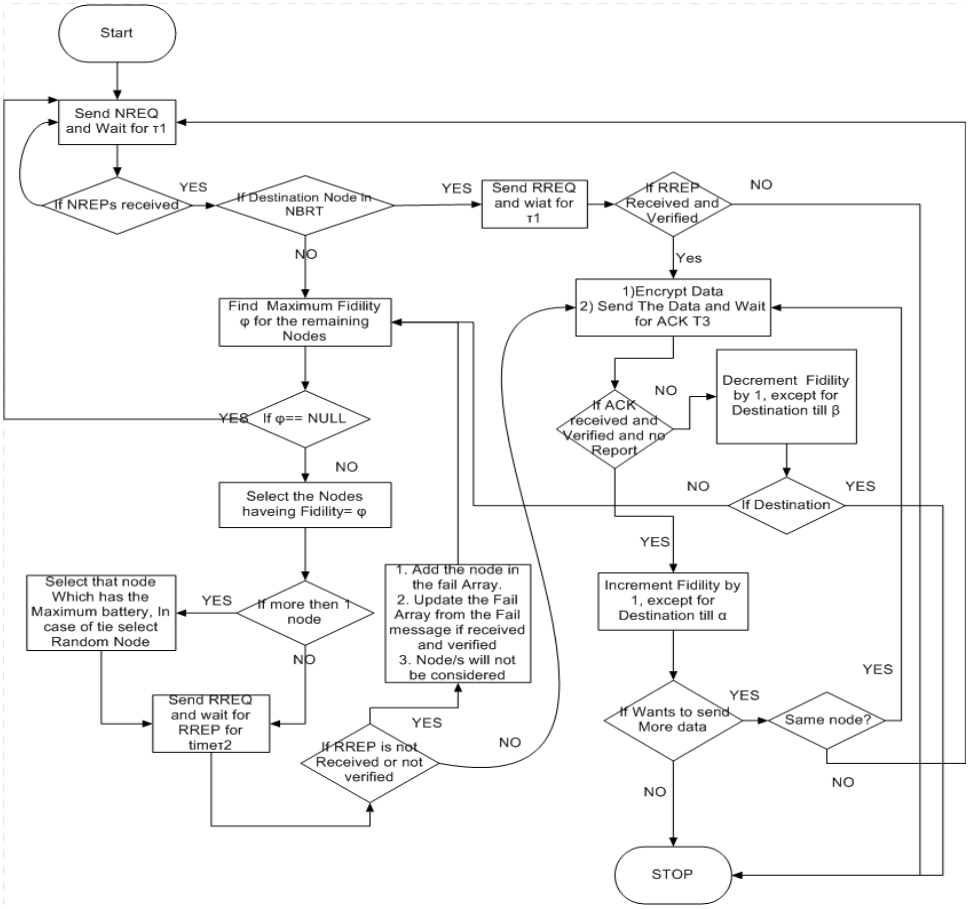

Figure 5: Flowchart for data routing in sender

5.2. Intermediate Node

An Intermediate node can receive all kinds of packets like NREQ, RREQ and Data Packet, as shown in Figure 6.

Section A:

The section A represents the part of sending NREP to the last seen address, as shown in Figure 10.

Step 1: Start.

Step 2: Send NREP to Last Seen Address (sending node).

Step 3: Stop.

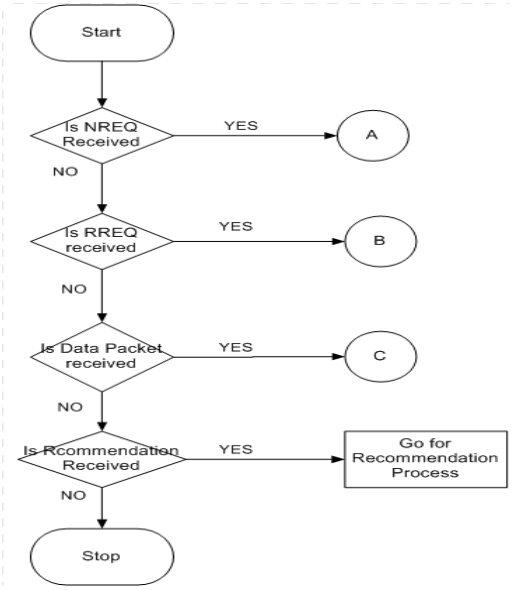

Figure 6: Flowchart for intermediate node Section A

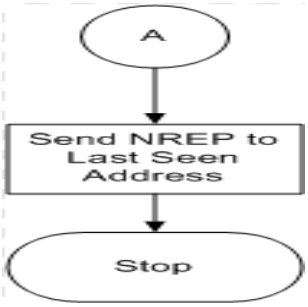

Figure 7: Flowchart for intermediate node Section B

The section B represents the part of sending RREP to the last seen address, as shown in Figure 11.

Step 1: Start.

Step 2: Send NREQ and Wait for τ_1.

Step 3: Check whether NREP is received or not. If yes, then repeat Step 3.1 & 3.2 for time T

Step 3.1 If a node is an existing Neighbor in the Neighbor Table then

Step 3.1.1:Update the entries in the neighbor table

Step 3.2: Else

Step 3.2.1 Enter the new neighbor in the table with Fidelity initialized to 0.

Step 3.3 : Check If the destination node is in the neighbor table then

Step 3.3.1: Send RREQ to that Node an wait for time τ_1.

Step 3.3.2: If RREP is received then

Step 3.3.2.1: Save the Address and send the RREP to the last seen address, Goto Step 5.

Step 3.3.3: Else, Goto Step 4.

Step 3.4: Else

Step 3.4.1: Find the Maximum_Fidelity φ from the remaining nodes.(If there are no nodes in the neighbor table then Maximum_Fidility=NULL).

Step 3.4.2: If no node selected, i.e , φ =NULL, then Goto Step 4

Step 3.4.3: Select the nodes with Fidelity equal to the Maximum_Fidelity.

Step 3.4.4: If there is more than one nodes then

Step 3.4.5.1: Select the node with maximum battery power, in case of tie select any random node

Step 3.4.6: Send the RREQ to the selected Node and wait for τ_2.

Step 3.4.7: If RREP is received, then Goto Step 5
Step 3.4.8: Else
Step 3.4.8.1: Add the node in the fail Array.
Step 3.4.8.2: Update the Fail Array from the Fail message if received and verified
Step 3.4.8.3: Node/s will not be considered.
Step 3.4.8.4: Go to Step 3.4.2
Step 4: Send Fail Message to the last seen address
Step 5: Stop

Section C:
The section C represents the part of sending ACK to the last seen address, as shown in Figure 8.
Step 1: Start.
Step 2: Select Next Hop which is received from the Route Reply Packet.
Step 3: Forward the Data and wait for ACK for τ_3 =2*(HOP_COUNT)*(AVG_DELAY)
Step 4: Check whether ACK received or not. If yes,
Step 4.1: Forward the ACK to Last Seen Address.
Step 4.2: Increment the fidelity by 1, if the Maximum fidelity α is yet to be reached except for the destination node.
Step 5: Else,
Step 5.1 : Is Report Received
Step 5.1.1: Forward this Report Back to the last seen address
Step 5.2: Else
Step 5.2.1: Sign and Send Report.
Step 5.2.2: Broadcast Recommendation
Step 5.3: Decrement the fidelity by 1 till the Minimum fidelity β is yet to be reached except for the destination node.
Step 7: Stop.

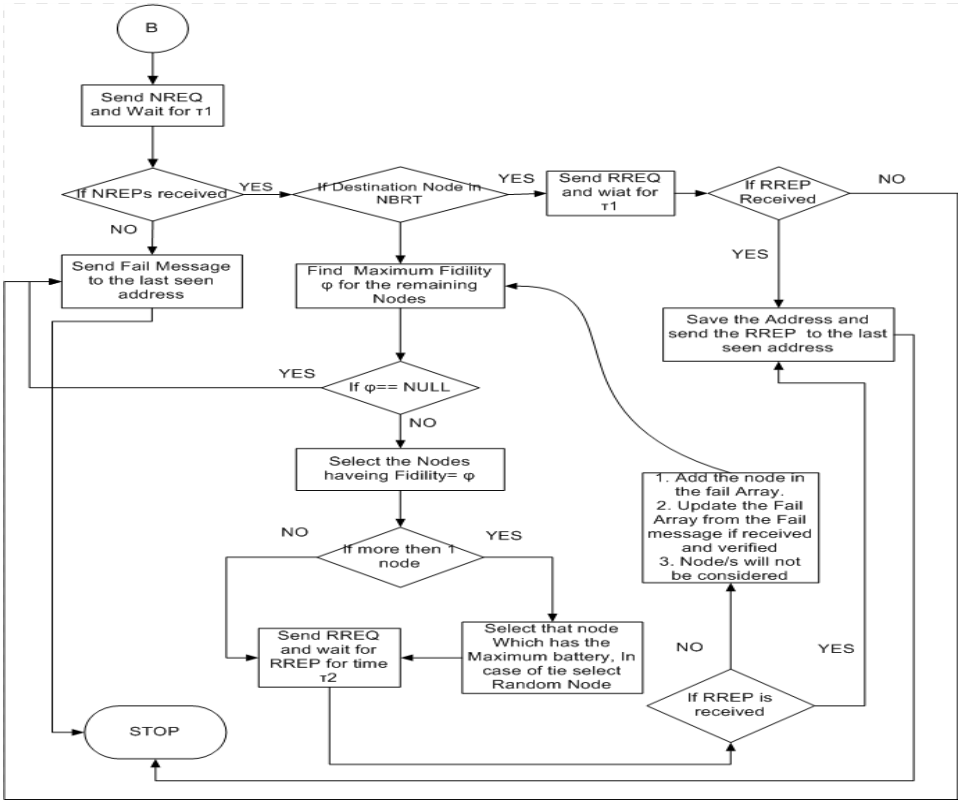

Figure 8: Flowchart for intermediate node-Section B

5.3. Destination Node

The destination node on receiving packets will perform steps as shown in Figure 13.

Step 1: Start

Step 2: Is NREQ received

Step 2.1: Send NREP back to the last seen address

Step 2.2: Goto Step 2

Step3: Is RREQ received

Step 3.1:The node Signs the RREP packet with Source's Public Key

Step 3.2: Send and Goto Step 2

Step 4: Is Data Packet received and verified

Step 4.1: Decrypt the Data Packet with its own Private Key

Step 4.2: Sign the ACK and send

Step 4.3: Go to Step 6

Step 6: Stop

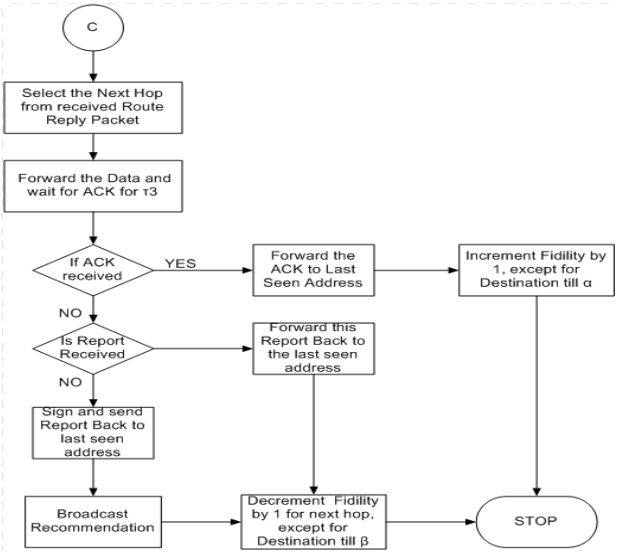

Figure 9: Flowchart for intermediate node- Section C

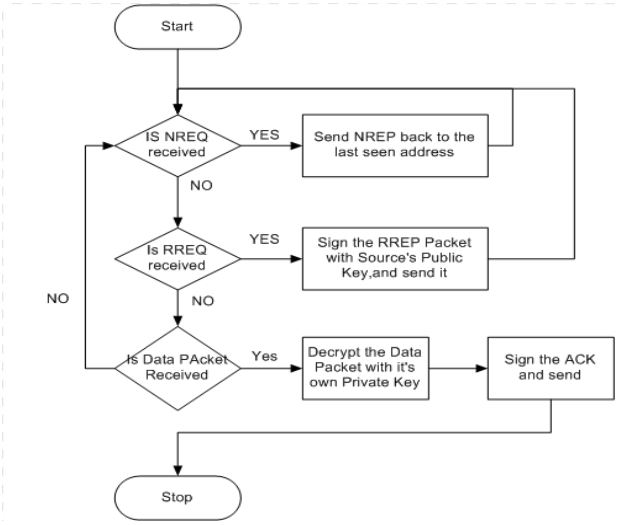

Figure 10: Flowchart for destination node

5.4. Recommendation Process

This process is initiated by any node whether it is a source or intermediate or destination node whenever it receives a Recommendation packet form its neighboring node. The flowchart for a node on receiving recommendation is represented in the Figure 14.

A. Sending node–

Step 1. Node puts the address of the neighbor from which it was supposed to receive acknowledgment (ACK) as the Culprit in the Recommendation.
Step 2. The node broadcasts Recommendation to all its neighbors.

B. Receiving node–

Step 1: Initialize the count as 0
Step 2: Check for Recommendation
Step 3: Is recommendation verified, if no
    Step 3.1: Goto Step 9
Step 4: Is the node in the culprit array is its neighbor table, if no
    Step 4.1: Goto Step 9
Step 5: Has it already been recommended by the same sender, if yes
    Step 5.1: Goto Step 9
Step 6: Else
    Step 6.1: Add the address in the blacklist and increment the counter by 1
    Step 6.2 : Decrement the Fidelity of that node by 1.
Step 7: Is counter >= 3 if no, then Goto Step 2
Step 8: Else
    Step 8.1: The recommended node is removed from NBRT
Step 9: Discard Report and Do Nothing
Step 10: Stop

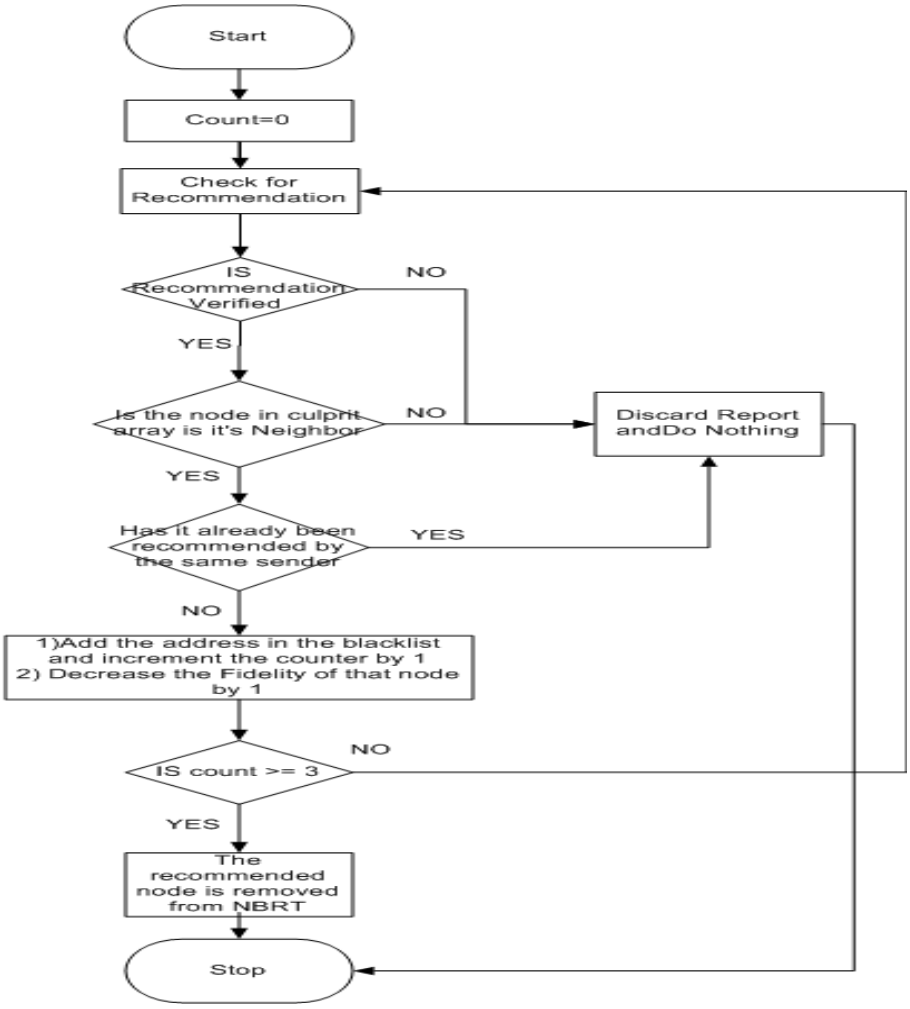

Figure 11: Flowchart for recommendation system

6. Result and Discussion

6.1. Performance Matrices

The proposed protocol can be compared with the existing security-based routing algorithms on the account of different performance metrics.

6.1.1 Packet Delivery Fraction (PDF): This is the output of total number of received data packets divided by the total number of sent data packets, as shown in Equation 23.

$$\text{PDF} = \frac{Number\ of\ Received\ Data\ Packets}{Number\ of\ sent\ Data\ Packets} \quad (23)$$

6.1.2 Normalized Routing Load (NRL): This is the total number of routing packets sent divided by the total number of data packets received. This accounts for the overhead of the routing protocols. The number of total routing packets includes the number of route request packets (RREQ), route reply packets (RREP), acknowledgement packets, etc., as shown in Equation 24.

$$\text{NRL}=\frac{Number\ of\ Sent\ Routing\ Packets}{Number\ of\ Received\ Data\ Packets} \quad (24)$$

This metric gives an estimation of how efficient a routing protocol is, since the number of routing packets sent per data packet gives an idea of how well the protocol keeps the routing information updated. The higher the NRL metric is, the higher the overhead of routing packets and consequently the lower the efficiency of the protocol.

6.1.3 End to End Delay: This is the average time involved in delivery of data packets from the source node to the destination node. To compute the average end-to-end delay, add every delay for each successful data packet delivery and divide that sum by the number of successfully received data packets , as shown in Equation 25.

$$\text{Average End to End Delay}= \frac{\sum Time\ Recived-Time\ Sent}{Total\ Data\ Packets\ receiveds} \quad (25)$$

6.1.1. Simulation Environment

Global Mobile Information System Simulator (GloMoSim), is a simulator environment which uses parallel discrete-event simulation based on Parsec [1,16]. We consider nodes moving in a 500*500 meter region and we change the number of nodes from 10 to 50, with 20% malicious nodes, with a simulation time of 25sec. The mobility model is the random waypoint model [2], with minimal speed of 1 m/s, and the maximal speed of 10m/s. The pause time is 30s. We have considered a two way propagation model. We neglect over-hearing of peer-to-peer packets. The RTS/CTS option is turned off in the MAC layer. The 100-120 bits headers which will get appended by other layers have been neglected, since it will be same for all MANETs with same standards. The CBR Data packets are of size 20 bytes. The protocol uses an RSA cryptographic algorithm, which uses 512 bits.

6.1.2. Result and Analysis

We have done an extensive study of our protocol by comparing it with the most popular secure routing protocols. We plot consider the PDF,NRL and End-to-End delay as parameters to observe the performance of our protocol.

Traditionally, the shortest path to a destination (in terms of number of hops) is considered to be the best routing path. AODV explicitly seeks shortest paths using the hop count field in the route request/reply packets. ARAN, on the other hand, assumes that the first route discovery packet to reach the destination must have traveled along the best path (i.e., the path with the least congestion). Again TAODV uses a trust concept and have extra packets for recommendation packets, which makes it protocol heavy. MFBOD on the other hand is an upgraded model of FBOD, so as to overcome the problems portrayed in Section 2. Each data point in Figures 15-26, is an average of 10 simulation runs with identical configuration but different randomly generated mobility patterns.

In Figures 12, 16 and 20 we consider the performance parameters in a benign environment with varied node speeds. In Figures 13, 17 and 21 we consider the performance parameters in a malicious environment with varied node speeds. In Figures 14, 18 and 22 we consider the performance parameters in an environment with 20%, malicious nodes with variable number of nodes. In Figures 15, 19 and 23 we consider the performance parameters in an environment with varied malicious activity.

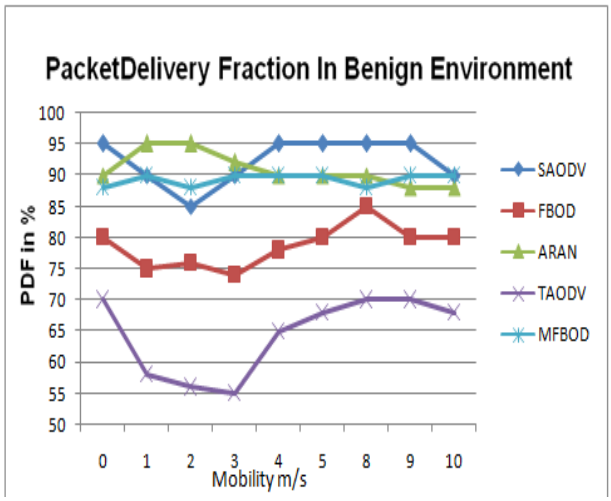

Figure 12: Packet delivery Fraction in Benign Environment with varied mobility

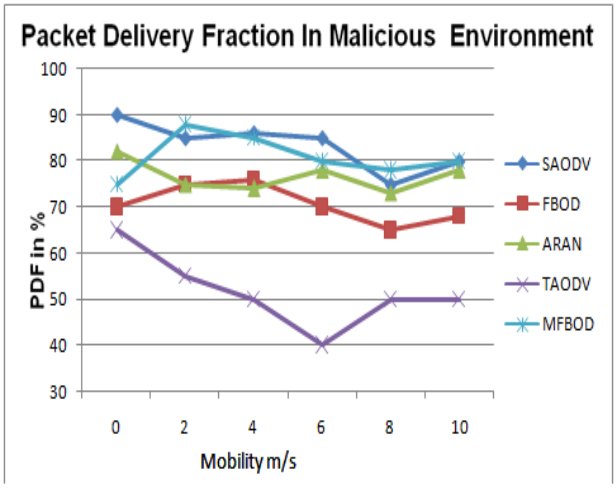

Figure 13: Packet delivery Fraction in Malicious Environment with varied mobility

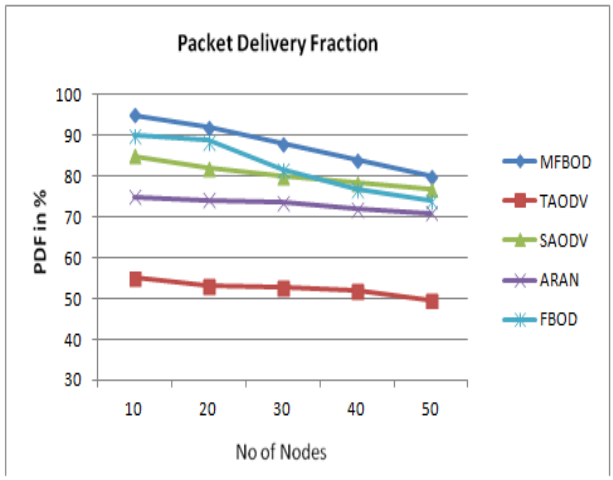

Figure 14: Packet delivery Fraction in Benign Environment with varied number of nodes

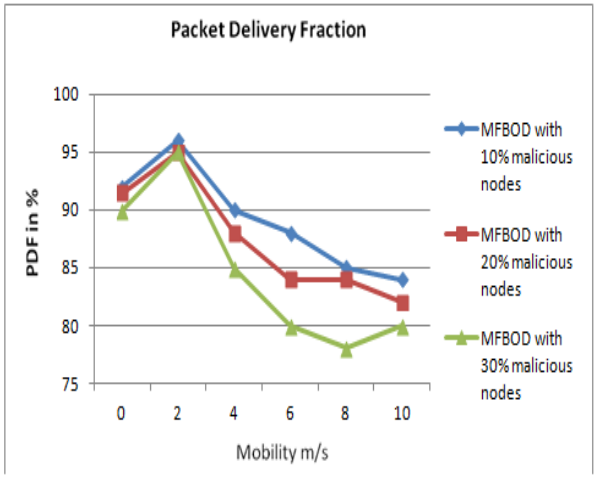

Figure 15: Packet delivery Fraction in Malicious Environment with varied malicious activity

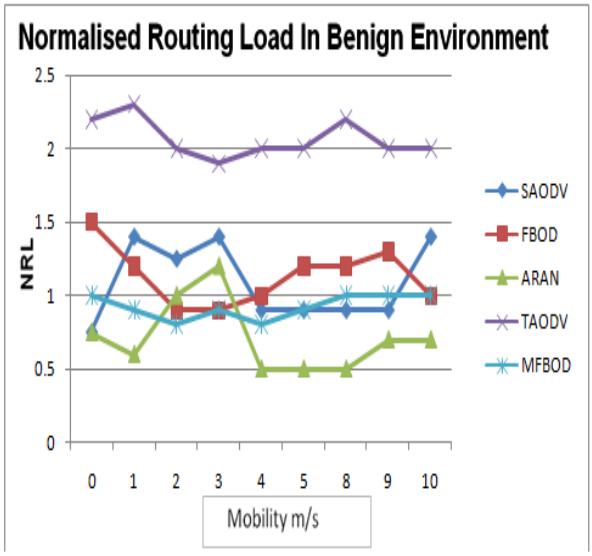

Figure 16: Normalize Routing Load in Benign Environment with varied mobility

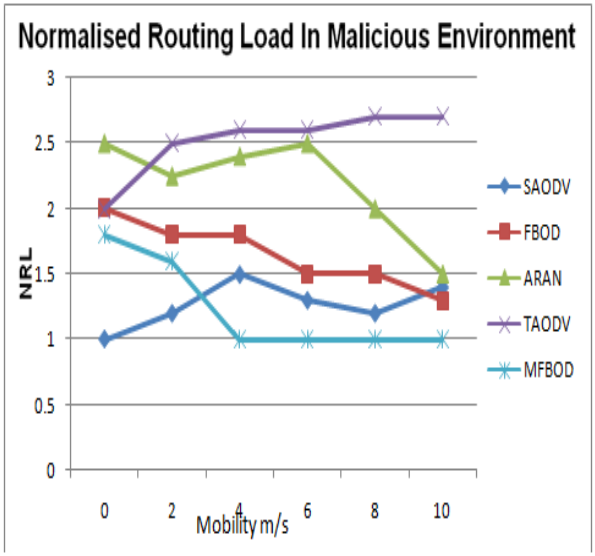

Figure 17: Normalize Routing Load in Malicious Environment with varied mobility

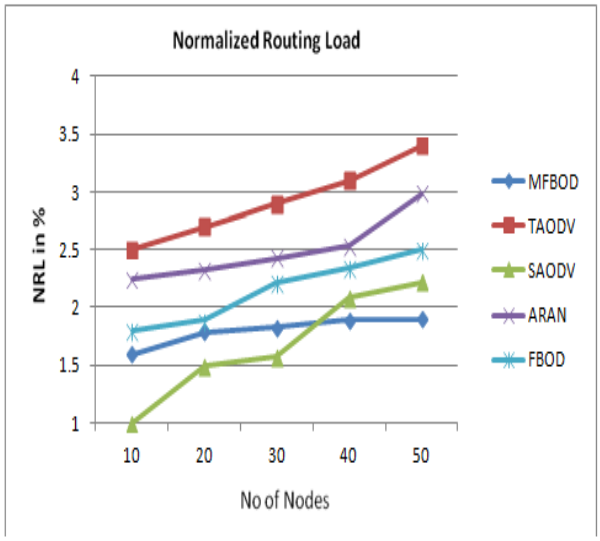

Figure 18: Normalize Routing Load in Benign Environment with varied number of nodes

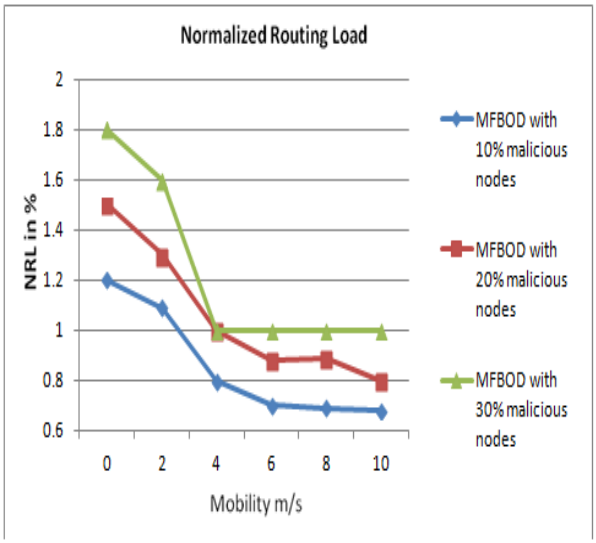

Figure 19: Normalize Routing Load in Malicious Environment with varied malicious activity

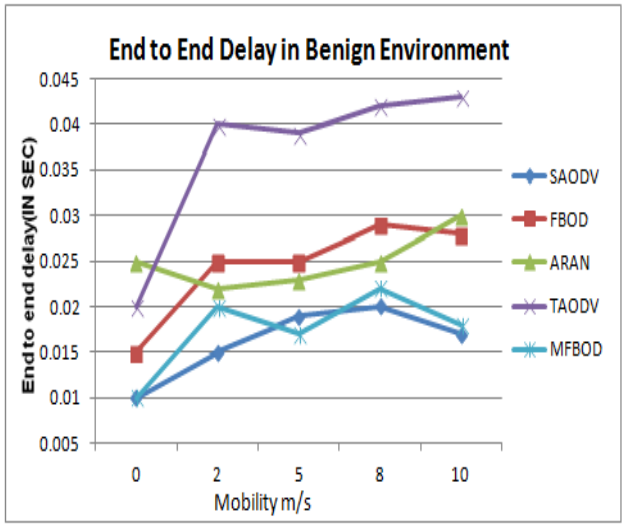

Figure 20: End to End Delay in Malicious Environment with varied mobility

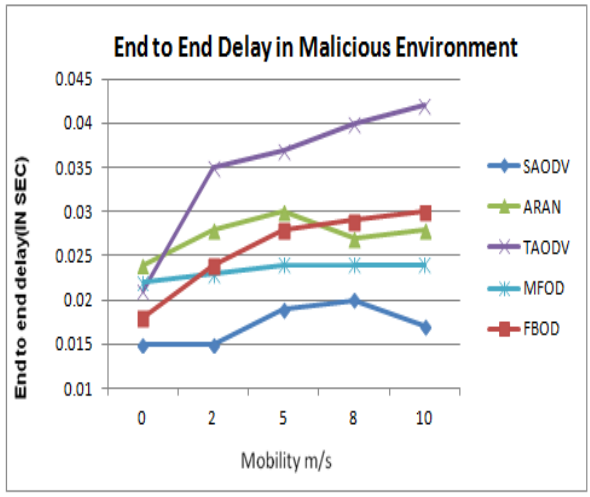

Figure 21: End to End Delay in Malicious Environment with varied mobility

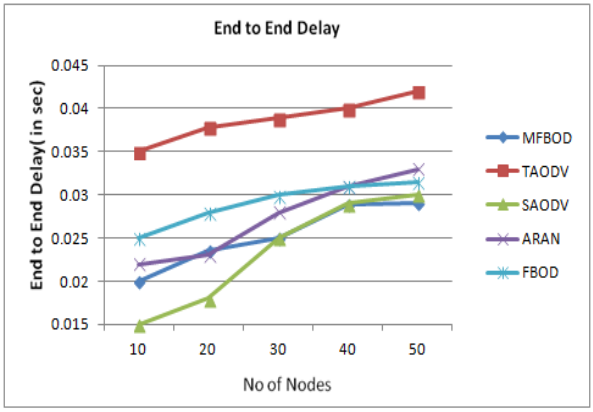

Figure 22: End to End Delay in Malicious Environment with varied number of nodes

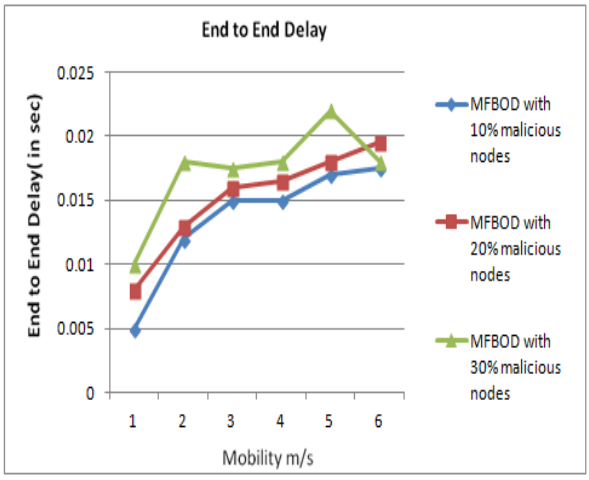

Figure 23: End to End Delay in Malicious Environment with varied malicious activity

7. Conclusion

Our proposed protocol has many unique features which makes it stand different from other existing secure on-demand protocols. Firstly, it is a lightweight protocol and doesn't require any flooding of extra packets or extra memory, as in the case of TAODV and ARAN. Secondly, it is a unicast protocol, thereby making the network prone to many attacks. This dedicated single route is obtained from the fidelity values, hence selecting the most secure node, eventually obtaining a secure path from source to destination. This secure route selection mitigates attacks like wormhole and rushing attack. As the fidelity of other nodes increases the chances of blackhole node getting selected will decrease. Moreover, the count value monitors the greyhole and blackmail attacks quite efficiently. In our protocol, fidelity parameter ensures that only trustworthy nodes are present in

the network. Thus, Sybil attack is reduced to some extent. Nodes wait for a fixed time period for RREP, and ACK packets to arrive, hence jellyfish attacks are reduced. We have used new lightweight packets like NREQ and NREP, to update the neighbor table periodically. The use of the busy flag prevents the cycling of RREQ packets. Again with packets like Report and Recommendation in hand the malicious nodes are effectively and quickly identified and eliminated from the network. Once the malicious nodes are eliminated the NRL decreases to that as in the case of benign environment. We can see from the performance metrics that our protocol works better in a malicious environment than other popular secure routing protocols, with high PDF, low NRL and average End-to-End delay.

8. Reference

[1] Bagrodia R., Meyerr R., *PARSEC: A Parallel Simulation Environment for Complex System*, UCLA technical report, 1997.

[2] Broch, J., Maltz A.D., Johnson B.D., Hu C.Y., Jetcheva J., A performance comparison of multi-hop wireless ad hoc network routing protocols, *Journal of Mobile Computing and Networking*, 1998, 85–97.

[3] Capkun S.,Buttya L., Hubaux J.P., Self-Organized Public-Key Management for Mobile Ad Hoc Networks, *IEEE Transactions On Mobile Computing*, 2, 1, 2003,1-13.

[4] Corson S., Macker J., Mobile Ad hoc Networking (MANET): routing protocol performance issues and evaluation considerations, *Network Working Group*, RFC:2501,1999.

[5] Eastlake D., Jones P., US secure hash algorithm 1 (SHA1). *Network Working Group*.RFC-3174, 2001.

[6] Grobler T. L., Penzhorn W. T., Fast decryption methods for rsa cryptosystem, *In 7th AFRICON Conference in Africa*,1, 2004,361-364.

[7] Huang Q., Cukier J., Kobayashi H., Liu B., Zhang J., Fast authenticated key establishment protocols for self-organizing sensor networks, *Mitsubishi Electric Research Laboratories*,2004.

[8] Lamport L., Password authentication with insecure communication. *Communications of ACM*, 24,11,1981,770–772.

[9] Nekkanti R.K., Lee C.W., Trust based adaptive on demand ad hoc routing protocol. *In: Proceedings Of The 42nd Annual Southeast Regional Conference*, 2004, 88–93.

[10] Rivest R.L., Shamir A., Adleman, L., A method for obtaining digital signatures and public-key cryptosystems, *Communications of ACM*, 21,2, 1978, 120-126.

[11] Saha H.N., Bhattacharyya D., Banerjee B., Mukherjee S., Singh R. and Ghosh D., Self-Organized key management based on fidelity relationship list and dynamic path. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*,3,7,2014,97-100.

[12] Saha H.N, Bhattacharyya D., Banerjee B., Mukherjee S., Singh R., Ghosh D., A review on attacks and secure routing protocols in MANET, *In CIBTech, International Journal of Innovative Research and Review (JIRR)*, 1, 2, 2013,12-31.

[13] Saha H N., Bhattacharyya D., Banerjee P K., Fidelity based on demand secure (FBOD) routing in mobile adhoc network, *International Journal of Advanced*

Computer Science and Applications,Special Issue on Wireless & Mobile Networks(IJACSA), 2011,26-34.

[14] Sanzgiri K., Dahill B., Levine B., Belding-Royer E., A secure routing protocol for ad hoc networks. *In: Proceedings of 10th IEEE International Conference on Network Protocols (ICNP)*, 2002,78-87.

[15] Zapata M., Asokan N., Securing ad hoc routing protocols. *In: Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2002,1-10.

[16] Zeng, X., Bagrodia. R., Gerla M., GloMoSim: a library for parallel simulation of large-scale wireless networks, *In Proc. of 12th Workshop on Parallel and Distributed Simulation (PADS)*,1998,154 - 161.

Received 29.10.2014, accepted 20.06.2015