I. J. Computer Network and Information Security, **2015, 8, 39-48**
Published Online July 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2015.08.05

Hardware Implementation of Fidelity based On Demand Routing Protocol in MANETs

Himadri N. Saha[1], Rohit Singh[2], Debika Bhattacharyya[3]
[1, 2, 3] Department of Computer Science and Engineering, Institute of Engineering & Management, Kolkata, 700091, India
Email: him_shree_2004@yahoo.com, roh9singh@yahoo.in, bdebika@yahoo.com

Abstract—With the advent of real applications in fields like defense and domestic fields, MANET is becoming more popular. MANET does not require any infrastructure, moreover it can behave as mobile networks. These features have boosted up the popularity of MANET in the community. As more and more fields get dependent on MANET, the system needs to be more robust and less expensive. For example, in defense field security is the major issue, while in the domestic field maintaining the QoS is the major issue. To commercialize MANET the routing protocols need to be lightweight, secure and the hardware on which it is to be implemented should be low cost at the same time. We propose a lightweight, secure and efficient routing model for MANETs; which uses fidelity to allocate trust to a neighbor, thereby taking the decision whether to send data via that secure neighbor or not. It also uses new packets like report and recommendation that help the protocol to detect and eliminate the malicious nodes from the network. To observe the results we implement this protocol in hardware, on the Arduino platform in ZigBee network. We observe that our protocol exhibits high packet delivery fraction, low normalize routing load and low end to end delay, compared to existing secure routing protocols. Thereby, maintaining a constant tradeoff between the QoS and the security of the network.

Index Terms—Mobile Ad hoc Networks (MANETs), Secure, Low Cost, Fidelity, Hardware Implementation, Arduino, Zigbee.

I. Introduction

With the advent of wireless networks, the applications of MANETs [1] are wide ranging from search and rescue operations to personal area networks. Such applications are characterized by the lack of communications infrastructure and central authority. While doing so often the quality of service or the security of the data has to be compromised. These properties make MANET quite applicable in many fields [2,3], like in a battlefield, rescue operations and personal area networks.

There are some simulations which have been done taking into consideration the real life applications of MANETs. Sharmila et.al [4] have done a hardware implementation of secure AODV, and have proposed a novel technique by using a Virtex IV device from Xilinx family. The delay and power consumption has been compared with AODV in the hardware platform. Dalu et.al [5] have proposed a physical implementation of a topology control algorithm for MANETs. The proposed algorithm maintains the topology without any control message. Passarella and Delmastro [6] have proposed a real implementation of Group Communication Applications. They present a prototype implementation of a Whiteboard Application. The prototype includes the networking support required by the Whiteboard, and thus can be used to test it in a real test bed.

We have used the Arduino board because of its scalability, reliability and easy accessibility. Based on the IEEE 802.15.4 LR-WPAN standard, the ZigBee standard has been proposed to interconnect simple, low rate, and battery powered wireless devices. The deployment of ZigBee networks is expected to facilitate numerous applications, such as home-appliance networks, home health care, medical monitoring, consumer electronics, and environmental sensors. Hence, aiding in building up an effective and secure routing scheme.

The ZigBee standard [15], designed to interconnect simple devices that previously have not been networked, is the latest attempt to address this wireless network vision. In the context of a business environment, this wireless movement can facilitate better automated control of facilities and assets. Moreover, there are also many applications for home-appliance networks, as well as in the area of home health care, consumer electronics, and environmental sensors.

ZigBee is a network and application layer specification developed by a multi-vendor consortium called the ZigBee Alliance [16]. Various ZigBee compliant product prototypes and application scenarios have already been developed by the industry, yet the performance and the supporting facilities of ZigBee networks have not been thoroughly evaluated. Routing in a ZigBee enabled network is very similar to the one in a Mobile Adhoc NETwork (MANET). In both cases, maintaining an end-to-end route is challenging since the network topology may change very frequently due to node failures, mobility, and many other factors.

Various MANET routing protocols have been proposed in the last few years [17, 18, 19, 20, 21, 22, 23]. Among them, Adhoc On-demand Distance Vector Routing (AODV) [3] and Dynamic Source Routing (DSR)

[4] are two of the most popularly deployed schemes. These routing algorithms of MANET aim to figure out the best route, even if the network is highly dynamic, toward a given destination at any time by consuming minimal messages/time overhead. Moreover, every participating node in MANET routing is implicitly assumed to be MANET router capable, and assumed to operate with the same set of functionalities.

Developing and testing real life applications for ad hoc network environments still demand a cardinal attention to the MANET research community. Moreover, hardware simulation lets one obtain the accurate and exact results. In software simulation what seem to be cheap might be an overhead in practical scenarios. The motivation of this paper is to present a low cost simulation of our proposed secure routing protocol, so that it can be used in personal area networks. In Section 2, we review the related secure routing protocols. In section 3 we present the packet structures used. In Section 4, we explain fidelity and explain the protocol in Section 5. In Section 6 we explain the hardware environment of Arduino and ZigBee along with the experimental results in Section 7. We provide a comparison of our protocol with some of the popular secure routing protocols, in Section 8 and present the conclusion in Section 9.

II. Related Work

In this section we review the existing secure routing protocols. There exist many secure routing protocols in MANET, as reviewed in [24]. These secure protocols cannot mitigate all kinds of attack faced by MANET networks. These protocols are more subjected in detecting and eliminating certain class of attacks. These protocols while mitigating attacks degrade the QoS of the network to a significant extent. This shortcoming demand a more secure protocol, which can mitigate majority of the attacks, such that the QoS is not effected.

Sanzgiri et.al [7] have proposed Authenticated Routing for Ad hoc Networks (ARAN), which uses asymmetric cryptography. Since, it uses public key encryption confidentiality is guaranteed and network structure is not exposed. Though the protocol maintains a high PDF, it requires extra memory, along with high processing overhead for encryption. It is still vulnerable to attacks like a black hole, wormhole and rushing attacks. Zapta et.al [8] 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 possibilities of MIM [9] attacks by invader nodes. Papadimitratos et.al have proposed SRP [25], which maintains a security association in between the source and the destination. It can prevent fabrication and loops created by malicious nodes. But, it suffers from cache poisoning and wormhole attacks. Wan et.al [26] have presented a protocol (UBSOR-Unobservable Secure on-Demand Routing Protocol) which achieves high privacy in reactive routing. It hides the content of the packets by encryption methods. However, it needs third parties to establish the key, and cannot handle wormhole attacks.

Li et.al [10] have proposed a Trusted AODV (TAODV) routing protocol. It uses trust recommendation and later on combining these to derive a logical conclusion. It exchanges, trust via two packets called TREQ and TREP, which is an extra overhead. The computational overhead of each authentication operation is high, and it may even lead to high traffic when there are many malicious nodes. Saha et.al [11] have proposed a routing protocol, which is based on the concept of fidelity. Fidelity is an integer number that is associated with each node. The approach reduces the computational overhead to a lot extent. 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. Dhurandher et.al [27] have presented a protocol (FACES-Friend-Based Routing Protocol) which determines trust of the nodes by sending challenges and sharing friends' lists. Challenges are sent to authenticate the nodes, and accordingly they are placed in friend list or question mark list. Friends are rated on the basis of the amount of data they transmit and rating obtained from other friends. But, it fails to combat wormhole or rushing attacks. Moreover, the control overhead is increased due to periodic flooding of challenge packet, and periodic sharing of friend list.

Our contribution is to provide a secure, reliable and low cost hardware protocol for MANETs. This gets implemented through fidelity. A second level of reliability is obtained through recommendations and report packets. This not only helps to identify the malicious nodes, but also eliminate them from the network. Hence, maintaining a good QoS for the network. Our main goal of the protocol is to build a low cost MANET, which is used effectively and cheaply, in a secured manner; both in fields like defense and domestic.

Table 1. Packet Formats

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, Current address, Last Hop 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, Current address, Last Hop Address, Culprit, Digital Signature}
Recommendation	{Source Address, Culprit, Digital Signature}

III. Proposed Packet Structure

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 1.

The attributes associated with the packets are explained below:

- Hop Count: It is the number of hops a packet takes from originator node to current receiving node. It has a size of 4 bits.
- Message Count: It is the total number of hops a packet must take to send data from the sender node to the destination node. This is the hop count value of the RREQ packet when it has reached the destination node. It has a size of 4 bits.
- Source Address: It is a 32 bit IP Address of the node which wants to send data.
- Destination Address: It is a 32 bit IP Address of the destination node to which the node wants to send the data.
- Battery Power: It is a 12 bit value which signifies the battery power left for the node replying with the NREP packet.
- Current Address: It is a 32 bit IP Address of the node sending/forwarding the packets, for a particular source-destination pair.
- Next Hop Address: It is a 32 bit IP address of the next node to which the packet is meant for. Packets like RREQ and Data use it for forward communication.
- Last Hop Address: It is a 32 bit IP address of the last node to which a reply has to send. Packets like RREP, Report, Fail Message and ACK use it for backward communication.
- Fail Array: It is a 32 bit IP addresses of the nodes that have failed to find a route for a particular source-destination pair. Since, there can be many common neighbors, in any two adjacent nodes. Other intermediate nodes can avoid sending RREQ packets again to these nodes. Hence, reduces the routing load.
- Digital Signature: It is the signed digest (hash value), using SHA-1 as the hashing algorithm.
- Encrypted Message: The message is encrypted with the public key of the destination node, using the RSA algorithm. The destination node on receiving the data, decrypts it by using its private key.
- Culprit: It is a 32-bit IP Address of the node, which has failed to send the ACK packet.

IV. Fidelity Definition

In this section we explain the fidelity and the decisions associated with it. Fidelity 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 varies with respect to node to node. Let, node B has a fidelity value φ_{BA} with respect to A, while C has a value φ_{CA} with respect to A. If $\varphi_{BA} > \varphi_{CA}$, 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. We assume that the source and the destination are none malicious, hence the fidelity does not increase or decrease in the case of a source or destination.

$$\varphi = f\ (\text{ACK, Report, Recommendation}) \qquad (1)$$

If ACK packet is received and verified, then the fidelity is incremented; which indicates the reception of data packets by the destination node. Moreover, if no ACK packet is received within timeout or a report packet is received, then the fidelity is decreased. If recommendation packets are received uniquely from a neighbor node, then the fidelity is decreased and a counter value is also increased. This count value is a check for blacklisting the culprit node from the network. Once 3 unique recommendations are received from three different neighbors, the node gets blacklisted. The choice of count value as 3 is because of observed data. At count=3 it has been observed that the malicious nodes are effectively expelled from the network, by all the nodes in most effective time. We have simulated and found out the average time required by a network to eliminate a malicious node from its network for 10 simulations, with count values as 1,3,5,7 and 9, as shown in Table 2. We have considered the Node Traversal Time T = 5ms.

Table 2 Time required (in ms) for different Black List counts and number of nodes

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

Assuming no infrastructure for charging battery is available, relying on a node too much for transmission of the same packet can sometimes prove futile, since the battery power of every node is finite. Thus, 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, if a node with a high fidelity value starts behaving maliciously, it would take a lot of time to bring down that node's fidelity value. Hence, a maximum limit on the fidelity needs to be calculated, as shown in Equation 2. The power consumed by an intermediate node can be calculated by considering the worst case scenario, i.e, all packets are dropped. Similarly, a minimum fidelity can be calculated with Equation 2. In case of minimum fidelity the value will be a negative value, which will directly depend on number of protocol packets, report and recommendation packets a node can send.

$$\varphi = \left\lfloor \frac{\text{Intial Battery Power of Intermediate Node} - \text{Total power consumed for receiving and transmitting packets}}{\text{Total power consumed to send data and acknowledgement packets}} \right\rfloor \qquad (2)$$

V. PROPOSED PROTOCOL

In this section, we have discussed the flow diagram along with the routing details of our protocol, referring to Fig. 1. We have used a new self organized key management scheme as proposed in [12], since it uses less memory space which supports our objective of making a lightweight secure routing protocol.

We have explained the flow diagram of our FBOD model using an example with a network containing 5 nodes A, B, C, D and E; where A is the source node and D is the destination node. We assume that the source and destination nodes are non malicious nodes, otherwise no communication can be successful. We have explained the algorithm by dividing it into 3 parts: Source Node,Intermediate Node and Destination Node.

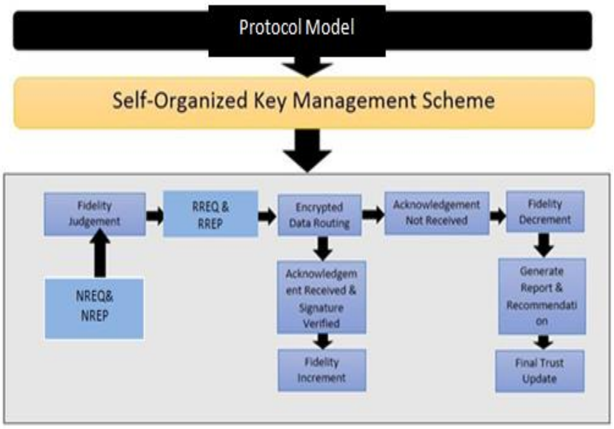
Protocol Model
Self-Organized Key Management Scheme
Fidelity Judgement
RREQ & RREP
Encrypted Data Routing
Acknowledgement Not Received
Fidelity Decrement
NREQ& NREP
Acknowledgement Received & Signature Verified
Generate Report & Recommendation
Fidelity Increment
Final Trust Update

Fig.1. Fidelity Based On-Demand

A. Source Node

The sender node selects a node and routes data as show in Fig 2-9 and explained as follows:

```
Step 1: Start.
Step 2: Send NREQ and wait for $\tau_1$=2* Average_Delay
        for neighbors to reply with NREP.
Step 3: Enter the new neighbor in the table with Fidelity
        initialized to 0. If old neighbors, then activate that
        node in the table and consider the old fidelity value.
Step 4: If the destination node is in the neighbor table
    Step 4.1: Send RREQ and wait for $\tau_1$.
    Step 4.2: If RREP is received and verified.
        Step 4.2.1: Goto Step 6.
    Step 4.3: Else Goto Step 10.
Step 5: Else
    Step 5.1: If (Fidelity_Judgement () ==False)
        Step 5.1.1: Goto Step 10.
Step 6: Encrypt Data and wait For ACK for time
        $\tau_3$=2*Message_Count*Avarage_Delay
Step 7: If ACK received and verified.
    Step 7.1: Increment the fidelity by 1, only if the old
    fidelity is less than maximum fidelity, except for the
    destination node.
    Step 7.2: Goto Step 9.
Step 8: If ACK is not received or Report packet is
        received and verified.
    Step 8.1: Decrement the fidelity by 1, only if the old
    fidelity is greater than the minimum fidelity, except
    for the destination node.
Step 9: If Source wants to send data.
    Step 9.1: If to the same destination node
        Step 9.1.1: Goto Step 6.
    Step 9.2: Else, Goto 2.
Step 10: Stop.
```

a. Fidelity_Judgement()

```
Step 1: Find the neighbor with maximum fidelity $\varphi$ from
        the set of neighbor nodes.
Step 2: If there is tie, then find the node with maximum
        battery power. If still tie exists, then select random
        node form the set.
Step 3: Send the RREQ to the selected Node and wait for
        $\tau_2$=2*Network_Diameter*Avarage_Delay.
Step 4: If RREP not received within $\tau_2$ or Fail Message is
        received.
    Step 4.1: Add the node/s in the Fail Array.
    Step 4.2: If Recommendation is received.
        Step 5.4.2.1: Call Recv_Recco()
Step 5: Else Goto Step 9.
Step 6: Find a set of neighbor nodes which are not present
        in the Fail Array and Blacklist.
Step 7: If the set is empty.
    Step 7.1: Return False
Step 8: Else Goto Step 1
Step 9: Return True
```

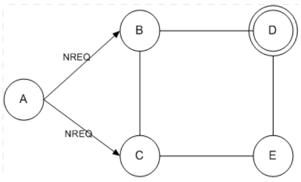

Fig.2. Node A broadcasts NREQ Packets

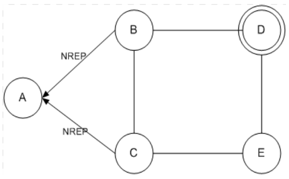

Fig.3. Node B, C sends NREP packet

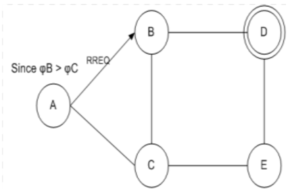

Fig.4. Node A selects and sends RREQ to node B

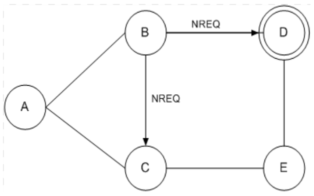

Fig.5. Node B sends NREQ

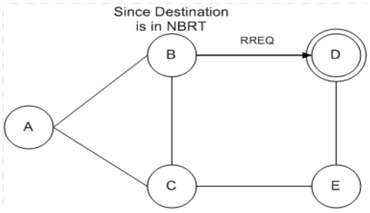

Fig.6. Node B sends RREQ to D

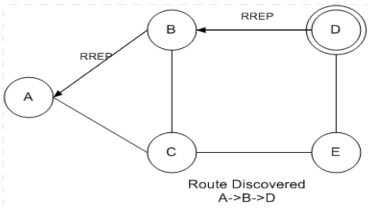

Fig.7. RREP is sent from D to A

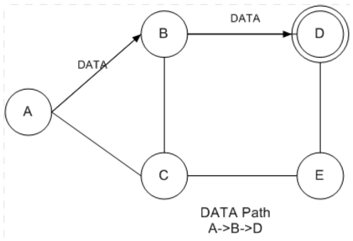

Fig.8. Data is sent to D

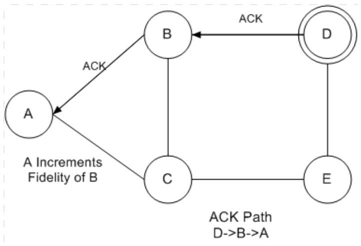

Fig.9. ACK packet is sent back to A

B. Intermediate Node

An intermediate node can receive all kinds of packets. After it receives RREQ from the sender/neighbor intermediate node, it routes data as shown in Fig as shown in Fig 5-9.

```
Step 1: Start
Step 2: If NREQ received
        Step 2.1: Send NREP to last seen address
Step 3: If RREQ received
        Step 3.1: Call Intermediate_Node_Seletion()
Step 4: If Data Packet received
        Step 4.1: Call Forward_Data()
Step 5: If Recommendation recived
        Step 4.1: Call Recv_Recco()
Step 6: Stop
```

a. Intermediate_Node_Seletion()

```
Step 1: Start.
Step 2: Send NREQ and Wait for $\tau_1$.
Step 3: Enter the new neighbor in the table with Fidelity
        initialized to 0. If old neighbors, then activate that
        node in the table and consider the old fidelity value.
Step 4: If the destination node is in the neighbor table.
    Step 4.1: Send RREQ to that Node an wait for $\tau_1$.
    Step 4.2: If RREP is received
        Step 4.2.1: Goto Step 6.
    Step 4.3: Else Goto Step 8.
Step 5: Else
    Step 5.1: If (Fidelity_Judgement()==False)
        Step 5.1.1: Goto Step 8.
Step 6: Save the address as next hop address.
Step 7: Send the RREP to the last seen address.
Step 8: Stop
```

b. Forward_Data()

```
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

        $\tau_3 = 2 * (\text{MESSAGE\_COUNT} - \text{HOP\_COUNT}) * (\text{AVG\_DELAY})$

Step 4: If ACK received.
    Step 4.1: Forward the ACK to Last Seen Address.
    Step 4.2: Increment the fidelity by 1, only if the old
    fidelity is less than maximum fidelity, except for the
    destination node.
Step 5: Else,
    Step 5.1: If 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: Node puts the address of the
        neighbor in the Culprit of the Recommendation.
    Step 5.3: Decrement the fidelity by 1, only if the old
    fidelity is greater than the minimum fidelity; except
    for the destination node.
Step 6: Stop.
```

c. Recv_Recco()

```
Step 1: If recommendation verified
    Step 1.1: Goto Step 9
Step 2: If the node in the culprit array is its neighbor table
    Step 2.1: Goto Step 9
Step 3: If it has already been recommended by the same
    sender,
    Step 3.1: Goto Step 9
Step 4: Else
```

```
        Step 4.1: Add the address in the blacklist and
        increment the counter by 1
        Step 4.2 : Decrement the Fidelity of that node by 1.
Step 5: If counter >= 3
Step 6: Else
        Step 6.1: The recommended node is removed from
        neighbor table.
Step 7: Stop
```

C. Destination Node

The destination node on receiving packets will perform as follows:

```
Step 1: Start
Step 2: If NREQ received
       Step 2.1: Send NREP back to the last seen address
Step 3: If RREQ received
       Step 3.1: The node Signs the RREP packet digest
           with Source's Public Key
Step 4: If Data Packet received
       Step 4.1: Decrypt the Data Packet with its own
           Private Key.
       Step 4.2: If decryption is successful
           Step 4.2.1: Sign the ACK and send to the last
           seen address.
Step 5: Stop
```

VI. Node Architecture

In Fig. 10 the architecture of a node has been shown along with the pin diagram of ZigBee in Fig. 11.

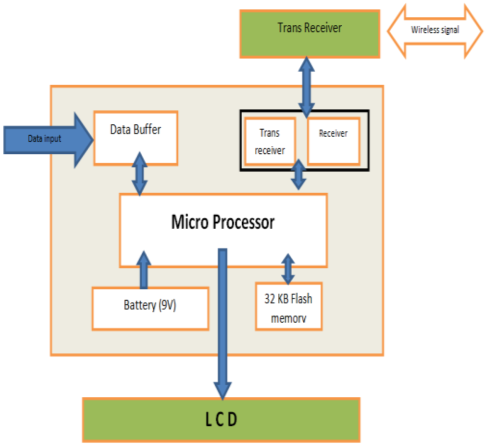

Fig.10. Architecture of the a node

The node contains a micro-processor, trans-receiver and LCD to display the details. The high-performance Atmel 8-bit; AVR RISC-based microcontroller; combines 32 KB ISP flash memory, with read-while-write capabilities. It consists of 1KB EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. In one node, the coded algorithm is uploaded in the microprocessor through a USB drive to the Arduino board. The micro-processor communicates with the different serial ports according to the algorithm.

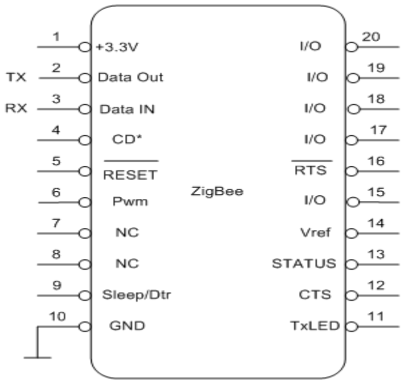

Fig.11. Pin diagram of ZigBee

The Rx and Tx are receiver port and transmitter port respectively. The main function of Tx and Rx is to transmit data to the peripheral devices, in our case it is the ZigBee trans-receiver. The Tx of Arduino is connected to the Data in of ZigBee and Rx is connected to the Data out of the ZigBee module. The ZigBee modules, transmits and receives the wireless signal from other nodes and the micro-processor checks the received data and takes appropriate action. The first step to configure the hardware is to configure the ZigBee module, which must be paired with the Arduino Board. The ZigBee will only detect signals from a same PAN ID, which must be set same for all the trans-receivers. All the trans-receivers must have a Network Id which will uniquely identify every node in the network. Fig. 12 represents the real routing nodes, and Fig. 13 shows a node with the battery connected with it.

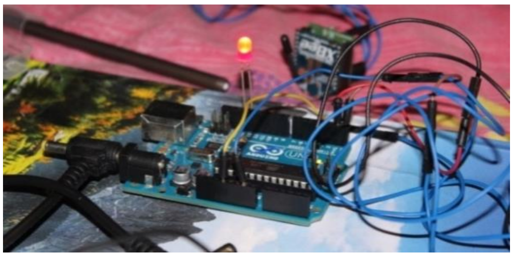

Fig.12. Circuit of a node

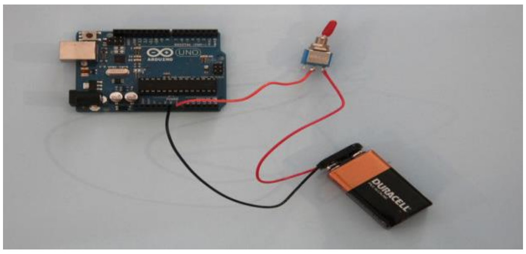

Fig.13. Router node with battery

VII. Experimental Results

We have simulated the protocol on the hardware, with all the transmitters belonging to the same PAN ID. While setting up the ZigBee modules it is to be kept in mind that all the nodes must belong to the same network ID, otherwise the transceiver will not detect any signals from the other nodes. We have taken the id of the nodes as 1, 2 etc., but it can be taken as the IP address of the nodes.

In our simulation, we have considered that only one node is sending data and one node is receiving data, the other nodes act as a routing node. Simple cryptographic symbols are used in the routing algorithm, which can be custom designed according to the use of the network. Nodes move in a 50*26 meter region, with each node's transmission range as 15m.

In the first simulation, we consider three nodes, as shown in Fig. 14. The destination node is not in the source's range, so the source sends a request to the nearest intermediate node, i.e., Node 1. Node 1 finds the destination node in its neighbor table, and sends the request directly. The destination replies, which is forwarded back to the node. After, the source node has received the ACK, it increases the fidelity of Node 1 by one. Node 1, does not increase the fidelity of the destination node, since it has been assumed that the destination node is non-malicious.

In the next simulation, we consider four nodes, as shown in Fig. 15. The source node now has two neighbor nodes. Since, Node 2 has fidelity zero, the source sends the request to the destination through Node 1. After the source node receives a reply from the destination, it forwards the data from the same route. Let us assume that Node 1 is a malicious node, with greyhole attack; then it will drop the ACK packet coming in from the destination node. After the waiting time for the source node is over, it reduces the fidelity of Node 1 by one. The source node sends a route request to Node 2, as shown in Fig. 16. Node 2 sends the data successfully and its fidelity is increased by one.

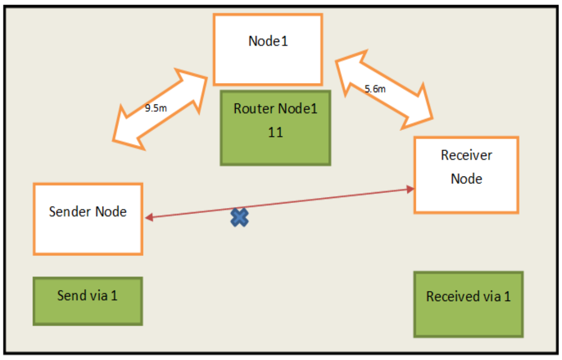

Fig.14. Simulation with three nodes

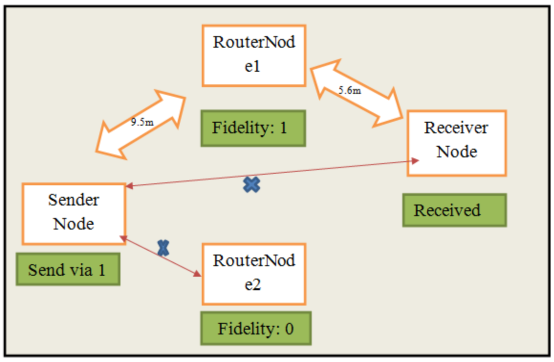

Fig.15. Simulation with four nodes

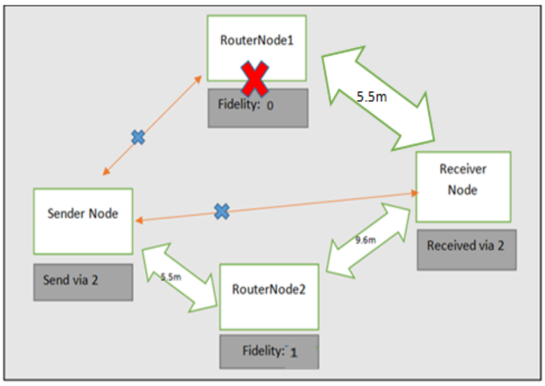

Fig.16. A malicious node is added in the network

VIII. Results & Comparison

An extensive simulation model having scenario of 10 mobile nodes is used to study inter-layer interactions with an area of 50 meter x 26 meter, with each node's range as 15 m. We have considered Node 1 as the source and Node 10 as the destination node, as shown in Fig. 17. We change the number of nodes from 2 to 10, with the mobility model as a random waypoint model. The average speed is 1 m/s with pause time of 30 seconds.

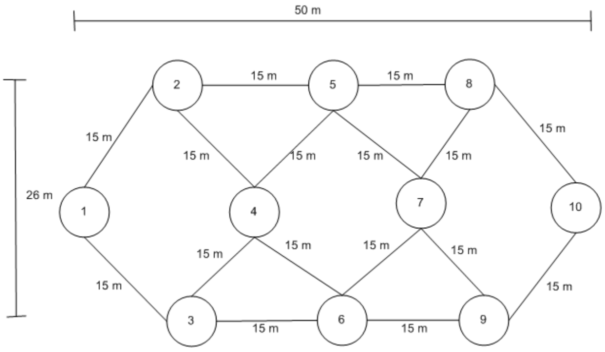

Fig.17. Node Placements for Hardware Simulation

When all 10 nodes start routing and few transmissions have taken place, the nodes 2, 6, 8 are made malicious, and they start their attack one after another. We have altered the positions of the intermediate nodes randomly and taken the average value of all such node placements. The same scenario has been also used for performance evaluation of other secure protocols with which our protocol has been compared i.e. ARAN, SAODV, TAODV. We consider these protocols as they are well known among the secure on demand routing protocols. Moreover, we try to show that our protocol stands way better than the other secured protocol.

First, we compute the packet delivery fraction (PDF) for all the protocols as showed in Fig. 18, 19. The graph shows that FBOD shows an average PDF of 89.6%, which is decreased to 83.25% in malicious environment. Other protocol shows fluctuations in benign and fall in a malicious environment, since none can eliminate the malicious nodes. FBOD on the other hand, uses packets like report and recommendation to blacklist the malicious nodes. Once the malicious nodes get blacklisted, the packet delivery fraction increases, as in the case of FBOD.

Second, we compute the normalized routing load (NRL) for the protocols as shown in Fig. 20, 21. In the benign environment, the average NRL for FBOD protocol is 0.82, which increases to 1.05 in malicious environment. TAODV shows high NRL, due to its extra packets to build trust. SAODV and ARAN comparatively shows average NRL, since with inclusion of malicious nodes lot of authentication process has to take place. In case of FBOD, though fidelity it measures the trust of the neighbor, as well as eliminates these malicious nodes from the network.

Finally, we calculate the end to end delay for the protocols in benign environment as shown in Fig. 22, 23. As the number of nodes increase, the end to end delay increases. Our protocol shows an average delay of 15.2 sec in benign and 20.9 sec in malicious environment. Our protocol shows a smaller increase in the end to end delay, compared to other protocol, since we can effectively detect and eliminate malicious nodes, there by bringing the network back to stability. Moreover, we don't use heavy packets like TAODV, or heavy authentication schemes like SAODV and ARAN, which increases the delay.

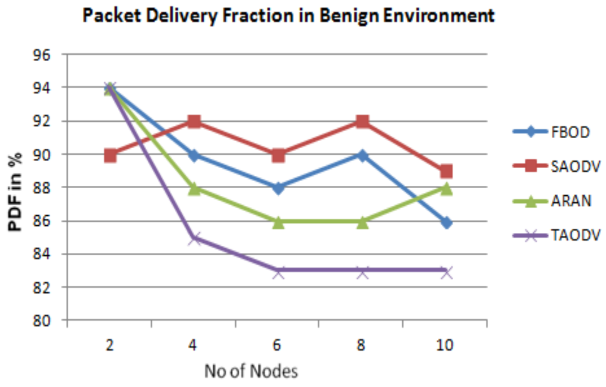

Fig.18. Packet Delivery Ratio Vs Number of Nodes

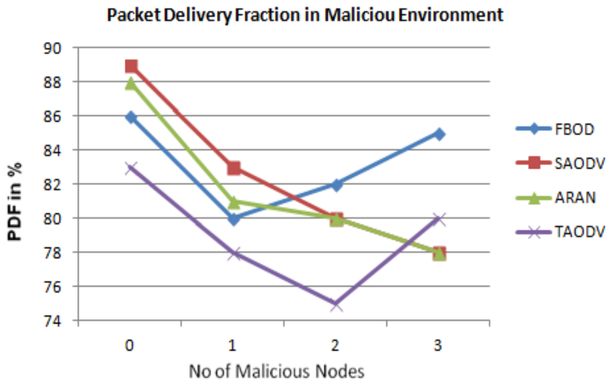

Fig.19. Packet Delivery Ratio Vs Number of Malicious Nodes

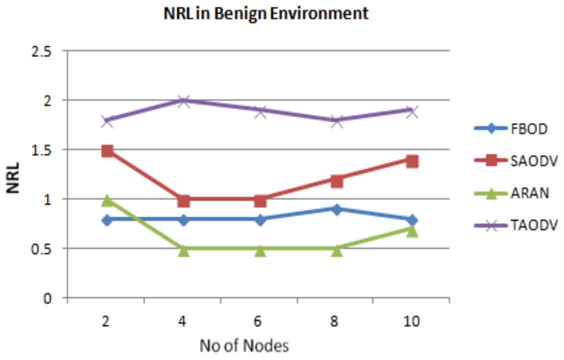

Fig.20. Normalized Routing Load Vs Number of Nodes

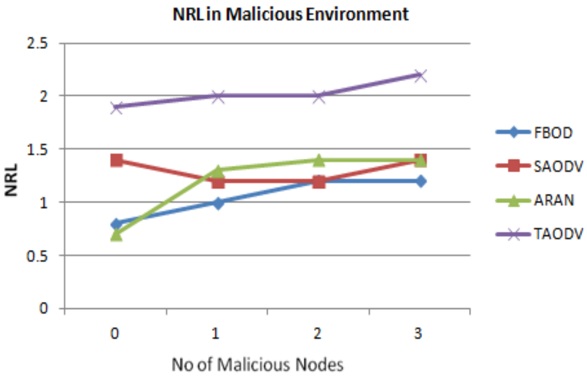

Fig.21. Normalized Routing Load Vs Number of Malicious Nodes

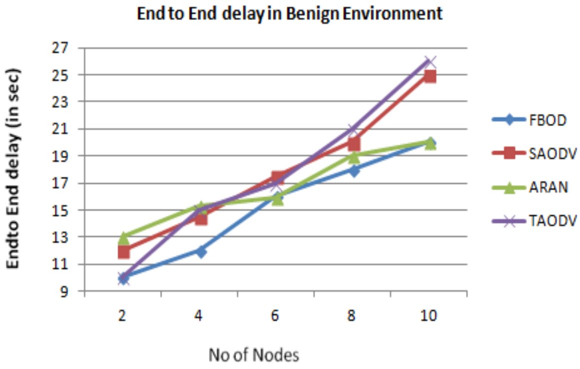

Fig.22. End to End Delay Vs Number of Nodes

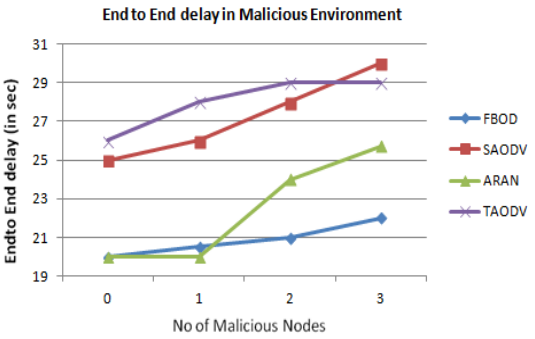

Fig.23. End to End Delay Vs Number of Malicious Nodes

FBOD bypasses the route containing the blackhole, greyhole, wormhole and jellyfish attacks by the use of fidelity and blacklisting concept.

Black hole [28] is any node which silently discards the data traffic without informing the source node. Suppose node A is the source and B is the destination. Let there be two paths between A and B, one through node C another through node D. Let node D be the black hole. During network initialization, both C and D are assigned fidelity value 0. If node D is selected as the intermediate node by source A, which silently discards the data packet, and no acknowledgement is received by node A. Then, according to the protocol, node D's fidelity is decreased to -1. On the next attempt, node C has greater fidelity, i.e. 0 than node D, therefore node C is selected and after successful reception of acknowledgement its fidelity becomes 1. This process continues and node A will select node C always, due to higher fidelity value thus bypassing node D permanently. Grayhole [28] is a node that can switch from behaving trustfully to behaving maliciously and the reverse as well. Due to its vacillating nature it is difficult to identify, the attacker. In our protocol, once the node starts behaving maliciously, its fidelity starts decreasing. Moreover, it will be recommended as a culprit node by others nodes. As soon as the count is 3, the greynode gets blacklisted, and it gets removed completely from the network.

In wormhole attack [29], a tunnel is created using a number of colluding nodes. When an attacker receives packets at one point in the network, it tunnels them to another point in the network, and then relays them into the network from that point. These advertised routes are much shorter than the actual routes which go through the wormhole tunnel. In our protocol, the fidelity parameter of the nodes does not allow colluding attackers to stay on the network for long. When acknowledgement from source is, not received within the timeout period, the fidelity of the responsible node is decreased and later the worm hole route is avoided.

Jellyfish [30] affects packet end-to-end delay and the delay jitter but not packet delivery ratio or throughput. A jellyfish attacker first needs to intrude into the multicast-forwarding group. It then delays data packets unnecessarily for some amount of time before forwarding them. This results in significantly high end-to-end delay and thus degrades the performance of real applications. In this protocol, due to the delay caused by the attacker, the acknowledgement's waiting time will get over and time to live (TTL) will expire. As the route was unsuccessful in communicating the data, the fidelity is decreased and a new route is selected.

IX. Conclusion

Our proposed model has many unique features which makes it stand different from other existing secure on-demand protocols. FBOD is a lightweight protocol and doesn't require any flooding of extra packets or extra memory, which is not in the case of TAODV and ARAN. Secondly, it is a unicast protocol, thereby making the network free from many attacks. The secure route selection mitigates attacks like wormhole and rushing attack, which is not in the case of SAODV. 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. The use of the busy wait prevents the cycling of RREQ packets. Packets like report and recommendation help in quickly identifying malicious nodes and eliminating them from the network. Once the malicious nodes are eliminated, the NRL decreases back to that in the case of benign environment. We can have observed that our hardware implementation works better in malicious environment than other popular secure routing protocols, with high PDF, low NRL and average End-to-End delay; hence making it commercially viable.

References

[1] S. Corson andz J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," *Network Working Group*, RFC: 2501, January 1999.

[2] M. Frodigh, P. Johansson and P. Larsson, "Wireless ad hoc networking: the art of networking without a network," *Ericsson Review*, No. 4, pp. 248-263, 2000.

[3] K. Komali, V. Mahesh and R.Y. Kumar, "A novel secured protocol for data transmission in ad hoc networks using clustering," *International Journal of Computer Science and Information Technologies (IJCSIT)*,Vol. 5(5),pp. 6567-6571, 2014.

[4] S. Sharmila1, G. Umamaheshwari and M. Ruckshana,"Hardware implementation of secure aodv for wireless sensor networks," *ICTACT Journal On Communication Technology*, Vol.1, Issue 04,pp.218-229, December 2010.

[5] S. Dalu, M.K. Naskar and C.K. Sarkar,"Implementation of a topology control algorithm for manets using nomadic community mobility model," *Industrial and Information Systems*,pp.1-5,2008.

[6] A. Passarella and F. Delmastro, *Multi-hop Ad hoc Networks from Theory to Reality*, Nova Science Publishers,ch.9,2007, pp.153-177.

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

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

[9] R.K. Guha, F. Zeeshan and M. Shahabuddin, "Discovering man-in-the-middle attacks in authentication protocols," *Military Communications Conference, MILCOM IEEE*, pp 29-31, October 2007.

[10] R.K. Nekkanti and C.W. Lee, "Trust based adaptive on demand ad hoc routing protocol," *In: Proceedings of the 42nd Annual Southeast Regional Conference*, pp 88–93,2004.

[11] H.N. Saha, D. Bhattacharyya, P.K. Banerjee,"Fidelity based on demand secure (FBOD) routing in mobile adhoc network," *Advances in Parallel Distributed Computing*, Springer Berlin Heidelberg, pp 615-627, 2011.

[12] H.N. Saha, D. Bhattacharyya, B. Banerjee, S. Mukherjee, R. Singh and D. Ghosh, "Self-organized key management based on fidelity relationship list and dynamic path," *International Journal of Application or Innovation in Engineering & Management*, Vol 3, Issue 7, pp.97-100, July 2014.

[13] J.C. Cano and P. Manzoni, "Evaluating the energy-consumption reduction in a manet by dynamically switching-off network interfaces," *In Proceedings of the 6th IEEE Symposium on Computers and Communications*, pp 186-191, Juiy 2001.

[14] M. Tarique and R. Islam, "Minimum energy dynamic source routing protocol for mobile ad hoc networks," *International Journal of Computer Science and Network Security*,Vol 7, Issue 11,pp.304-311,2007.

[15] Z. Alliance, "Zigbee specification v1.0," June 2005.

[16] "Zigbee alliance," Available: http://www.zigbee.org/

[17] C. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," *In Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, 1999.

[18] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, Kluwer Academic Publishers, 1996, vol. 353, ch. 5, pp. 153–181.

[19] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, and T. Clausen, "Optimized link state routing protocol (olsr)," *Network Working Group*, RFC:3626, October 2003.

[20] I. Chakeres, E. Belding-Royer and C. Perkins, "Dynamic manet ondemand routing protocol (dymo)," *Mobile Ad hoc Networks Working Group*, Internet Draft, version: draft-ietf-manet-dymo-10, October 2005.

[21] V. Park and S. Corson, "Temporally-ordered routing algorithm (tora) version 1," *Mobile Ad hoc Networks Working Group*, Internet Draft, version: draft-ietf-manet-tora-spec- 03, June 2001.

[22] Z. J. Haas, M. R. Pearlman and P. Samar, "The zone routing protocol (zrp) for ad hoc networks," *Mobile Ad hoc Networks Working Group*,Internet Draft, version: draft-ietf-manet-zone-zrp- 04, July 2002.

[23] B. N. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," *In Proceedings of ACM Mobicom*, pp. 243–254, August 2000.

[24] H.N. Saha, D. Bhattacharyya, B. Banerjee, S. Mukherjee, R. Singh and D. Ghosh, "A Review On Attacks And Secure Routing Protocols In MANET," *International Journal of Innovative Research and Review (JIRR), CIBTech*, Vol. 1, No. 2, October-December 2013.

[25] P. Papadimitratos and Z.J. Haas,"Secure Routing for Mobile Ad Hoc Networks.," *In Proceedings of International conference on SCS Communication Networks and Distributed Systems (CNDS)*, pp.112-118, January 27-31, 2002.

[26] Z. Wan, K. Ren and M. Gu, "USOR: An unobservable secure on-demand routing protocol for mobile ad hoc networks," *IEEE Trans. Wireless Communication.*, Volume 11, Issue 5, 2012, DOI: 10.1109/TWC.2012.030512.111562.

[27] S. K. Dhurandher, M. S. Obaidat, K. Verma, P. Gupta, and P. Dhurandher, "FACES: Friend-Based Ad Hoc Routing Using Challenges to Establish Security in MANETs Systems," *IEEE, Systems Journal,* Vol. 5, Issue 2, pp. 176-188, 2011.

[28] R. Kaur and P. Singh,"Review of black hole and grey hole attack," *The International Journal of Multimedia & Its Application (IJMA)*,Vol 6,No 6,December 2014.

[29] V. Mahajan, M. Natu, A. Sethi, "Analysis of wormhole intrusion attacks in MANETS," *In IEEE Military Communications Conference (MILCOM)*, pp. 1-7, 2008.

[30] A. Kaur, D.S. Wadhwa, "Effects of Jelly Fish Attack on Mobile Ad-Hoc Network's Routing Protocols," *Int. Journal of Engineering Research and Application (IJERA)*, Vol. 3, Issue 5, pp 1694-1700, September-October 2013.

Authors' Profiles

Himadri Nath Saha: He graduates from Jadavpur University. He did his post graduate degree from Bengal Engineering and Science University. He is Assistant Professor of Institute of Engineering and Management. His research interest is security in MANET.

Rohit Singh: He is a student of Institute of Engineering and Management and is currently pursuing B.Tech in Computer Science. His research interest is Wireless Network.

Prof.(Dr) Debika Bhattacharyya: She did Phd. from Jadavpur University in the dept. of ETCE. She is Head of Department in the Department of Computer Science Engineering at Institute of Engineering & Management. Her research Interest is security in MANET.

How to cite this paper: Himadri N. Saha, Rohit Singh, Debika Bhattacharyya,"Hardware Implementation of Fidelity based On Demand Routing Protocol in MANETs", IJCNIS, vol.7, no.8, pp.39-48, 2015.DOI: 10.5815/ijcnis.2015.08.05