A New Combination Approach To Secure MANETS Against Attacks

G. S. Mamatha[1] and Dr. S. C. Sharma[2]

[1]Department of Information Science and Engineering, R. V. College of Engineering, Bangalore, India
mamatha.niranjan@gmail.com
[2]Vice-Chancellor, Tumkur University, Tumkur, Karnataka, India
scsrvrd@yahoo.co.in

ABSTRACT

In Wireless communications the traffic across a mobile ad hoc network (MANET) can be highly vulnerable to security threats. Because of the features like unreliability of wireless links between nodes, constantly changing topology, restricted battery power, lack of centralized control and others, the mobile ad hoc networks are more prone to suffer from the malicious behaviors than the traditional wired networks. Therefore, it becomes very much necessary to pay more attention to the security issues in the mobile ad hoc networks. In this paper a new combination approach, which combines three techniques is discussed. This approach is used to identify and prevent the malicious nodes exhibiting different network layer attacks and is compared with a nearest approach through the experimental results and analysis. Performance evaluation is done based on few network parameters.

KEYWORDS

Wireless communications, MANET, Security, nodes, malicious and combination.

1. INTRODUCTION

A mobile ad hoc network (MANET) is a group of devices or nodes that transmit across a wireless communication medium. Cooperation of nodes is important to forward packets on behalf of each other when destinations are out of their direct wireless transmission range. There will be no centralized control or network infrastructure for a MANET to be set up, thus making its deployment quick and inexpensive. The nodes ability to move freely ensures a flexible and versatile dynamic network topology which is another important feature of a MANET. Some of the MANET applications includes emergency disaster relief (eMANETS [1]), military operations over a battlefield (vulnerable infrastructure), and wilderness expeditions (transient networks), and community networking and interaction between students during a lecture [2].

The inherent features of mobile ad hoc networks make them more vulnerable to a wide variety of attacks by misbehaving nodes. Such attacks can be listed as passive and active attacks. In active attacks, we mainly consider the internal attacks for network layer such as black hole attack, gray hole attack, worm hole attack, message tampering, routing attacks. A malicious node drops packets or generates additional packets solely to disrupt the network performance and prevent other nodes from accessing any network services (a denial of service attack) [2].
Misbehavior can be divided into two categories [3]: routing misbehavior (failure to behave in accordance with a routing protocol) and packet forwarding misbehavior (failure to correctly forward data packets in accordance with a data transfer protocol). In this paper we focus on the latter. Our approach consists of an algorithm that performs two tasks: a) enables packet

DOI: 10.5121/ijwmn.2010.2406

forwarding misbehavior detection through the principle of conservation of flow (PFC) [4], and b) enables the prevention of nodes that are consistently detected exhibiting packet forwarding misbehavior. A node that is accused of misbehavior is denied access to the network by its peers, which ignore any of its transmission attempts. Thus, misbehaving nodes are isolated from the rest of the network. Our criterion for judging detection of misbehavior on a node is the estimated percentage of packets dropped by using principle of flow conservation and a simple acknowledgement approach, which is compared against a pre-established misbehavior threshold. Any node dropping packets in excess of this threshold is deemed a misbehaving node while those below the threshold are considered to be correctly behaving [2]. These two are employed using AODV (Ad hoc on demand distance vector protocol) routing strategy. This approach detects and prevents misbehaving nodes (malicious) capable of launching any of the network layer attacks.

The paper discusses the framework and a relevant algorithm with AODV protocol implementation that deal with these attacks. We then demonstrate through experimental results that an appropriate selection of the misbehavior threshold will be able to identify the misbehaved and well-behaved nodes, as well as high robustness is assured against different degrees of node mobility in a network that is affected especially by black hole and/or gray hole attacks.

The paper is organized as follows. Section II describes related work carried out in the area of MANET security. Section III describes our algorithm for packet forwarding misbehavior identification and prevention, and Section IV presents a performance evaluation. Finally, the paper is concluded in Section V.

2. RELATED WORK

A lot of research has been carried out in both the route discovery security part of routing protocols, and on packet forwarding. This Section at first looks at different ways of securing the network against misbehaving nodes and data forwarding anomalies. Then a sneak review followed shows the work that attempts to identify and prevent misbehavior in data packet forwarding.

2.1. Routing and Packet Forwarding Protection

Secure routing protocols have been proposed based on existing ad hoc routing protocols. These eliminate some of the optimizations introduced in the original routing protocols because they can be exploited to launch different types of attacks. Examples of such protocols are the secure efficient distance vector (SEAD) routing [5] which is based on the destination sequenced distance vector (DSDV) [6], the secure ad-hoc on-demand distance vector (SAODV) routing protocol [7] [8] based on AODV [9] [10], and the secure on-demand routing protocol for ad hoc networks (Ariadne) [11] based on the dynamic source routing (DSR) protocol [12] and the timed efficient stream loss-tolerant authentication (TESLA) protocol proposed in [13]. Also extending DSR to provide it with security mechanisms is CONFIDANT (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks) [14]. These approaches only secure the path discovery and establishment functionality of routing protocols, thus the proposed approach complements them by securing the data forwarding functionality [2].

The routing protocol proposed in [15] offers resilience to Byzantine behavior (any action that results in the disruption or degradation of the data forwarding service) by an algorithm that allows the detection of an anomalous link after log n faults have occurred on a path, where n is the hop length of the path. In [16] when a node has broken the security mechanisms of a network is regarded as an intruder. Each node is able to detect signs of intrusion locally and neighboring nodes collaborate to further investigate malicious behavior. In both these approaches a node uses its own data to identify another node as an intruder. In contrast, the

proposed approach in this paper allows a node to detect anomalies in packet forwarding based on data acquired by other nodes in the network as well as on its own data [2].

2.2. Misbehavior Detection

In order to provide reliable network connectivity some work has been carried that aims to protect data packet forwarding against malicious attacks. The last paragraph of this section reviews some approaches that detect malicious behavior in the data forwarding phase. WATCHERS (Watching for Anomalies in Transit Conservation: a Heuristic for Ensuring Router Security) [4] is a protocol designed to detect disruptive routers in fixed networks through analysis of the number of packets entering and exiting a router. In this approach each router executes the WATCHERS protocol at regular intervals in order to identify neighboring routers that misroute traffic and avoid them. WATCHERS require the existence of at least one path not affected by disruptive routers between any two well behaved routers in the network. Although WATCHERS is based on the principle of conservation of flow in a network as our proposed algorithm, its design focuses only on fixed networks and is not applicable to mobile ad hoc networks [2].

SCAN (self-organized network layer security in mobile ad hoc networks) [3] focuses on securing packet delivery. It uses AODV [9] [10], but argues that the same ideas are applicable to other routing protocols. SCAN assumes a network with sufficient node density that nodes can overhear packets being received by a neighbor, in addition to packets being sent by the neighbor. SCAN nodes monitor their neighbors by listening to packets that are forwarded to them. The SCAN node maintains a copy of the neighbor's routing table and determines the next-hop node to which the neighbor should forward the packet; if the packet is not overheard as being forwarded, it is considered to have been dropped [2]. Whereas, in the proposed algorithm the nodes do not need to overhear transmissions to and from any neighbor in order to detect misbehavior.

Finally, in [17] a system that can mitigate the effects of packet dropping is proposed. This is composed of two mechanisms that are kept in all network nodes: a watchdog and a pathrater. The watchdog mechanism identifies any misbehaving nodes by promiscuously listening to the next node in the packet's path. If such a node drops more than a predefined threshold of packets the source of the communication is notified. The pathrater mechanism keeps a rate for every other node in the network it knows about. A node's rate is decreased each time a notification of its misbehavior is received. Then, nodes' rates are used to determine the most reliable path towards a destination, thus reducing the chance of finding a misbehaving node along the selected path. This work as described uses DSR but it is claimed it can easily be adapted to other source routing protocols. However, its applicability has not yet been addressed for distance-vector based routing protocols. Moreover, the watchdog might not detect a misbehaving node in the presence of ambiguous collisions, receiver collisions or nodes capable of controlling their transmission power. Such weaknesses are the result of using promiscuous listening to determine whether a node has forwarded a packet or not. This approach does not have similar kind of weaknesses since it is based on parameters obtained from nodes that are actually sending and receiving packets to and from the node whose behavior is under evaluation. Also, using path rater can be considered a reward for selfish nodes since the flow is diverted towards other nodes in the network while selfish nodes preserve their resources [2]. Whereas, the proposed approach denies access to the network to any of the node, that has been identified as malicious, thus discouraging them from dropping packets.

A different approach with a 2ACK scheme, which is a network-layer technique to detect misbehaving links and to mitigate their effects, is used in [18] [19]. It can be implemented as an add-on to existing routing protocols for MANETs, such as DSR (Dynamic Source Routing). The 2ACK scheme detects misbehavior through the use of a new type of acknowledgment packet, termed 2ACK. A 2ACK packet is assigned a fixed route of two hops (three nodes), in

the opposite direction of the data traffic route [18]. Whereas, the proposed algorithm just uses a simple acknowledgement approach instead a 2ACK scheme, which increases the overhead. In section IV we are giving the comparison results for the proposed algorithm (ACK+AODV+PFC) and the 2ACK+ DSR approach, which uses an on-demand protocol DSR instead of AODV [18].

Some of the recent works related o network security attacks can be mentioned as, Enhanced Intrusion Detection System for Discovering Malicious Nodes in Mobile Ad Hoc Networks. The main feature of the proposed system is its ability to discover malicious nodes which can partition the network by falsely reporting other nodes as misbehaving and then proceeds to protect the network by [20]. Detection and Accusation of Packet Forwarding Misbehavior in Mobile Ad-Hoc networks using flow of conservation mechanism and done with protocol less implementation [2]. WAP: Wormhole Attack Prevention Algorithm in Mobile Ad Hoc Networks without using any specialized hardware wormholes can be detected and isolated within the route discovery phase [21]. A Reliable and Secure Framework for Detection and Isolation of Malicious Nodes in MANET: This security framework involves detection of malicious nodes by the destination node, isolation of malicious nodes by discarding the path and prevention data packets by using dispersion techniques [22]. A Cooperative Black hole Node Detection Mechanism for ADHOC Networks [23]. DPRAODV: A Dynamic learning system against black hole attack in aodv based Manet [24], Shared Information Based Security Solution for Mobile Ad Hoc Networks [25].

3. ALGORITHM

The brief overview on the design aspects of the algorithm with AODV protocol implementation, a simple acknowledgement approach and principle of flow conservation is discussed as follows:

The proposed algorithm aims at efficient data forwarding in network and in that process monitors the misbehaving nodes or routes, so that such nodes or routes are avoided in data forwarding. There can be three functional modules for execution of the work as sender module, receiver module and intermediate node modules. The proposed system is developed by using a simple acknowledgement approach with two way communications. Once the sender sends the message it waits for the acknowledgement back from the receiver to confirm that the message has reached the receiver or not. Also the particular data frame formats which specify the various fields in the data and acknowledgement frames are presented. The routing takes place according to an on-demand protocol like AODV (Ad hoc on demand distance vector protocol). The malicious behavior which exhibits significant packet dropping is identified by principle of flow conservation. So only, the approach is a combination of 3 techniques as (ACK+AODV+ PFC).

The following list of events takes place when the data has been sent from the sender node:

1. AODV protocol initiates routing and selects the path based on the highest destination sequence number:
When a source has data to transmit to an unknown destination, it broadcasts a Route Request (RREQ) for that destination. At each intermediate node, when a RREQ is received a route to the source is created. If the receiving node has not received this RREQ before, is not the destination and does not have a current route to the destination, it rebroadcasts the RREQ. If the receiving node is the destination or has a current route to the destination, it generates a Route Reply (RREP). The RREP is unicast in a hop-by-hop fashion to the source. As the RREP propagates, each intermediate node creates a route to the destination. When the source receives the RREP, it records the route to the destination and can begin sending data. If multiple RREPs are received by the source, the route with the shortest hop count is chosen. As data flows from the source to the destination, each node along the route updates the timers associated with the routes to the source and destination, maintaining the routes in the routing table. If a route is not used for some period of time, a node cannot be sure whether the route is still valid; consequently, the

node removes the route from its routing table. If data is flowing and a link break is detected, a Route Error (RERR) is sent to the source of the data in a hop-by-hop fashion. As the RERR propagates towards the source, each intermediate node invalidates routes to any unreachable destinations. When the source of the data receives the RERR, it invalidates the route and reinitiates route discovery if necessary [26].

2. Sender connects to the nearest intermediate node.
The snippet below is describes the procedure of connecting to intermediate node:

```
Socket soc=null;
try
{
  soc=new Socket(currentnode,currentport);
}
catch(Exception e)
{
  if (currentport==4000)
        OptionPane.showMessageDialog(jf,"Error","connection
        error",JOptionPane.ERROR_MESSAGE);
  else
        JOptionPane.showMessageDialog(jf,"Error while connecting to
        centernode2","Error in connection",JOptionPane.ERROR_MESSAGE);
```

3. Dividing message/data into packets.
First the length of the msg is calculated, if it is less that 48 bytes then it generates the data frame according to the data frame format and sends it. Else the msg is divided in to packets of 48byte each. The pseudo code for the same is as follows:

```
st=0, end=48, split=0;
len=200, len1=len;
extract the first 48 bytes
…
While ( len<=48 )
 {
  len1=len-48
  if(len1<=48)
   {
    extract ( end, len )
    …  }
 else
  {
   split=end+48
   extract ( end, split )
    … }
 }
```

The variable *st* will point to the start of the message, *end* will be initialized to 48, *and split* variable keeps adding 48 to the *end*, to point to the next position where the message has to be *split*.

4. Creates data frame with destination address, sender name, hash code and message.
The data frame contains the following fields as shown below:

- Destination address: is taken from the text field, entered by the user.
- Host name: it is obtained using the following snippet of code.

```
InetAddress inta=InetAddress.getLocalHost();
Sender's hostname=ineta.getHostName();
```

- Hash code: the function *hashing()* is called with *msg* as the parameter, which calculates and returns the hash code.
- Message: *msg* is taken from the text area that is either manually entered by the user or browsed and copied from a text file.

5. Sends the packet.
Once the connection is established, BufferedReader and BufferedOutputStream are used to create the input and output stream that sends and receives packets in bytes. Functions *write ()* and *read ()* are used for sending the packet and receiving the acknowledgement.

6. Waits for acknowledgement.
The sender keeps waiting till acknowledgement is received from the intermediate node. The function *read ()* reads the acknowledgement written by the intermediate node to the sender in to the string object *chstr*, and returns the number of bytes read. The following snippet shows the infinite loop that is used for waiting.

```
While (true)                              ///read ACK
{
readcnt=in1.read(chstr);
if(readcnt <=0)
        continue;
else
        break;
}
```

7. Calculates time taken and the number of packets lost.
The moment the message is sent the time is saved in *start*, which is long variable, and once the acknowledgement has reached the time is again noted in the long variable *end*.

```
Start=System.currentTimeMillis();
end=System.currentTimeMillis();
```

The total time taken for the message to be sent and acknowledgement to reach back is calculated from *end-start.*

Every time a packet is sent, there is a counter *cpkt*, which is incremented. If the total time taken exceeds the wait time limit which is 20msec, a counter *cmiss* that keeps count of packets lost is incremented. This uses the principle of flow conservation for calculating the (*cmiss/cpkt)* ratio which is explained in the following step.

8. Chooses the intermediate node.
Once the whole message is sent, a packet called *"done"* is sent by the sender to mark the end of the message. If the ratio of (*cmiss/cpkt)* exceeds 20%, the link is said to be misbehaving. And if the acknowledgement field that is extracted from the *ack* packet sent by the destination matches "CONFIDENTIALITY LOST" then we consider that the message is modified. If the ratio of (*cmiss/cpkt)* is less than 20% and the acknowledgement field extracted is *"ACK"* then the link is considered to be working properly. Thus the sender displays appropriate information message indicating the behaviour of the link.

If the link is misbehaving or the confidentiality of the message is lost, there has to be a switch in the intermediate node used. This is done so that in the next session, a faithful communication is carried out. In case the link is learnt to be working properly then the same link is used for the further sessions of sending messages.

4. EXPERIMENT ANALYSIS AND RESULTS

The proposed algorithm was practically implemented and tested in a lab scenario. Through the experimental analysis it is found that the algorithm exactly shows the results for two attacks. To analyze the algorithm with three mechanisms combined as (AODV+ACK+PFC), two laptops

are connected at both the ends in between 30 numbers of intermediate nodes with WI-FI connection.

The underlying MAC protocol defined by IEEE 802.11g was used with a channel data rate of 2.4 GHZ. The data packet size can vary up to 1024 bytes. The wireless transmission range of each node was 100 m. Traffic sources of constant bit rate (CBR) based on TCP have been used. The following Table 1 shows the results for the experiment conducted:

Table 1. Summary of Results

Number of nodes	Cpkt	Cmiss	Cmiss/Cpkt	Time taken in Seconds	Link Status
30	158	4	0.025	1000	Proper
30	4	4	1.0	1065	Misbehaving

According to Table 1 results, the algorithm exactly works in the way as explained in previous section. The ratio (Cmiss/Cpkt) will be taken as limit of tolerance, which is fixed to 20 % and the transmission time is fixed to 20 ms (RTT=*end-start*). The total number of nodes considered is 30, wherein the and nodes will be treated as source and destination respectively. When a message is transmitted from source to destination, every time a packet is sent, the counter Cpkt will be incremented by one. Then the total size of the first message sent will be 158 bytes. After the message is received by the destination node, it prepares acknowledgement frame accordingly and sends back to source node. If this transmission time of sending back the acknowledgement exceeds the time limit of 20 ms i;e RTT, then the link is said to be misbehaving other wise if the acknowledgement is received within 20 ms , then the link is properly behaving. The above results show that, the first message is 158 bytes of length and it has lost 4 packets. So the ratio will be calculated to 0.025, which is less than 0.2 (20%), the limit of tolerance for which PFC is applied. The transmission time is also < 20 ms, as shown in Table 1. Then the link is said to be working properly without any malicious nodes presence. In the similar way for the second message, which is of 4 bytes in length, has lost 4 packets, which leads to the ratio to be 1.0, which is greater than 0.2. The transmission time also exceeds 20 ms, so only the link is said to be misbehaving, where the presence of malicious node is identified and prevented.

4.1. Performance Analysis

We have considered two of the network parameters for evaluating the performance with the combined (AODV+ACK+PFC) scheme. Further it can be extended to a few more parameters based upon the node density in the network. The algorithm can also be extended to identify and prevent few more network layer attacks.

- Packet delivery ratio – the ratio of the number of packets received at the destination and the number of packets sent by the source.

- Routing overhead – The number of routing packets transmitted per data packet delivered at the destination.

The following table 2 shows the comparison of results by the proposed combined approach with that of a nearest approach namely (2ACK+DSR) approach [18]. The (2ACK +DSR) approach is taken because it also uses a 2 hop acknowledgement scheme. The parameter evaluation is done for two of the network parameters as mentioned above.

Table 2. Comparison Results

Approach	Link Status	Packet Delivery ratio	Routing Overhead
AODV+ACK+PFC	Proper	100%	Low
	Misbehaving	95%	Low
2ACK+DSR[18]	Proper	98%	High
	Misbehaving	91%	High

From table 2 comparison results for the link status considered as working properly and misbehaving it is observed that even when the misbehaviour is high the packet delivery ratio is 100% by (AODV+ACK+PFC) scheme compared to (2ACK+DSR). For example, the (2ACK+DSR) scheme delivered over 91% of data packets even when misbehaviour ratio (pm) = 0.4. The rest of the packets were dropped because no well-behaved routes could be found from the source to the destination [18].

We compared the routing overhead of the (2ACK+DSR) with that of (AODV+ACK+PFC) scheme. The overhead of 2ACK increases with the increase of misbehavior percentage. This is because more RERR (the misbehavior report) and RREQ packets are sent to report misbehaviors and to find alternate routes in a more hostile network environment [18]. The high routing overhead in the (2ACK+DSR) is due to the transmission of extra acknowledgment packets. The extra routing overhead of the AODV scheme is caused due to the extra route discovery process. Since AODV maintains only active routes to control traffic, the overhead will be minimized. Also the use of flow conservation mechanism reduces the nodes overhearing.

Another important fact can be considered with respect to the approach is the power consumption of the nodes in the network. When compared to (2ACK +DSR) approach which has got more overhead and less packet delivery ratio, obviously uses more power for two hop acknowledgement. In this proposed scheme since the acknowledgement is simple one hop, the transmission time will get reduced. The overall transmission for sending and receiving data happens in just few milliseconds, the time constraint is also overcome and reduces power consumption.

5. CONCLUSIONS

The MANETS security issues foster new ideas and approaches as it has got potential widespread applications in military and civilian communications. In these networks there will be more dependence on the cooperation of all its nodes to perform networking functions. Thus, makes it highly vulnerable to malicious nodes. One such misbehavior is related to routing of packets. When such misbehaving nodes take part in the route discovery process, but refuse to forward the data packets, routing performance may be degraded severely. In this paper, we have investigated the performance degradation caused by such malicious nodes (misbehaving) in MANETS. We have proposed and evaluated a technique called, (AODV+ACK+PFC) to detect and mitigate the effect of such routing misbehavior.

An immediate enhancement for this scheme can be done by evaluating for more number of nodes and network parameters. Through simulations it can be compared with the nearest methods. Further the scheme can also be extended for identifying and preventing more number of network layer attacks; so that the approach can be made more robust against attacks.

ACKNOWLEDGEMENTS

The authors would like to thank all nears and dears who have been source of inspiration to them.

REFERENCES

[1] Emmanouil A. Panaousis, Tipu A. Ramrekha, Grant P. Millar and Christos Politis, " Adaptive and Secure Routing protocol for Emergency Mobile Ad Hoc Networks", International Journal of Wireless and Mobile Networks (IJWMN), Vol-2, No-2, May 2010.

[2] Oscar F. Gonzalez,God win Ansa, Michael Howarth, and George Pavlou, "Detection and Accusation of Packet Forwarding Misbehavior in Mobile Ad-Hoc networks", Journal of Internet Engineering, Vol-2, 2008, pp. 1.

[3] H. Yang, J. Shu, X. Meng, and S. Lu, "SCAN: Self-organized network-layer security in mobile ad hoc networks," IEEE Journal on Selected Areas in Communications, Vol. 24, No. 2, February 2006, pp. 261-273.

[4] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. A. Olsson, "Detecting disruptive routers: a distributed network monitoring approach", in Proc. Symposium on Security and Privacy, May 1998.

[5] Y. C. Hu, D. B. Johnson, and A. Perrig, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," in Proc. 4th IEEE workshop on Mobile Computing Systems & Applications, New York, USA, June 2002.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers", in Proc. ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, London, UK, September 1994.

[7] M. Guerrero-Zapata and N. Asokan, "Securing ad hoc routing protocols", in Proc. 3rd ACM Workshop on Wireless Security, New York, USA, 2002

[8] M. Guerrero-Zapata, "Secure ad hoc on-demand distance vector (SAODV) routing", Internet Draft, IETF Mobile Ad Hoc Networking Working Group, February 2005.

[9] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A secure routing protocol for ad hoc networks", in Proc. 10th IEEE International Conference on Network Protocols, Paris, France, November 2002.

[10] C. E. Perkins, "Ad hoc on-demand distance vector (AODV) routing", Request For Comments (RFC) 3561, July 2003, Available online at: http://www.ietf.org/rfc/rfc3561.txt .

[11] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks", in Proc. 8th ACM International Conference on Mobile Computing and Networking, Atlanta, USA, September 2002.

[12] D. B. Johnson, D. A Maltz, and Y. C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)", Internet Draft, IETF MANET Working Group, July 2004.

[13] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels", in Proc. IEEE Symposium on Security and Privacy, Berleley, USA, May 2000.

[14] S. Buchegger, and J. Le Boudec, "Performance analysis of the CONFIDANT protocol," in Proc. 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Lausanne, Switzerland, June 2002.

[15] B. Awerbuch, D. Holmes, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilient to Byzantine failures", in Proc. 3rd ACM Workshop on Wireless Security, New York, USA, 2002.

[16] Y. Zhang, and W. Lee, "Intrusion detection in wireless ad-hoc networks", in Proc. 6th ACM International Conference on Mobile Computing and Networking, Boston, USA, August 2000.

[17] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks", in Proc. 6th ACM International Conference on Mobile Computing and Networking, , Boston, USA, August 2000, pp. 255-265.

[18] Kejun Liu, Jing Deng, Pramod K. Varshney, and Kashyap Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs", IEEE Transactions on Mobile Computing, Vol-6 , Issue 5, May 2007, pp. 536-550.

[19] T.V.P.Sundararajan, Dr.A.Shanmugam," Performance Analysis of Selfish Node Aware Routing Protocol for Mobile Ad Hoc Networks", ICGST-CNIR Journal, Volume 9, Issue 1, July 2009.

[20] Nasser, N.; Yunfeng Chen, "Enhanced Intrusion Detection System for Discovering Malicious Nodes in Mobile Ad Hoc Networks", IEEE International Conference on Communications, ICC apos; Vol-07 , Issue 24-28, June 2007 , pp.1154 – 1159.

[21] Sun choi, Doo-young Kim, Do-hyeon Lee, Jae-il Jung, "WAP: Wormhole Attack Prevention Algorithm in Mobile Ad Hoc Networks", International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, Vol-0 , ISBN = {978-0-7695-3158-8}, 2008, pp.343-348 .

[22] S.Dhanalakshmi, Dr.M.Rajaram, "A Reliable and Secure Framework for Detection and Isolation of Malicious Nodes in MANET", IJCSNS International Journal of Computer Science and Network Security, Vol-8, No.10, October, 2008.

[23] Moumita Deb, "A Cooperative Black hole Node Detection Mechanism for ADHOC Networks", Proceedings of the World Congress on Engineering and Computer Science, 2008.

[24] Payal N.Raj and Prashant B. swadas, " DPRAODV: A Dynamic learning system against blackhole attack in AODV based MANET", International Journal of Computer ScienceIssues, Vol-2, 2009.

[25] Shailender Gupta and Chander Kumar, "Shared Information BasedSecurity Solution forMobile Ad Hoc Networks", International Journal of Wireless and Mobile Networks (IJWMN), Vol-2, N0-1, Feb 2010.

[26] Ian D. Chakeres, Elizabeth M. Belding-Royer, "AODV Routing Protocol Implementation Design", Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04), Vol-7, 2004, pp. 698 – 703.

Authors

G. S. Mamatha has completed her MTech from Visveswaraya technological University in the year 2004 in the field of Computer Science and Engineering. She is currently pursuing her Ph.D in Avinashi Lingam University for women; Coimbatore. She has 6 Years of academic experience in R.V.C.E. She is a member of ISTE. Her area of research includes Network security, Software Engineering and Multimedia systems.

Dr. S. C. Sharma is Vice-Chancellor of Tumkur University, Tumkur, Karnataka. He pursued PhD in Mechanical Engineering from Mysore University, Doctor of Science in CSE from Kuvempu University, Doctor of Engineering from Avinashi Lingam University. State Government of Karnataka Nominee as Member of Execut Research Review Committee, Associate Editor, Research Journal Editorial Boar Avinashi Lingam University, Coimbatore, Tamil nadu. His specialization are materials, Metal Casting, Internet Enabled Automated System.