Improving AODV Protocol against Blackhole Attacks

Nital Mistry, Devesh C Jinwala, *Member, IAENG,* Mukesh Zaveri

***Abstract*—The proliferation of Mobile Adhoc Networks (MANETs) help to realize the nomadic computing paradigm with ubiquitous access. Though they ensure self-maintainable, dynamic and temporary topology, the MANETS also suffer from constraints in power, storage and computational resources. In addition, the pervasiveness, ubiquity and the inherent wireless nature, warrant appropriate security provisions in these networks that becomes difficult to support, amidst the lack of sufficient resource strengths. As a result, the MANETs are more vulnerable to various communications security related attacks.**

In this paper, therefore, we attempt to focus on analyzing and improving the security of one of the popular routing protocol for MANETS viz. the Ad hoc On Demand Distance Vector (AODV) routing protocol. Our focus specifically, is on ensuring the security against the Blackhole Attacks. We propose modifications to the AODV protocol and justify the solution with appropriate implementation and simulation using NS-2.33. Our analysis shows significant improvement in Packet Delivery Ratio (PDR) of AODV in presence of Blackhole attacks, with marginal rise in average end-to-end delay.

***Index Terms*—AODV, Blackhole attack, MANET, Routing protocols, Security.**

I. Introduction

In the present era, the study of MANETs has gained a lot of interest of researchers due to the realization of the nomadic computing paradigm [1]. A Mobile Adhoc Network (MANET), as the name suggests, is a self-configuring network of wireless and hence mobile devices that constitute a network capable of dynamically changing topology. The network nodes in a MANET, not only act as the ordinary network nodes but also as the routers for other peer devices [2]. The dynamic topology, lack of a fixed infrastructure and the wireless nature make MANETs susceptible to the security attacks. To add to that, due to the inherent, severe constraints in *power*, *storage* and *computational* resources in the MANET nodes, incorporating sound defense mechanisms against such attacks is also non-trivial. Therefore, the traditional security mechanisms and protocols – including those for the wired networks - are not directly applicable and require a careful relook [2].

We attempt revisiting the routing protocols applicable in MANETs, in this research exercise and investigate whether it is possible to strengthen the existing attempts on devising secure routing protocols for MANETs. The routing protocols are especially susceptible in MANETs because of the major reliance on the cooperative routing algorithms employed for establishing the network routes, with underlying assumptions about the sanctity of the peer network nodes.

The network layer in MANETs is susceptible to various attacks viz. eavesdropping with a malicious intent, spoofing the control and/or data packets transacted, malicious modification/alteration of the packet contents and the Denial-of-service (DoS) attacks viz. Wormhole attacks, Sinkhole attacks, Blackhole attacks [4].

Amongst these, in this paper, we attempt in analyzing and improving the security of the routing protocol AODV [5] against the Blackhole attacks. As we describe further in section 2, several attempts exist in the literature that propose a secure version of AODV to resist the Gray hole and Blackhole attacks. However, as per our view, none of the proposed attempts safeguards AODV against the Blackhole DoS attacks.

We propose an algorithm to counter Blackhole attack against the AODV routing protocol. As testified by our results and analysis described in section 5, we observe that the proposed modification to secure AODV is indeed effective in preventing the Blackhole attacks with marginal performance penalty.

The rest of the paper is organized as follows: In Section 2, we briefly describe the working of the AODV routing protocol, the Blackhole attack and then survey of the related work in the area. In section 3, we discuss our solution to AODV algorithm. In section 4, we discuss the methodology of evaluating our solution and the metrics used to compare the algorithm relative to the existing traditional AODV. In Section 5, we describe the simulation results and analyze the same. Finally, we conclude in Section 6 with future scope.

II. Theoretical Background and Related Work

A. Overview of AODV

AODV is a state-of-the-art routing protocol that adopts a purely *reactive* strategy: it sets up a route on-demand at the start of a communication session, and uses it till it breaks, after which a new route setup is initiated [1]. AODV uses Route Request (RREQ), Route Reply (RREP) control messages in Route Discovery phase and Route Error (RERR) control message in Route Maintenance phase. The header information of this control messages can be seen in detail in [3].

In general, the nodes participating in the communication can be classified as source node, an intermediate node or a

Manuscript received on January 1, 2010.

Nital H. Mistry is with Shree Sad Vidya Mandal Institute of Technology, Bharuch, India – 395007. (e-mail: misnit_22@yahoo.com).

Devesh C. Jinwala is with the S V National Institute of Technology, Ichchhanath, Surat, India – 395007. (e-mail: dcjinwala@acm.org).

Mukesh A. Zaveri is with the S V National Institute of Technology, Ichchhanath, Surat, India – 395007. (e-mail: mazaveri@gmail.com).

destination node. With each role, the behavior of a node actually varies. When a source node wants to connect to a destination node, first it checks in the existing route table, as to whether a fresh route to that destination is available or not. If a fresh enough route is available, it uses the same. Otherwise the node initiates a Route Discovery by broadcasting a RREQ control message to all of its neighbors. This RREQ message will further be forwarded (again broadcasted) by the intermediate nodes to their neighbors. This process will continue until the destination node or an intermediate node having a fresh route to the destination, receives this message. At this stage eventually, a RREP control message is generated. Thus, a source node after sending a RREQ waits for RREPs to be received.

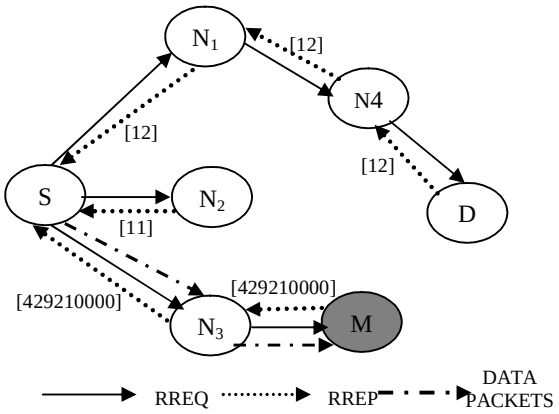


Fig. 1 Traversal of Control Messages in AODV

In Fig. 1, we illustrate a typical scenario of the protocol packet exchanges, depicting the generation and traversal of RREQ and RREP control messages. The node S is assumed to be the source node desiring to communicate with node D. Thus, as per the explanation earlier, node S would generate the RREQ control message and broadcast it. The broadcasted RREQ control message is expected to be received by the nodes N_1, N_2 and N_3.

Assuming that the node N_2 has a route to node D in its route table, the node N_2 would generate a RREP control message and update its routing table with the accumulated hop count and the destination sequence number of the destination node. Destination Sequence Number is a 32-bit integer associated with every route and is used to decide the freshness of a particular route. The larger the sequence number, the fresher is the route [6]. Node N_2 will now send it to node S (Destination Sequence Number is shown in square bracket in Fig. 1). Since node N_1 and node N_3 do not have a route to node D, they would again broadcast the RREQ control message. RREQ control message broadcasted by node N_3 is also expected to be received by node M (assumed to be a malicious node). Thus, node M being malicious node, would generate a false RREP control message and send it to node N_3 with a very high destination sequence number, that subsequently would be sent to the node S.

However, since, the destination sequence number is high, the route from node N_3 will be considered to be fresher and hence node S would start sending data packets to node N_3. Node N_3 would send the same to the malicious node. The RREQ control message from node N_1, would eventually reach node D (destination node), which would generate RREP control message and route it back. However, since the node S has a RREP control message with higher destination sequence number to that route, node S will ignore two genuine RREP control messages.

How the source node processed the incoming RREPs for consideration is shown in Fig. 2. After a source node receives a RREP message, it calls *ReceiveReply(Packet P)* method - one of the crucial function of AODV. The manner in which the RREP control message is handled is explained in the pseudocode of the *ReceiveReply(Packet P)* function of AODV in Fig. 2.

```
At Source Node: AODV

1 ReceiveReply (Packet P){
2  if(P has an entry in Route Table){

3   select Dest_Seq_No from routing table

4     if(P.Dest_Seq_No > Dest_Seq_No){

5       update entry of P in routing table

6       unicast data packets to the route
        specified in RREP
7     }
8     else {
9        discard RREP
10    }
11  }
13  else {

14    if(P.Dest_Seq_No >= Src_Seq_No){

15      Make entry of P in routing table
16    }
17    else {

18      discard this RREP

19    }
20  }
21 }
```

Fig. 2 RecvReply pseudocode

For every RREP control message received, the source node would first check whether it has an entry for the destination in the route table or not. If it finds one, the source node would check whether the destination sequence number in the incoming control message is higher than one it sent last in the RREQ or not. If the destination sequence number is higher, the source node will update its routing table with the new RREP control message; otherwise the RREP control message will be discarded.

In Route Maintenance phase, if a node finds a link break or failure, then it sends RERR message to all the nodes that uses the route.

B. Blackhole Attack

A Blackhole attack is one of the active DoS attacks possible in MANETs. In this attack, a malicious node sends a *false* RREP packet to a source node that initiated the route discovery, in order to pose itself as a destination node or an immediate neighbor to the actual destination node. In such a case, the source node would forward all of its data packets to

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the malicious node, which originally were intended for the genuine destination. The malicious node, eventually may never forward any of the data packets to the genuine destination. As a result, therefore, the source and the destination nodes became unable to communicate with each other [6].

Since AODV treats RREP messages having higher value of destination sequence number to be fresher, the malicious node will always send the RREP having the highest possible value of destination sequence number. Such RREP message, when received by source node is treated afresh, too. The fallout is that there is a high probability of a malicious node attempting to orchestrate the Blackhole attacks in AODV.

C. Related Work

There indeed have been numerous attempts published in the literature that aim at countering the Black attacks. We survey them in the following.

In [7], the authors discuss an approach in which the requesting node waits for the responses including the next hop details, from other neighboring nodes for a predetermined time value. After the timeout value, it first checks in the CRRT (Collect Route Reply Table) table, whether there is any repeated *next-hop-node* or not. If any repeated *next-hop-node* is present in the reply paths, it assumes the paths are correct or the chance of malicious paths is limited. The solution adds a delay and the process of finding repeated next hop is an additional overhead.

In [8], the authors discuss a protocol viz. DPRAODV to counter the Blackhole attacks. DPRAODV checks to find whether the RREP_Seq_No is higher than the threshold value. In this protocol, the threshold value is dynamically updated at every time interval. If the value of RREP_Seq_No is found to be higher than the threshold value, the node is suspected to be malicious and is added to a list of blacklisted nodes. It also sends an ALARM packet to its neighbors with information about the blacklisted node. Thus, the neighbor nodes know that RREP packets from the malicious node are to be discarded. That is, if any node receives the RREP packet, it looks over the list to check the source of the received message. If the reply is from the suspected node, the same is ignored. Thus, the protocol though successful, suffers from the overhead of updating threshold value at every time interval and generation of the ALARM packets. The routing overhead, as a result is higher.

In [9], the authors discuss a protocol that requires the intermediate nodes to send RREP message along with the next hop information. When the source node get this information, it sends a RREQ to the next hop to verify that the target node (i.e. the node that just sent back the RREP packet) indeed has a route to the intermediate node and to the destination. When the next hop receives a *FurtherRequest*, it sends a *FurtherReply* which includes the check result to the source node. Based on information in *FurtherReply*, the source node judges the validity of the route.

In this protocol, the RREP control packet is modified to contain the information about next hop. After receiving RREP, the source node will again send RREQ to the node specified as next hop in the received RREP. Obviously, this increases the routing overhead and end-to-end delay. In addition, the intermediate node needs to send RREP message twice for a single route request.

In [10], the authors describe a protocol in which the source node verifies the authenticity of a node that initiates RREP by finding more than one route to the destination. When source node receives RREPs, if routes to destination shared hops, source node can recognize a safe route to destination.

All solutions discussed above, involve additional overhead on either/both intermediate and destination nodes in one or the other way. Since the mobile nodes in MANETs suffer from limited battery life, processing power and storage, it is essential to devise a protocol that aims at reducing the overhead on intermediate and destination nodes. In addition, the process of selecting secure root, should involve minimum possible rise in end-to-end delay.

III. The Modified AODV

A. The Proposed Solution

The solution that we propose here is designed to prevent any alterations in the default operations of either the intermediate nodes or that of the destination nodes. The approach we follow, basically only modifies the working of the source node, using an additional function *Pre_ReceiveReply(Packet P)*. The pseudocode of the same is shown in Fig. 3. Apart from this, we also added a new table Cmg_RREP_Tab, a timer MOS_WAIT_TIME and a variable Mali_node to the data structures in the default AODV protocol, as explained further.

In the original AODV protocol, by default, the source node accepts the first fresh enough RREP request coming to it. As compared, in our approach (Fig. 3), we store all the RREPs in the newly created table viz. Cmg_RREP_Tab until the time, MOS_WAIT_TIME. Based on the heuristics, we initialize MOS_WAIT_TIME to be half the value of RREP_WAIT_TIME – the time for which source node waits for RREP control messages before regenerating RREQ. In our solution, the source node after receiving first RREP control message waits for MOS_WAIT_TIME. For this time, the source node will save all the coming RREP control messages in Cmg_RREP_Tab table.

Subsequently, the source node analyses all the stored RREPs from Cmg_RREP_Tab table, and discard the RREP having presumably very high destination sequence number. As before, the node that sent this RREP is suspected to be the malicious node. Once, such malicious node is identified, our solution selects a reply having highest destination sequence number from Cmg_RREP_Tab table. It does so, by calling our own method viz. the *Pre_ReceiveReply()* method.

The proposed solution maintains the identity of the malicious node as Mali_node, so that in future, it can discard any control messages coming from that node. Now since malicious node is identified, the routing table for that node is not maintained. In addition, the control messages from the malicious node, too, are not forwarded in the network. Moreover, in order to maintain freshness, the Cmg_RREP_Tab is flushed once an RREP is chosen from it.

Thus, the operation of the proposed protocol is the same as that of the original AODV, once the malicious node has been detected. This is testified by the call to the default ADOV

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

routine R*eceiveReply(Packet p)* viz. in line number 14 in Fig. 3.

B. Analysis

The overhead in the proposed algorithm is in the form of a new table of size 6 bytes, a variable Mali_Node of size 2 bytes and a timer variable of size 10 bytes. Thus, the additional memory overhead is not more than 20 Bytes as compared to the AODV. This is worthy for the rise in Packet delivery Ratio (PDR) (discussed in Section IV B).

The Proposed Algorithm : at Source Node:

```
1 Pre_ReceiveReply (Packet P){

2    t0 = get(current time value)
3    t1=t0 + MOS_WAIT_TIME
4    while(CURRENT_TIME <= t1){
5      Store P.Dest_Seq_No and P.NODE_ID In
       Cmg_RREP_Tab table
     }
6    while (Cmg_RREP_Tab is not empty) {
7      Select Dest_Seq_No from table
8      if (Dest_Seq_No >>>=Src_Seq_No){
9          Mali_Node=Node_Id
10         discard entry from table
11     }
12   }
13   select Packet q for Node_Id having
     highest value of Dest_Seq_No
14   ReceiveReply(Packet q)
15  }
```

Fig. 3 Pseudocode of Our Solution

In addition, the solution does not add any control message to existing AODV neither it needs to even regenerate any control messages. So, there are minimal chances of rise in Normalized Routing Overhead i.e. in the ratio of number of control packets to data transmissions in a simulation. The overhead in time in the proposed solution is in terms of the MOS_WAIT_TIME and the time required to execute *Pre_ReceiveReply()*.

IV. Methodology of Evaluation

A. Simulation Environment

For the simulations, we use NS-2 (v-2.33) network simulator. NS-2 provides faithful implementations of the different network protocols. At the physical and data link layer, we used the IEEE 802.11 algorithm. The channel used is Wireless Channel with Two Ray Ground radio propagation model. At the network layer, we use AODV as the routing algorithm. Finally, UDP is used at the transport layer. All the data packets are CBR (continuous bit rate) packets. The size of the packet is 512 bytes. The packets transmission rate is 0.2 Mbps.

The connection pattern is generated using *cbrgen* and the mobility model is generated using *setdest utility. Setdest* generates random positions of the nodes in the network with specified mobility and pause time. The terrain area is 800m X 800m with number of nodes varying from minimum 10 to maximum 80 with chosen maximum speed up to from 10 m/s to 70 m/s. The simulation parameters are summarized in table 1.

Each data point represents an average of ten runs. The same connection pattern and mobility model is used in simulations to maintain the uniformity across the protocols.

Table 1 Simulation Parameters

Parameter	*Value*
Simulator	Ns-2(ver.2.33)
Simulation Time	100 s
Number of nodes	10 to 80
Routing Protocol	AODV
Traffic Model	CBR
Pause time	2 s
Mobility	10 - 70 m/s
Terrain area	800m x 800m
Transmission Range	250m
No. of malicious node	1

B. Metrics used for Simulation

To analyze the performance of our solution, various contexts are created by varying the number of nodes and node mobility. The metrics used to evaluate the performance of these contexts are given below.

Packet Delivery Ratio: The ratio between the number of packets originated by the "application layer" CBR sources and the number of packets received by the CBR sink at the final destination.

Average End-to-End Delay: This is the average delay between the sending of the data packet by the CBR source and its receipt at the corresponding CBR receiver. This includes all the delays caused during route acquisition, buffering and processing at intermediate nodes, retransmission delays at the MAC layer, etc. It is measured in milliseconds [8].

V. Simulation Results and Analysis

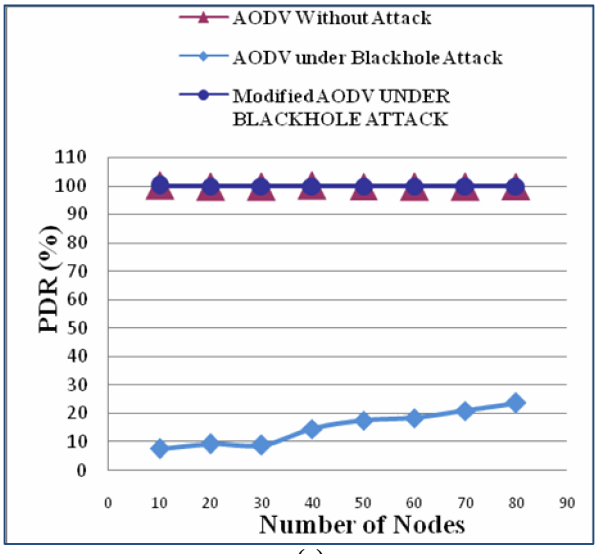
To evaluate the packet delivery ratio, End-to-End Delay and Normalized Routing Overhead; simulation is done with nodes with the source node transmitting maximum 1000 packets to the destination node. Fig. 4 shows the graphs when network size (number of nodes) is varying. It can be seen from the Fig. 4 (a), that PDR of AODV drops by 81.812 % in presence of Blackhole attack. The same increases by 81.811 % when our solution is used in presence of Blackhole attack. At the same time, Fig. 4 (b) shows that the rise in End-to-End delay is 13.28 %.

Fig. 4 shows the graphs when mobility of nodes is varying. It can be seen from the Fig. 5 (a), that PDR of AODV drops by 70.867 % in presence of Blackhole attack. The same increases by 70.877 % when our solution is used in presence of the attack. At the same time, Fig. 5 (b) shows that the rise in End-to-End delay 6.28 %.

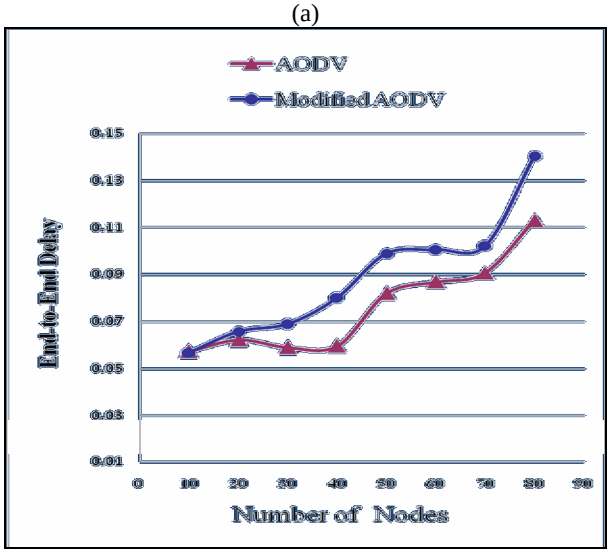
VI. Conclusion and Future Work

With the fact that the default AODV protocol is susceptible to the Blackhole attacks, in this research exercise, we attempt at investigating the existing solutions for their viability.

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a)

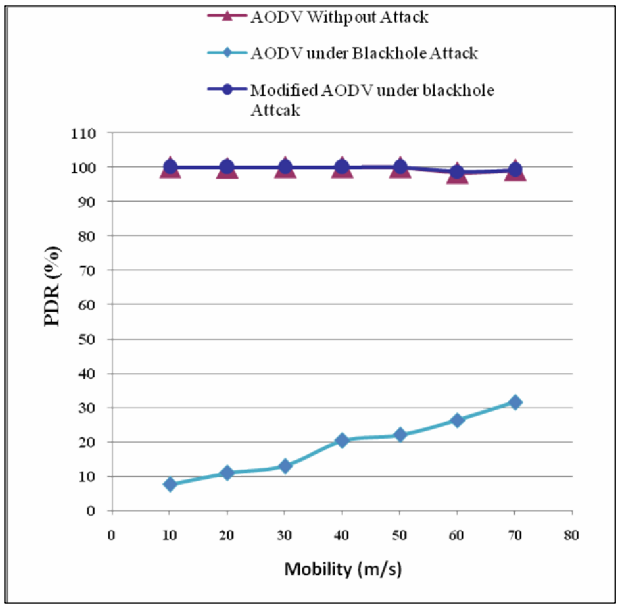


(b)

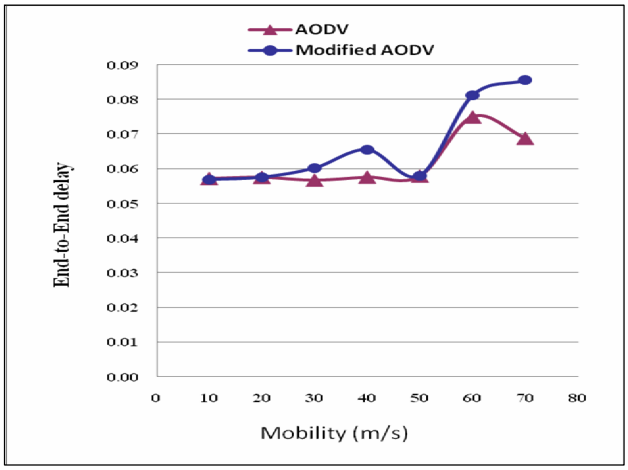
Fig. 4 Effect of Network Size

Having justified a need for further improvements, we propose an algorithm to counter the Blackhole attack on the routing protocols in MANETs. We successfully analyze and demonstrate that with trivial additional overhead in terms of a new MOS_WAIT_TIME variable and a new Cmg_RREP_Tab table, we are able to counter the Blackhole attacks on the AODV protocol. From the experimental results, we conclude that the proposed solution achieves a very good rise in PDR with acceptable rise in end-to-end delay. Moreover, the proposed algorithm does not entail any hidden overhead on either the intermediate nodes or the destination nodes. Thus, as compared to the other approaches discussed in section II, we believe the proposed algorithm is simple and efficient in implementation.

We also emphasize that though the proposed algorithm is implemented and simulated for the AODV routing algorithm, it can also be further trivially extended for use by any other routing algorithms, as well. As part of our future endeavor, we aim to study the impact of varying pause time on the protocol efficiency. In addition, we would also attempt to investigate the impact of varying network size and node mobility on Normalized Routing Overhead in the protocol.



(a)



(b)

Fig. 5 Effect of Mobility

REFERENCES

[1] Gianni A. Di Caro, Frederick Ducatelle, Luca M. Gambardella. "A simulation study of routing performance in realistic urban scenarios for MANETs". In: *Proceedings of ANTS 2008, 6th International Workshop on Ant Algorithms and Swarm Intelligence*, Brussels, Springer, LNCS 5217, 2008

[2] Ebrahim Mohamad, Louis Dargin. "Routing Protocols Security." In: Ad Hoc Networks". A Thesis at Oakland University School of Computer Science and Engineering.

[3] C. Perkins. "(RFC) request for Comments-3561", Category: Experimental, Network, Working Group, July 2003.

[4] N.H.Mistry, D. C. Jinwala, M. A. Zaveri. "Prevention of Blackhole Attack in MANETs". In: *Proceedings of EPWIE-2009, Gujarat, India*, pp.89-94, July 2009.

[5] Charles E. Perkins and Elizabeth M. Royer. "Ad-Hoc On-Demand Distance Vector Routing." In: *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pages 90–100, February 1999.

[6] Satoshi Kurosawal, Hidehisa, Nakayama, Nei Kato, Abbas Jamalipour and Yoshiaki Nemoto. "Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method." In:

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

International Journal of Network Security, Vol. 5, No.3, pp.338–346, Nov. 2007.

[7] Latha Tamilselvan, V Sankaranarayanan. "Prevention of Blackhole Attacks in MANET." In: *Proceedings of the 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*, pp. 21-21, Aug. 2007.

[8] Payal N. Raj, Prashant B. Swadas. "DPRAODV: A Dyanamic Learning System Against Blackhole Attack In Bodv Based Manet." In: *International Journal of Computer Science Issues*, Vol.2, pp 54-59, 2009.

[9] H. Deng, W. Li, and D. P. Agrawal. "Routing Security in Adhoc Networks." In: *IEEE Communications Magazine*, Vol. 40, No. 10, pp. 70-75, Oct. 2002.

[10] M. A. Shurman, S. M. Yoo, and S. Park, "Black hole attack in wireless ad hoc networks." In: *Proceedings of the ACM 42nd Southeast Conference (ACMSE'04)*, pp 96-97, Apr. 2004.

ISBN: 978-988-18210-4-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)