WDA: Wormhole Attack Detection Algorithm based on measuring Round Trip Delay for wireless Ad hoc networks

Anuradha[1] Dr. Puneet Goswami[2] Gurdeep Singh[3]
[1]M.Tech [2]Professor & HOD
[1]Department of Computer Science & Engineering
[1,2]Galaxy Global Imperial Technical Campus [3]MMU, Sadopur, Ambala

Abstract— The recent advancements in the wireless arena and their wide-spread utilization have introduced new security vulnerabilities. The wireless media being shared is exposed to outside world, so it is susceptible to various attacks at different layers of OSI network stack. For example, jamming and device tampering at the physical layer; disruption of the medium access control (MAC) layer; routing attacks like Blackhole, rushing, wormhole; targeted attacks on the transport protocol like session hijacking, SYN flooding or even attacks intended to disrupt specific applications through viruses, worms and Trojan Horses. Wormhole attack is one of the serious routing attacks amongst all the network layer attacks launched on MANET. Wormhole attack is launched by creation of tunnels and it leads to total disruption of the routing paths on MANET. In this paper, Wormhole detection algorithm (WDA) is proposed based on modifying the forwarding packet process that detects and isolates wormhole nodes in ad hoc on demand distance vector (AODV) routing protocol.

Keywords: Wormhole, MANET, Attack, Detection

I. INTRODUCTION

A mobile ad hoc network (MANET) consists of mobile hosts that can forward packets for neighbors. Every node could be router in these networks and is responsible for organizing and controlling the network. Many critical applications of MANET, such as military tactical communication or emergency rescue operations require a secure cooperative environment [1]. Due to the wireless nature of communications in MANETs, the security threats are more than corresponding wired environment. The unique features of MANET like low profile autonomous terminals, bandwidth constrained and dynamic configuration give unsatisfactory results of effects of applying the security techniques like access control and authentication that are used in wired networks to wireless and mobile networks. Thus, achieving security for MANET has gained significant attention in the past few years.

Among several possible attacks in wireless networks, wormhole is one of the dangerous attacks. In wormhole attack, an attacker intercepts packets at one location and tunnels them to another location within the wireless network. Any routing protocol that relies on network topology for routing packets can't work normally and is prone to wormhole attack. Because of this reason, the detection of wormhole attack has become an essential issue. Wormhole attack when used against an on-demand routing protocol like AODV or DSR increases the probability of choosing routes through the wormhole nodes.

This paper is organized as follows: Section 2 discusses the related work. Section 3 elaborates wormhole attack in detail and detection algorithm is proposed in Section 4. The simulation environment details are shown in section 5 and section 6 concludes our results.

II. RELATED WORK

Some of the previous work for defense against wormhole attack is listed below. Jen et al. [2] provided a Multipath wormhole attack model to prevent wormhole attack in MANETs. MHA (Multipath Wormhole Attack Analysis) consisted of three steps: 1) considering hop count values of all routes; 2) choosing a reliable set of paths for transmitting data; 3) sending data packets randomly by routers through paths calculated in step 2 according to decreasing the level of packet as sent by wormhole tunnel. This method minimizes the level of using the path consisting wormhole nodes even though it can't completely avoid wormhole nodes in the path chosen. The simulations were done on AODV routing protocol and did not use any specialized hardware.

Jain et al. [3] presented a novel trusted-base scheme to detect wormhole attack, where a trust model based on Dynamic Source Routing (DSR) was used to detect wormhole attack. In DSR protocol, the control packets store the address list of each node that it has to traverse. In this scheme, the wormhole attack is identified by using effort-return based trust model in which each node following DSR routing calculated trust levels in other nodes.

Choi et al. [4] introduced wormhole attack prevention (WAP) model for preventing the wormhole attack. In this prevention technique, all nodes need to monitor the neighbor's behavior by using a special list known as neighbor list after broadcasting or forwarding RREQ. From the respond packet, if received, it can detect the path under wormhole attack. Once wormhole node is detected, it is the responsibility of source node to record them in the Wormhole List and avoid them taking part in routing. Furthermore, the WAP method can detect both hidden and exposed attack without any external hardware devices.

In [5], authors developed a simple and efficient distributed algorithm using communication graph for wormhole detection in wireless ad hoc and sensor networks without making unrealistic assumptions. Their algorithm performed well in relatively dense or regular networks but gave false positives in sparse or random networks.

Lu et al. [6] presented Multi-Dimensional Scaling (MDS) scheme in which each node locally collects its neighborhood information and reconstructs the neighborhood sub-graph by MDS. Potential wormhole nodes are detected by validating the legality of the reconstruction of neighborhood sub-graph. Further, a

refinement process is introduced to filter the suspect nodes and to remove false positives.

Chaurasia et al. [7] proposed an efficient method to detect a wormhole attack called modified wormhole detection AODV protocol. Wormhole attack is detected by using number of hops along different paths from source to destination and delay of each node along these paths.

Banerjee et al. [8] proposed a cluster based Wormhole attack avoidance scheme that used DSR as an underlying protocol. In order to avoid attacking during the route discovery phase, hierarchical clustering with a novel hierarchical 32-bit node addressing scheme is used.

Song et al. [9] proposed a statistical approach as defense against wormhole attack. Each sensor node collected the recent history of number of neighbors and detected if the current neighbor count shows abrupt increase as compared to the normal ones.

III. WORMHOLE ATTACK

Wormhole attack is one of the most severe routing attacks in wireless networks. In this attack, an attacker node intercept packets at one location, tunnels them to another node at some other location of the network, where it is retransmitted in the network by a colluding attacker [10]. The tunnel can be established either by using out-of band private wired link or logical link via packet encapsulation technique. The colluding attacker nodes create an illusion that two remote regions of a network are in direct connection through nodes that appear to be neighbors thus violating security. Based on the tunneling mechanism used, wormhole attack can be classified into following two categories:

- Out-of-band wormhole: In this, the colluders create a direct link between the two nodes so it requires specialized hardware to support the communication between them. The tunneled packets arrive faster due to high speed private link than the multi-hop packets. It enhances capacity of the communication channel.
- In-band wormhole: It does not require any external communication medium, specialized hardware or special or special routing protocol. The packets reach much slower than out-of-band wormhole. It can be launched by any node in the network to another colluder and are more likely to occur in real world. It consumes network bandwidth and capacity thus degrading network performance.

The existence of wormhole tunnel disrupts the normal routing procedure in several ways. The attackers can attract a significant amount of network traffic from their surroundings. If the attackers do not drop any data packets and keep the wormhole tunnel active all the times, they are rendering useful service to the network. But actually they are responsible for disrupting the normal flow of data packets by selectively dropping, spoofing or modifying packets, recording packets for later analysis and generating unnecessary routing activities by making wormhole link up/down periodically. The paths attracted by wormhole nodes are having different advertised and actual routes. The advertised routes are much shorter than the actual routes which go through the wormhole tunnel. For instance, consider the path between nodes S and D in Figure 1. The advertised route from source to destination traverses nodes in the order from S to W1 to E to D, but the actual route taken by packets goes through nodes S, W1, B, W2, E and D. In this way, actual path between nodes S and D is different from that of advertised path. As seen from Figure 1, actual path and the advertised path between nodes F and G stay the same, but overlap with the wormhole tunnel. Abnormal network behavior exhibited by wormhole tunnel can be further exploited to devise defense mechanism against it.

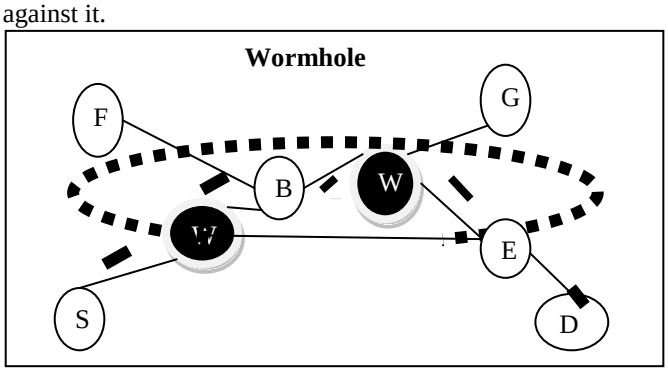


Fig. 1: Wormhole attack

IV. WORMHOLE DETECTION ALGORITHM (WDA)

We have used Ad hoc on-demand distance vector routing (AODV) protocol for wormhole detection. In our proposed wormhole detection algorithm, a wormhole tunnel is suspected between two nodes if the RTD (Round Trip Delay) between the nodes is greater than the threshold value. RTD calculated by a node for a packet is the time difference between the route request (RREQ) packet sent to the neighboring node and the route reply (RREP) message received. The source node is responsible for calculating the RTD between all the successive nodes in the path established during the route discovery phase. RREP message format is appended with field RTD_value. Each intermediate node receiving RREP inserts their RTD values to assist in calculating delay by the source node.

$$RTD(N) = \left(Time\ of\ receiving\ RREP\ by\ node\ N - Time\ of\ sending\ RREQ\ by\ node\ N\right)$$

The delay is expected to increase with increasing wormhole nodes because more the number of paths getting attracted towards the wormhole, more traffic pass through the wormhole tunnel. This will increase the delay at these tunnel nodes. Thus, delay criterion suits well for detecting the wormhole attack in this case. Enough space is allocated for

RTD_value field depending on number of hops.

$$\text{Size of RTD\_value } field = \text{Bits to represent RTD} * (\text{Hop count} - 1)$$

A node then forwards the RREP message to the next hop along the reverse path. Each intermediate node receiving RREP message will calculate RTD and embed it into RTD_value field and forward it to the next hop along the reverse path. When the RREP message reaches the source node, it contains the RTD values of all the intermediate nodes. RTD between the adjacent nodes N1 and N2 is calculated by the source node as:

$$RTD\ (N1, N2) = RTD\ (N1) - RTD\ (N2)$$

A wormhole is suspected if the RTD between two nodes is greater than the RTD between the other nodes in the established path. But RTD can also increase due to longer delays caused by congestion or queuing delays. Therefore a confirmation mechanism is needed to detect whether the long delays caused is due to congestion or queuing delays. Queue status is used for detecting congestion at a node level. After successful detection of wormhole attack, the source node broadcasts an alert message to its neighborhood with wormhole node id to isolate wormhole nodes from the network so that their packets are discarded. The detailed algorithm for processing RREQ and RREP by a mobile node is outlined as below:

AODV Receive RREQ

```
{
if (source address and broadcast ID pair already in request buffer)
        discard RREQ
else
        add source address and broadcast ID pair to request buffer
if (no route to source in routing table)
        create a route entry for source address
else
{
        if (source seqno in RREQ > source seqno in route entry)
                update route entry for source address
        if ((source seqno in RREQ = source seqno in route entry) AND (hop count in RREQ < hop count in route entry))
                update route entry for source address
}
        if (a node is destination of RREQ)
{
        Calculate size of RTD_value field using

Size of RTD_value field = Bits to represent RTD *
        (Hop count − 1)
        Create a RREP packet with allocating space for RTD_value field
        Unicast RREP to source of request
}
else
{
        if ((have unexpired route to destination) AND (destination seqno in route entry >= destination seqno in RREQ))
{
        Calculate size of RTD_value field using
        Size of RTD_value field
                = Bits to represent RTD * (Hop count − 1)
        Create a RREP packet with allocating space for RTD_value field
        Unicast RREP to source of request
}
else
        broadcast RREQ to its neighboring nodes
}
}
```

AODV Forward RREP

```
{
        if (route to requested destination does not exist)
                create a route entry for requested destination
        else if (destination seqno in RREP > destination seqno in route entry)
                update-route entry for requested destination
        else if ((destination seqno in RREP = destination seqno in route entry) AND (hop count in RREP < hop count in entry))

        update route entry for requested destination

        if (route to requesting source exists)
{
        Calculates RTD and insert in RTD_Value field
        Forward RREP to requesting source
}
}
```

AODV Receive RREP by source node

```
{
        {
        Calculate RTD of successive nodes as
        RTT (N1, N2) = RTT (N1) − RTT (N2)
        for each RTD (N1, N2) pair
        {
                If RTD (N1, N2) > threshold
                Check anamoly
        }
                else
                Create a route entry for destination
        }
}
        else if (destination seqno in RREP > destination seqno in route entry){
        for each RTD (N1, N2) pair
        {
                If RTD (N1, N2) > threshold
                Check anamoly
        }

                Update route entry for destination
        }
        else if ((destination seqno in RREP = destination seqno in route entry) AND (hop count in RREP < hop count in entry))
        {
                for each RTD (N1, N2) pair
                {
                        If RTD (N1, N2) > threshold
                                Check anamoly
                }

        Update route entry for destination
        }
else
        discard RREP
}
```

AODV check anomaly


```
{
        If delay increases due to congestion
                Proceed as normal
        Else
        {

                Wormhole attack is detected and RREP is
                discarded
                Source node creates wormhole lists and
                broadcasts an alert message to isolate
                wormhole nodes from the network
        }
}
```

V. SIMULATION ENVIRONMENT

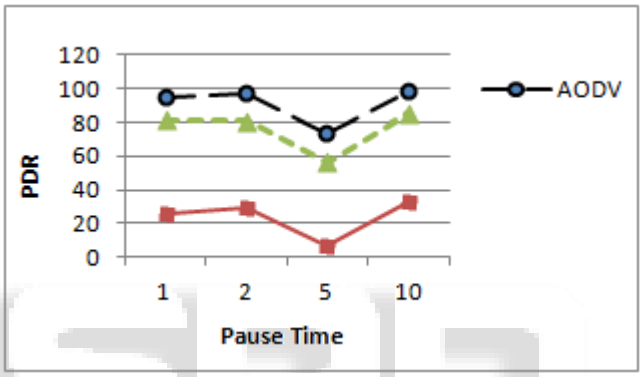
The simulations were carried out using network simulator (Ns-2), a discrete event driven simulator [11]. This section presents the topology and different parameters used in the simulation process. This simulation process considered a wireless network of nodes which are placed within a 1000m X 1000m area. CBR (constant bit rate) traffic is generated among the nodes. The simulation runs for 100 seconds. Keeping all other parameters constant, pause time and number of nodes are varied to observe the behavior of performance metrics.

Parameter	Value
Simulation area	1000m x 1000m
Antenna	Omni antenna
Number of nodes	25, 35, 45, 55
Speed	5 m/s
Pause Time (sec)	1, 2, 5, 10
Max queue length	50
Traffic	CBR (Constant bit rate)
Routing protocol	AODV
Transport Layer	UDP
Data Packets	512 bytes/packet
Data Rate	2Mbps
Mobility model	Random Way Point
Wormhole nodes	2

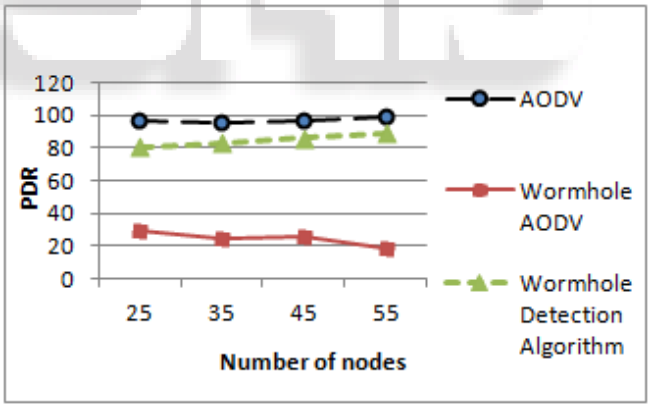
Table 8.1 Important Simulation Parameters

A. Effect of Wormhole Attack and proposed Algorithm on PDR

PDR is the ratio of packets received at destination node to that of packets sent by source node. It decreases with increase in wormhole nodes. It is clear from Fig. 2 and 3 that PDR of AODV is heavily affected by the wormhole nodes where as the PDR of Wormhole Detection Algorithm are immune to it. This graph confirms that while proposed AODV is secure against wormhole nodes, AODV is not. Pause time is the time for which mobile nodes wait at a destination before moving to other destination. Low pause time signifies high mobility as the node will have to wait for lesser time duration. Higher pause time leads to slower detection time and higher accuracy. This is because the longer the node stays at one place; it can collect enough neighbor count evidence in that location to declare the wormhole with more precision. However, if the wormhole-attacked area is the last one to be visited in the cyclic monitor, the detection time is higher since it is delayed more as it spends time in its previous locations. This is mainly due to the fact that our protocol detects the attacker and allows the source nodes to avoid it. By avoiding the attacker, our protocol finds shortest paths, and so, delivers more packets. PDF drops from 94.35 to 26.10 in presence of wormhole attack and with our scheme it improves to 81.21 in Figure 2.



Graph 1: PDR vs Pause time



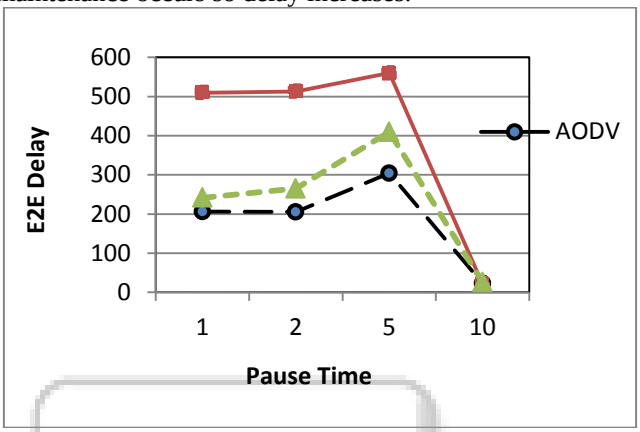
Graph 2: PDR vs Number of nodes

The PDR decreases in the case of AODV that is subject to an attack. This is due to the fact that the number of correctly received packet is very less than the number of transmitted packets. Our proposed algorithm is independent of number of nodes and shows consistent performance as nodes varies from 25 to 55 with an average PDF of 84.86 close to PDF of normal AODV.

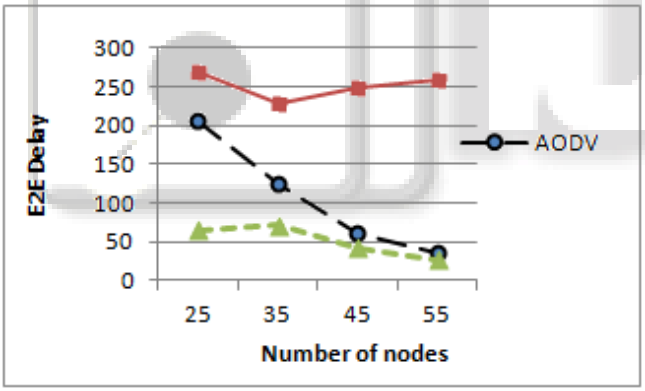
B. Effect of Wormhole Attack and proposed Algorithm on Average E2E Delay

Average End-to-End delay is average time taken for successfully transmitting data packets across MANET from source to destination which can be calculated by summing the time taken by all received packets at destination divided by their total numbers. It includes all kinds of delays like buffering during the route discovery latency, queuing at the

interface queue, retransmission delay at the Medium Access Control, the propagation and the transfer time. The average end-to-end delay The Average End-to-End Delay should be less for high performance. E2E delay increases in AODV under wormhole attack as expected. As pause time increases from 1 to 5 seconds, E2E delay increases from 206.35 ms to 304.12 ms in normal AODV, 509.97 ms to 560.07 ms in AODV under wormhole attack and 241.55 ms to 409.65 ms as shown in Fig 4. Our proposed algorithm does not incur much additional delay. There is abrupt decrease in average E2E delay as pause time reaches 10 second because more pause time means low mobility. Whenever node changes its direction or speed, route maintenance occurs so delay increases.



Graph 3 Average E2E Delay vs Pause time

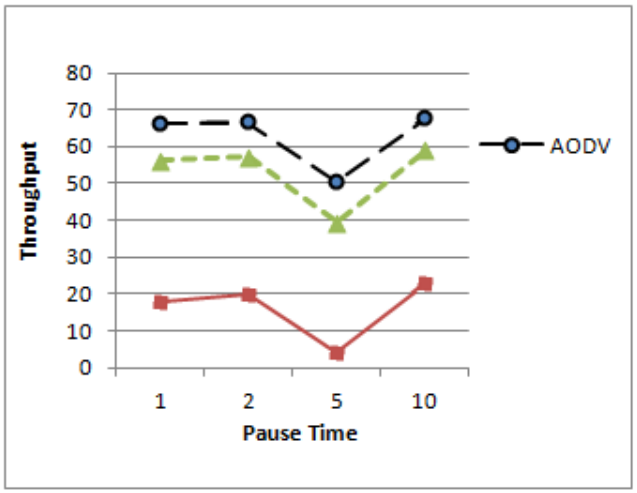


Graph 4 Average E2E Delay vs Number of nodes

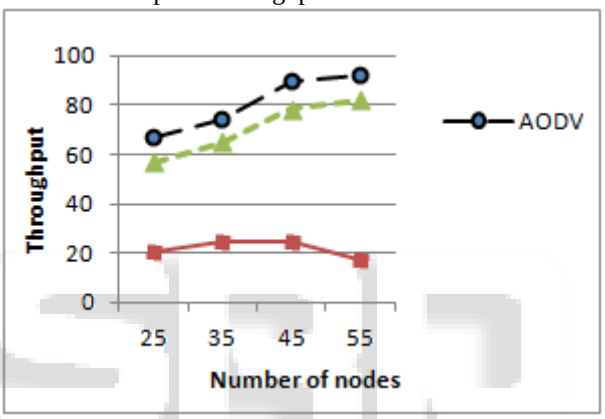
C. Effect of Wormhole Attack and proposed Algorithm on throughput.

Throughput is the ratio of the total data received from source to the time it takes till the receiver receives the last packet. Fig 6 and 7 shows the effect of pause time and number of nodes on the throughput. There is huge difference between the throughput for AODV and AODV under attack. High pause time means less mobility and more stable network but when pause time increases then the node will not move and throughput decreases. With AODV, without attack, its throughput is higher than in the case with under attack because of the packets discarded by the wormhole nodes.

The throughput of network drops from 66.10 kbps to 18.03 kbps with wormhole attack and rises to 56.10 kbps with proposed algorithm. The throughput of network increases from 66.95 kbps to 91.85 kbps for normal AODV as number of nodes increases from 25 to 55, for AODV under AODV attack throughput decreases from 20.16 kbps to 17.12 kbps. With our proposed algorithm, throughput increases from 57.11 kbps to 82.21 kbps.



Graph 5 Throughput vs Pause time



Graph 6 Throughput vs Number of nodes

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we have simulated the self-contained in-band wormhole attack and detection algorithm. We have also explained theoretically some metrics affecting wormhole attack that helps in developing the strategy for the detection of wormhole attack. Finally, we have presented the simulation results which shows

Having simulated the wormhole attack, we saw that the PDR falls drastically in the ad-hoc network. Our proposed solution tries to eliminate the wormhole effect with minimum increase in average end to end delay and the detection accuracy of our solution is quite high. Our proposed algorithm works well with node mobility, and does not require any strict clock synchronization and its network performance is independent of network density and pause time. The proposed method is equally effective for higher speed networks such as VANETs.

As part of the future work, we can integrate packet drop and round trip delay for detecting wormhole attack to improve the detection ratio. We would like to study Blackhole, Jellyfish and Sybil attacks in comparison with wormhole attack. These can be categorized on the basis of network performance degradation caused by the network.

REFERENCES

[1] Rajakumar, P.; Prasanna, V.T.; Pitchaikkannu, A "Security attacks and detection schemes in MANET," Electronics and Communication Systems (ICECS), 2014 International Conference on , vol., no., pp.1-6, 13-14 Feb. 2014.
[2] Jen, S.-M., et al. (2009). "A Hop-Count Analysis Scheme for Avoiding Wormhole Attacks in MANET." Sensors 9(6): 5022-5039.
[3] Jain, S. and S. Jain (2010). "Detection and prevention of wormhole attack in mobile adhoc networks." networks 1793: 8201.
[4] Choi, S., et al. (2008). WAP: Wormhole attack prevention algorithm in mobile ad hoc networks. Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08. IEEE International Conference on, IEEE.
[5] Ban, X., Sarkar, R., Gao, J.: Local connectivity tests to identify wormholes in wireless networks. In: Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing. pp. 65–78 (2011)
[6] Lu, X., Dong, D., Liao, X.: MDS-detection using local topology in wireless sensor networks. International Journal of Distributed Sensor Networks 2012, 1–9 (2012).
[7] Chaurasia, U.K.; Singh, V., "MAODV: Modified wormhole detection AODV protocol," Contemporary Computing (IC3), 2013 Sixth International Conference on , vol., no., pp.239,243, 8-10 Aug. 2013.
[8] Banerjee, S., & Majumder, K. (2014). Wormhole Attack Mitigation In Manet: A Cluster Based Avoidance Technique. International Journal of Computer Networks & Communications, 6(1), pp. 45-60.
[9] S. Song, H. Wu, and B. Choi, "Statistical ormhole detection for mobile sensor networks," in ICUFN, Conference on Ubiquitous and Future Networks, pp. 322-327, July 2012.
[10] Y. Hu, A. Perrig, and D. Johnson, Wormhole attacks in wireless networks. IEEE Journal on Selected Areas in Communications, 2006. 24(2): p. 370-380.
[11] T. Issariyakul and E. Hossain, "Introduction to Network Simulator NS2," Springer, US, 2009.

