NTTM: Novel Transmission Time Based Mechanism to Detect Wormhole Attack

Kumar Chanchal and D.K. Lobiyal

School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi.
{chanchal.ck1,lobiyal}@gmail.com

Abstract. The cooperative nature and absence of infrastructure gives rise to lot of scope for research in the area of Mobile Ad-hoc Networks (MANETs). The dynamic topology, absence of central control and broadcast nature of communication open security threats for MANETs. Many security attacks have been identified by the researchers, but wormhole attack is one of the most devastating attacks. Novel Transmission Time based Mechanism (NTTM) detects wormhole attacks by keeping every node under the surveillance of its neighbors. Based on the Round Trip Time (RTT) computed by each node on a route, the source node computes RTT between each neighbor. If the RTT between a pair of nodes is more than the threshold value, it is assumed that there is wormhole attack between these nodes. The performance of NTTM is evaluated using dynamic source routing (DSR) protocol under wormhole attack.

Keywords: MANETs, Wormhole attack, RTT, NTTM, DSR.

1 Introduction

A number of threats and their countermeasures in the area of MANETs have been identified by the researchers. In all possible threats, wormhole attack is the most devastating attack and it is lunched at the time of route discovery phase. Two malicious nodes located at different positions form a secret tunnel (wormhole). One malicious node captures the control as well as data packets from the location near the source. It directs these packets to move through the tunnel towards other colluding nodes placed at other locations in the network. These colluding nodes in turn drop or replay back the packets into the network.Although the length of the tunnel is large but it creates an illusion that there exists a shortest path between the source and the destination. Therefore, source node chooses the shortest path through the tunnel to send its data. By using this link, malicious nodes launch variety of attacks against the data flow such as selective dropping, reply attacks, eavesdropping etc.[1,2].

Wormhole or tunnel can be formed either by packet encapsulated channels (also known as In-band-channel) or out-of-band channels. In packet encapsulated channels, malicious node captures the route message and inserts it in data packet payload. This packet is transmitted using legitimate nodes towards other malicious node. The

K. Singh, A.K. Awasthi, and R. Mishra (Eds.): QSHINE 2013, LNICST 115, pp. 485–495, 2013.

malicious node draws the routing message from packet payload and further braodcast it to the destination. In Out-of-Band channel, a special channel either a direct wired link or a long range wireless link can be used to form tunnel between malicious nodes.

Wormhole attacks can be classified broadly into hidden wormhole attacksand exposed wormhole attacks. In hidden wormhole attacks, legitimate nodes are unware of malicious nodes. The malicious nodes don't upadate hop count field in packet header i.e. only legitimate nodes change the hop length during route establishment. In exposed wormhole attacks, they are aware of the fact that malicious nodes are forwarding packets. But they actually do not know that they are malicious nodes. Here, attackers neither modify packet header nor the content of the packet. The nodes simply add its own MAC address in the header of the packet and forwards it. By extracting information from packet header, the malicious node obtains necessary information about the sender of a packet [2-4].

In this paper we have proposed an effiecent and secure mechanism to detect wormhole attack known as NTTM over DSR protocol. NTTM uses route discovery mechanism of DSR protocol with some modifications. The RTT between the destination and each node in the path is computed by the neighbor nodes of the destination. Finally, RTT value is forwarded to the source that will declare about the presence of wormhole aftersome computation.

The remainder section of this paper is arranged as follows. Section 2 describes the work that has been done related to the detection of wormhole attack. The problem is defined in section 3. Section 4 provides the proposed work in detail. The performance of NTTM evaluated by using simulations is presented in section 5. Section 6 concludes the work carried out along with discussions on possible future extensions.

2 Related Works

Many methods have been proposed to defend against wormhole attacks in which either existing protocol is modified like AODV or special hardware is used such as directional antennas [5,6].

Su et al. [1] introduced Wormhole Avoidance Routing Protocol (WARP) based on AODV protocol. WARP keeps multiple link disjoint paths into consideration. The malicious nodes have great tendency to get involved in the path discovery process. WRAP uses this characteristic of malicious nodes to detect the wormholes attack. Each node records anomaly value of its neighboring nodes. The probability of involvement of a node among the multi joint path is known as its anomaly value. If the anomaly value of a particular node exceeds the threshold value, its neighbor node declares it as a malicious node and it further, discards all the requests coming from that node to form a route. It may be possible that legitimate nodes may be considered as malicious and isolated by their neighbors.

Phuong et al. [3] proposed transmission time based mechanism using Round Trip Time (RTT) of packets between each neighbor to detect wormhole attacks. It is tested over AODV protocol. The destination node modified the format of RREP packet in AODV by adding an extensional part. The size of extensional part is according to hop count field in RREQ packet. A source node calculates RTT between it and eachof theneighbors after obtaining the information regarding the RTT time of each node. If RTT value between any pair of neighbor's nodes exceeds the threshold time limit, it shows the existence of the wormhole on the route. There are the high chances of inserting false information by the malicious nodes.

Hu et al. [7] introduced a general mechanism to defend against wormhole attack in which small amount of information called Leash is added into a packet. Leash restricts a packet's maximum allowed transmission distance. Leash can be geographical or temporal [4, 7]. To form geographical leash, every node should know their positions. After receiving a packet, receiver calculates the maximum distance between sender and itself. It also records its receiving time. If the distance exceeds the maximum limit, the node discards the packets. The temporal leash uses a special off-the-shelf hardware based on LORAN-C, and WWVB in place of loose clock synchronization to provide tight time synchronization. It is implemented through Timed Efficient Stream Loss tolerant Authentication (TESLA) with Instant Key (TIK) disclosure protocol. It requires extremely tight time synchronization and Global Positioning System (GPS).

Wing et al. [8] brought in an End-to-End detection of wormhole attack which calculates "minimum hop count" from every node to the destination. The EDWA can work with both AODV and DSR routing protocols with a constraint that only destination node can reply to the RREQ packet. Based on the position of source and the destination, source node calculates the shortest path in terms of hop length. If the estimate hop count value is greater than hop count value of RREP packet, it is assumed that there is wormhole somewhere on the path. To identify the end point of wormhole, source node sends Trash packet. If the large increment (more than one) is observed in hop count between any pair of neighbors, it shows that this pair of nodes comprises the end points of a wormhole. The EDWA mechanism performs better when the source and destination are not far away.

3 Proposed Work

This section describes the detail of proposed work. The RRT experienced by the packet while travelling through tunnel is large. This characteristic is considered as key point for this proposed model.

3.1 Route Discovery

In Fig. 1 shown below, node *S* wants to send data to node *D*. The sending node *S* triggers route discovery and broadcasts a RREQ packet into network. The source node stores T_{RREQ} (time of broadcasting RREQ packet) of the RREQ packet. Each RREQ packet contains address of sender, receiver as well as broadcast ID which are used to discard the duplicate packets received at a node.

Node 1, 2 and 3 receive RREQ packet broadcast by node S. Each Node matches destination address recorded in RREQ with its IP address (step 6). The steps mentioned in the brackets are of NTTM algorithm described next. None of them (1, 2 and 3) is destination, so they processes the RREQ packet and put their IP address in

RRL. They rebroadcast the RREQ packet in to network. The node *S* hears the RREQ packet again as shown in Fig.1 and find its address at (n-1)th position in RRL (step 9). Therefore, it stores T_{RREQ} of each node and drops the RREQ packets broadcasted by nodes (step10).

At node 4 two RREQ packets are received through node 1 and node 2. Suppose RREQ packet broadcast by node 2 reaches first at node 4, which will discard RREQ packet (step 5) broadcast by node 1 because of duplication (same broadcast ID, same originator ID). Similarly, node 5 receives RREQ packet from node 3 and node 2. RREQ is received first through 2. They are neither destination nor the RREQ packet has their address in RRL. Therefore, they append their address in RRL tail and broadcast RREQ packet again. Now node 2 satisfies the condition of step 9 for both node 4 and node 5. Hence, node 2 stores T_{RREQ} of the both nodes.

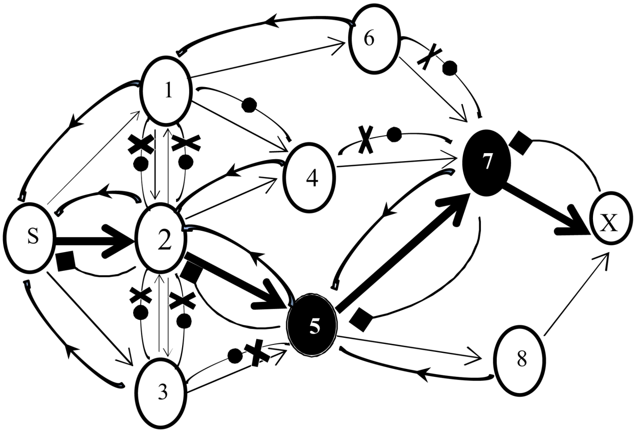

Fig. 1. Route Discovery in NTTM

W	= Wormhole Node
X	= Legitimate Node
x ⌒ y	= Node X save broadcast time of RREQ packet by node Y
x ⌒● y	= Node X reject RREQ packet broadcasted by node Y
x ■⌒ y	= Route Reply sent by node Y to Node X
→	= Broadcasting direction of RREQ packets
➔	= Actual path obtained
✖	= Shows the rejection by any particular node

A node keeps the record of T_{RREQ} of its neighboring node till it gets the information regarding T_{RREP}(Time of receiving RREP at any node) of that particular node. In this way RREQ packet reaches at destination node D through the intermediate nodes (2, 5 and 7). Thus the route established is S---2---5---7---D.

Now destination node responds by generating RREP packet. Node D copies the RRL in RREP packet and attaches an extensional space to record the RRT between each neighbor on the route. The RREP packet is unicast back and received by node 7. This node satisfies the condition mentioned in step 21 and simply put the time of receiving of RREP T_{RREP} at the corresponding space in RREP packet without any computation. When the RREP packet proceeds further according to RRL, it will be received by node 5. Node 5 could not meet step 21, so it extracts$(T_{RREP})_7$ *i.e.* time of receiving RREP at any node7 from extensional part in RREP. The RTT of its neighbor 7 computed by node 5 (through step 22 to 24 of algorithm) and put it back at the place from where $(T_{RREP})_7$ it was extracted. Similarly each intermediate node repeats steps from 20 to 24 till RREP packet reached back at source.

3.2 NTTM Algorithm

1. If a node wants to send a data, it initiates route discovery process.
2. The Source node generates RREQ and put its own IP address into Record Route List (RRL) option in RREQ packet as an originator.
3. Source node broadcasts RREQ packet and store T_{RREQ}.
4. Each node in the transmission range of sender node receives RREQ packet.
5. If a node receives a packet with same source ID, broadcast ID and hop length greater or equal than already received packet, then drop the packet.
6. Otherwise node receiving RREQ matches destination (target) IP address.
7. If target IP address matches with receiver IP address go to step 15.
8. Otherwise each neighbor node starts checking RRL.
9. If IP address at $(N-1)^{th}$ position in RRL matched with receiver IP address. // where N is the number of addresses stored in RRL in RREQ packet at any time T.
10. Then stores T_{RREQ} and drop the packet. //neighbor node hears RREQ broadcasting of other nodes.
11. Otherwise node continuously further searches RRL.
12. If node receiving RREQ packet, IP address matched at other position in RRL, then drop RREQ packet.
13. Otherwise receiver node appends its address at tail of RRL in RREQ packet and broadcast it further.
14. Repeat the step 4 to 13 for each intermediate node till destination.
15. When the RREQ packet arrived at destination.
16. Destination generate RREP packet to respond RREQ packet.
17. The RRL is reversed and copied into RRL of RREP packet.
18. An extensional part is added into basic DSR RREP packet by destination. // To store RTT calculated for each node on the route by its neighbor.
19. Destination unicasts RREP packet along reversed RRL.

20. Next node in RRL receives RREP packet.
21. If receiver node IP address is at 2nd position in reversed RRL (RREP RRL) or second last on RRL of RREQ, then store $(T_{RREP})_x$ at appropriate position in extensional part of RREP and forward back the packet further. // Receiver node is just before the destination node.
22. Otherwise if nodereceiving RREP packet is addressed at kth position on reveres RRL, then receiver node extracts T_{RREP} from extensional part.
23. Then receiver node calculate RTT of its neighbor addressed at (k-1)th position on reverse RRL or at (k+1)th position on RRL of RREQ using T_{RREQ} value stored in step 10 and T_{RREP} extracted by receiver from extensional part using equation 1given below.

$$(RTT)_{x,\,d} = \text{abs}\,((\text{RREQ Time})_x - (\text{RREP Time})_x) \tag{1}$$

24. The node calculating RTT of corresponding neighbor, stores backRTT valueat position from where T_{RREP} is extracted.
25. Repeat steps 20 - 24 till RREP packet reached at source node.
26. When RREQ packet arrives at source, the source node repeats step 23, 24 for one time and calculate RTT value of its neighbor's node on the route.
27. Now source extracts RTT value of each node from extensional part and calculate RTT between each neighbor node on the route using equation 2

$$(RTTneighbors)_{x\,y\,=}\ \text{abs}\,((RTT)_{x,\,d} - (RRT)_{y,\,d}) \tag{2}$$

28. The source node compares the value of RTTneighbors calculated in step 27 with threshold value.
29. If RTTneighbors is more than threshold value then source node declares that the presence of wormhole on the route.

3.3 Computation of RTT in NTTM

Each node overhears the RREQ broadcastby its neighbors as shown below in Fig. 2 and keeps record of T_{RREQ} in its cache. While receiving RREP packets every node inserts receiving time T_{RREP} into extensional part of RREP packet for further computation.

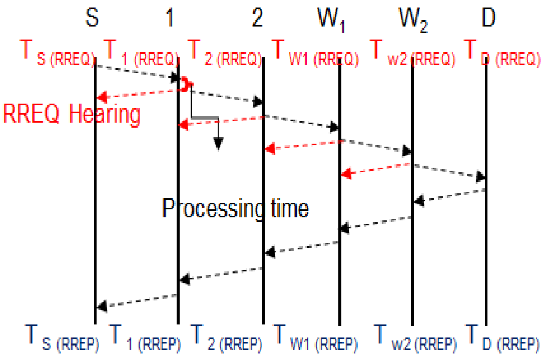

Fig. 2. Timing Diagram of Hearing of RREQ in NTTM

The value of RTT between each node and destination is computed according to the equation 1 mentioned above. The value of RTT between neighboring nodes on the path is computed according to equation 2 given above.

Each node extracts T_{RREP} of its neighbor from extensional part and after calculating its neighbor RTT inserts back RTT in place of T_{RREP}. Here, we assumed that the time at which RREQ sent by a particular node is same as the time of hearing the broadcast of RREQ by its neighbor. The RREQ broadcasting time and RREP receiving time are listed in Table 1.

Table 1. RREQ and RREP Time Record in NTTM

Nodes	Node Hearing RREQ broadcast	RREQ Hearing Time($TH_{(RREQ)}$)	RREP receiving Time ($T_{(RREP)}$)
S	S	-	30
2	S	2.5	27
5	2	5.5	23
7	5	13	15.5
D		-	-

Now the RTT values between each node and the destination is computed using equation 1 and the values are lsited in Table 2.

Table 2. Computation of RRT of Each Node in NTTM

Nodes	Node Computing RTT	RREQ sending Time ($T_{(RREQ)}$)	RREP Receiving Time($T_{(RREP)}$)	RRT of Node
S	S	0	29	29
2	S	2.5	26	23.5
5	2	5.5	23	17.5
7	5	13	16	3

The source node extracts the value of the RTT between each node and the destination node from the extensional part of RREP. It computes the RTT between each neighbors and compares these values with thresold value.The values of RTT between each neighbors are listed in Table 3.

Table 3. RTTneighbors Computation in TTM at Source Node

$(RTT)_{x\ d}$	$(RTT)_{y,\ d}$	$(RTTneighbors)_{x\ y}$
33	28	5 (RTT_{S1})
28	24	4 (RTT_{12})
24	18	6 (RTT_{2W1})
18	4	14 (RTT_{W1W2})

The value of RTT between node 5 and node 7 is comparatively very high as shown in table 3. Therefore, it shows the presence of wormhole attack between node 5 and node 7.

4 Simulations and Experimental Evaluation

To evaluate the performance of NTTM, the simulation is carried out using QualNet 5.0.2 Network simulator. The “All Pass” model is used to launch the wormhole attack. It is assumed that the time of receiving RREQ packet at a node (who calculates RTT) is same as the time of broadcasting RREQ packet by its neighboring node for which RTT is calculated. All nodes are working in promiscuous mode. The diameter of the network is small. The simulation parameters usedare listed in Table 4.

Table 4. Network Simulator Parameters

Parameters	Value
Simulation Time	1000sec
Simulation Repetition	100
Routing protocol	DSR
MAC Layer	802.11
Packet Size	512 bytes
MAC Protocol	802.11
Data Rate	2Mbps
MAC propagation delay	1 µs
Terrain Size	1500 x 1500
Network layer protocol	IPv4
Mobility Model	Random waypoint
Data Traffic Type	CBR
Maximum buffer size forpackets	50 packets
Antenna Model	Omnidirectional
Antenna Height	1.5metres
Noise Factor (SNR)	10.0
Transmission Power	15dBm
Transmission range	367metres

The simulation results were recorded in text file and graphs were generated using Microsoft Office Excel 2007. The trend is observed through the line graph between node’s speed verses packet delivery Ratio (PDR). DSR under the wormhole attack and NTTM under wormhole attack were compared in terms of PDR with different node mobility.

Under the wormhole attack, PDR value decreased consistently as compared to DSR protocol. The NTTM model performed better as shown below in Fig. 3. It showed maximum growth in PDR of 11% at the mobility rate of 35m/s while the worst performance was observed at mobility rate of 25m/s where the growth was only approximately 5%.

At high mobility, the topology changes very rapidly. Therefore, the frequency of route breakage is very high. It is very difficult to build new routes in such conditions. As the speed of nodes increase, PDR falls downs. Initially it fell down very rapidly as shown in Fig. 3. But on further increment in the mobility of nodes, frequency of route breakages get saturated. Therefore, the metric values fell relatively low. From the results, it is evident that the NTTM model performed better and showed a significant growth in PDR as compared to DSR protocol under wormhole attack.

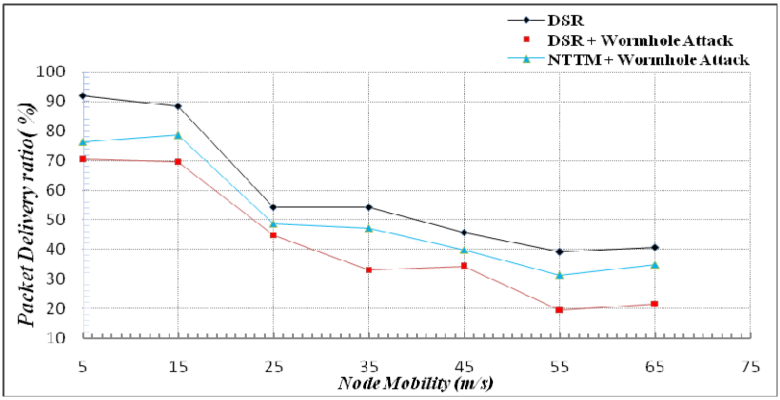

Fig. 3. Packet Deliver Ratio verses Node Mobility

Threshold value played an important role in NTTM model. The threshold value is picked up with respect to the RTT between real neighbors which was observed 14ms and it is then incremented further. If the nodes were at critical position in the network, the nodes experienced large delay could be considered as malicious nodes. Thus, low threshold value result in high false positive. At high threshold value the wormhole attack launched with small tunnel length were undetected and slowly damage the network, therefore false negatives increases.

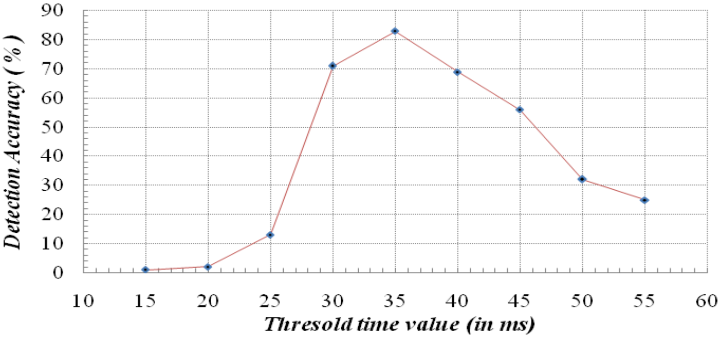

Fig. 4. Detection Accuracy of NTTM

As in the Fig. 4, the detection accuracy graph showed an extreme increment after 25s and achieved best detection accuracy at 35s. At 35ms, both false positive and false negative were low.Therefore,35ms were chosen as threshold limit value.

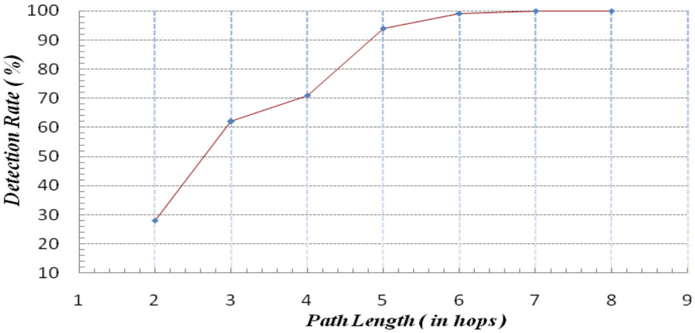

Fig. 5. Detection Rate in NTTM

As shown above in Fig. 5, the detection rate increased exponentially with respect to tunnel length up to 5 hops and saturated after the tunnel length exceeds 6 hops. As the tunnel length increases, the RTT value between the malicious nodes also increases and finally theRTT value between malicious nodes exceeds threshold. Therefore, wormhole attacks can be detected easily and accurately.

5 Conclusion and Future Scope

The study regarding the wormhole attack leads us to draw the conclusion that Wormhole attack is most dangerous attack in MANETs. By identifying the wormhole attack during route discovery, NTTM avoids the chances of damages in the network due to attacks. It was observed that the PDR value on average falls by 17% when DSR protocol under wormhole attack is compared with DSR protocol without wormhole attack at different node mobility. However, the results are improved by 9% under NTTM model. The accuracy and detection rate of NTTM model improved with the incrementin tunnel length.

In future, the difference between the sending time of RREQ packet and receiving time of RREQ packet at its neighbor can be taken into consideration which was assumed negligible in NTTM. NTTM can be implemented over other routing protocols like TORA, DSDV etc.

References

1. Su, M.Y.: WARP: A wormhole-avoidance routing protocol by anomaly detection in mobile Ad-hoc networks. Int. J. of Computer and Security 29, 208–224 (2010)
2. Taheri, M., Naderi, M., Barekatain, M.B.: New Approach for detection and defending the Wormhole Attacks in Wireless Ad Hoc Networks. In: Proc. 18th ICEE, Iran, pp. 331–335 (2010)
3. Van Tran, P., Canh, N.T., Lee, Y.-K., Lee, S.: Transmission Time-based Mechanism to Detect Wormhole Attacks. In: Proc. of IEEE 2nd Asia-Pacific Services Computing Conference, Seoul, pp. 172–178 (2007)

4. Chiu, H.S., Lui, K.S.: DELPHI: wormhole detection mechanism for ad hoc wireless networks. In: Proc. of IEEE 1st Symposium on Wireless Pervasive Computing, China, pp. 6–11 (2006)
5. Perkins, C.E., Royer, E.M., Das, S.R.: Ad hoc on-demand distance vector (AODV) routing. IETF Internet draft, MANET Working Group (2004)
6. Gupta, S., Kar, S., Dharmaraja, S.: WHOP: Wormhole attack Detection Protocol using Hound Packet. In: Proc. IEEE on Innovations in Information Technology, United Arab Emirates, pp. 226–231 (2011)
7. Hu, Y.C., Perrig, A., Johnson, D.B.: PACKET LEASHES: A defense against wormhole attacks in wireless ad hoc networks. In: Proc. IEEE INFOCOM 2003, USA, vol. 3, pp. 1976–1986 (2003)
8. Wang, X., Wong, J.: An end-to-end detection of wormhole attack in wireless Ad-hoc networks. In: Proc. 31st Computer Software and Applications, vol. 1, pp. 39–48. IEEE Computer Society, Washington, DC (2007)