Mitigating Large Propagation Delay by Mitigating Wormhole Attack in Mobile Ad Hoc Network

Ranjeeta Siwach, Vanditaa Kaul

Computer Science & Engineering, B.S.Anangpuria Institute of Technology & Management

Abstract- Wormhole attacks, in which colluding attackers with out-of-band communication links record packets (or bits) at one location and replay at another, cause far away nodes to consider themselves as neighbours to one another.

Under this attack, two faraway malicious nodes can collude together using either wired link or directional antenna, to give an impression that they are only one hop away. We presents NEVO, in which nodes passively monitor (overhear) the forwarding of broadcast type packets by their neighbours and use the send and overhear times of trans- missions of these packets, to mitigate these wormhole attacks. NEVO does not require synchronized clocks, special hardware support, or any special capability. NEVO can detect almost all instances of wormhole attacks and is virtually independent of the routing protocol used. NEVO has a disadvantage that it does not consider the impact of clock drift. On the other hand, NEVO uses network layer verification, which takes more time to complete, and thus require clock drift correction. In this paper, we use an approach which consider the impact of clock drift and do not require any correction in network layer.

Index Terms- MANET**,** wormhole, NEVO

I. Introduction

mobile Ad Hoc Networks is the most popular networks widely used in various applications. It consists of mobile nodes where each node communicates with each other. The control of nodes is not administrated by any access point. Due to this, the network is easily impersonated by several attacks like active and passive attacks. These attacks degrade the performance of networks i.e. means network connectivity, network availability and communication coverage.

In a wormhole attack, two or more colluding attackers create a private communication channel (wormhole) between them and replay packets heard at one end of the link at the other end. This can cause two far away nodes to consider themselves as neighbours to each other or have a distorted view of the number of hops between them. This is a particularly hard attack using which even a handful of malicious nodes can conduct traffic analysis of packets or disrupt connections by dropping packets when needed. Wormhole attack can be launched in hidden or in participation mode. The wormhole attacks are powerful since the attackers neither needs to neither compromise any of the network nodes nor have any knowledge of the security mechanisms or routing protocols in use.

The security mechanisms used for wired network such as authentication and encryption are futile under hidden mode wormhole attack, as the nodes only forward the packets and do not modify their headers. Wormhole attacks are considered to be difficult to prevent since the malicious nodes are invisible and may use special high-speed links to cover large distances [1]. Wormhole attackers can also simply record the traffic for later analysis.

It is easy to see that wormhole attack that span long distances (multiple hops) can be prevented if each node that its neighbour is truly within its radio range.

According to this observation, there is a network layer based countermeasure in which nodes passively monitor or overhear [5] the forward- ing of certain types of broadcast packets by their neighbours and use the timing information of these broadcast packets to ensure that routes are established through true neighbours only. We call it NEVO (Neighbour Verification by Overhearing). NEVO re- quires broadcasts among neighbors, which are commonly used in ad hoc wireless networks, and local timestamps of broadcast packets sent or received by the medium access control (MAC) layer, which do not require any changes to the MAC protocol but may require a firmware upgrade to enable MAC layer to automatically send this information to the network layer. NEVO does not rely on special hardware support such as directional antennas or ultrasound transmitters/receivers, special capabilities such as clock synchronization or GPS coordinates, geometric inconsistencies, or statistical methods. Therefore, NEVO is a practical solution to mitigate wormhole attacks. NEVO works with all ad hoc network routing protocols.

Implementation of NEVO together with a previously proposed on demand secure routing protocol SOR [9], which prevents route Falsification and tunnelling attacks by compromised insider nodes, in the Glomosim simulator.

II. WORMHOLE ATTACK IN MANET

In this attack, an attacker receives packets at one location in the network and tunnels packets to another location in the network, where the packets are resent into the network. This tunnel between two colluding attackers is referred to as a wormhole. It could be established through a single long-range wireless link or even through a wired link between the two colluding attackers.

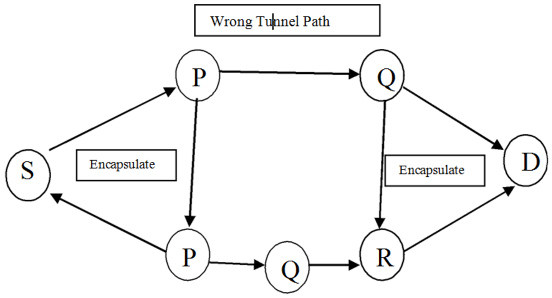


FIGURE1: WORMHOLE ATTACK

Due to the broadcast nature of the radio channel, the attacker can create a wormhole even for packets not addressed to itself.

Example In figure1. **P** and **Q** are two malicious nodes that encapsulate data packets and falsified the route lengths. Suppose node **S** wishes to form a route to **D** and initiates route discovery. When **P** receives a Route Request from **S,** ***Q*** encapsulates the Route Request and tunnels it to **Q** through an existing data route, in this case {**P** --> **P** --> **Q** --> **R** --> Q}. When Q receives the encapsulated Route Request for **D** then it will show that it had only travelled {**S** --> **P** --> **Q** --> **D**}. Neither P nor **Q** updates the packet header. After route discovery, the destination finds two routes from **S** of unequal length i.e. one is of 4 and another is of 3. If **Q** tunnels the Route Reply back to **P**, **S** would falsely consider the path to **D** via P is better than the path to **D** via R. Thus, tunnelling can prevent honest intermediate nodes from correctly incrementing the metric used to measure path lengths.

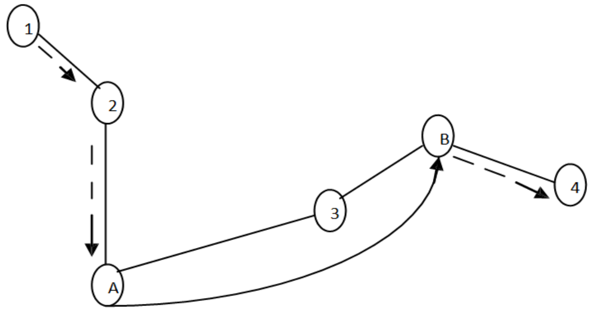


Figure.1.1. A wormhole attack performed by colluding malicious nodes A and B

Though no harm is done if the wormhole is used properly for efficient relaying of packets, it puts the attacker in a powerful position compared to other nodes in the network, which the attacker could use in a manner that could compromise the security of the network. The wormhole attack is particularly dangerous against many ad hoc network routing protocols in which the nodes that hear a packet transmission directly from some node consider themselves to be in range of that node. Performance of wormhole attack can be shown in figure 2.In this attack, an attacker receives packets at one point in the network, "tunnels" them to another point in the network, and then replays them into the network from that point.

Due to the nature of wireless transmission, the attacker can create a wormhole even for packets not addressed to it, since it can overhear them in wireless transmission and tunnel them to the colluding attacker at the opposite end of the wormhole. Two malicious nodes share a private communication link between them. Worm hole can eavesdrop the traffic, maliciously drop the packets, and perform man-in- the-middle attacks against the network protocols.

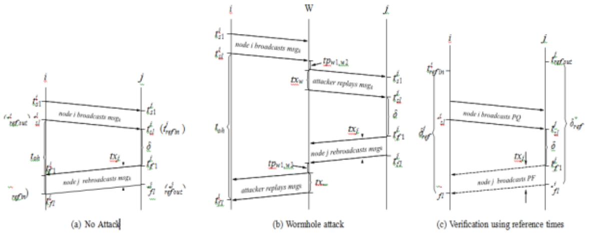
III. NEVO

We illustrate the approach of NEVO using Fig. 1.3 in which node i broadcasts a packet, and one of its neighbors, node j, rebroadcasts (forwards) it. We assume half-duplex wireless channels. Let t denote the time it takes a packet to traverse one hop and δ denote the time taken by j to process the packet and acquire the channel before transmitting it. Then, node i overhears j's forwarding in $t + \delta$ seconds after it completed its broadcast if node j is a true neighbor. On the other hand, it takes at least $3t + \delta$ seconds to overhear j's forwarding via a wormhole.

a. Timing Analysis of Wormhole Attacks

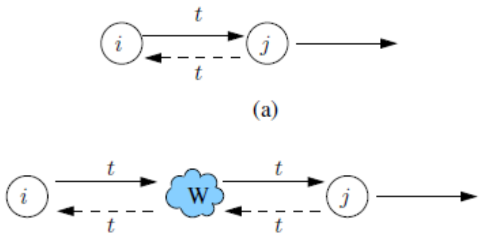
For a more rigorous timing analysis, we use Figs. 2 (a) and (b) and the following notation.

- $ts1$ ($ts1$): the local time of node i (node j) at the time the first bit of the message is broadcasted by node i (heard by node j).



(a) No Attack (b) Wormhole attack (c) Verification using reference times

Fig. 2. Timing analysis of packet transmissions.



(a)

(b)

Fig.1.3. Detection of out-of-band wormholes using passive monitoring. Node i sends a packet to node j and then passively monitors node j's forwarding.

(a) normal case; (b) attack case with a wormhole, W, between node i and
node j. W is formed by one or multiple colluding attackers.

- t^i_{sl} (t^j_{sl}): the local time of node i (node j) at the time the last bit of the message is broadcasted by node i (heard by node j).

• t^i_{f1} (t^j_{f1}): the local time of node *i* (node *j*) at the time the first bit of the message is overheard by node *i* (forwarded by node *j*).

• t^i_{fl} (t^j_{fl}): the local time of node *i* (node *j*) at the time the last bit of the message is overheard by node *i* (forwarded by node *j*).

• tx_j: the transmission time for the forwarded message by node *j*. It includes preamble and MAC headers. Note that
$t^j_x = t^j_{fl} - t^j_{f1} = t^i_{fl} - t^i_{f1}$.

• δ: the message delay at node *j*, $\delta = t^j_{fl} - t^j_{sl}$

.• tx_w: the additional transmission delay incurred to replay the message by a wormhole attacker. This can be as low as one bit time to as much as *txj* .

• $tp_{i,j}$: the message propagation delay between nodes *i* and node *j*.

• t_{oh}: the overhear time, i.e., the time delay for node *i* to overhear node *j*'s forwarding after it broadcasted the message. I.e., $t_{oh} = t^i_{fl} - t^i_{sl}$.

• *R*: maximum radio transmission range in meters.

• *Sp*: radio signal propagation speed; $Sp < c$, where *c* is the speed of light in free space.

If nodes *i* and *j* are true neighbors, see Fig. 2 (a), the propagation delay between them can be estimated as follows.

$t_{oh} = t^i_{fl} - t^i_{sl} = tx_j + \delta + 2tp_{i,j}$ (1)

$t_{oh} - tx_j - \delta = 2tp_{i,j}$ (2)

Note that node *i* knows $tx_j = t^i_{fl} - t^i_{f1}$. In practice, δ, the processing delay at node *j*, varies a lot. However, if node *i* knows δ (suppose, node *j* gave this information in a separate message), then node *i* can verify if the following condition
holds.

$t_{oh} - tx_j - \delta = 2tp_{i,j} \leq 2R/Sp.$ (3)

In the normal case without attack, (3) is satisfied.

In the case of a wormhole link between nodes *i* and *j*, see Fig. 2 (b), the time to overhear, denoted as *t'oh*, is given by
$t'_{oh} = t^i_{fl} - t^i_{sl} = tx_j + \delta + 2(tx_w + tp_{i,w1} + tp_{w1,w2} + tp_{w2,j})$
(4)

where *w*1 and *w*2 are the two endpoints of the wormhole.

IV. Message Transmission Sequence in NEVO

The neighbor verification consists of three message transmissions between two encountered nodes, as shown in Fig. 3 — (1) node *i* broadcasts a control packet, called probe query (PQ) targeted to node *j*; (2) after node *j* receives the PQ from node *i*, it rebroadcasts (forwards) this query packet as its probe forward (PF) packet; then (3) node *j* sends node *i* a unicast packet, called probe reply (PR), which contains the processing delay, δ. After receiving PR from node *j*, node *i* can decide whether to accept node *j* as a true neighbor according to (3). To prevent wormhole attackers from fabricating probe packets, nonces are added to PQ and PF packets and a message authentication code to the PR packet. The message formats for PQ, PF, and PR are given as follows:

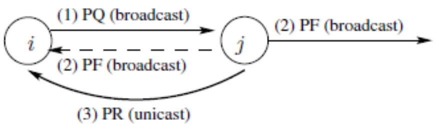


Figure3:Message Transmission Sequence used By NEVO

PQi = {PQ, *i, j, ni*} (5)
PFj = {PF, *i, j, ni, nj*} (6)
PRj = {PR, *i, j, ni, nj, δ,Mij*} (7)

where *ni* and *nj* are nonces generated by nodes *i* and *j* respectively,

Mij is the message authentication code computed over{PR,*i, j, ni, nj* , *δ*} using a shared cryptographic key betweennode *i* and *j*. Note that digital signatures may be used instead of message authentication codes.

Optimization: The broadcast message PF relayed might not be overheard by node *i* due to collision. Even if node *i* receives the PR from node *j* but did not overhear *j*'s forwarding, it cannot decide whether there is a wormhole link between them. Then the neighbor verification fails, and node *i* needs to retry the neighbor verification sequence. To reduce the number of retries, we introduce the concept of reference times. The reference times are local timestamps corresponding to a previous successful event such as the last time nodes *i* and *j* verified that they were true neighbors. Consider Fig. 2 (a) in which node *i* initiates the neighbor verification. When node *I* verifies that node *j* is its neighbor, node *i* records *tisl* and *tifl* as the reference times *tirefout* and *tirefin*, respectively. Also node *i* sends a special unicast packet to notify node *j* of theirtrue connectivity so that node *j* can record its local timestamps *tjsl* and *tjfl* during that verification event as its reference times *tjrefin* and *tjrefout*, respectively. These reference times are usedfor future neighbor verifications as follows.

We add one more fields to PR, *δjref* = *tjfl*−*tjrefout*, which is the delay between the time to send the last bit of the PF and the reference time, as shown in Fig. 2 (c). If node *i* receives *a* PR from node *j* and did not overhear the related PF, it can estimate *tifl* = *tirefin* +*δjref* and then verify node *j* accordingto (3). This works only if nodes *i* and *j* verified each other by overhearing both ways and established the reference times.Node *j* sets *δjref* = 0, which indicates that reference time is invalid for neighbor verification, if reference times with node *I* are not established. The reference times *tirefout* and *tjrefin* are used for the case where node *j* initiates the verification of node *i*. Potentially, the references times may be updated whenever node *i* verifies node *j* without using *δjref* ; however, such updates should be done infrequently to reduce the overhead.

V. Disadvantage of the NEVO

The message delay large *δ* in PR packet is measured at node *j* and used by node *i for* neighbor verification. If *δ* is, then relative clock drifts of the network cards can cause false positives.

Usually, a clock is characterized by its “skew,” i.e., relative speed with respect to a reference clock, as well as an “off- set”, i.e., the time difference from the reference clock at a particular time (time 0 of the reference clock is often used). In the neighbor verification of NEVO, clock at node *i* can be considered as the reference clock (denoted as *t*) and the clock (denoted as *tj* (*t*)) at node *j* satisfies

$$tj\ (t) = aj\ t + bj \quad , \qquad (8)$$

where *aj* denotes *skew* and *bj* denotes *offset*. Assume node *i* sends two broadcast messages to node *j* successfully and each node records the times of sending or receiving the first bit of the message.

NEVO is based on the True Link protocol [9]. True Link does not have the clock skew problem, however, since it reserves the channel and completes its verification related transmissions within a few microseconds. The disadvantage is it requires modifications to the 802.11 protocol. On the other hand, NEVO uses network layer verification, which takes more time to complete, and thus requires clock drift correction.

VI. PROPOSED WORK

As we know that NEVO does not consider the impact of clock drift and it requires modification to the 802.11
Protocol and also uses network layer verification, which Take more time to complete and thus require clock drift correction. we propose an approach in which we are considering the impact of clock drift and which do not require the change in 802.11 protocol and also no change in network layer verification which is very time consuming.

REQUESTING NODE

1. Enters request in its own queue (ordered by time stamps)
2. Send a request to every node.
3. Wait for replies from all other nodes.
4. if own request is at the head of the queue and all replies have been received send the data
5. After sending the data send a release message to every node

OTHER NODES

1. After receiving a request, enter the request in the request queue (ordered by time stamps) and reply with a time stamp.
2. After receiving release message, remove the corresponding request from the request queue.
3. If own request is at the head of the queue and all replies have been received.

REFERENCES

[1] T. Sakthivel and R. M. Chandrasekaran, “Detection and Prevention of Wormhole Attacks in MANETs using Path Tracing Approach”, European Journal of Scientific Research, ISSN 1450-216X, Vol.76, No.2, 2012, pp.240-252.

[2] Shalini Jain, Dr.Satbir Jain, “Detection and prevention of wormhole attack in mobile adhoc networks”, International Journal of Compute Theory and Engineering, Vol. 2, No. 1 February, 2010, pp.78-86.

[3] S. Madhavi and K. Duraiswamy, “WAS-DP: Wormhole Attack in SAODV-Detection and Prevention”, European Journal of Scientific Research, ISSN 1450-216X, Vol.77, No.4, 2012, pp.560-569.

[4] Revathi Venkataraman, M. Pushpalatha, T. Rama Rao and Rishav Khemka, “A Graph-Theoretic Algorithm for Detection of Multiple Wormhole Attacks in Mobile Ad Hoc Networks”, International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009, pp.220-222.

[5] Xu Su, Rajendra V. Boppana, “Mitigating Wormhole Attacks using Passive Monitoring in Mobile Ad Hoc Networks”, IEEE Conferences, 2008, pp.1-5.

[6] Issa Khalil, Saurabh Bagchi, Ness B. Shroff, “MOBIWORP: Mitigation of the wormhole attack in mobile multihop wireless networks”, Ad Hoc Networks, 6, 2008, pp.344–362

[7] J. Eriksson, S. V. Krishnamurthy, and M.Faloutsos, “TrueLink: A practical countermeasure to the wormhole attack in wireless networks,” in Proceedings of IEEE ICNP, 2006.

[8] S. Marti, T. J. Giuli, K. Lai, and M.Baker, “Mitigating routine misbehaviour in mobile ad hocnetworks,” in Proceedings of MOBICOM, pp. 255–265, August 2000.

[9] R. V. Boppana and X. Su, “Secure routing techniques to mitigate insider attacks in wireless ad hoc networks,” IEEE Wireless Hive Networks Symposium, 2007.

AUTHORS

First Author – Ranjeeta Siwach, Computer Science & Engineering, B.S.Anangpuria Institute of Technology & Management
Second Author – Vanditaa Kaul, Computer Science & Engineering, B.S.Anangpuria Institute of Technology & Management