World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:12, 2008

# Analysis of Detecting Wormhole Attack in Wireless Networks

Khin Sandar Win, Department of Engineering Physics, Mandalay Technological University, Pathein Gyi, Mandalay.

***Abstract*—In multi hop wireless systems, such as ad hoc and sensor networks, mobile ad hoc network applications are deployed, security emerges as a central requirement. A particularly devastating attack is known as the *wormhole attack*, where two or more malicious colluding nodes create a higher level virtual tunnel in the network, which is employed to transport packets between the tunnel end points. These tunnels emulate shorter links in the network. In which adversary records transmitted packets at one location in the network, tunnels them to another location, and retransmits them into the network. The wormhole attack is possible even if the attacker has not compromised any hosts and even if all communication provides authenticity and confidentiality. In this paper, we analyze wormhole attack nature in ad hoc and sensor networks and existing methods of the defending mechanism to detect wormhole attacks without require any specialized hardware. This analysis able to provide in establishing a method to reduce the rate of refresh time and the response time to become more faster.

***Keywords*—Ad hoc network, Sensor network, Wormhole attack, defending mechanism.

## I. Introduction

Wireless Sensor Networks (WSNs) are rapidly emerging as a new field of research. WSNs are built with a large number of tiny and inexpensive sensor nodes that are equipped with low-bandwidth radios. In a Mobile Ad Hoc Network (MANET), each node serves as a router for other nodes which allows data to travel by utilizing multi hop network paths without relying on wired infrastructure. Unlike wired networks where the physical wires prevent an attacker from compromising the security challenges especially for military applications, emergency rescue operations, and short-lived conference or classroom activities. Security of such network is a major concern[3]. The open nature of the wireless medium makes it easy for outsiders to listen to network traffic or interfere with it. These factors make sensor networks potentially vulnerable to several different types of malicious attacks. These malicious nodes can carry out both Passive and Active attacks against the network. In passive attacks a malicious node only eavesdrop upon packet contents, while in active attacks it may imitate, drop or modify legitimate packets[1]. A typical example of particularly devastating security active attack is known as a wormhole attack. In which, a malicious node captures packets from one location in the network, and tunnels them to another malicious node at a distant point, which replays them locally. The wormhole attack can affect network routing, data aggregation and clustering protocols, and location-based wireless security systems. Finally, the wormhole attack can be launched even without having access to any cryptographic keys or compromising any legitimate node in the network in [2].

The rest of this paper is organized as follows; section 2 presents the significance of wormhole attack nature; section 3 studies analysis of detection and countermeasure of wormhole attacks and presents discussion and summary. In section 4 presents our proposed model and in section 5 followed by the simulation setup and results. Section 6 concludes the paper.

## II. Significance of Wormhole Attack and background

### *A. Problem statement*

This section describes wormhole attacks nature and problem statement. A wormhole attack is a particularly severe attack on MANET routing where two attackers connected by a high-speed off-channel link called the wormhole link. The wormhole link can be established by using a network cable and any form of “wired” link technology or a long-range wireless transmission in a different band. The end-point of this link (wormhole nodes) is equipped with radio transceivers compatible with the ad hoc or sensor network to be attacked. Once the wormhole link is established, the adversary record the wireless data they overhear, forward it to each other, and replays the packets through the wormhole link at the other end of the network. Replaying valid network messages at improper places, wormhole attackers can make far apart nodes believe they are immediate neighbors, and force all communications between affected nodes to go though them.

In general, ad hoc routing protocols fall into two categories: *proactive routing protocols* that rely on periodic transmission of routing updates, and *on-demand routing protocols* that search for routes only when necessary[4]. A wormhole attack is equally dangerous for both proactive and on-demand protocols.

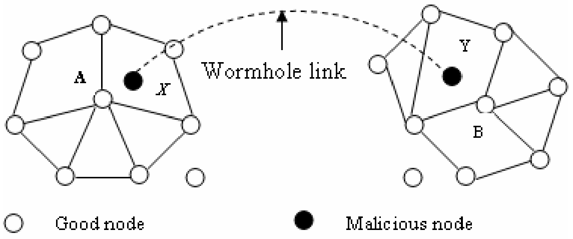


Fig.1. A network under a wormhole attack.

It should be nodded that wormholes are dangerous by themselves, even if attackers are diligently forwarding all packets without any disruptions, on some level, providing a communication service to the network. With wormhole in place, affected network nodes do not have a true picture of the network, which may disrupt the localization-based schemes, lead to the wrong decisions, etc. Wormhole can also be used to simply aggregate a large number of network packets for the purpose of traffic analysis or encryption compromise. Finally, a wormhole link is simply unreliable, as there is no way to protect what the attackers can do and when. Simply put the wormholes are compromising network security whether they are actively disrupting routing or not.

## III. Solutions to Wormhole attacks and Countermeasurements

In an ad hoc network, several researchers have worked on pretending and detecting wormhole attacks specifically. In section A we discuss a technique called 'packet leashes', which allows preventing packets from traveling farther than radio transmission range. In section B explain about wormhole prevention methods that rely on Round Trip message Time (RTT). Finally, in section C we discuss wormhole detection or prevention techniques suitable for only particular kinds of networks and in D discuss summary of wormhole discovery methods.

### *A. Packet leashes*

Packet Leash in[5], [6], [7] is a mechanism to detect and defend against wormhole attacks. The mechanism proposes two types of leashes for this purpose: Geographic and Temporal. In Geographic Leashes, each node knows its precise position and all nodes have a loosely synchronized clock. Each node, before sending a packet, appends its current position and transmission time to it. The receiving node, on receipt of the packet, computes the distance to the sender and the time it took the packet to traverse the path. The receiver can use this distance anytime information to deduce whether the received packet passed through a wormhole or not. In Temporal Leashes, all nodes are required to maintain a tightly synchronized clock but do not rely on GPS information. When temporal leashes are used, the sending node append the time of transmission to each sent packet $t_s$ in a packet leash, and the receiving node uses its own packet reception time $t_r$ for verification. The sending node calculates an expiration time $t_e$ after which a packet should not be accepted, and puts that information in the leash. To prevent a packet from traveling farther than distance L, the expiration time is set to:

$$t_e = t_s + (L/c) - \Delta \quad (1)$$

Where $c$ is the speed of light and Δ is the maximum clock synchronization error. All sending nodes append the time of transmission to each sent packet. The receiver compares the time to its locally maintained time and assuming that the transmission propagation speed is equal to the speed of light, computes the distance to the sender. The receiver is thus able to detect, whether the packet has travelled on additional number of hops before reaching the receiver. Both types of leashes require that all nodes can obtain an authenticated symmetric key of every other node in the network. These keys enable a receiver to authenticate the location and time information in a received packet.

### *B. Time-of-flight*

Another set of wormhole prevention techniques is similar to temporal packet leashes in [6], is based on the time of flight of individual packets. One possible way to prevent wormholes, as used by Capkun et al in [9] is to measure round-trip travel time of a message and its acknowledgement, estimate the distance between the nodes based on this travel time, and determine whether the calculated distance is within the maximum possible communication range.

The basis of all these approaches is the following. The Round Trip Travel Time (RTT) δ of a message in a wireless medium can, theoretically, be related to the distance $d$ between nodes, assuming that the wireless signal travels with a speed of light $c$:

$$d = \delta * c/2 \quad (2)$$
$$\delta = 2d/c \quad (3)$$

The neighbour status of nodes is verified if $d$ is within the radio transmission range $R$:

$R > d$ ($d$ within transmission range)
$$R > \delta * c/2 \quad (4)$$
$$\delta < 2R/c \quad (5)$$

In essence, the use of RTT eliminates the need for tight clock synchronization required in temporal leashes: a node only uses its own clock to measure time. When a de-facto standard of wireless ad hoc networks 802.11 Medium Access Control (MAC) protocol is used, such calculations are downright impossible. 802.11 imposes a short wait time of 10$\mu$s (SIFS) between the reception of a packet and sending of 802.11 acknowledgement. When 802.11 is used, transmission range $R$ is generally about 300 meters. The speed of light $c$ is $3\times10^{-8}$ $m/s$. Then, from equation 4:

$$\delta = 2d/c = 600m/3 \times 10^{-8}\ m/s$$
$$= 0.000002s = 2 \times 10^{-6} = 2\mu s \quad (6)$$

Therefore, the RTT is an order of magnitude smaller than the delay required by the protocol. We could, of course, account for this processing time by modifying formula 4 in the following manner:

$$\delta = 2d/c + S \quad (7)$$

where S is SIFS(Short Inter frame Space). However, note that wormhole attackers are not limited by the rules of the network, and could send their packets without 802.11-imposed delay.

Approaches based on RTT that one node sends a packet to another, the answer should arrive very shortly, ideally within the amount of time a wireless signal would travel between the nodes. If there is a wormhole attacker involved, packets end up traveling farther, and thus can not be returned within a short time.

### C. Specialized techniques

A wide variety of wormhole attack mitigation techniques have been proposed for specific kinds of networks: sensor networks, static networks, or networks where nodes use directional antennas. In this section, we describe and discuss such techniques, commenting on their usability and the possibility of their use in general mobile MANETs.

Hu and vans propose a solution to wormhole attacks for ad hoc networks in which all nodes are equipped with directional antennas in [10]. In this technique nodes use specific 'sectors' of their antennas to communicate with each other. Each copule of nodes has to examine the direction of received signals from its neighbour. Hence, the neighbour relation is set only if the directions of bothpairs match. This extra bit of information makes wormhole discovery and introduces substantial inconsistencies in the network, and can easily be detected. Wang and Bhargava [11] introduce an approach in which network visualization is used for discovery of wormhole attacks in stationary sensor networks. In their approach, each sensor estimates the distance to its neighbours using the received signal strength. All sensors send this distance information to the central controller, which calculates the network's physical topology based on individual sensor distance measurements. With no wormholes present, the network topology should be more or less flat, while a wormhole would be seen as a 'string' pulling different ends of the network together.

Lazos et al [12] proposed a 'graph-theoretical' approach to wormhole attack prevention based on the use of Location-Aware 'Guard' Nodes (LAGNs). Lazos uses 'local broadcast keys' - keys valid only between one-hop neighbours - to defy wormhole attackers: a message encrypted with a local key at one end of the network can not be decrypted at another end. Lazos proposes to use hashed messages from LAGNs to detect wormholes during the key establishment. A node can detect certain inconsistencies in messages from different LAGNs if a wormhole is present. Without a wormhole, a node should not be able to hear two LAGNs that are far from each other, and should not be able to hear the same message from one guard twice.

Khalil et al [2] propose a protocol for wormhole attack discovery in static networks they call LiteWorp. In LiteWorp, once deployed, nodes obtain full two-hop routing information from their neighbours. While in a standard ad hoc routing protocol nodes usually keep track of their neighbours are, in LiteWorp they also know who the neighbours' neighbours are, - they can take advantage of two-hop, rather than one-hop, neighbour information. This information can be exploited to detect wormhole attacks. Also, nodes observe their neighbours' behavior to determine whether data packets are being properly forwarder by the neighbour.

Song et al [14] proposes a wormhole discovery mechanism based on statistical analysis of multipath routing. Song observes that a link created by a wormhole is very attractive in routing sense, and will be selected and requested with unnaturally high frequency as it only uses routing data already available to a node. These factors allow for easy integration of this method into intrusion detection systems only to routing protocols that are both on-demand and multipath.

### D. Summary of wormhole attack.

TABLE 1: SUMMARY OF WORMHOLE DISCOVERY METHODS

| Method | Requirements | Commentary |
|---|---|---|
| Packet leashes, geographical, [6] | GPS coordinates of every node;Loosely synchronized clocks (ms) | Robust, straightforward solution; inherits general limitations of GPS technology |
| Packet leashes, temporal [5], [7] | Tightly synchronized clocks (ns) | **Impractical**; required time synchronization level not currently achievable in to sensor networks |
| Packet leashes, end-to-end [8] | GPS coordinates; Loosely synchronized clocks (ms) | Inherits limitations of GPS technology |
| Time of flight [9] | Hardware enabling one-bit message and immediate replies without CPU involvement ([9]) | **Impractical**; likely to require MAC-layer modifications |
| Directional antennas [10] | Directional antennas on all nodes [10] or several nodes with both GPS and directional antennas [13] | Good solutions for networks relying on directional antennas, but not directly applicable to other networks |
| Network visualization [11] | Centralized controller | Seems promising; Works best on dense networks; Mobility not studied; Varied terrains not studied |
| Localization[12] | Location-aware 'guard' Nodes | Good solution for sensor networks; Not readily applicable to mobile networks. |
| LiteWorp [2] | none | Applicable only to static stationary networks; **Impractical** |
| Statistical analysis [14] | no requirements | Works only with multi-path on-demand protocols; |

## IV. IMPROVED ALGORITHMS

In this section, algorithms used in the DaW –Defence against Wormhole security model, monitoring nodes, calculation of trust and wormhole detection are discussed. Whenever routing takes place in the network, analysis of the frequencies of links in different routes is done. If any of the links are suspicious, then the available trust information is used to check if the link is that of a wormhole. In the trust model used, nodes monitor neighbours based on their packet drop pattern and not on the measure of number of drops.

### *A Techniques for Wormhole Detection*

There are several simple techniques to detect wormholes in a network but these have some basic flaws which are discussed in the current section.

- **Link Frequency Analysis.** Analysis of the link frequency is a simple method to detect a wormhole in a network. Abnormally high frequency of a link could suggest that it can be a wormhole luring traffic into it. But in the case of cluster networks where the bottleneck links offer comparable delays as that of a wormhole in the network, the traffic might be equally distributed between the bottleneck link and the wormhole link and there is no way to find whether there is a wormhole and if found, it will be difficult to identify the wormhole link.
- **Trust Based Model.** Another significant method to detect wormholes is by the use of trust information. Nodes can monitor the behaviour of their neighbour and rate them. Assuming that a wormhole drops all the packets it receives as in blackholes, a wormhole in such a system should have the least trust level and can be easily eliminated. Drops in bottleneck in a network could be due to congestion, which could be triggered by improper routing, high TCP window sizes, sudden bursts of traffic from a node etc. But all these drops occur in bursts and network gets reconfigured after congestion. For example, if there are a lot of drops in TCP, the window size is decreased. Hence, the drop of packets in bottleneck is generally high only during congestion after which it is brought down again.

### *B. Monitoring Neighbours.*

In this security model, nodes go into promiscuous mode immediately after sending a packet to their neighbour. They monitor to check if the neighbour is transmitting it to the intended sender or dropping it. This can be found by listening to the packet header of the retransmission. If the destination is not transmitting to the intended destination or if the packet is simply dropped, then the source counts this as a drop. Hence every node in the network keeps track of the number of packets that are sent and dropped for each of its neighbours. This information is stored periodically for different intervals. For each neighbour, a node monitors the number of packets dropped $D_p$ and packets sent $S_p$ to it in that interval. I – 1, I – 2, I – 3, etc., are various intervals for which the observations are made. The size of the observation window or the number of interval information that is stored is dependent on the memory available in each sensor node with accuracy as the tradeoff. The information collected is stored in the form of an array in the node as shown in Table.2. Each of the rows in the array represents a neighbour and each column is the interval in which the observation is made. ($S_p1$, $D_p1$) is the packets sent and dropped by the corresponding neighbour in I -1 and ($S_p2$, $D_p2$) is the packets sent and dropped by the corresponding neighbour in I -2 and so on.

TABLE. 2. STORAGE OF NEIGHBOUR INFORMATION

| | *I-1* | *I-2* | *I-3* | *I-4* | *I-5* | . | . | . | *I-m* |
|---|---|---|---|---|---|---|---|---|---|
| n1 | $(S_p1,D_p1)$ | $(S_p2.D_p2)$ | $(S_p3,D_p3)$ | $(S_p4,D_p4)$ | $(S_p5,D_p5)$ | . | . | . | $(S_pm,D_pm)$ |
| n2 | $(S_p1,D_p1)$ | $(S_p2.D_p2)$ | $(S_p3,D_p3)$ | $(S_p4,D_p4)$ | $(S_p5,D_p5)$ | . | . | . | $(S_pm,D_pm)$ |
| n3 | $(S_p1,D_p1)$ | $(S_p2.D_p2)$ | $(S_p3,D_p3)$ | $(S_p4,D_p4)$ | $(S_p5,D_p5)$ | . | . | . | $(S_pm,D_pm)$ |
| n4 | $(S_p1,D_p1)$ | $(S_p2.D_p2)$ | $(S_p3,D_p3)$ | $(S_p4,D_p4)$ | $(S_p5,D_p5)$ | . | . | . | $(S_pm,D_pm)$ |
| n5 | $(S_p1,D_p1)$ | $(S_p2.D_p2)$ | $(S_p3,D_p3)$ | $(S_p4,D_p4)$ | $(S_p5,D_p5)$ | . | . | . | $(S_pm,D_pm)$ |
| .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| nj | $(S_p1,D_p1)$ | $(S_p2.D_p2)$ | $(S_p3,D_p3)$ | $(S_p4,D_p4)$ | $(S_p5,D_p5)$ | . | . | . | $(S_pm,D_pm)$ |

### *C. Trust Evaluation.*

The packet dropped versus packets sent by a wormhole node is given in Fig. 2. The bar chart shows the region where the drops occur. The correlation is very high with the packets sent.

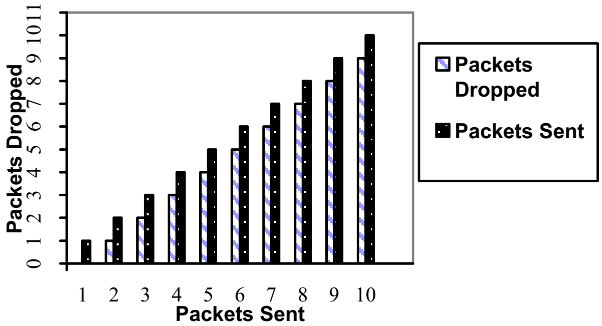


Fig.2. Packets dropped versus packets sent

In this algorithm, Carl Pearson's correlation coefficient has been made use of to calculate the correlation between packets sent and that which are dropped. The expression for trust is as follows:

$$t_a^b = \frac{\sum_{SD} s_i\, d_i - \frac{\sum_S s_i \sum_D d_i}{n}}{\sqrt{\left[\sum_S s_i^{\,2} - \frac{\left[\sum_S s_i\right]^2}{n}\right]\left[\sum_D d_i^{\,2} - \frac{\left[\sum_D d_i\right]^2}{n}\right]}}$$

Where,

$t_a^b$ - correlation coefficient of node b with respect to node a
$s_i$ - packets sent by a to b
$d_i$ - Number of packet from node a dropped by node b
n - number of intervals in the observation window
S – set of packets sent at different intervals by node a to node b
D – set of packets dropped by node b which were sent by node a at different intervals

$t_a^b$ indicates the drop pattern and hence is a measure of the trust. Simulation experiments have been conducted to know the typical value of correlation coefficient for a wormhole node. Our studies show that the correlation coefficient is generally more than 0.9 for wormholes. The correlation coefficient is calculated for all the neighbours and the trust vector of a node is constructed. Trust vector of a node is the vector containing the $t_a^b$ values of each of its neighbours. Fig.4 shows the Trust vector of a network with n nodes.

### D. *Algorithm for Detection of the Wormhole.*

With the trust information available through neighbour monitoring, it is simple to detect the wormhole. The algorithm for detection of Wormhole is run during the routing phase. The procedure for wormhole detection is described by means of a flowchart given in Fig. 3.

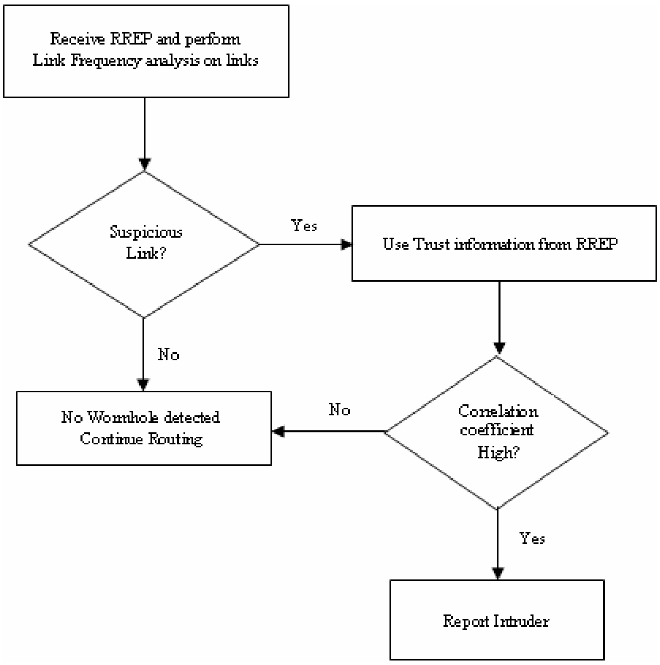

Receive RREP and perform Link Frequency analysis on links
Suspicious Link?
Yes
Use Trust information from RREP
No
No Wormhole detected Continue Routing
No
Correlation coefficient High?
Yes
Report Intruder

Fig. 3 Flowchart for detection of wormhole

[Broadcast RREQ]
1. Support nodes want to transmit to node d. It broadcasts the route request RREQ as shown in Fig.5. It attaches its trust vector in the header.

[Append Trust Vectors]
2. Every nodes in the intermediate path also attaches it's trust vector in the RREQ message.

[Send RREP]
3. Destination node receives the RREQ and sends a route reply RREP. It copies the Trust Vectors from the RREQ into the RREP.

[Check for suspicious link]
4. Source receives various RREP coming through different routes. Check if there is a link with very high frequency using the following expression:

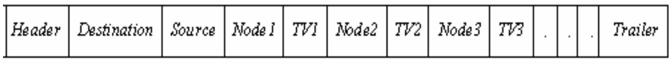
$$P_i = n_i / N, \text{ for all } I_i$$
$$P_{max} = max (p_i)$$

Where, $R$ is the set of all obtained routes, $I_i$ is the i$^{th}$ link, $n_i$ is the number of times that $I_i$ appears in $R$, N is the total number of links in $R$, and $P_i$ is the relative frequency that $I_i$ appears in $R$.

[Confirm wormhole]
5. *If* $P_{max} > P_{threshold}$(0.2), check the trust information available in the RREP of that route. If the value of correlation coefficient for packets dropped to that sent is > t $_{threshold}$ (0.9), then the node is malicious, inform the operator.
else continue with routing process.

| *Ni* | *NEIGHBOUR(Ni)=(Nx,Ny,Nz)* | *TRUST VECTOR(NI)=(tx,ty,tz)* |
|---|---|---|

Fig. 4 Trust vector



| *Header* | *Destination* | *Source* | *Node1* | *TV1* | *Node2* | *TV2* | *Node3* | *TV3* | . | . | . | *Trailer* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Fig. 5 Format of RREQ

## V. PERFORMANCE EVALUATION

### A. *Simulation Setup*

The performance of DaW was evaluated against existing method of link frequency analysis. Simulations are performed in ns-2 network simulator [15]. We have implemented both Link Frequency analysis and DaW on DSR routing protocol. The wormholes have multiple interfaces, one for communicating with the sensor nodes and another wired interface to the colluding wormhole. The simulations are done for various numbers of nodes in each of the cluster. Simulation parameters are given in Table.3. The wormhole nodes tunnel only RREQ and RREP packets between them and all other packets through the route are dropped as in the case of a blackhole. Nodes monitor their neighbour by going into promiscuous mode. Each interval spans over a period of 20 seconds and at any time a maximum of 5 intervals are observed and are used for trust evaluation. The size of the interval and the number of intervals observed both are variables and can be changed based on the available resources.

**Table. 3. Simulation Parameters**

| | |
|---|---|
| Examined Protocol | DaW, LF analysis |
| Simulation time | 900 seconds |
| Simulation area | 200×200 m |
| Number of nodes | 16, 24, 32, 40, 48, 56 |
| Transmission range | 50 m |
| Traffic type | CBR(UDP) |
| Maximum Connections | (7*n-26)/4, where n is the no. of nodes |
| MAC | 802.11 |
| Payload size | 512 bytes |
| Packet rate | 2 pkt/sec |
| Number of source nodes | 6 |
| Number of wormhole | 2 |

The sources are connected with a constant bit rate application. The simulations are run for various cluster sizes for duration of 900 seconds.

### *B. Precision of Alarms.*

The results of the simulations in terms of the total number of alarms raised and the genuine alarms out of them are tabulated. This gives us an idea of how reliable an alarm raised by the protocol is. The precision is defined as follows:

$$Precision of\ Alarms = \frac{Number\ of\ alarms\ for\ genuineWormholes}{Total\ number\ of\ Alarms}\%$$

The total number of alarms might include apart from genuine wormhole. The precision is decided by the proportion of genuine wormholes detected. Based on the simulations, the graph of Precision of Alarms versus the number of nodes is plotted in Fig.6. From the results of the simulations, it is clear that the normal link frequency based approach could mislead into believing bottleneck links as wormholes. The precision decreases with the increase in the size of the network. This is due to the fact that the number of possible routes between two nodes increases as the network becomes bigger.

There are a large number of links which have comparable high frequency. Link Frequency analysis cannot make an accurate detection and it simply gives an alarm for a wormhole. DaW on the other hand, uses the trust information to verify whether a suspicious link is a wormhole. Also use of time-scale information in trust evaluation enables DaW to clearly identify the behavior of a wormhole which existing Trust based models would fail as they keep track of amount of drops rather than the pattern of drop.

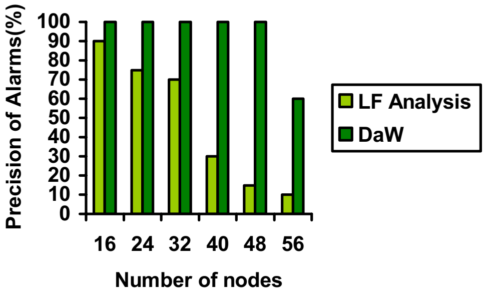


Fig. 6. Precision of Alarms

### *C. Wormholes Detected.*

DaW relies on the fact that the wormhole links would have a high frequency. But, suppose a wormhole is not active or if it doesn't have much traffic passing through it, then the wormhole may not be detected by the algorithm at all. The number of wormholes detected with respect to the size of the network is shown in Fig.7. As the size of the network increases, the number of possible routes between source and destination increases. The network may have many wormholes, but not all may be active during the simulation. This is evident when the size of network becomes 48 and 56, one of the wormholes is never chosen as part of some route, hence, is never detected.

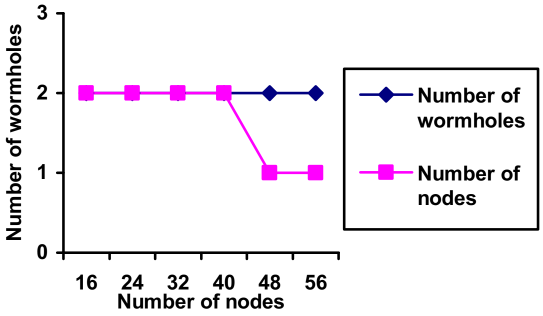


**Fig.7 Number of wormholes detected**

### *D. Amount of False Positives.*

False positive alarms are raised when a genuine network link is detected as a wormhole. The percentage of false positive is calculated as:

$$\text{Percent of False Positives} = \frac{Number\ of\ Real\ Links\ detected\ as Wormholes}{Total\ number\ of\ Real\ Links}\%$$

The percent of false positive alarms for the total number of real connections versus the number of nodes is plotted in Fig. 8.

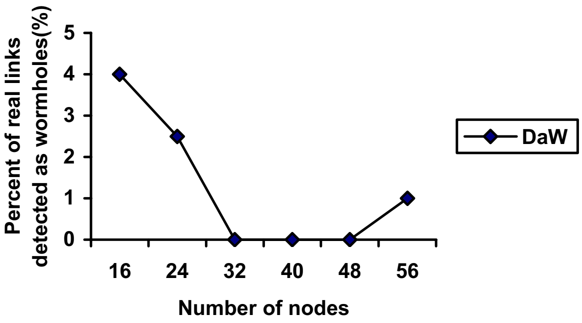


Fig.8. Amount of False Positive alarms

The links which are close to the wormhole tend to have high link frequencies. All these links also become suspicious. Suppose these links contain the wormhole node in them, SaW might falsely detect these links as a wormhole connection as the wormhole will have a low trust value.

## VI. CONCLUSION

In this paper, we address the various solutions available for wormhole attack in wireless Ad hoc and sensor networks. More specifically, we address algorithms used in the DaW security model that incorporates a detection and defense mechanism against the wormhole attack. The performance of DaW in terms of precision of alarms, amount of false positive has been found to be good. The alarms were found to be more precise than LF analysis. The performance of secure in multi hop wireless systems with the help of ns-2 simulations and our routing protocol can efficiently defend against the wormhole attack and achieve low delay.

## REFERENCES

[1] R.E.Kassi, A.Chehab, and Z. Dway, "DAWWSEN: A Defense Mechanism against Wormhole Attacks in Wireless Sensor Networks", in proceeding of the second International conference on innovations in information Technology (ITT' 05), UAE, September 2005.

[2] T. Park and K. Shin, "LISP: A Lightweight Security Protocol for Wireless Sensor Networks", in proceedings *of ACM transaction on Embedded Computing systems*, August 2004.

[3] Y.-C. Hu, A. Perrig, A Survey of Secure Wireless Ad Hoc Routing, Security and Privacy Magazine, IEEE, vol. 2, issue 3, pp. 28-39, May 2004.

[4] D. Johnson, D. Maltz, and J. Broch, "The dynamic Source routing Protocol for Multi hop Wireless Ad hoc Networks," in Ad Hoc networking, C. Perking, Ed., Addson-Wesley, 2001.

[5] Y.-C. Hu, A. Perrig, D. B. Johnson, "Wormhole Attacks in Wireless Networks," Selected Areas of Communications, IEEE Journal on, vol. 24, numb. 2, pp. 370- 380, 2006.

[6] Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks", in *proceedings of INFOCOM*, 2004.

[7] W. Weichao, B. Bharat, Y. Lu, X. Wu, Wiley Interscience, "Defending agains Wormhole Attacks in Mobile Ad Hoc Networks," Wireless Communication and Mobile Computing, January 2006.

[8] S. Capkun, L. Buttyan, J.-P. Hubaux, SECTOR: Secure Tracking of Node Encounters in Multi-Hop Wireless Networks, October 2003, Processings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks.

[9]. On the Survivability of Routing Protocols in Ad Hoc Wireless Networks, A. Baruch, R. Curmola, C. Nita-Rotaru, D. Holmer, H. Rubens, Converence on Security and Privacy for Emerging Areas Communications, SecureComm 2005, September 2005

[10] L. Hu, D. Evans, Using Directional Antennas to Prevent Wormhole Attacks, 14 Proceedings of the 11th Network and Distributed System Security Symposium, pp. 2003.

[11] W. Wang, B. Bhargava., Visualization of wormholes in sensor networks, Proceedings of the 2004 ACM workshop on Wireless Security, pp. 51-60, 2004.

[12] L. Lazos, R. Poovendran, Serloc: Secure Range-Independent Localization for 21- 30, Wireless Sensor Networks, Proceedings of the ACM Workshop on Wireless Security, pp. October 2004.

[13] L. Lazos, R. Poovendram, C. Meadows, P. Syverson, L.W. Chang, Preventing Wormhole Attacks on Wireless Ad Hoc Networks: a Graph Theoretical Approach, IEEE Communication Society, WCNC 2005

[14]. N. Song, L. Qian, X. Li, Wormhole Attack Detection in Wireless Ad Hoc Networks: a Statistical Analysis Approach, Parallel and Distributed Processing Symposium, 2005, Proceedings of, 19th IEEE International IPDPS'05, 04-08 April 2005, pp.

[15] ns-2 homepage: www.isi.edu/nsnam/ns/index.html