E2SIW: An Energy Efficient Scheme Immune to Wormhole Attacks in Wireless Ad Hoc Networks

Sanjay Kumar Dhurandher
CAITFS, Division of Information Technology
Netaji Subhas Institute of Technology
University of Delhi, Delhi, India
E-mail: dhurandher@rediffmail.com

Isaac Woungang
Department of Computer Science
Ryerson University
Toronto, Ontario, Canada
E-mail: iwoungan@scs.ryerson.ca

Abhishek Gupta
CAITFS, Division of Information Technology
Netaji Subas Institute of Technology
University of Delhi, Delhi, India
E-mail: abhishekg.nsit@gmail.com

Bharat K. Bhargava
Department of Computer Sciences
Purdue University
West Lafayette, Indiana, USA
E-mail: bb@cs.purdue.edu

***Abstract*—Wormhole attacks are considered as a severe security threat in multi-hop wireless ad hoc networks. In this paper, we propose an Energy-Efficient Scheme Immune to Wormhole attacks (our so-called E2SIW). This protocol uses the location information of nodes to detect the presence of a wormhole, and in case a wormhole exists in the path, it finds alternate routes involving the nodes of the selected path so as to obtain a secure route to the destination. The protocol is capable of detecting wormhole attacks employing either hidden or participating malicious nodes. Simulations are conducted, showing that E2SIW can detect wormholes with a high detection rate, less overhead, and can consume less energy in less time, compared to the De Worm wormhole detection protocol, chosen as benchmark.**

Keywords*—*wormhole attacks, ad hoc networks, energy-efficiency, detection, prevention, malicious node, hidden node

I. INTRODUCTION

Wireless ad hoc networks consist of nodes that cooperate dynamically to establish routes using wireless links without the use of a centralized authority. Each node acts as a router that selects the next node to which data must be sent to accomplish efficient routing. Due to the principal characteristics of wireless ad hoc networks such as dynamic topology, stringent resource availability, lack of centralized authority [1, 2], they are vulnerable to various security threats. One of the most severe types of attacks that can be launched against these networks is the wormhole attack.

A wormhole attack is a type of a collaborative attack where the attacker uses two malicious transceivers to degrade the performance of the network or analyze the network traffic [3]. These transceivers constitute the end points of the wormhole. The endpoints are connected using a high-speed link (called a tunnel (see Fig. 1)). Packets are captured from one endpoint (node) and are tunneled to the other end (malicious node) in some other part of the network, where they are replayed, typically without modification. Fig. 1 illustrates a network topology affected by a wormhole. Nodes 1, 2, 3, 4 and 5 are in the transmission range of M1. Nodes A, B, C and D are in the transmission range of M2. Nodes 1, 2, 3, 4 and 5 will consider nodes A, B, C and D as their immediate neighbors due to the presence of wormhole.

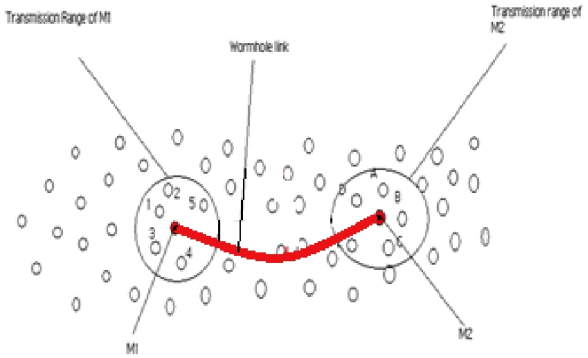

Figure 1: Illustration of a wormhole attack.

Wormhole attack is considered as one of the most severe form of attacks on ad hoc networks. The severity of wormhole attacks lies in the fact that it is capable of disrupting a significant amount of network traffic. A severity analysis of a wormhole attack has been done in [4], where the authors showed that in shortest path routing

978-0-7695-4652-0/12 $26.00 © 2012 IEEE
DOI 10.1109/WAINA.2012.85

protocols, two strategically located malicious nodes can disrupt an average 32% of all communications across the network, when the nodes of the network are distributed uniformly.

When the wormhole targets a particular node in the network, it can disrupt an average 30% to 90 % (based on the location of the target) of all communication between the target node and all other nodes in the network. In a network of grid topology, it has been proved that 40% to 50% of all communication can be disrupted if the wormhole is placed along the diagonal of the grid. The above discussion illustrates the severity of wormhole attacks in wireless ad hoc networks**.**

An effective wormhole attack must attract a significant amount of network traffic by providing a perceived short-cut through the network. Hence, routes going through the wormhole must be shorter than alternate routes through valid network nodes. Therefore, the effectiveness of the wormhole will increase with the increase in length of the wormhole link. This observation is the basis of E2SIW, our wormhole detection protocol. E2SIW uses the location information of the nodes and the routing variations between neighboring nodes along a path from a source to a destination to detect wormhole attacks. The proposed protocol is simple, localized, and requires no synchronization or a special hardware other than the Global Positioning System (GPS).

The rest of the paper is organized as follows. Section II describes related works on wormhole detection schemes. Our proposed E2SIW protocol is described in Section III. In Section IV, the simulation results comparing E2SIW against the De Worm detection protocol are presented. Finally, Section V concludes our work.

II. RELATED WORK

Wormhole attacks are considered as one of the most devastating types of attacks against wireless ad hoc networks. These types of attacks are detrimental against both On Demand and Pro-active routing protocols. A wide variety of wormhole attack mitigation techniques have been proposed in the literature.

Hu et al. [5] proposed a mitigation solution to prevent wormhole attacks that suggested the use of geographical or temporal packet leashes. A geographical leash requires each node to know its own location and all nodes to have loosely time synchronized clocks. The nodes need to securely exchange location information. A sender node can then ensure that the receiver is within a certain distance and detect discrepancies therein.

With temporal leashes, all nodes must have tightly synchronized clocks. The receiver will compare the receiving time with the sending time attached with the packet. It can determine if the packet has travelled too far in too little time and detect the wormhole attack. The proposed mitigation solution is robust and reliable, however there are some issues attached to this solution. For instance, the nodes require tightly synchronized clocks; special hardware is needed to achieve tight time synchronization between the nodes which makes the setup complex and costly; each node requires predicting the sending time and computing the signature while having to timestamp the message with its transmission time.

In [6], Hu and Evans proposed a solution to detect wormhole attacks in ad hoc networks in which all nodes are equipped with directional antennas. In their technique, nodes use specific 'sectors' of their antennas to communicate with each other. Each pair of nodes has to examine the direction of the received signals from its neighbor. Hence, the neighbor relation is set only if the directions of both pairs match. This additional bit of information introduces substantial inconsistencies in the network, leading to wormhole detection. The protocol uses a special hardware that adds some expenses and complexity, as well as a need for a special customization.

In [7], Khalil et al. proposed a protocol for wormhole attack discovery in static networks (so-called LiteWorp). Once LiteWorp is deployed, the nodes obtain full two-hop routing information from their neighbors. In addition to the fact that nodes keep track of their neighbors (as in standard ad hoc routing protocols), a node knows who are the neighbors of its neighbors and can take advantage of two-hop neighborhood information, rather than just one-hop neighborhood information. This information is then exploited to detect wormhole attacks. Also, a node can observe its neighbor's behavior to determine whether the data packets are being properly forwarded by this neighbor node.

In [8], Hayajneh et al. proposed a protocol called De Worm that uses the routing discrepancies between the neighbors along the path from the source to the destination to detect a wormhole. The De Worm scheme is based on the observation that for a wormhole to have a successful impact on the network, it must attract a significant amount of network traffic towards itself and the routes going through the wormhole must be shorter than the alternate routes going through the valid network nodes. Each node on the route selected by the routing protocol runs the wormhole detection algorithm, which

involves the acquisition of routes for a target node by all the one-hop neighbors of the node running the algorithm. The protocol is run on all the nodes of the route until a wormhole is detected. The protocol is not energy efficient as it has a large control packets overhead associated with it. The fact that all the one-hop neighbors of a node running the De Worm try and find routes to a target node makes the protocol time consuming.

In [9], Wang et al. proposed a more generic mechanism that classifies wormhole attacks, in the sense that it is end-to-end and does not rely on trust among neighbors. However, this mechanism requires a high computation and storage power since period wormhole detection packets are transmitted and the responses are used to compute each node's position and velocity. To alleviate this burden, a complementary mechanism (so-called Cell-based Open Tunnel Avoidance (COTA) was proposed to manage the detection information.

In [10], Wang et al. proposed a distributed mechanism referred to as Dis-VoW that can be used to detect wormhole attacks in underwater sensor networks. The approach consisted in visualizing the distortions in edge lengths and angles among neighboring sensors. Based upon these distortions, a wormhole indicator is defined to identify the fake neighbor connections. Dis-VoW does not depend on any special hardware.

Unlike previous works, our proposed E2SIW protocol is simple, localized, and does not require synchronization.

III. PROPOSED E2SIW PROTOCOL

E2SIW uses the location information of nodes to detect the presence of a wormhole in the selected route and finds an alternate path to a target node that bypasses the wormhole, thus preventing the attack. To attract a large amount of data traffic, the route through the wormhole should have a smaller number of hop counts as compared to other routes. The effectiveness of the wormhole will depend on its ability to shorten the route for a source-destination pair and thus, will increase with the increase in the length of the wormhole. The alternative routes found out will differ significantly in the number of hop counts as compared to the route that passes through the wormhole. In order to acquire alternate routes that are not affected by wormholes, our E2SIW scheme tries to find out the alternate routes between nodes that are a short distance apart in a hop-by-hop fashion along the route. The algorithm for the proposed protocol is explained in the sequel.

A. Network and Attack Model.

Before presenting our proposed protocol, a brief description of the network and attack model follows. Let's consider an ad hoc network consisting of n nodes. The network can have at most *(n)(n-1)* source destination node pairs. The network is assumed to be symmetric in nature and each node in the network is assumed to be equipped with a GPS (Global Positioning System) module, which provides its location information. The information provided by GPS is assumed to be error free.

The wormhole is defined as two nodes *M1* and *M2* in the network, and it is assumed that the wormhole is an out of band, which uses a high speed link to connect the nodes *M1* and *M2*. The wormhole nodes can be hidden or non-hidden. By hidden, we mean that these nodes are not part of the network, that is, they do not disclose any of their information (such as address, etc) to the network. One hop neighbors of *M1* are linked to the one hop neighbors of *M2* via the wormhole.

In the case of non-hidden malicious nodes, the endpoints of the wormhole take part in the routing as legitimate nodes, and thus, they use their identities for routing purpose, i.e. they share their addresses with other nodes in the network. It is also assumed that the wormhole nodes are not capable of maliciously changing the data passing through them or are not capable of any other advanced behavior.

B. Algorithm

We use the following terminologies:

- R_{SD}*:* Route selected by the routing algorithm for the source - destination node pair *(S, D).*
- *R(i):* Node on the route *i* hops away from the source *S*
- *T(R(i)):*Target node for *the R(i)* node. It is a node along the route that is *2* hops away from the *R(i)* i.e. *R(i+2)*
- *Z (R(i)):* One hop neighbor of *R(i)* that is nearest to *R(i).*
- *N(R(i)):* Neighbor list of node *R(i).*
- $H_{(X-Y)}$*:* Number of hops of the route between *X* and *Y.*

The proposed protocol works as follows:

Step 1: Initialize *i = 0.*

Step 2: The node *R(i)* will set the target node *TR(i)* to be the node two hops away in the route selected, i.e. *T(R(i))= R(i+2).*

Step 3: R(i) will broadcast a "HELLO packet". The nodes that will hear the "HELLO packet" (one-hop neighbors of *R(i)*) will reply back to *R(i)*. The replies will contain their respective positions i.e. their GPS coordinates.

Step 4: R(i) will calculate its distance from all its neighbors and creates a table containing the distance information of the neighbor nodes.

Step 5: R(i) will mark node *R(i+1)* and node *Z(R(i))* one of the neighbors of *R(i)* that is at minimum distance from *R(i)* and that is not *R(i+1)*. *R(i+1)* is known to *R(i)* during the route discovery phase of the routing protocol. *R(i)* also creates a list *N(R(i))* containing the addresses of the one-hop neighbors of *R(i).*

Step 6: *R(i)* unicasts the list (*N(R(i)), T(R(i))*) to *Z(R(i))* and asks it to find a route to the target node *T(R(i))*, such that the route does not include any other node in *N(R(i))* and the node *R(i)*. Node *Z(R(i))* will run the network routing algorithm to find a route to *T(R(i))*. *T(R(i))* will reply back with a packet containing the number of hops $H_{(Z(R(i))-T(R(i)))}$ of the route found. *Z(R(i))* will inform *R(i)* about the number of hops of the discovered route.

Step 7: R(i) will test for the existence of a wormhole by comparing the length $H_{(Z(R(i))-T(R(i)))}$ to a threshold value.

if ($H_{(Z(R(i))-T(R(i)))} \geq threshold$)

Wormhole detected

else

Wormhole not detected.

The *threshold* value can be calculated based on parameters such as the number of hops of the selected route (R_{SD}), the transmission range of the genuine nodes, to name a few. The accuracy of the protocol depends on the threshold value. Smaller values will increase the detection rates but accuracy will be decreased i.e. the number of false positives will be increased. Higher values will result in accurate detection, but with lesser detection rate. Small length wormholes will escape the detection algorithm. If a wormhole is detected, then, go to Step 9 else, go to Step 8.

Step 8: Increment '*i*' by 1 and go back to Step 2.

Step 9: The node routes the data through an alternate path between *Z(R(i))* and the *T(R(i))* that has been discovered during the wormhole detection process in the above steps.

E2SIW is a modified version of the De Worm detection protocol [8]. De Worm uses multiple route acquisitions to detect a wormhole. In the De Worm scheme, all the one-hop neighbors of *R(i)* are asked to explore the routes to *T(R(i)),* which results in large control packets overhead. E2SIW uses the location information of nodes to limit the number of route acquisitions to just one, resulting in a significant decrease of the control packets overhead as well as in the wormhole detection time.

ii) Special Case

In the case where wormhole is not yet detected and E2SIW is currently running on *R(i)*, where *R(i)* is the last node before the destination on the route, the protocol will work as follows:

Step 1: R(i) creates a neighbor list *N(R(i))* by broadcasting a "HELLO packet" and selects the nearest neighbor *Z(R(i))*. It also asks node D (destination node) to provide its neighbor list *N(D)*.

Step 2: R(i) selects a target node *T(R(i))* belonging to *N(D)* and unicasts the list (*N(R(i)), N(D), T(R(i))*) to *Z(R(i))* and then asks it to find a route to *T(R(i))* such that the route must not contain any node belonging to *N(D)* and *N(R(i))*.

Step 3: T(R(i)) replies back with a packet containing the number of hops of the route $H_{(Z(R(i))-T(R(i)))}$. *Z(R(i))* informs *R(i)* about the number of hop counts of the selected route.

Step 4: R(i) tests the existence of a wormhole by comparing the value $H_{(Z(R(i))-T(R(i)))}$ to the selected threshold value as discussed previously.

C. Analysis

E2SIW has several advantages over the existing methods of mitigating wormhole attacks. Our method not only detects a wormhole, but it is also applicable to fault tolerance as it prevents the wormhole attack by finding alternate paths bypassing the wormhole. Our protocol is simple and is free of the use of any special hardware other than the GPS, which provides the nodes with location information. There is no compromise of data packets in the wormhole detection process. The detection process starts after the route reply has been received by the source node from the destination node. No data packets are sent before the whole process is finished for a hop. The detection process can be applied on demand i.e. when there is a need to verify the presence of a wormhole in the network.

In the case of non-hidden wormhole, the participation of malicious nodes of the wormhole in the detection process can be avoided by making the target node 3 hops away (instead of 2 hops away as in our case). In this case, at the hop where a wormhole will be detected, the target node will be a node one-hop away from the second malicious node, thus will help avoiding the participation of malicious nodes.

IV. PERFORMANCE EVALUATION

To assess the performance of our proposed E2SIW protocol, E2SIW has been compared against the De Worm

protocol [8]. Both protocols were implemented on top of the AODV routing protocol for ad hoc networks.

A. Simulation Setup

The simulations were run using GloMoSim [11]. We considered a topology with N nodes placed uniformly in 1200 m x 1200 m square area. N belongs to the set {140, 160, 180, 200}. Varying the number of nodes had the effect of changing the node degree as the terrain dimension remains constant. The transmission power of the nodes was kept constant at 176 m. A source-destination node pair was chosen randomly from the set of nodes. The malicious nodes of the wormhole were chosen such that the wormhole is able to incorporate itself in the route to the destination.

B. Simulation Results

i) Time Taken to Detect the Wormhole

This is the time taken by the protocol to successfully detect a wormhole. For both protocols, the wormhole is detected when the algorithm runs on a node that is a neighbor of a malicious transceiver. Thus, the time taken to detect a wormhole depends on the location of the wormhole along the route. The results are shown in Fig. 2.

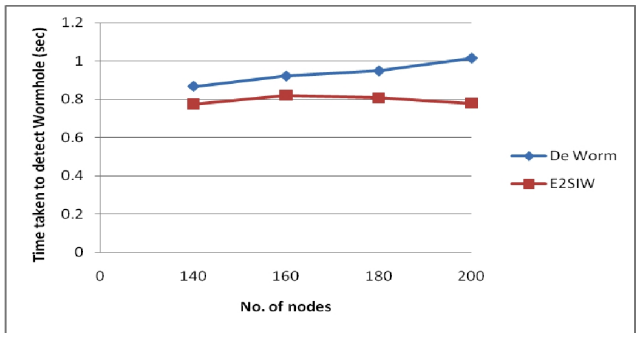

Figure 2: Variation of time taken to detect wormhole with the number of nodes in the network.

In Fig. 2, it can be observed that the time taken to detect a wormhole is significantly less in the case of E2SIW as compared to that of De Worm. The average time taken to detect a wormhole by the De Worm protocol is 0.93 sec whereas it is 0.79 in the case of E2SIW, thus giving a drop of 15.15%. The drop in the detection time can be explained by the fact that E2SIW uses a single route acquisition at each hop for the detection purpose whereas the De Worm uses multiple route acquisitions (depending on the number of one-hop neighbors) at each and every hop until a wormhole is detected. Thus, De Worm waits until it receives all the replies from the target node at each and every hop making it a time consuming process.

ii) Energy Consumed

This is the total amount of energy consumed in running the protocol. It includes the energy requirement in transmission of control packets, data packets, and energy required in processing and maintaining the data in the memory. It is assumed that each transmission requires *2Joules* of energy (the values chosen are simulation dependent and may vary from simulation to simulation). The processing required to create a neighbor list consumes *1Joule* of energy. The results for this metric are depicted in Fig. 3.

In Fig. 3, it can be observed that there is a significant decrease in energy consumption in E2SIW compared to the De Worm protocol. The average energy required in the De Worm protocol is 3937. In E2SIW, the value drops to 1502, thus giving a drop of 61.84%. This large difference can be attributed to the fact that at each hop, E2SIW makes a single route acquisition for the detection purpose whereas De Worm makes multiple route acquisitions (number of one-hop neighbours - 1) at each hop. The route acquisition process uses the RREQ mechanism of the AODV protocol that involves flooding of the RREQ packets in the network, resulting to the significant difference that is observed in the values of the number of transmissions in the two protocols.

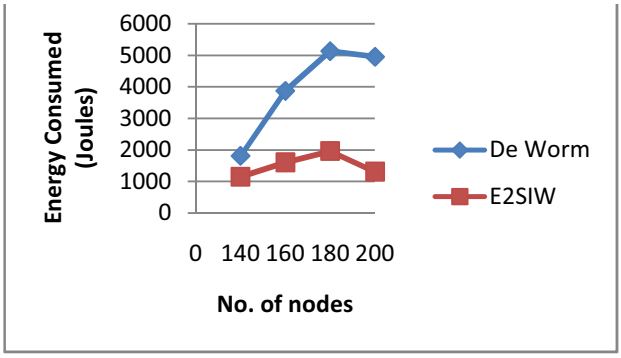

Figure 3: Variation of energy required with number of nodes in the network

iii) Number of Control Packets Originated

We vary the number of nodes and study its impact on the number of control packets transmitted, where control packets include: RREQs, RREPs, HELLO packets, HELLO packet replies, and RERRs. The results are shown in Fig. 4.

In Fig. 4, it can be observed that the average number of control packets originated in the case of De Worm is 381 while in E2SIW, it is 60. In addition, the number of control packets originated is proportional to the overhead created by the control packets in the network. Thus, the control overhead due to the control packets in E2SIW is much less than that in the case of the De Worm scheme.

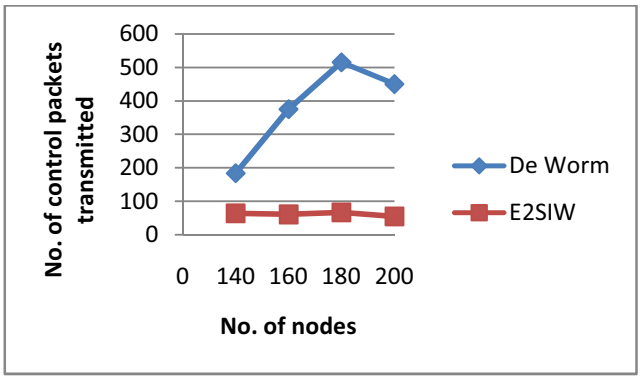

Figure 4: Variation number of control packets transmitted with the number of nodes.

iv) Wormhole Detection Percentage

The detection percentage is the number of times the wormhole is detected out of the number of times the simulations were run. In our case, we ran the simulations 10 times, each time changing the sender destination pair and accordingly creating a wormhole in between. The results are depicted in Fig. 5.

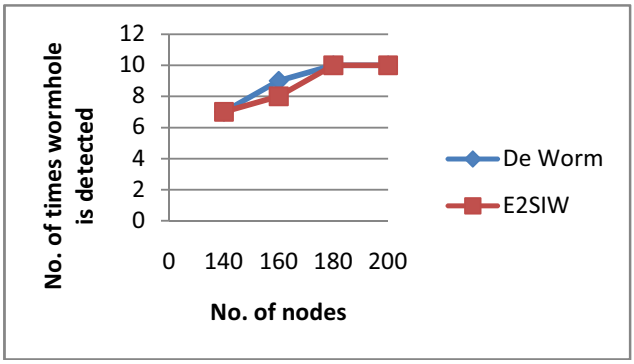

Figure 5: Percentage detection of wormhole with the number of node.

In Fig. 5, it can be observed that E2SIW has a high detection rate. The detection rate increases with the increase in node degree (i.e. in the number of nodes) as the connectivity increases with the increase in node degree. Also, the detection is 100% in the case that 180 nodes and 200 nodes are used.

V. Conclusion

We have proposed E2SIW, a routing protocol immune to wormhole attacks. E2SIW uses a simple location information and alternate route finding techniques to detect and prevent wormhole attack in ad hoc networks. Our simulation results have shown that E2SIW has a high detection rate and less energy requirements compared to the De Worm protocol. We have also contributed in reducing the overhead associated with the control packets.

Most of the work done so far in this topic assumes that the wormhole nodes are not capable of maliciously changing the data passing through them. But this may not always be the case. The design of the mitigation solutions keeping in mind that intelligent malicious nodes may exists is the need of the hour.

Acknowledgment

This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC), currently held by the 2nd author, under Ref# 119200.

References

[1] H. Deng, W. Li, D. P. Agarwal, "Routing security in wireless ad hoc networks," IEEE Communications Magazine, Vol. 40, pp. 70-75, 2002

[2] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in Mobile Ad Hoc Networks: Challenges and Solutions", IEEE Wireless Communications, Vol. 11, pp. 38-47, 2004.

[3] C. H. Vu, A. Soneye, "An Analysis Collaborative Attacks on Mobile Ad Networks" Master Thesis, Computer Science, School of Computing, Blekinge Institute of Technology, Sweden, June, 2009.

[4] M. Khabbazian, H. Mercier, V. K. Bhargava, "Severity Analysis and Countermeasures for the Wormhole Attack in Wireless Ad Hoc Networks" in IEEE Transactions on Wireless Communications, Vol. 8, No. 2, Feb. 2009.

[5] Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in Proc. of IEEE INFOCOM 2003, San Francisco, USA, Mar. 30 – April 3, 2003.

[6] L. Hu and D. Evans,"Using Directional Antennas to Prevent Wormhole Attacks", in Proc. of 11th Annual Network and Distributed Systems Security Symposium (NDSS'03), San Diego, CA, USA, Feb. 6-7, 2003.

[7] S. Khalil, S. Bagchi, N. B. Shroff, "Liteworp: Detection and isolation of the wormhole attack in static multihop wireless networks," Computer Networks, Vol. 51, No. 13, pp. 3750–3772, 2007.

[8] T. Hayajneh, P. Krishnamurthy, D. Tipper, "DeWorm: A Simple Protocol to Detect Wormhole Attacks in Wireless Ad hoc Networks", in Proc. of 3rd Intl. Conference on Network and System Security (NSS 2009), Gold Coast, Australia, Oct. 19-21, 2009.

[9] W. Wang, B. Bhargava, Y. Lu, X. Wu, "Defending against Wormhole Attacks in Mobile Ad Hoc Networks", Wiley Journal on Wireless Communications and Mobile Computing, pp. 483-503, 2006.

[10] W. Wang, J. Kong, B. Bhargava, and M. Gerla, "Visualization of Wormholes in Underwater Sensor Networks: A Distributed Approach", Int. Journal of Security and Networks, Vol. 3, No. 1, 10-23, 2008.

[11] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for the parallel simulation of large-scale wireless networks", Proc. of he 12th Workshop on Parallel and distributed Simulation. PADS'98, Banff, Alberta, Canada, May, pp. 154–161, 1998.