SEEEP: Simple and Efficient End-to-End protocol to Secure Ad hoc Networks against Wormhole Attacks

Neelima Gupta
Department of Computer Science,
University of Delhi
ngupta@cs.du.ac.in
http://people.du.ac.in/ ngupta/

Sandhya Khurana
Department of Computer Science,
University of Delhi
skhurana@cs.du.ac.in

Abstract

In this paper, we present a very simple and efficient end-to-end algorithm to handle wormhole attacks on ad hoc networks. We provide a lower bound on the minimum number of hops on a good route. Any path showing lesser hop-counts is shown to be under attack. Our algorithm requires every node to know its location. With very accurate GPS available, this assumption is not unreasonable. Since our protocol does not require speed or time, we do not need clock synchronization.

In the absence of any error in the location, there are no false alarms i.e. no good paths are discarded. We have shown that the effect of error in the location information is negligible. The storage and computation overhead is low. For a path of length l, it takes only $O(l)$ space and time which is less as compared to other end-to-end algorithms like Wang etal [8]. Their algorithm uses $O(lm)$ storage and $O(lm^2)$ computation time, where m is the number of packets examined. Since their protocol uses speed to detect wormholes, they assume the clocks to be loosely synchronized.

1 Introduction

Ad-hoc networks have been proposed to support scenarios where no wired infrastructure exists. Several types of attacks on ad hoc networks have been discussed in literature. Some of these (blackhole or grey holes attack, rushing attack, wormhole attacks) cripple the network by disrupting the route of the legitimate packets while others (flooding attack) inject too many extra packets in the system thereby consuming system resources like bandwidth, memory/computational power of nodes.

In this paper, we address the problem of detecting wormhole attacks in ad hoc networks. Since the mobile devices use a wireless medium to transmit information, the malicious nodes can eavesdrop the packets, tunnel them to another location in the network and retransmit them at the other end. Attackers may use out of band channel, high power transmission, packet relay or encapsulation technique to tunnel packets to colluding nodes. The tunnel so created forms a wormhole. The tunneling procedure generates an illusion that the two nodes more than one hop away are in the neighborhood of each other. We call the two nodes as the victim nodes. Since most of the routing protocols maintain a neighbourhood set at each node, false information about a node's neighbour can severly affect the discovered route. If the routing protocol uses the number of hop-counts to compute the shortest path, it prevents the routes longer than three hops to be discovered between the victim nodes. If the routing protocol uses the round trip delay to compute the shortest path and there exists a fast transmission path (out of band channel) between the two ends of the wormhole, it prevents normal multi-hop routes to be discovered since the tunneled packets travel much faster through the wormhole than through the normal route. Hence the route is established through the wormhole. Once a route has been established through malicious nodes it may drop or compromise packets.

Several protocols based on the use of special hardware like directional antennas, GPS and synchronized clocks have been designed to secure the ad hoc networks against wormhole attacks. The protocols based on trust your neighbor policy can not be used for the purpose as the wormhole attacks the neighbourhood relationships. Wang etal [8] have proposed a mechanism requiring only end to end trust. They require that the nodes know their positions and assume loosely synchronized clock. Each node attaches a (P, t) pair where P is the location of the node at time t. The destination checks if there is a conflict in the information sent by various nodes. It computes the moving speed of a node by examining its position at various times. If the

speed is found to be more than a certain threshold v, they declare a wormhole on the path. However, if a malicious node buffers a packet for time t, a wormhole may remain undetected if the distance between the two malicious nodes is less than the product $v * t$. To detect such attacks they perform cross packet validation. As a result the protocol incurs a lot of storage and computation overhead. If the path length is l and m packets are examined, it requires $O(lm)$ storage and $O(lm^2)$ computation time. To reduce this overhead they have proposed another Cell-based Open Tunnel Avoidance(COTA) mechanism in which they divide the network area into a number of cells and the time into equal time slots. For every node they store only one record for one (cell no, time slot) pair thereby achieving a reduction of $O(m)$ factor in storage requirement and computation time. However, there is a trade-off between the storage requirement vs number of false positives/detection capability and also between the computation time vs number of false positives/detection capability.

In this paper, we propose an end-to-end mechanism wherein we provide a lower bound on the minimum number of hops on a good route. Any path showing lesser hop-counts is shown to be under attack. Our protocol requires that every node in the network is equipped with a GPS and that every node knows its location. We assume that nodes are equipped with secret keys which provide secrecy and authenticity of message between the source and the destination. The protocol does not require clock synchronization. The storage and computation overhead is low. We do not store more than one packet at the destination. Hence the protocol requires only $O(l)$ space and time.

The idea is very simple. If d is the length of a path between the source and the destination in terms of the distance traveled by a packet and r is the communication range between any two nodes then the packet must travel at least $\lceil d/r \rceil$ hops. We show that if the length k of the path in terms of the the number of hop counts is less than $\lceil d/r \rceil$, then there is a wormhole on the path. Conversely, we show that if there is a wormhole on a path and the length of the tunnel is $\geq (k/2 + 2)r$ then $k < \lceil d/r \rceil$. In the absence of any error in the location, there are no false alarms.

When the source node sends a wormhole detection packet, each node attaches its location and d is calculated by adding the distance traveled by the packet in each hop. With the GPS accurate upto 15 feet available, we will show that the effect of error in the location information is negligible. The idea works well for closed wormholes where nodes do not lie about their position. However, in open or half-open wormhole a malicious node may show a large hop-count, big enough to escape the test or may lie about its position. Our protocol checks a node from lying too much about its position by checking if two consecutive nodes on the path are in direct range of each other. To detect a malicious node lying about the hop-count every intermediate node attaches its *id* to the packet, recomputes the MAC code using a secret shared key between itself and the destination. If a malicious node lies about the hop-count, it will have to generate and attach a THL (traversed hop list) to each packet. Though the node may be able to generate a fake list of *id*s, it will not be able to generate their MAC code as it neither has their keys nor enough computational power. All the checks are performed by the destination and intermediate nodes do not verify anything.

Our scheme can be included in the route discovery process as well as used once a data path has been established to examine the path for the presence of wormhole, from time to time. It can be used as a plug-in for any existing routing protocol like DSR or AODV.

2 Related Work

The wormhole attack in wireless networks was independently introduced by Dahill [1], Papadimitratos [5], and Hu [3]. In [4], authors have described different types of wormholes depending upon the techniques used to tunnel the packets between the colluding nodes: wormhole using encapsulation, wormhole using out-of-band channel, wormhole with high power transmission, and wormhole using packet relay.

A partial approach to defend ad hoc networks against wormhole attacks is to use a secret method for modulating bits over wireless transmissions. Another approach, known as RF watermarking, authenticates a wireless transmission without decoding the data, by instead modulating the RF waveform in a way known only to authorized nodes.

Hu etal [3] have introduced the notion of a packet leash as a general mechanism for detecting and thus defending against wormhole attacks. The packet leash approach works by specifying a maximum allowable distance that a packet can travel. The receiver detects the wormhole attack if it finds that a packet has traveled more than the allowed distance. A leash is any information that is added to a packet designed to restrict the packet's maximum allowed transmission distance. They describe two types of leashes: geographical leashes and temporal leashes. A geographical leash ensures that the recipient of the packet is within a certain distance from the sender. A temporal leash ensures that the packet has an upper bound on its lifetime, which restricts the maximum travel distance, since the packet can travel at most at the speed of light. The packet leash approach requires precise knowledge of location or tightly synchronized clocks.

Several approaches to defend against wormhole attacks, require special hardware like directional antennas, GPS and synchronized clocks. Hu and Evans [2] have presented a solution that assumes the use of bidirectional antennas being

used for communication between the mobile nodes rather than the communication being omni-directional. They work by keeping an authentic set of neighbors at every node. If a node receives a message from another node, it checks if it is in the neighborhood set of the node, it accepts it else discards it. A node validates its neighborhood set with the help of directional information shared between the nodes.

Poovendran and Lazos [6] proposed a graph theoretic model for characterizing a wormhole attack. Wang and Bhargava [7] have proposed a solution in which they do not require any special hardware in the nodes. They take the distance matrix between the network nodes as an input and reconstruct the network by calculating the virtual position for each node. Detection method focuses on the shape of the network. For example, a wormhole that pulls two nodes at extreme ends close to each other through the fake connection results in a bend in the structure of the network. The wormhole is located by detecting this bending feature.

3 Notations Used

If pairwise keys are used to encrypt the message, K_{AB} denote the symmetric shared key between the nodes A and B. $MAC_{K_{AB}}(M)$ represents the encrypted MAC code on the message M using the key K_{AB}.

Every node A can find its geographic location denoted by P_A. The maximum error in location is denoted by δ. If a packet is forwarded by a node A at recorded location P_A and it arrives at node B at recorded location P_B then the real distance d_{AB} traveled by the packet between A and B lies between $||P_A - P_B|| - 2\delta$ and $||P_A - P_B|| + 2\delta$.

4 SEEEP

The end-to-end protocol proposed here assumes that only the source and the destination trust each other. The assumption holds in most of the conditions. Once a route has been established, existence of wormholes is examined several times during the lifetime of the route. The detection packets may be sent separately or the information may be attached to the routing packets or the data packets.

Let d denote the length of a path between the source and the destination measured in terms of the distance traveled by the packet on the path. Let r be the communication range between any two nodes. The protocol is based on a very simple idea that any packet from source to destination must travel at least $\lceil d/r \rceil$ hops. For example, if $d = 9m$ and $r = 2m$ then Figure 1 shows that a packet from the node s to t must travel through the nodes $n_1, n_2 \ldots n_4$ resulting in a hop count of 5.

When the source node sends a wormhole detection packet, each node attaches its location and d is calculated

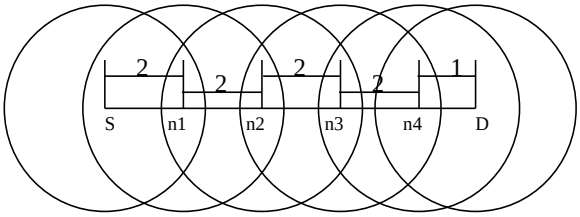

Figure 1. Example to illustrate the lower bound

by adding the distance traveled by the packet in each hop. The idea works well for closed wormholes where nodes do not lie about their position. However, in open or half-open wormhole a malicious node may show a large hop-count, or may lie about its position. In section 4.1, we will show how to check a malicious node from lying too much about its position. To detect a malicious node lying about the hop-count every intermediate node attaches its *id* to the packet, recomputes the MAC code using a secret shared key between itself and the destination. If a malicious node lies about the hop-count, it will have to generate and attach a THL (traversed hop list) to each packet. Though the node may be able to generate a fake list of *id*s, it will not be able to generate their MAC code as it neither has their keys nor enough computational power. All the checks are performed by the destination and intermediate nodes do not verify anything.

Let d be the length of a path P between the source S and the destination D in terms of the distance traveled and, r be the communication range between any two nodes. Let k be the number of hops on P. Then, we prove the following two theorems one of which provides an upper bound on the length of the wormhole tunnel. The wormhole is detected if the length of its tunnel is greater than this bound.

Theorem 4.1 *If $k < \lceil d/r \rceil$ then there is a wormhole on the path.*

Proof 4.1 *We will prove the result by proving that on a normal good path, number of hops is at least $\lceil d/r \rceil$. See Figure 1. Clearly, the number of nodes on P is minimum when the nodes are placed as far apart as possible and P lies along a straight line between S and D. Since two consecutive nodes on P cannot be placed farther than r, distance traveled is $\leq kr$ or $k \geq \lceil d/r \rceil$. Thus, if $k < \lceil d/r \rceil$ there must be a wormhole tunnel of length greater than r on P.*

However, the converse of the above theorem is not true in general. That is, there may be a wormhole on a path and $k \geq \lceil d/r \rceil$. There may be lots of closely placed nodes between S and another node u and then there is a long tunnel between u and D. For example in Figure 2, if $r = 2m, d = 10m$ so that $\lceil d/r \rceil = 5$, but $k = 7$. Let

$dist(S, u_1) = 1m$, $dist(u_1, u_2) = (1+\epsilon)m$, $dist(u_2, u) = 1m$, $dist(u, M1) = (1+\epsilon)m$, $dist(M2, v) = 1m$, and $dist(v, D) = (1+\epsilon)m$ then $dist(M1, M2) = (4-3\epsilon)m$. The reason is that there are 6 nodes covering a distance of $(6+3\epsilon)m$ with a long tunnel of length $(4-3\epsilon)m$. Thus there is a wormhole on the path but $k > \lceil (d/r) \rceil$.

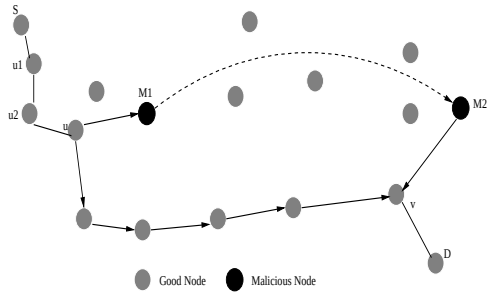

Figure 2. Path through wormhole

In the following lemma, we will bound the number of good nodes that may occur on a good path spanning some distance. The idea is if n_1, n_2, n_3 are on some path then n_3 must be outside the range of n_1. This property is satisfied in most of the routing protocols like AODV and DSR. In AODV, Suppose n_1 broadcasts a route request packet. If both n_3 and n_2 are in the range of n_1, both of them will receive the packet. If n_3 is also in the range of n_2, it will later receive the packet from n_2 but will discard it as a duplicate. Hence no path will be setup through n_1, n_2, n_3.

Lemma 4.1 *Let S_i denote the interval $(ir, (i+1)r]$ and $d \in S_i$ for some i then $k \leq 2i+1$.*

Proof 4.1 *We'll prove the claim by induction on i.*

For $i = 0, d \leq r$, clearly then D is neighbor of S and $k = 1$. Let the result holds for $i \leq t$. That is for any node D_i whose distance d_i from S along P satisfies $ir < d_i \leq (i+1)r$, the number of hops k_i from S to D_i satisfies $k_i \leq 2i+1$ for all $i \leq t$. Let D_{t+1} be a node whose distance d_{t+1} from S along P satisfies $(t+1)r < d_{t+1} \leq (t+2)r$. Consider the part Q of P between S and D_{t+1}. Let D_l be the neighbor of D_{t+1} on Q, then either $d_l \in S_i$ for some $i \leq t$ or $d_l \in S_{t+1}$ In the first case, induction applies and and hence $k_l \leq 2i+1$. Then $k_{t+1} = k_l + 1 \leq 2i+2 \leq 2t+2 \leq 2(t+1)+1$. In the second case, we cannot apply induction. In this case, let $D_{l'}$ be the neighbor of D_l on Q. Then, $d_{l'} \in S_i$ for some $i \leq t$ and hence $k_{l'} \leq 2i+1$. l' cannot be in S_{t+1} for else D_{t+1} would be in the range of $D_{l'}$ and hence they would be neighbors. Thus $k_{t+1} = k_{l'} + 2 \leq 2i+3 \leq 2t+3 = 2(t+1)+1$.

From the above lemma it follows that $k < 2d/r + 1$ or $d > (k-1)r/2$. In the following theorem, we show that the converse of Theorem 4.1 holds if the tunnel is long enough.

Theorem 4.2 *If there is a wormhole on a path and the length of the tunnel is $\geq (k/2+2)r$ then $k < \lceil d/r \rceil$.*

Proof 4.2 *Suppose there is a wormhole on a path $S = u_1, u_2, \ldots u_{k+1} = D$. Since there is a wormhole, there exists a pair of vertices u_i, u_{i+1} which form a wormhole. Also, the distance between u_i and u_{i+1} is $\geq (k/2+2)r$, by assumption. Then,*

$$d = dist(S, u_i) + dist(u_i, u_{i+1}) + dist(u_{i+1}, D)$$
$$> \frac{(i-1-1)}{2} r + (k/2+2)r + \frac{(k-i-1)}{2} r$$
$$= (2k+1)r/2 > kr$$
$$\Rightarrow k < d/r \leq \lceil d/r \rceil.$$

Theorem 4.1 shows that if $k < \lceil d/r \rceil$ then we are sure that there is a wormhole on the path. Theorems 4.1 and 4.2 can be combined to give the following algorithm: discard a path if $k < \lceil d/r \rceil$. Theorem 4.1 guarantees that no good path is discarded and Theorem 4.2 guarantees that wormhole of length equal to roughly half the length of the entire path are detected.

When the source sends a wormhole detection packet, it includes the source *id*, the destination *id*, message if any, its location, hop-count field set to 1, in the packet and encrypt it with say MAC code using K_{SD}, the shared key between the source and the destination. Each intermediate node A attaches its *id* to the THL (traversed hop list), stores its location in the packet, increments the hop-count and encrypt it with the MAC code using K_{AD}, the shared key between the node and the destination. When the destination receives the detection packet, it calculates the distance traveled by the packet using the location information and checks the hop-count announced by the path. If it is less than $\lceil d/r \rceil$, it detects a wormhole on the path and broadcasts a message informing the source to abort sending data packets on the path.

4.1 Check the attacker from lying

The above scheme requires that each node attaches its location information in the detection packet. The scheme works fine in closed wormhole where no node lies about its position. However, in half-open or open wormhole an attacker (or colluding attackers) may lie about its (their) position(s). To check an attacker from lying, destination also verifies whether two consecutive nodes are in direct communication range of each other. Consider Figure 2, to announce that the distance between $M1$ and $M2$ is small, one or both of $M1$ and $M2$ may lie about their position. In either case, at least one of them will go out of the range of communication of its good neighbor and hence the wormhole will be detected.

An attacker may also lie about its hop-count from S. It may put a large value in the hop-count of the detection packet. Let $d = 20m$ and $r = 2m$. Since $M1$ and $M2$

are colluding $M1$ may have an idea of the location of $M2$. Let d_{M1M2} denote the distance between $M1$ and $M2$. Let $d_{M1M2} = 16m$. Then $M2$ may increment the hop-count by $d_{M1M2}/r = 8$. The destination will then get the packet with the right hop count value 10, and hence the wormhole will go unnoticed. To detect such wormholes, we use the THL in the detection packet. The attacker may be able to generate a fake list of *id*s, but it will not be able to generate their MAC code. Hence by examining the THL, wormhole will be detected.

4.2 Detection of wormhole at the destination

When the detection packet reaches the destination, it performs the following operations:

1. It verifies that all MAC codes have been computed correctly.
2. It verifies that all pairs of consecutive nodes are in direct range of communication with each other.
3. Extracts the locations of all the nodes from the packet and computes d by adding the distance traveled by the packet per hop. If the hop-count in the detection packet is less than $\lceil d/r \rceil$, it broadcasts a message to inform the source to discard the route.
4. Else, it will examine the THL in the detection packet. In case there is a wormhole on the path and it has announced a fake hop-count, it will not have a valid THL. Hence the wormhole will be detected.

4.3 Effect of error in the location information

Every node is equipped with a global positioning system (GPS) so that it knows its geographic location. The effect of accuracy of location information is negligible. In a very few cases some good short paths may remain undiscovered.

Let k denote the number of hops on a path from the source S and the destination D. Let d be the traveled distance as calculated by the destination and let d' be the real distance between S and D.

Let P_i and P_{i+1} denote the recorded location of two consecutive nodes u_i and u_{i+1} on the path and let P'_i and P'_{i+1} be their real positions. Then $||P_i - P_{i+1}||$ the recorded distance traveled by the packet lies between $||P'_i - P'_{i+1}|| - 2\delta$ and $||P'_i - P'_{i+1}|| + 2\delta$, where δ is the maximum error in the location information of any node. Summing it over all the hops we get that d lies between $d' - 2k\delta$ and $d' + 2k\delta$.

If $d = d' - 2k\delta$, then we are putting a looser lower bound on the number of hops of a good path. A wormhole may go undetected if it shows a hop count greater than $\lceil d/r \rceil = \lceil (d' - 2k\delta)/r \rceil$ but less than $\lceil d'/r \rceil$ even if its tunnel is long. However, in a practical scenario, with very accurate GPS, the value of $2k\delta/r$ is a much smaller quantity and its effect is not damaging. For example, if the real distance is $1250m, r = 250m$ and $\delta = 5m$ (> 15 feet). Let $k = 10$, then the recorded distance could be $1150m$. We rightly discard the paths with hop counts less than $5 = \lceil 1150/250 \rceil$.

If $d = d' + 2k\delta$, then we are putting a tighter lower bound on the number of hops of a good path. Hence it will not affect the wormhole detection capability of the algorithm but we may have false positives. That is, we may miss some good short paths. For example, if in the above scenario, the recorded distance is 1350m then it discards all paths of length less than $6 = \lceil 1350/250 \rceil$ and hence good paths of length 5 are also discarded.

5 Security analysis

Our protocol is able to detect closed wormholes as well as open and half-open wormholes. Most of the algorithms designed to defend the ad hoc networks against various types of attacks suffer from false positives, (i.e. a good path is suspected to be under attack and is discarded) and false negatives (i.e. a path under attack escapes detection). Theorem 4.1 guarantees that in the absence of any error in the location, our algorithm does not give false alarms. In the previous section we showed that even in presence of error, wormhole detection capability of the protocol is not affected, however in a very few cases there may be some false alarms. Some wormholes of relatively short length ($< (k/2 + 2)r$) may escape detection.

6 Overhead

In this section we present the overhead due to storage, communication , and computation incurred by our protocol and compare it with other algorithms.

6.1 Storage and Communication Overhead

If there are k nodes on the path, then the size of the packet is $O(k)$. Hence the communication time per packet per hop is $O(k)$ which is same as that of end-to-end mechanism of Wang etal. Since in our protocol we do not need to perform any cross packet validation, we do not store more than one packet at the destination. No storage is used at the intermediate nodes and only $O(k)$ storage is used at the destination. This is much less than $O(km)$ space, where m is the total number of packets examined, used by end-to-end mechanism of Wang etal. COTA proposed by them saves space by storing only c_1 number of packets instead

	Storage	Computation	Comm. Overhead
End to End Mechanism Wang etal [8]	$O(km)$	$O(km+km^2)$	$O(k)$
COTA	$O(c_1k)$	$O(c_2km)$	$O(k)$
SEEEP	$O(k)$	$O(k)$	$O(k)$

Table 1. Table of Comparison

of all the m packets, where c_1 is a constant. The storage space used by COTA is then $O(c_1k)$. The constant c_1 decreases as a factor called *sensitivity* increases, by more than a linear rate. That is, c_1 and hence the amount of storage space can be made arbitrarily small by making *sensitivity* large. However, large sensitivity leads to large number of false positives. Thus there is a trade-off between the storage space and the number of false positives/detection capability (small *sensitivity* leads to missing the detection of some real wormholes).

6.2 Computation Overhead

Computing the MAC code at the intermediate nodes and verifying them at the destination does not take much time. Checking whether consecutive nodes are in direct range or not involves only $O(k)$ pairs. Hence this step takes $O(k)$ time. Similarly, computing the distance between consecutive nodes and adding them to compute the distance between the source and the destination requires only $O(k)$ computation time. Examining the THL of length $O(k)$ will take only $O(k)$ time. This is much less than the $O(km^2)$ time of end-to-end mechanism and $O(c_2km)$ time of COTA. Again the constant c_2 decreases as the *sensitivity* increases, by more than a linear rate. That is, c_2 and hence the computation time can be made arbitrarily small by making *sensitivity* large. Thus there is also a trade-off between the computation time and the number of false positives/detection capability.

Table 1 summarizes the comparison of our protocol with the end-to-end protocol and COTA.

7 Conclusion and future work

We have presented a very simple end-to-end algorithm to handle wormhole attacks on ad hoc networks. We have suggested to discard a path with hop count less than $\lceil d/r \rceil$. In the absence of any error in the location, there are no false alarms i.e. no good paths are discarded. However wormhole tunnels of length less than $(k/2+2)r$ may be missed. We have shown that the effect of error in the location information is negligible. The protocol does not require clock synchronization. The storage and computation overhead is low.

In the future work, we intend to reduce the length of the tunnel beyond which the wormhole may be detected. One approach to achieve this is to relax the bound on the hop count. For example, if we discard the path if the hop-count is less than $2d/r$ instead of $\lceil d/r \rceil$ we will be able to identify wormholes of shorter length. But this introduces a number of false positives. The real challenge would be to reduce the length of the tunnel without discarding too many good paths. Another extension to the work would be to the case when nodes have variable ranges.

References

[1] B. Dahill, B. Levine, E. Royer, and C. Shields. A secure routing protocol for ad hoc networks. In *Tech report 02-32*. Dept. of Computer Science, University of Massachusetts, Amherst, 2001.

[2] L Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2004.

[3] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Proceedings of INFOCOM*, 2003.

[4] Issa Khalil, Saurabh Bagchi, and Ness B. Shroff. Liteworp: A lightweight countermeasure for the wormhole attack in multihop wireless networks. In *Proceedings of International Conference on Dependable Systems and Networks (DSN)*, 2005.

[5] P. Papadimitratos and Z. Haas. Secure routing for mobile ad hoc networks. In *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002.

[6] Radha Poovendran and Loukas Lazos. A graph theoretic framework for preventing the wormhole attack in wireless ad hoc networks. In *ACM Journal on Wireless Networks (WINET)*, volume 13, pages 27 – 59. ACM, 2007.

[7] W. Wang and B. Bhargava. Visualization of wormholes in sensor networks. In *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, 2004.

[8] Weichao Wang, Bharat Bhargava, Yi Lu, and Xiaoxin Wu. Defending against wormhole attacks in mobile ad hoc networks. In *Wiley Journal Wireless Communications and Mobile Computing (WCMC)*, volume 6, pages 483 –503. Wiley, 2006.