FEEPVR: First End-to-End protocol to Secure Ad hoc Networks with variable ranges against Wormhole Attacks

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

Neelima Gupta
Department of Computer Science,
University of Delhi
ngupta@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 with variable ranges of communication. Most of the existing approaches focus on the prevention of wormholes between neighbors that trust each other. The known end-to-end mechanisms assume that all the nodes of the network have same communication range. To the best of our knowledge this is the first attempt to handle variable ranges of the nodes in an ad hoc network where only the source and the destination are assumed to trust each other. 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 and can be ignored most of the times. The storage and computation overhead is low. For a path of length l, it takes only $O(l)$ space and time.

1 Introduction

Ad-hoc networks [9] have been proposed to support scenarios where no wired infrastructure exists. They can be set up quickly where the existing infrastructure does not meet application requirements for reasons such as security, cost, or quality. Examples of applications for ad hoc networks range from military operations, emergency disaster relief to community networking and interaction between attendees at a meeting or students during a lecture. In Mobile Ad hoc Networks (*MANET*) each node has limited wireless transmission range, so the communication depends on the cooperation of intermediate nodes. Most routing protocols in ad hoc networks rely on implicit trust-your-neighbor relationship to route packets among participating nodes. Lack of infrastructure, central controlling authority and the properties of wireless links make Mobile Ad hoc Networks (*MANET*s) vulnerable to threats in security. Attacks range from passive eavesdropping in which the attacker may get access to secret information thereby violating the confidentiality to active impersonation, message replay, and message distortion. Attacks may be by an external source which is not a part of the network and hence does not have valid signatures or could be from a compromised node within the network. Chances of a node being compromised in a hostile environment (e.g., a battlefield) with relatively poor physical protection are non-negligible. Therefore, we should not only consider attacks from outside a network, but also take into account the attacks by compromised nodes within the network. Since the external attackers do not have valid digital signatures, erroneous routing information can be identified using cryptographic schemes. However, an erroneous message signed by a compromised node cannot be distinguished from a correct message from a non-compromised node using digital signatures.

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 neighborhood set at each node, false information about a node's neighbor can severely 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.

In [2], we presented an end-to-end mechanism to secure ad hoc networks with nodes of same communication range against wormhole attack. In this paper, we extend our previous work to allow nodes to have variable communication ranges. Most of the existing approaches focus on the prevention of wormholes between neighbors that trust each other. The only other end-to-end mechanism proposed is due to Wang et al [11]. Their algorithm also assumes that all the nodes of the network have same communication range. To the best of our knowledge this is the first attempt to handle variable ranges of the nodes in an ad hoc network where only the source and the destination are assumed to trust each other.

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, where l is the length of the path in terms of the number of hops.

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_{max} is the maximum communication range between any two nodes then the packet must travel at least $\lceil d/r_{max} \rceil$ hops. We show that if the length k of the path in terms of the the number of hop counts is less than $\lceil d/r_{max} \rceil$, 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 $\geq \frac{(2pk+3)}{2p} r_{max}$ then $k < \lceil d/r_{max} \rceil$ where $p = \frac{r_{max}}{r_{min}}$ and r_{min} is the minimum communication range between any two nodes. We assume that maximum allowed communication range r_{max} is known to all the nodes in the network. In the absence of any error in the location, there are no false alarms. With simulation, we will show that wormholes of tunnel length $< \frac{(2pk+3)}{2p} r_{max}$ are also detected in many scenarios if the tunnel length is not too short (when good nodes are placed reasonably far apart).

When the source node sends a wormhole detection packet, each node attaches its location and range to the packet; distance d is calculated at the destination 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 and can be ignored most of the times. The idea works well for closed wormholes where nodes do not lie about their position or range. 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 or range. Our protocol checks a node from lying too much about its position or range 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.

The paper is organized into 8 sections. In Section 2 and 3, we define the problem and give the state of the art for the problem respectively. Section 4 summarizes the notations used in the paper. In section 5 we discuss our algorithm. The wormhole detection capability of the algorithm is discussed in section 6. Computation and storage overhead are presented in section 7. Finally we conclude the paper in section 9 suggesting directions for future improvement.

2 Statement of Problem

The wormhole attack in wireless networks was independently introduced by Dahill [1], Papadimitratos [7], and Hu [5]. In [6], 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.

1. Wormhole using encapsulation: The source node broadcasts a route request packet, received by the malicious node $M1$, which encapsulates it and forwards it to $M2$ via good nodes . $M2$ demarshals the packet and broadcasts it further to the destination. Note that due to the packet encapsulation, the hop count does not increase during the traversal through the good nodes. See figure 1.

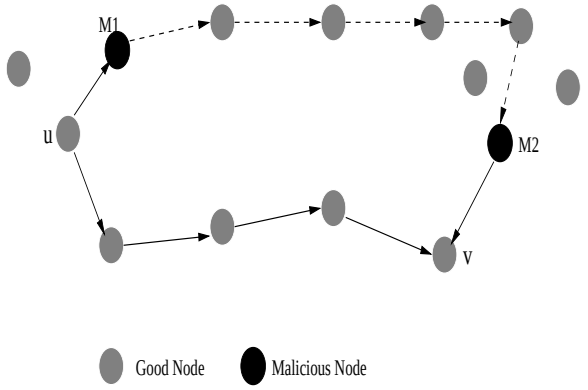

Figure 1. wormhole using encapsulation

2. Wormhole using out of band channel: The two colluding nodes communicate directly via an out-of-band high-bandwidth channel using a long-range directional wireless link or a direct wired link. See figure 2.

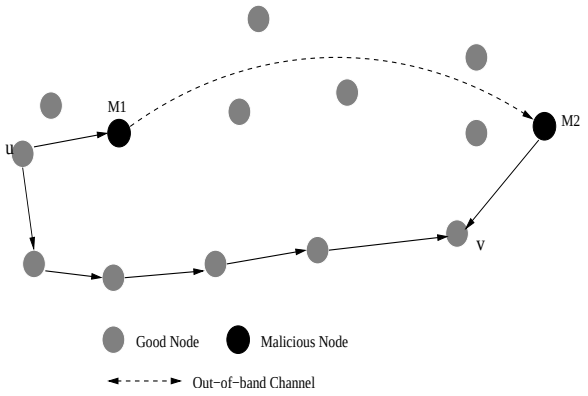

Figure 2. wormhole using out-of-band channel

3. Wormhole using high power transmission: Malacious nodes have a high power antenna and hence distant nodes receive the route request packet faster from the malicious nodes than through the normal multi-hop route increasing the chance of malicious node to get inserted in the route. See figure 3.

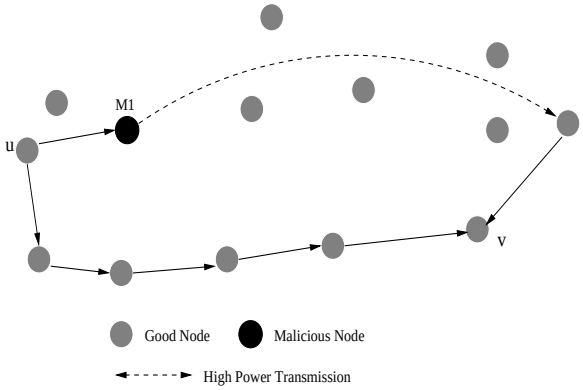

Figure 3. wormhole using high power transmission

4. Wormhole using packet relay: A malicious node relays packets between two non-neighbor nodes creating an illusion that they are neighbors.

In [11], Wang etal have classified wormholes depending upon whether one, both or none of the two colluding nodes at the end of the tunnel are visible to the good nodes (we will call them the victim nodes). See Figure 4. They describe the wormhole as closed if none of them is visible; u and v get the illusion that they are direct neighbors of each other. It is called half-open, if the malicious node near u is visible to it but the other end is not visible to v; two hops path is established between u and v. Finally, they call the wormhole to be open if both the ends are visible to u and v, i.e. three hops path is established between them through the wormhole.

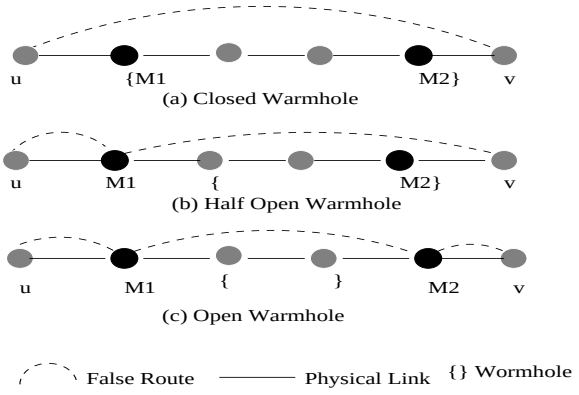

Figure 4. Types of wormhole

Now consider a node S trying to establish a route to destination D. It is possible to establish a path between

S and D through the wormhole as shown in Figure 5. Once a path through a wormhole is established malicious nodes may drop or compromise the data packets.

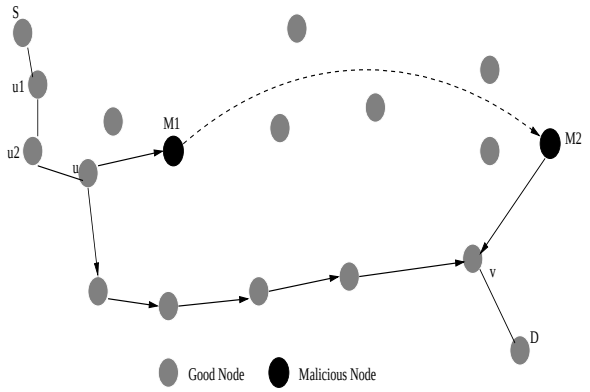

Figure 5. Path through wormhole

3 Related Work

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 [5] 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 [4] 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. The approach of Khalil et al in [6] is based on *trust your neighbor* where a *guard node* monitors the traffic going in and out of its neighbors and informs all other nodes if any of its neighbors misbehaves.

Poovendran and Lazos [8] proposed a graph theoretic model for characterizing a wormhole attack and derived the necessary and sufficient conditions for any candidate solution to prevent wormholes. In this approach, a small fraction of the nodes needs to be equipped with a GPS receiver. Wang and Bhargava [10] 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.

In [11], Wang et al 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 computes the moving speed of a node by examining its positions at various times. If the speed is found to be more than a certain threshold, they declare a wormhole on the path. In [2], we have presented an algorithm in which 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. However, both these algorithms assume fixed range for the nodes.

4 Assumptions and Notations

4.1 Network Assumptions

We assume that the authentication of keys can be performed by pairwise secret keys or digital signatures. We also assume that the network drops packets only due to wormhole. We do not assume any clock synchronization.

4.2 Node Assumptions

We assume that different nodes may have different communication ranges. However, links are assumed to be bidirectional i.e. if a node A can hear another node B then B can also hear A. Every node is equipped with a global positioning system (GPS) so that it knows its geographic location. Though the computation power of the nodes is limited, it is enough to carry out the computations required

by security mechanism such as calculation and verification of digital signatures, and calculation of the MAC code.

4.3 Model of attacker

The attackers do not have the capability to acquire the secret keys nor the computation power to compute MAC codes. The attacker may use encapsulation, out-of-band channel, high power transmission or packet relay to tunnel the packets through long distances without interfering with the signals sent by the good nodes. The attackers have a total control over the wormholes. Once a path is established through a wormhole, the attacker may forward the packets or choose to drop them.

4.4 Notations

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 $||P_A - P_B|| + 2\delta$. Range of a node A is denoted by r_A.

5 FEEPVR

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.

We suggest modification to our previous work so that the nodes are allowed to have different communication ranges. Most of the arguments there go through for variable ranges with suggested modifications. Some of the arguments (Example in Figure 6 and Theorem 5.1) hold trivially when r is replaced with r_{max}. Lemma 5.1 clearly holds if r_{min} is used instead of r. We modify Theorem 5.2 to take these changes into account. We repeat these results here for the sake of completeness.

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_{max} be the maximum 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 $\lceil d/r_{max} \rceil$ hops. For example, if $d = 9m$ and $r_{max} = 2m$ then figure 6 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. If the range of any of these nodes was lesser than r_{max} the number of hop counts would only be more.

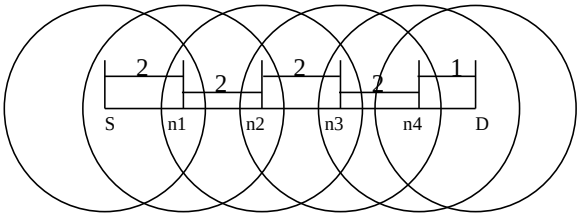

Figure 6. Example to illustrate the lower bound

When the source node sends a wormhole detection packet, each node attaches its location and range; d is calculated at the destination 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 or range. In section 5.1, we will show how to check a malicious node from lying too much about its position or range. 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_{max} and r_{min} be the maximum and the minimum 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 5.1 *If $k < \lceil d/r_{max} \rceil$ then there is a wormhole on the path.*

Proof 5.1 *We will prove the result by proving that on a normal good path, number of hops is at least $\lceil d/r_{max} \rceil$. See figure 6. 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_{max}, distance traveled is $\leq kr_{max}$ or $k \geq \lceil d/r_{max} \rceil$. Thus, if $k < \lceil d/r_{max} \rceil$ there must be a wormhole tunnel of length greater than r_{max} 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 $k \geq \lceil d/r_{max} \rceil$. 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 5, if $r_{max} = 2m, d = 10m$ so that $\lceil d/r_{max} \rceil = 5$, 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 $k > \lceil d/r_{max} \rceil$.

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 5.1 *Let S_i denote the interval $(ir_{min}, (i+1)r_{min}]$ and $d \in S_i$ for some i then $k \leq 2i + 1$.*

Proof 5.1 *We'll prove the claim by induction on i.*

For $i = 0, d \leq r_{min}$, 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 $ir_{min} < d_i \leq (i+1)r_{min}$, 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 $(t+1)r_{min} < d_{t+1} \leq (t+2)r_{min}$. 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 $k < 2d/r_{min} + 1$ or $d > (k-1)r_{min}/2$. In the following theorem, we show that the converse of Theorem 5.1 holds if the tunnel is long enough.

Theorem 5.2 *If there is a wormhole on a path and the length of the tunnel is $\geq \frac{2kp+3}{2p} r_{max}$ then $k < \lceil d/r_{max} \rceil$, where $p = r_{max}/r_{min}$.*

Proof 5.2 *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 $\geq \frac{2kp+3}{2p} r_{max}$, by assumption. Then,*

$$
\begin{aligned}
& d = dist(S, u_i) + dist(u_i, u_{i+1}) + dist(u_{i+1}, D) \\
& > \tfrac{(i-1-1)}{2} r_{min} + \tfrac{2kp+3}{2p} r_{max} + \tfrac{(k-i-1)}{2} r_{min} \\
& = \tfrac{(i-1-1)}{2} \tfrac{r_{max}}{p} + \tfrac{2kp+3}{2p} r_{max} + \tfrac{(k-i-1)}{2} \tfrac{r_{max}}{p} \\
& = \tfrac{2p+1}{2p} k r_{max} > k r_{max}. \\
& \Rightarrow k < d/r_{max} \leq \lceil d/r_{max} \rceil.
\end{aligned}
$$

Theorem 5.1 shows that if $k < \lceil d/r_{max} \rceil$ then we are sure that there is a wormhole on the path. Theorems 5.1 and 5.2 can be combined to give the following algorithm: discard a path if $k < \lceil d/r_{max} \rceil$. Theorem 5.1 guarantees that no good path is discarded and Theorem 5.2 guarantees that long wormholes are always detected. With simulation, we will show that when nodes are placed reasonably far apart the length of the tunnel is not too short and we are able to detect wormholes of length $< \frac{(2pk+3)}{2p} r_{max}$ also.

When the source sends a wormhole detection packet, it includes the source *id*, the destination *id*, the path *id*, message if any, its location, its range, 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 and range 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. The delivery of wormhole detection packet is shown in figure 7. 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 $\lceil d/r_{max} \rceil$, it detects a wormhole on the path and broadcasts a message informing the source to abort sending data packets on the path.

In section 5.3, we will show that with the GPS accurate upto 15 feet, the effect of error in the location information does not affect the detection capability of our protocol. However, the error may sometimes lead to false positives.

5.1 Check the attacker from lying

The above scheme requires that each node attaches information about its location and range 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

$S : HC = 1$
$h_s = MAC_{K_{SD}}(S, D, M, id, P_S, r_S, HC)$
$S- > A : (S, D, M, id, P_S, r_S, HC, h_s)$
$A : HC++$
$A : h_A = MAC_{K_{AD}}(ReceivedPacket, A, P_A, r_A, HC)$
$A- > B : (ReceivedPacket, A, P_A, r_A, HC, h_A)$
$B : HC++$
$B : h_B = MAC_{K_{BD}}(ReceivedPacket, B, P_B, r_B, HC)$
$B- > D : (ReceivedPacket, B, P_B, r_B, HC, h_B)$

Figure 7. Delivery of wormhole detection packets

(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 4 (c), to announce that the distance between $M1$ and $M2$ is small, one or both of $M1$ and $M2$ may lie about its(their) position(s). 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. This holds true even if $M1$ or $M2$ or both simultaneously lie about its (their) location(s) and range(s).

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.

5.2 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.

   (a) If the hop-count in the detection packet is less than $\lceil d/r_{max} \rceil$, it broadcasts a message to inform the source to discard the route. (Once it has been verified in step 2 that nodes are not lying about their locations and ranges, if r_{max} is not known, it may be computed from the packets received.)

   (b) 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.

5.3 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 $\lceil d/r_{max} \rceil = \lceil (d' - 2k\delta)/r \rceil$ but less than $\lceil d'/r_{max} \rceil$ even if its tunnel is long. However, in a practical scenario, with very accurate GPS, the value of $2k\delta/r_{max}$ is a much smaller quantity and its effect is not damaging. For example, if the real distance is $1250m, r_{max} = 250m$ and $\delta = 1m$. Let $k = 10$, then the recorded distance could be $1230m$. We rightly discard the paths with hop counts less than $5 = \lceil 1230/250 \rceil$.

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 1270m then it discards all paths of length less than $6 = \lceil 1270/250 \rceil$ and hence good paths of length 5 are also discarded.

6 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 5.1 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 ($< \frac{2kp+3}{2p} r_{max}$) may escape detection.

7 Overhead

In this section we present the overhead due to storage, communication , and computation incurred by our protocol.

7.1 Storage, Communication and Computation 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 $O(k)$. No storage is used at the intermediate nodes and only $O(k)$ storage is used at the destination.

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.

8 Simulation Results

We tested the performance of our wormhole detection approach through simulation. The simulations were carried out in NS2.

8.1 Simulation setup

We implemented the proposed protocol using Network Simulator NS2 [3]. AODV is chosen as the routing protocol and is updated to combine with detection mechanism. Source initiates the routing request with attached location information and its range. When the intermediate node forwards the routing request it also attaches its location along with its range to the request packet. When the destination receives the request it computes the distance traveled by the RREQ packet using location information attached by each node on the path. Finally, it calculates $\lceil d/r_{max} \rceil$ and compares it with the received hop count in RREQ packet. If $\lceil d/r_{max} \rceil$ is less than or equal to the received hop count, it sends RREP otherwise it discards RREQ packets.

Simulation Duration	1000 seconds
Simulation area	$1500m * 1500m$
Number of mobile nodes	50
Transmission Range	$150m - 275m$
Movement model	Random waypoint
Traffic type	CBR(UDP)
Data payload	512 bytes
Pause Time	500 seconds

Table 1. Simulation Parameters

8.2 Results

Network topology was generated in two ways. One was user generated and another was generated randomly in NS2.

In NS2 generated topology, we considered a connection scenario between nodes say s and t. Six paths were found between s and t. Wormholes of varying tunnel length were created on one of these paths. It was found that whenever the tunnel length was greater than $\frac{2kp+3}{2p} r_{max}$, the wormhole was always detected and the shortest wormhhole free path was established. In many cases, we were also able to detect and isolate wormholes of length shorter than this. For example for $p = 1.8, k = 3, ((2pk + 3)/2p) r_{max} = 1050$, we were able to detect wormholes of length $900m$. We studied more connection scenarios and similar results were obtained.

We also created a topology in which we studied wormhole on more than one paths for a connection scenario. Our algorithm was able to detect and isolate both of them and established a wormhole free path.

9 Conclusion and future work

We have presented a very simple end-to-end algorithm to handle wormhole attacks on ad hoc networks with variable communication ranges. We have suggested to discard a path with hop count less than $\lceil d/r_{max} \rceil$. 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 $\frac{2kp+3}{2p} r_{max}$ may be missed. We have shown that the effect of error in the location information is negligible and can be ignored most of the times. 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 $2d/r_{max}$ instead of $\lceil d/r_{max} \rceil$ 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.

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] N. Gupta and S. Khurana. Seeep: Simple and efficient end-to-end protocol to secure ad hoc networks against wormhole attacks.

[3] http://www.isi.edu/nsnam/ns/. The network simulator.

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

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

[6] 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.

[7] 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.

[8] 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.

[9] Ram Ramanathan and Jason Redi. A brief overview of ad-hoc networks: Challenges and directions. In *IEEE Communications Magazine*, pages 20–22. IEEE, May 2002.

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

[11] 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.