

Using Directional Antennas to Prevent Wormhole Attacks

Lingxuan Hu

David Evans

Department of Computer Science
University of Virginia
Charlottesville, VA
[lingxuan, evans]@cs.virginia.edu

Abstract

Wormhole attacks enable an attacker with limited resources and no cryptographic material to wreak havoc on wireless networks. To date, no general defenses against wormhole attacks have been proposed. This paper presents an analysis of wormhole attacks and proposes a countermeasure using directional antennas. We present a cooperative protocol whereby nodes share directional information to prevent wormhole endpoints from masquerading as false neighbors. Our defense greatly diminishes the threat of wormhole attacks and requires no location information or clock synchronization.

1. Introduction

Wireless ad hoc networks have properties that increase their vulnerability to attacks. Wireless links are inherently vulnerable to eavesdropping and message injection, as well as jamming attacks. Constraints in memory, computing power, and battery power in mobile devices can impose trade-offs between security and resource consumption.

Routing in ad hoc wireless networks is an especially hard task to accomplish securely, robustly and efficiently. Many proposed routing protocols are focused on energy, and provide no protection against an adversary. Some secure routing protocols also have been proposed. However, due to the unpredictability of ad hoc networks, it is hard to detect behavior anomalies in route discovery. In particular, proposed routing protocols cannot prevent wormhole attacks. In a wormhole attack, an attacker introduces two transceivers into a wireless network and connects them with a high quality, low-latency link. Routing messages received by one wormhole endpoint are retransmitted at the other endpoint. Attackers can exploit wormholes to build bogus route information, selectively drop packets, and create routing loops to waste the energy of network.

Wireless ad hoc networks typically assume omnidirectional antennas. In this paper, we consider devices with directional antennas. Directional antennas have been shown to improve efficiency and capacity of wireless networks. Several MAC protocols [4, 14, 12, 20] and routing protocols [5, 2, 17] have been proposed that take advantage of directional antennas.

Next we provide background on secure routing protocols and previous work on preventing wormhole attacks. Section 3 considers wormhole attacks and analyzes their effectiveness. Section 4 introduces directional antennas and describes the antenna model we use. Section 5 describes our protocols for verifying neighbor relationships. Section 6 considers the impact of our protocol on network connectivity and routing performance, and Section 7 analyzes the impact of directional errors. Section 8 concludes.

2. Background

Several secure routing protocols have been proposed for wireless ad hoc networks. Papadimitratos and Haas [23] present the SRP protocol that secures against non-colluding adversaries by disabling route caching and providing end-to-end authentication using an HMAC primitive. SEAD [7] uses one-way hash chains to provide authentication for DSDV [21]. Ariadne [8] uses an authenticated broadcast technique [22] to achieve similar security goals on DSR [11]. Marti et al. [16] examine techniques to minimize the effect of misbehaving nodes through node snooping and reporting, but it is vulnerable to blackmail attacks. ARRIVE [13] proposes probabilistic multi-path routing instead of single path algorithm to enhance the robustness of routing. These secure routing protocols are still vulnerable to wormhole attacks which can be conducted without having access to any cryptographic keys.

Wormhole attacks depend on a node misrepresenting its location. Hence, location based routing protocols have the potential to prevent wormhole attacks [15].

Localization may be done using globally accessible beacons that broadcast known locations (that may be pre-configured or determined using GPS [29]). Recently there has been some research to build localization system using localized protocols [19, 1, 10, 18]. The location service itself may become the attack target. Localization systems generally require some seed nodes that know their own positions, which may not be possible in all network environments. Sastry, Shankar, and Wagner propose a protocol that verifies a node is located within a particular region [28]. If an attacker acquires a wormhole endpoint within the region, the attacker could make other nodes also appear to be within the region. An adversary who acquires a wormhole endpoint within the region has already violated their security requirements.

A previous approach for detecting wormhole attacks is to use packet leashes [9]. A temporal packet leash places a bound on the lifetime of a packet that restricts its travel distance. The sender includes the transmission time and location information in the message, and the receiver checks that the packet could have traveled the distance between the sender and itself within the time between reception and transmission. Since radio transmissions travel at the speed of light, temporal packet leashes require tightly synchronized clocks and precise location knowledge.

Our approach to preventing wormhole attacks is for nodes to maintain accurate information about their neighbors (nodes within one hop communication distance). This is simpler than using location since each node need only maintain a set of its neighboring nodes. A message from a non-neighboring node is ignored by the recipient. Note that any protocol used to maintain accurate neighbor sets may itself be vulnerable to wormhole attacks, so our goal is to design a neighborhood discovery protocol that is not vulnerable to wormhole attacks. The security of our protocol will rely on using directional antennas to obtain relative direction information, and cooperation among nodes to verify possible neighbors.

3. Wormhole Attacks

In a wormhole attack, an attacker forwards packets through a high quality out-of-band link and replays those packets at another location in the network [9, 15]. Figure 1 shows a basic wormhole attack. The attacker replays packets received by X at node Y , and vice versa. If it would normally take several hops for a packet to traverse from a location near X to a location near Y , packets transmitted near X traveling through the wormhole will arrive at Y before packets traveling through multiple hops in the network. The attacker can make A and B believe they are neighbors by forwarding routing messages, and then selectively drop data messages to disrupt communications between A and B .

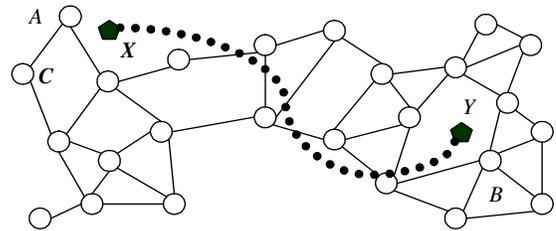


Figure 1. Wormhole attack. The adversary controls nodes X and Y and connects them through a low-latency link.

For most routing protocols, the attack has impact on nodes beyond the wormhole endpoints' neighborhoods also. Node A will advertise a one-hop path to B so that C will direct packets towards B through A . For example, in on-demand routing protocols (DSR [11] and AODV [24]) or secure on-demand routing protocols (SEAD [7], Ariadne [8], SRP [23]), the wormhole attack can be mounted by tunneling ROUTE REQUEST messages directly to nodes near the destination node. Since the ROUTE REQUEST message is tunneled through high quality channel, it arrives earlier than other requests. According to the protocol, other ROUTE REQUEST messages received for the same route discovery will be discarded. This attack thus prevents any other routes from being discovered, and the wormhole will have full control of the route. The attacker can discard all messages to create a denial-of-service attack, or more subtly, selectively discard certain messages to alter the function of the network. An attacker with a suitable wormhole can easily create a sinkhole that attracts (but does not forward) packets to many destinations. An intelligent attacker may be able to selectively forward messages to enable other attacks.

To show how much damage a single wormhole can cause to routing, we simulated randomly distributing nodes in a rectangular region and used the shortest path algorithm to find the best route between any node pairs. If a wormhole is formed, some far away nodes will appear to be neighbors and some node pairs will be able to find a "shorter" path through the wormhole. Hence the route between them is disrupted by the wormhole. In simulation experiments, a single wormhole with two randomly placed endpoints disrupts over 5% of all network routes.

A more intelligent attacker may be able to place wormhole endpoints at particular locations. Strategically placed wormhole endpoints can disrupt nearly all communications to or from a certain node and all other nodes in the network. In sensor network applications, where most communications are directed from sensor nodes to a common base station, wormhole attacks can be particularly devastating. If the base station is at the corner of the network, a wormhole with one endpoint near

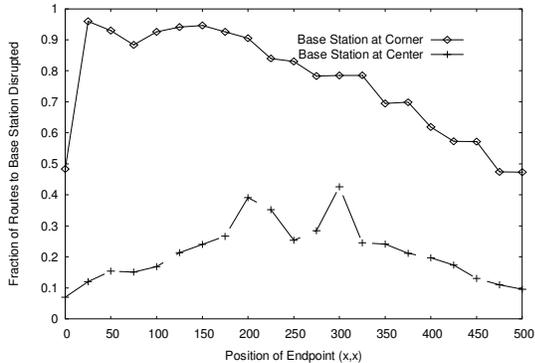


Figure 2. Impact of Wormhole Attack. A strategically placed node can disrupt a substantial fraction of communications. The position of the second endpoint moves diagonally across the network (position 250 means the second endpoint is at the center of the network; 0 means it is in the bottom left corner).

the base station and the other endpoint one hop away will be able to attract nearly all traffic from sensor nodes to the base station. If the base station is at the center of the network, a single wormhole will be able to attract traffic from a quadrant of the network. Figure 2 shows the effectiveness of a wormhole in disrupting communications from sensor nodes to a base station. One endpoint of the wormhole is within one hop of the base station; the position of the second endpoint varies along the x axis. When the base station is in a corner of the network, a wormhole with the second endpoint near the base station can effectively disrupt all network communications. If the second endpoint is placed in the opposite corner, approximately half of the nodes in the network will send messages for the base station to the wormhole.

4. Directional Antennas

Directional antenna systems are increasingly being recognized as a powerful way for increasing the capacity and connectivity of ad hoc networks [25, 26]. Transmitting in particular directions results in a higher degree of spatial reuse of the shared medium. Further, directional transmission uses energy more efficiently. The transmission range of directional antennas is usually larger than that of omnidirectional antennas, which can reduce hops for routing and make originally unconnected devices connected.

Antenna systems may be broadly classified into switched and steered antennas systems. Steered antennas provide fine directional control, but are too expensive for most sensor network applications. Switched antennas provide less precise directional control, but can be inexpensive and moderately small, particularly at COTS

frequencies such as 2.4GHz or 5 GHz [27]. In switched antenna systems, one of several fixed directional antennas can be selected using a switch. The transceiver can then choose between one or more beams for transmitting or receiving [25].

Most work on directional antennas has focused on media access protocols [14, 12, 4, 20]. Directional antennas can increase spatial reuse and reduce packet collisions and negative effects such as deafness [4]. Routing protocols using directional antennas can outperform omnidirectional routing protocols [2, 5, 17]. In this paper, we show that directional antennas also have security benefits. In particular, directional information can be used to mitigate wormhole attacks.

We adopt the antenna model in [5]. The model assumes an antenna with N zones. Each zone has a conical radiation pattern, spanning an angle of $2\pi/N$ radians. The zones are fixed with non-overlapping beam directions, so that the N zones may collectively cover the entire plane as shown in Figure 3. When a node is idle, it listens to the carrier in omni mode. When it receives a message, it determines the zone on which the received signal power is maximal. It then uses that zone to communicate with the sender.

The zones are numbered 1 to N oriented clockwise starting with zone 1 facing east. This orientation is established with respect to the earth's meridian regardless of a node's physical orientation. This is achieved in modern antennas with the aid of a magnetic needle that remains collinear to the earth's magnetic field. It ensures that a particular zone always faces the same direction.

When sending messages, a node can work in omni or directional mode. In omni mode signals are received with a gain G^o , while in directional mode with a gain of G^d . Since a node in directional mode can transmit over a longer distance, $G^d > G^o$. The omnidirectional and directional gains can be estimated from [25]. For example, when the number of zones is 6, and the omni transmission range is 250m, then the directional transmission range is 450m [5]. For our simulations, we use the same ratio between omni and directional transmission distances, but scale the ranges to 40m and 72m.

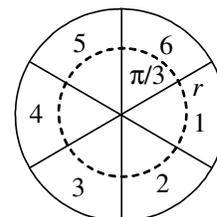


Figure 3. Directional Antenna with 6 zones. Each zone is a wedge with radius r spanning $\pi/3$ radians. Zone 1 always faces east. The dashed circle shows the omnidirectional communication radius.

5. Protocols

Our approach to detecting wormhole attacks depends on nodes maintaining accurate sets of their neighbors. An attacker cannot execute a wormhole attack if the wormhole transmitter is recognized as a false neighbor and its messages are ignored. One important property of directional antennas is a node can get approximate direction information based on received signals. Next we document our assumptions about the network. Then, we describe three increasingly effective protocols for preventing wormhole attacks. As directional information is added, attacks become increasingly difficult to execute successfully. The first protocol, *directional neighbor discovery*, does not rely on any cooperation between nodes, and cannot prevent many wormhole attacks. By sharing information among neighboring nodes, the *verified neighbor discovery* protocol can prevent wormhole attacks where the attacker controls only two endpoints and the victim nodes are at least two hops distant. Finally, the *strict neighbor discovery* protocol prevents wormhole attacks even when the victim nodes are nearby.

5.1 Assumptions

We assume all non-wormhole communication channels are bidirectional: if A can hear B , then B can hear A . This is not always the case in wireless networks, especially if battery power and physical characteristics of antennas vary. With our protocol, unidirectional links cannot be established.

We assume a mechanism is available to establish secure links between all pairs of nodes and that all critical messages are encrypted. Several efficient mechanisms have been proposed for establishing secure link keys in ad hoc networks [6, 3, 22].

Since using directional antennas can increase the transmission range significantly, we only have a modest requirement on network density. For example, when the density in omni mode is 10, then the density in directional mode will be 32.4, which is enough for our protocol to work efficiently. If the network is sparse, our protocol can still prevent wormhole attacks, but there is an increased likelihood nodes will be disconnected from the network (as discussed in Section **Error! Reference source not found.**).

We use the following notation:

$A, B, C\dots$	Legitimate nodes
X, Y	Wormhole endpoints
R	Nonce
$E_{KAB}(M)$	Message encrypted by key shared between nodes A and B
$zone$	The directional element,

\hat{zone}	which ranges from 1–6 as shown in Figure 3
$zone(A, B)$	The opposite directional element. For example, if $zone=1$ then $\hat{zone}=4$.
$neighbors(A, zone)$	Zone in which node A hears node B
	Nodes within one (directional distance) hop in direction $zone$ of node A .

5.2 Directional neighbor discovery

The directional neighbor discovery protocol does not prevent many wormhole attacks, but it forms the basis for our other protocols.

Immediately after deployment, nodes will have no known neighbors. Each node will randomly choose a time and periodically use neighbor discovery protocol to update its neighbor set. We call the node that initiates the protocol the *announcer*.

From Figure 3, one obvious observation is if node A is in node B 's $zone$ direction, then node B is in node A 's opposite direction \hat{zone} (for example, if $zone=1$, $\hat{zone}=4$). We summarize this as:

$$A \in neighbors(B, zone) \Rightarrow B \in neighbors(A, \hat{zone})$$

This relies on all nodes having the same antenna orientation due to their common magnetic orientation. Because of measurement imprecision, it is possible that the actual zone will be off by one in either direction. For simplicity of this presentation, we assume this observation holds for now. In Section 7, we consider the impact of directional inaccuracies.

The simple directional neighbor discovery protocol works in three steps:

1. $A \rightarrow \text{Region}$ HELLO $| ID_A$
The announcer A broadcasts a HELLO message that includes its identity. This is done by transmitting the message in every direction, sequentially sweeping through each antenna in the antenna array.
2. $N \rightarrow A$ $ID_N | E_{K_{NA}}(ID_A | R | zone(N, A))$
All nodes that hear the HELLO message send their node ID and an encrypted message to the announcer. The message contents are encrypted with a key shared between the announcer and the sender, which the sender can determine based on knowing its own node ID and that of the announcer. The encrypted message contains the announcer's ID, a random challenge nonce, and the zone in which the message was received.
3. $A \rightarrow N$ R

The announcer decrypts the message and verifies that it contains its node ID. It further verifies that it heard the message in the opposite zone from the zone reported by the neighbor. That is, $zone(A, N) = \hat{zone}(N, A)$. If it is correct, it adds the sending neighbor to its neighbor set for zone (A, N) . If the message was not received in the appropriate zone, it is ignored. Otherwise, the announcer transmits the decrypted challenge nonce to the sending neighbor. Upon receiving the correct nonce, the neighbor inserts the announcer into its neighbor set.

A node only modifies its neighbor set through the neighbor discovery protocol, either as a result of responses to its own announcements or through announcements it receives from its neighbors. When a message outside the neighbor discovery protocol is received the receiving node will check that its sender is in its neighbor set. The node will ignore messages from nodes that are not members of its neighbor set. Hence, if the neighbor sets are maintained correctly, a wormhole attack cannot be conducted because nodes will not accept messages that arrive from nodes that are not in their neighbor set. (This assumes all messages are authenticated using link keys. If some messages are not authenticated, an attacker can impersonate a legitimate neighbor and send those messages. Presumably, any sensitive control or routing messages would be authenticated.)

However, the neighbor discovery protocol itself is vulnerable to wormhole attacks. An attacker with a wormhole can establish a false distant neighbor by sending challenges and responses through the wormhole. An adversary with two transceivers, one near the announcer and another in a distant area of the network, can tunnel the announcer's HELLO message to the distant area through out-of-band channel. The wormhole node rebroadcasts the message, and receives challenges from neighboring nodes. It tunnels those challenges through the wormhole, and transmits them to the announcer. To the announcer,

the challenges appear to be perfectly legitimate, so the nodes are added and the decrypted nonces are transmitted. The adversary tunnels the responses through the wormhole, and transmits them to the senders. The distant sending nodes will appear as neighbors to the announcer, and the announcer will be added to each sending node's neighbor set.

Figure 4 shows an example scenario. An attacker has established a wormhole between nodes X and Y . With omnidirectional antennas, the attacker can systematically forward all neighbor discovery messages through the wormhole to convince all nodes in regions I and II they are mutual neighbors. With directional antennas, the attack's effectiveness is reduced. Only node pairs that are in opposite directions relative to the wormhole in each region will accept each other as neighbors. For example, if the attacker forwards messages between A and B through the wormhole, the directional protocol would prevent A and B from accepting each other as neighbors since A hears B (retransmitted by X) in its zone 4, and B hears A in its zone 4 instead of the expected opposite zone 1. However, the attacker can mislead A and C into believing they are neighbors since they are on opposite sides of their respective wormhole endpoints. On average, one sixth of the links between the two regions will be accepted through the wormhole. This does not, however, greatly diminish the overall effectiveness of the wormhole attack. The nodes in opposite regions will advertise the paths they find through the wormhole to their neighbors, and indirectly draw traffic towards the wormhole. As illustrated in Figure 2, an adversary who can establish a single wormhole link may be able to disrupt nearly all network traffic.

5.3 Verified neighbor discovery protocol

Although the simple directional protocol does not sufficiently mitigate the effectiveness of wormhole attacks, it suggests that if nodes cooperate with their neighbors they can prevent wormholes since the attacker will only be able to convince nodes in particular regions that they are neighbors.

Assume the adversary has one transceiver at each end of the wormhole. As described in the previous section, it can only trick nodes that are in opposite directions from the wormhole endpoints into accepting each other as neighbors. Hence, nodes in other locations can establish the announcer's legitimacy. We call such nodes *verifiers*.

Consider node C in Figure 4. Since C receives A 's transmissions through Y in its zone 1 antenna, all of its neighbors in zone 1 should also be neighbors of A . If any of those nodes are in different directions from Y (such as node B in Figure 4), then the wormhole will not be able to convince them they are neighbors of A . Note however, that C could be on the other end of the wormhole, as

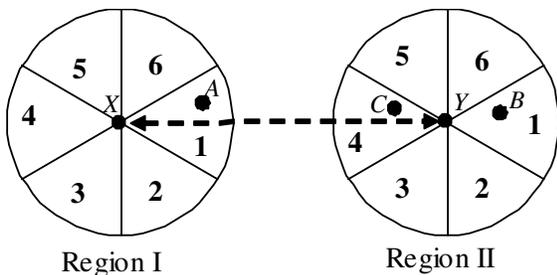


Figure 4. Directional Attack. The adversary establishes a wormhole between X and Y , and can trick A and C into accepting each other as neighbors by forwarding messages since they are in opposite zones relative to the respective wormhole endpoints.

shown in Figure 5. Here, B will hear A and C from the west through the wormhole ($\text{zone}(B, A) = \text{zone}(B, C) = 4$), and C will hear A directly from the east ($\text{zone}(C, A) = \wedge\text{zone}(A, C) = 1$) and C will hear B from the east through the wormhole ($\text{zone}(C, B) = \wedge\text{zone}(B, C) = 1$). Hence, we need a stricter requirement on verifiers to prevent verifiers from acting through the wormhole.

A valid verifier V for the link $A \leftrightarrow B$ must satisfy these properties:

1. $\text{zone}(B, A) \neq \text{zone}(B, V)$. Node B hears V in a different zone from node A , hence it knows A and V are in different locations, and both cannot be coming through a single wormhole endpoint.
2. $\text{zone}(B, A) \neq \text{zone}(V, A)$. Node B and V hear node A from different directions. A wormhole can deceive nodes in only one direction. So if both B and V are directionally consistent with A in different directions ($\text{zone}(B, A) = \wedge\text{zone}(A, B)$ and $\text{zone}(V, A) = \wedge\text{zone}(A, V)$), then they know A is not being retransmitted through a wormhole.

In Figure 5, C cannot act as a verifier since $\text{zone}(B, A) = \text{zone}(C, A)$, failing the second property. Node D can act as a verifier, since $\text{zone}(B, A) = 4 \neq \text{zone}(B, D) = 5$, and $\text{zone}(D, A) = 3 \neq \text{zone}(B, A) = 4$. Note, however, that the wormhole cannot convince D and A to accept each other as neighbors since $\text{zone}(D, A) = 3 \neq \wedge\text{zone}(A, D) = 4$. Hence, B will not be able to verify A as a neighbor through D .

We modify the original protocol to use verifier nodes to establish legitimate neighbor relationships. The first three steps are the same as in the simple neighbor discovery protocol:

1. $A \rightarrow \text{Region} \quad \text{HELLO} \mid ID_A$
2. $N \rightarrow A \quad ID_N \mid E_{K_{NA}}(ID_A \mid R \mid \text{zone}(N, A))$
3. $A \rightarrow N \quad R$

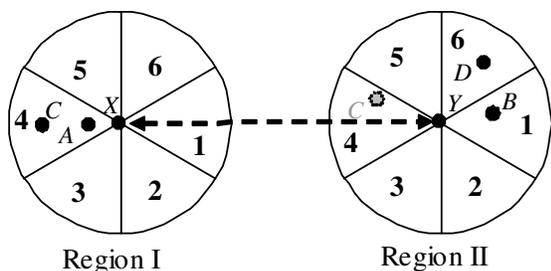


Figure 5. Verifiers. Node C cannot act as a verifier for the link $A \leftrightarrow B$ since the wormhole attacker could make a node on the other end of the wormhole appear. Node D could act as a verifier, since it satisfies the verifier properties.

These steps authenticate the nodes and their apparent relative positions, but do not establish that they are communicating without going through a wormhole. Next, the protocol uses a verifier node to confirm the link is not being created through a wormhole:

4. $N \rightarrow \text{Region} \quad \text{INQUIRY} \mid ID_N \mid ID_A \mid \text{zone}(N, A)$
All neighbor nodes that hear the HELLO message broadcast an inquiry in directions except for the received direction and opposite direction. So, if N received the announcement in zone 1, it will send inquiries to find verifiers to zones 2, 3, 5 and 6. The message includes $\text{zone}(N, A)$, so prospective verifiers can determine if they satisfy the verification properties by having heard A in a different zone.
5. $V \rightarrow N \quad ID_V \mid E_{K_{NV}}(ID_A \mid \text{zone}(V, N))$
Nodes that receive the inquiry and satisfy the verification properties respond with an encrypted message. This message confirms that the verifier heard the announcement in a different zone from N and has completed steps 1-3 for the protocol to authenticate A and its relative position.

To continue the protocol, N must receive at least one verifier response. If it does, it accepts A as a neighbor, and sends a message to A :

6. $N \rightarrow A \quad ID_N \mid E_{K_{AN}}(ID_A \mid \text{ACCEPT})$
After receiving the acceptance messages, the announcer adds N to its neighbor set.

The verified neighbor discovery protocol depends on both neighbor and verifier nodes receiving correct challenge responses from the announcer before either node will accept the announcer as a neighbor. Likewise, the announcer will not accept any neighbors until they have been mutually confirmed. The protocol is secure against wormhole attacks that involve two distant endpoints, since a wormhole can only deceive nodes to accept a particular neighbor if they are in the same relative direction from the wormhole, while the verified neighbor discovery protocol requires that a node receives confirmation from a verifier node in a different direction before accepting a new neighbor. Without acquiring key material, an attacker cannot create a wormhole since it must rely on forwarding messages to legitimate nodes through the wormhole to decrypt the nonce challenges. Those nodes will determine that the reception direction is inconsistent, and refuse to complete the protocol.

The verified neighbor discovery protocol is vulnerable, however, to a wormhole attack in which a single endpoint node acts as both a receiver and a retransmitter to deceive two nodes that are close to each other (but not within one hop) that they are neighbors. The next section describes this attack, and a variant of the protocol that prevents it.

5.4 Strict neighbor discovery protocol

Figure 6 shows the verifier region of two neighbor nodes. If there is a node in the shaded region, it can act as a verifier for A and B . However, the verifier region may still exist when two nodes are slightly out of radio range, and a smart adversary can use this to make them to be neighbors.

Figure 7 depicts the Worawannotai attack in which the adversary convinces two nearby (but not neighboring) nodes they are neighbors. Node B is located just beyond the transmission range of node A . There will be two areas that could have valid verifier for this protocol. If there is a valid verifier in those areas, the attacker can just put one node in between A and B (node X in Figure 7) and use it to listen to and retransmit messages between A and B . Nodes A and B will mistakenly confirm they are neighbors using verifier V , but the attacker will have control over all messages between A and B .

The Worawannotai attack will succeed only if the victim nodes (A and B in the figure) are unable to communicate directly, but are close enough to have a verifier that can hear both A and B . Assuming perfect transmission distances, this means A and B must be more than r distance apart, but less than

$$2r \cos \pi/6 = r\sqrt{3}$$

after which the size of the false verification region is zero. If A and B are aligned horizontally, the size of the areas that could contain false verifiers is

$$4 \int_{\frac{r+a}{2}}^{\frac{r\sqrt{3}}{2}} \left(\sqrt{1-x^2} - \frac{x}{\sqrt{3}} \right) dx$$

where $r + a$ is the distance between A and B . The maximum area is slightly less than 15% of the transmission area in the worst case where A and B are just over r distance apart (a is 0), and decreases substantially as the distance increases.

Another mitigating factor is that nodes A and B are within two hops through legitimate nodes (including the false verifier). This means if the data communication protocol uses acknowledgements, A will discover

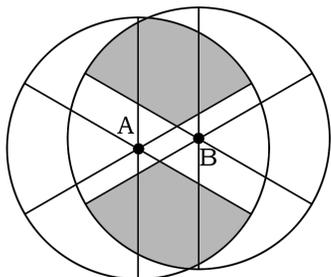


Figure 6. Verifier Region. The shaded area is the verifier region of node A and B in verified neighbor discovery protocol.

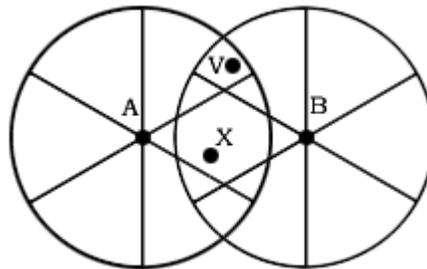


Figure 7. Worawannotai attack.

messages are not reaching B and can retransmit them through another node. This approach would not work against the distant wormhole attack, since A 's neighbors are likely to also forward packets directed towards B towards the wormhole endpoint.

To prevent the Worawannotai attack, we need to place additional constraints on verifiers. The strict neighbor discovery protocol exchanges the same messages as verified neighbor discovery protocol but has stricter requirements on verifiers. In strict protocol, a valid verifier V for the link $A \leftrightarrow B$ must satisfy these three properties:

1. $\text{zone}(B, A) \neq \text{zone}(B, V)$.
2. $\text{zone}(B, A) \neq \text{zone}(V, A)$.
3. $\text{zone}(B, V)$ cannot be both adjacent to $\text{zone}(B, A)$ and adjacent to $\text{zone}(V, A)$.

The first two conditions are the same as previous protocol, and they guarantee that the adversary cannot replay the confirmation message from verifiers. The third condition ensures that the verifier region is empty when two nodes are out of radio range, so the adversary cannot use this to conduct Worawannotai attack.

Figure 8 shows the verifier region of the strict protocol. Compared with Figure 6, we can see that the region between A and B is no longer contains verifiers. We next show that the shaded areas cannot contain any verifiers if A and B are further than r distance apart, and hence, the Worawannotai attack cannot succeed.

We consider the region above and left of A ; the proof for

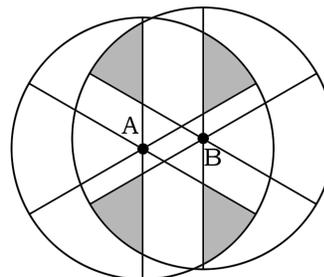


Figure 8. Strict Verifier Region. The shaded area is the verifier region of node A and B in strict neighbor discovery protocol

all other regions is equivalent. All potential verifiers V in that region have zone $(V, A) = 2$ and zone $(V, B) = 2$.

Let x_N denote the x coordinate of node N , d_{NM} the actual distance between nodes N and M , and θ_{NM} the angle between nodes N and M relative to the horizontal axis.

For a verifier V in the above left region, we have:

$$\begin{aligned} x_V &= x_A + d_{AV} \cos \theta_{AV} \\ x_V &= x_B + d_{BV} \cos \theta_{BV} \\ x_B &= x_A + d_{AB} \cos \theta_{AB} \end{aligned}$$

Because zone $(V, A) = 2$ and zone $(V, B) = 2$, we know θ_{AV} and θ_{BV} are between $\pi/2$ and $5\pi/6$. Hence, the minimum value of $\cos \theta_{AV}$ is 0 for $\pi/2$. So, we know $x_V \leq x_A$ and $x_V \leq x_B$. This makes sense since V must be to the left of both A and B . Substituting the expansion of x_B from the third equation into the second equation, we have,

$$x_V = x_A + d_{AB} \cos \theta_{AB} + d_{BV} \cos \theta_{BV}$$

Since zone $(A, B) = 1$, θ_{AB} is between $-\pi/6$ and $\pi/6$. Minimizing the values of the cosines, we have

$$x_V \geq x_A + d_{AB} \cos \pi/6 + d_{BV} \cos 5\pi/6.$$

The Worawannotai attack is possible only if $d_{AB} > r$ and $d_{BV} \leq r$. So, this implies $x_V > x_A$ which contradicts $x_V \leq x_A$ and proves that no false verifier could exist. Similarly, we can prove that all other three shaded regions are also empty if A and B are further than one hop apart.

5.5 Discussion

The strict neighbor discovery protocol prevents wormhole attacks when the adversary has only two endpoints. An attacker with multiple endpoints could selectively forward packets through different endpoints to establish false neighbors. In the extreme, an attacker who can surround a particular target node with wormhole endpoints can make messages arrive in any zone by transmitting them through a different endpoint. Our defense does not prevent multiple endpoint attacks, although it should be noted that the resources necessary to carry out such an attack are substantial.

The overhead associated with our protocol is minimal, the main cost is the potential loss of valid links (discussed in the next section). For pairwise key distribution, the typical process to securely discover one link (without resistance to wormhole attacks) involves node announcement, challenge and response (3 messages). Our protocol adds additional messages for inquiry, verification and acceptance.

One possible, but unrealistic, attack is to use magnets to attempt to disorient nodes. An attacker could use a magnet to control the orientation of a node to make it receive messages in the appropriate zone. This would require tight coordination between the wormhole retransmitting packets and the magnet controller. For the

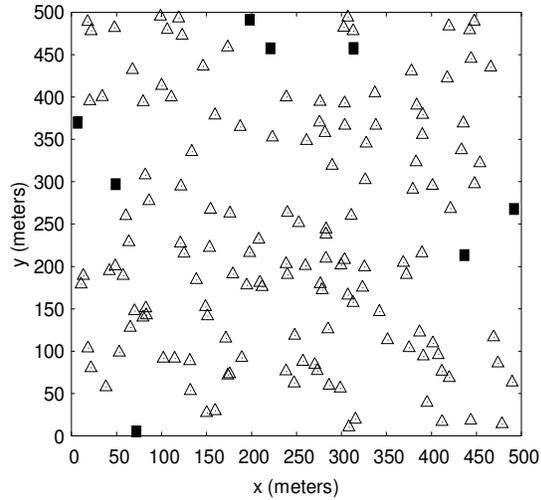
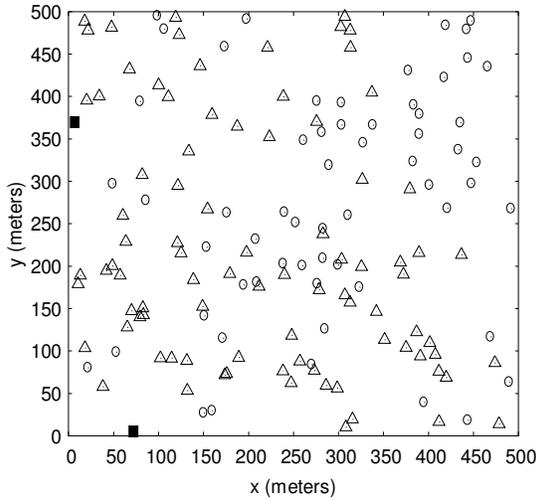
example in Figure 4, an enterprising attacker could retransmit node A 's announcement through the wormhole. Before transmitting B 's response, the attacker would use a magnet to disorient A one half rotation so its east zone is now facing west. Hence, A would accept B 's response as coming from the opposite direction (even though it actually came from the same direction). Note that the attacker would have to reorient A correctly before it sends out the next message. In practice, it is unlikely that a magnet attack could be done successfully because of the physical intrusiveness and timing precision needed. In most cases, it would be easier for an attacker to set up multiple endpoints.

6. Analysis

Our protocols have low overhead, but may effect the overall performance of the network by preventing legitimate links from being established. In this section we analyze the impact of our protocols on node connectivity and routing performance. In both the verified protocol and the strict protocol, it is possible that in step 3, there are no potential verifier nodes. Without a verifier node, N cannot distinguish legitimate neighbors from neighbors through a wormhole. Nodes near the perimeter of the network are especially prone to having no verifier nodes. In this case, an optimistic strategy will continue the protocol and accept the announcer without verification, while a conservative strategy will reject the announcer and stop the protocol. The first choice permits successful wormhole attacks while the second choice may prevent some legitimate nodes from joining the network. Because the damage a successful wormhole attack can cause is substantial, we adopt the more conservative choice: a node will only accept another node as a neighbor if it can be verified by at least one verifier.

Hence, our protocol prevents some legitimate links from being established. Losing links diminishes the effectiveness and efficiency of the network. If all links for a particular node are lost, it will be disconnected from the network and provide no value. The probability that a link is lost depends on the likelihood that a satisfactory verification node cannot be found. This probability scales inversely with the density of the network. For reasonably dense networks, there is a high probability that a legitimate link will be verified.

Figure 9 shows typical results from simulating a low density network with an average of 3 neighbors within the omnidirectional communication distance and 9.72 neighbors within the directional transmission distance. Under 14% of all links are lost, and only 1.3% of nodes are completely disconnected due to the verified neighbor protocol. The number of links lost increases substantially with the strict neighbor discovery protocol because of the reduced area for possible verifiers: the lost link and



a. Verified Protocol (Density=3, directional density = 9.7) b. Strict Protocol (Density = 3, directional density = 9.7)

Figure 9. Impact on Connectivity. Circular nodes establish all possible links; triangular nodes establish some links, but are unable to establish at least one possible link; solid square nodes are unable to establish any links.

disconnected nodes rates are 58% and 5.3% respectively. For a more typical network density of 10 neighbors (32.4 neighbors within the directional transmission distance), less than 0.5% of links are lost and no nodes are disconnected in verified neighbor discovery protocol, and 40% of links are lost and 0.03% nodes are disconnected in strict neighbor discovery protocol.

The closer nodes are to each other, the more likely they are find a verifier and establish a link. Figure 10 shows the impact of the distance between two nodes on their likelihood of establishing a link. For a reasonably dense network (omnidirectional density > 10), nearly all links are kept when the distance of nodes is within r in the verified protocol or within $0.6r$ in the strict protocol.

Few nodes are completely disconnected with either protocol, but the lost links can affect routing performance. Figure 11 shows the impact of our protocols on network

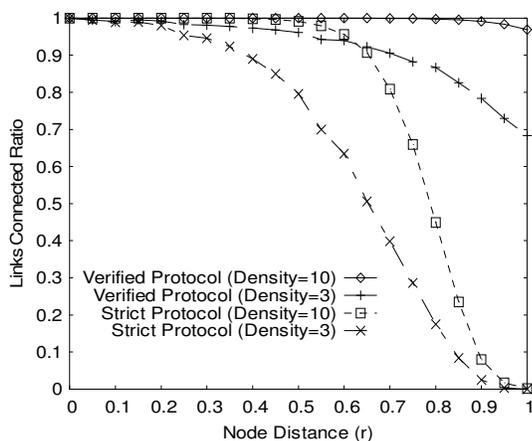


Figure 10. Node distance and connectivity.

routing. We use the shortest path routing to evaluate our protocol, and measure average route path length when the network density changes from 4 to 20. The results show that using the verified protocol for routing has virtually no impact on the path length: the path length is almost identical. The strict protocol will increase the path length around 20% when the network is reasonably dense (omnidirectional density > 10).

7. Directional Errors

So far, we have assumed nodes always hear each other in directly opposite directions (e.g., if node A hears node B in zone 1, node B hears node A in zone 4). In a typical deployment, this is often not the case. If nodes are near

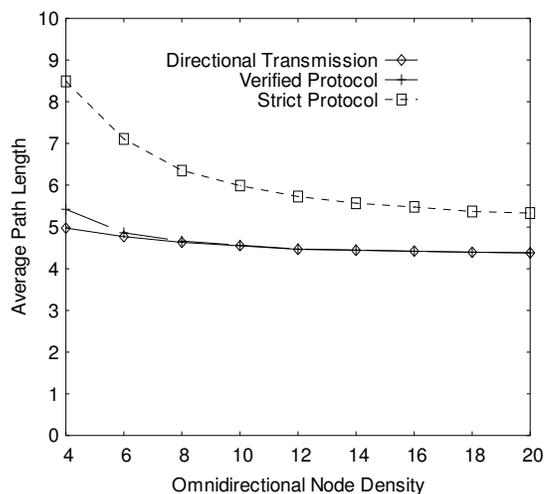


Figure 11. Impact on routing path length.

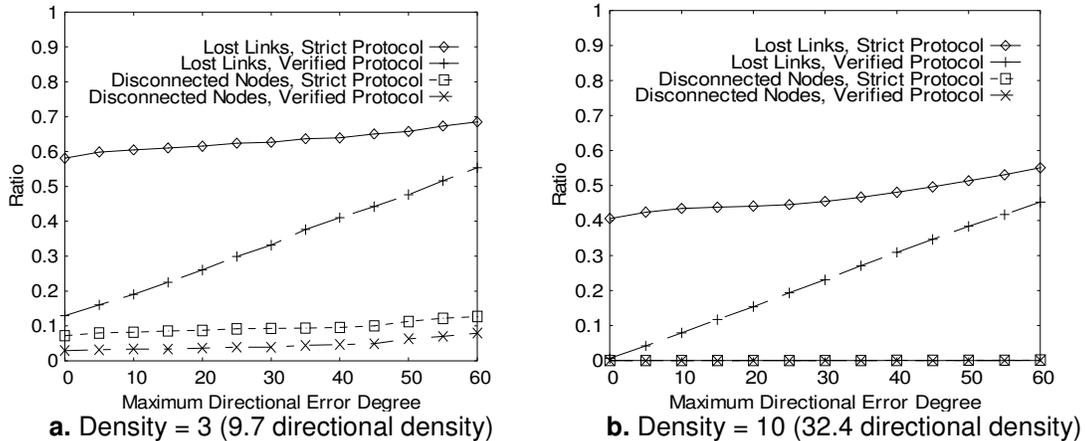


Figure 12. Impact of directional errors.

the transition point between two zones, small differences in node orientation, antenna alignment and gain, and transmission irregularities will lead to legitimate nodes appearing to be in the wrong zone. As result, some links between legitimate neighbors will be lost.

Figure 12 shows the impact of directional errors on network connectivity. We assume directional errors can be modeled by disorienting nodes by a random angle that is \pm the maximum directional error. Hence, for maximum directional error of 60 degrees, the average node is misoriented by $\frac{1}{2}$ zone. The fraction of links lost because of directional errors increases linearly with the directional error. For a reasonably dense network, even as the directional error reaches 60 degrees fewer than 0.1% of nodes are fully disconnected.

Another approach to directional errors would be to accept communications from directions that are not directly opposite the reception direction, but rather in an adjacent zone. This would eliminate most of the links lost to directional errors, but would substantially weaken the security of the protocol.

8. Conclusion

Wormhole attacks are a powerful attack that can be conducted without requiring any cryptographic breaks. An attacker who conducts a successful wormhole attack is in a position to disrupt routing, deny service to large segments of a network, and use selective forwarding to tamper with network applications. Directional antennas offer a promising approach to preventing wormhole attacks. They are less expensive than many mechanisms proposed for localization, and offer other advantages in addition to security including more efficient use of energy and better spatial use of bandwidth. The protocols we propose reduce the threat of wormhole attacks with minimal loss of network connectivity. Given the lack of

availability of other suitable defenses and the potential damage a successful wormhole attack can inflict, this tradeoff is desirable for many applications.

Acknowledgements

This work was supported by the National Science Foundation through CAREER (CCR-0092945) and ITR (EIA-0205327) awards. Chalermpong Worawannotai discovered the “Worawannotai attack” and suggested the solution. Section 5.3 of this version of the paper has been fixed from the proceedings version thanks to corrections provided by Yan Huang.

References

- [1] N. Bulusu, J. Heidemann and D. Estrin. *GPS-less Low Cost Outdoor Localization for Very Small Devices*. IEEE Personal Communications Magazine, October 2000.
- [2] S. Bandyopadhyay, K. Hausike, S. Horisawa and S. Tawara. *An Adaptive MAC and Directional Routing Protocol for Ad Hoc Wireless Networks Using ESPAR Antenna*. ACM/SIGMOBILE MobiHoc October 2001.
- [3] H. Chan, A. Perrig and D. Song. *Random Key Predistribution Schemes for Sensor Networks*. IEEE Symposium on Security and Privacy 2003.
- [4] R. Choudhury, X. Yang, R. Ramanathan and N. Vaidya. *Using Directional Antennas for Medium Access Control for Ad Hoc Network*. ACM MobiCom 2002, September 2002.
- [5] R. Choudhury and N. Vaidya. *Ad Hoc Routing Using Directional Antennas*. University of Illinois, Coordinated Science Laboratory, Technical Report, August 2002.
- [6] L. Eschenauer and V. Gligor. *A Key-Management Scheme for Distributed Sensor Networks*. ACM Conference on Computer and Communication Security, November 2002.
- [7] Y. Hu, D. Johnson, and A. Perrig. *SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks*.

- IEEE Workshop on Mobile Computing Systems and Applications, June 2002.
- [8] Y. Hu, A. Perrig and D. Johnson. *Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks*. ACM MobiCom 2002, September 2002.
- [9] Y. Hu, A. Perrig, and D. Johnson. *Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks*. INFOCOM 2003, April 2003.
- [10] T. He, C. Huang, B. Blum, J. Stankovic and T. Abdelzaher. *Range-Free Localization Schemes for Large Scale Sensor Networks*. ACM MobiCom 2003, September 2003.
- [11] D. Johnson, D. Maltz, and J. Broch. *The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*. In *Ad Hoc Networking*, C. Perkins, Ed. Addison-Wesley, 2001.
- [12] T. Korakis, G. Jakllari and L. Tassiulas. *A MAC protocol for full exploitation of Directional Antennas in Ad-hoc Wireless Networks*. MobiHoc 2003.
- [13] C. Karlof, Y. Li, J. Polastre. *ARREVE: Algorithm for Robust Routing in Volatile Environments*. Technical Report UCB//CSD-03-1233, March 2003.
- [14] Y. Ko, V. Shankarkumar and N. H. Vaidya. *Medium access control protocols using directional antennas in ad hoc networks*. IEEE INFOCOM, 2000.
- [15] C. Karlof and D. Wagner. *Secure Routing in Sensor Networks: Attacks and Countermeasures*. First IEEE International Workshop on Sensor Network Protocols and Applications, May, 2003.
- [16] S. Marti, T. J. Giuli, K. Lai, and M. Baker. *Mitigating routing misbehavior in mobile ad hoc networks*. ACM/IEEE International Conference on Mobile Computing and Networking, 2000.
- [17] A. Nasipuri, J. Mandava, H. Manchala and R. E. Hiromoto. *On Demand Routing Using Directional Antennas in Mobile Ad Hoc Networks*. IEEE Wireless Communications and Networking Conference (WCNC), September 2000.
- [18] D. Niculescu and B. Nath. *Ad Hoc Positioning System (APS) using AoA*. INFOCOM 2003.
- [19] R. Nagpal, H. Shrobe and J. Bachrach. *Organizing a Global Coordinate System from Local Information on an Ad Hoc Sensor Network*. 2nd International Workshop on Information Processing in Sensor Networks (IPSN '03), April, 2003.
- [20] A. Nasipuri, S. Ye, J. You, R.E. Hiromoto. *A MAC protocol for mobile ad-hoc networks using directional antennas*. IEEE Wireless Communications and Networking Conference, September 2000.
- [21] C. E. Perkins and P. Bhagwat. *Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers*. ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications, 1994.
- [22] A. Perrig, R. Szewczyk, V. Wen, D. Culler and D. Tygar. *SPINS: Security Protocols for Sensor Networks*. *Wireless Networks Journal*, September 2002.
- [23] P. Papadimitratos and Z. Haas. *Secure routing for mobile ad hoc networks*. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, January 2002.
- [24] C. Perkins and E. Royer. *Ad-Hoc On-Demand Distance Vector Routing*. IEEE Workshop on Mobile Computing Systems and Applications, February 1999.
- [25] R. Ramanathan. *On the Performance of Beamforming Antennas in Ad Hoc Network*. MobiHoc 2001, October 2001.
- [26] R. Ramanathan, Redi, Santivanez, Wiggins and Polit. *Ad Hoc Networking with Directional Antennas: A Complete Solution*. BBN Technical Report, 2003.
- [27] C. Santivanez and J. Redi. *On The Use of Directional Antennas for Sensor Networks*. Military Communications Conference (MILCOM 2003), October 2003.
- [28] N. Sastry, U. Shankar, and D. Wagner. *Secure Verification of Location Claims*. ACM Workshop on Wireless Security (WiSe 2003), September 19, 2003.
- [29] B. Wellenhoff, H. Lichtenegger and J. Collins. *Global Positions System: Theory and Practice*, Fourth Edition. Springer Verlag, 1997.