TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF ALGORITHMS

LIST OF FIGURES

Page

INTRODUCTION

Mobile ad hoc networks have been an attractive field of research for many years now. Their characteristics make them an excellent choice for emergency operations, vehicular communication, and short-live networks.

Ad hoc networks must deal with threats from both external agents and compromised internal agents. The lack of a central control and the fact that each node must forward packets of other nodes pose major security challenges. In such environments, it is difficult to ensure the confidentiality and integrity of the communications, as well as the availability of the services.

We focus here on the wormhole attack, one of the most serious threats to ad hoc networks. In this attack, two malicious nodes tunnel traffic from one end of the network to the other using an out-of-band link or an encapsulated link that takes advantage of the resources of the network. Their main goal is to attract traffic to drop, alter, or simply look at the packets later on.

Because of the characteristics of wormhole attacks, cryptographic solutions are not sufficient to prevent them. Numerous physical approaches have been proposed to secure the neighbor discovery process. Most of the solutions presented so far require that the nodes handle information about self-location, perform clock synchronization, or rely on information such as the trust relationship or on specialized antennas. Only a few solutions have been proposed to secure the overall end-to-end route discovery process.

Specifically, we propose a new approach to prevent Layer-3 wormhole attacks based on a multipath source routing protocol. The Witness Integration Multipath DSR (WIM-DSR) solution relies on the information provided by the routing protocol itself to determine whether or not there are some typical inconsistencies usually associated with these attacks. This solution does not require any cryptographic processing by the intermediate nodes if no attack takes place. Furthermore, it does not have any impact on packet length. These two features represent the main advantages over previous solutions (Hu, Perrig, and Johnson, 2003; Wang et al., 2006).

This thesis is structured as follows. Chapter 1 reviews the background and literature on ad hoc networks and the major routing protocol attacks. Chapter 2 describes the notion of a wormhole attack. Chapter 3 presents a review and analysis of wormhole attack solutions. In chapter 4, we present 4 WIM–DSR, which is an innovative protocol offering a new and inexpensive mechanism for preventing wormhole attacks. Finally, the results of the simulations are presented and analyzed in chapter 5.

CHAPTER 1

WIRELESS AD HOC NETWORK BACKGROUND

This chapter presents a general introduction to wireless ad hoc networks. Section 1.1 presents definitions of ad hoc networks and the forwarding process in these networks. Section 1.2 presents an introduction to routing protocols in ad hoc networks. Finally, in section 1.3, a review of the major threats and attacks to which ad hoc routing protocols are subject is presented.

1.1 Wireless Ad Hoc Networks

A wireless ad hoc network is a network deployed without any pre-existing infrastructure, which can be built on dynamically without any centralized administration (see Figure 1.1). The lack of a central authority and predefined infrastructure require all nodes in the network to participate actively in the common network functions, such as routing, addressing, security, etc. Nodes are free to move arbitrarily in the network, breaking and creating links dynamically, thereby constantly changing the topology of the underlying network.

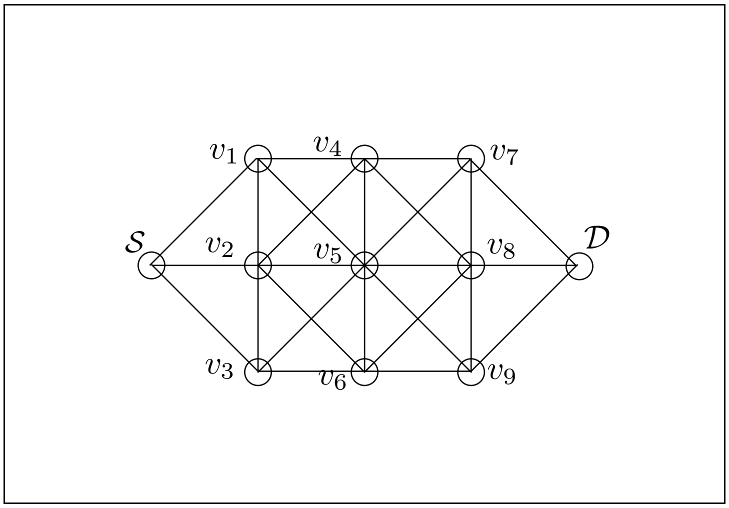


Figure 1.1 Wireless Ad Hoc Network.

The nodes participating in an ad hoc network must handle the routing process without the support or supervision of a central authority or any previously established infrastructure. As those nodes usually have a limited range of transmission, the packets exchanged between two nodes that are outside each other's transmission range must be forwarded on a hop-by-hop basis by intermediate nodes until the destination is reached.

Figure 1.2 shows how a packet having the node *S* as the source and the node *D* as the destination must be forwarded by the intermediate node *A*, since *S* is outside the transmission range of *D*.

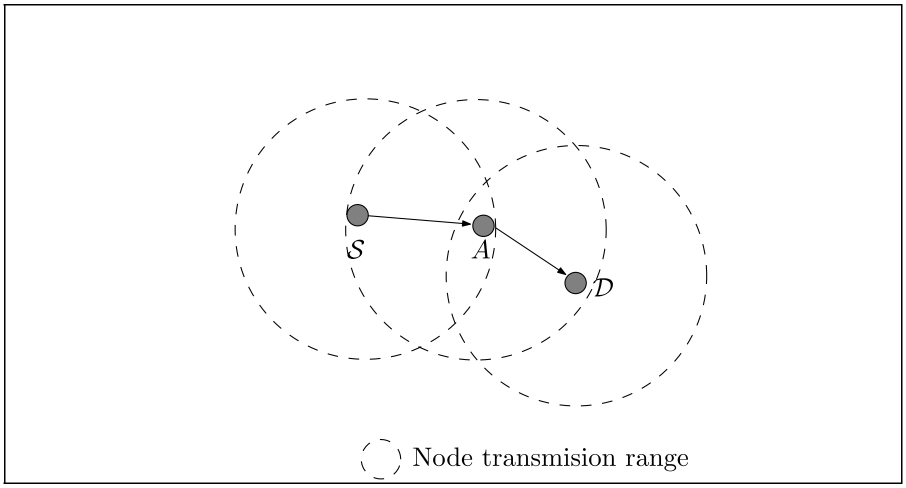


Figure 1.2 Packet forwarding in ad hoc networks.

Due to their self-organization and rapid deployment capabilities, mobile ad hoc networks can be used for several purposes, for example in battlefield communications, sensor networks, emergency relief scenarios, public social services, the virtual class room, etc.

1.2 Mobile Ad Hoc Routing Protocols

Unlike classical wired and wireless infrastructure networks, where packets are routed toward a destination using central authority devices (e.g. base stations or routers) and a predefined infrastructure, in ad hoc networks, the functions performed by those elements must be achieved by the nodes themselves.

The inherent characteristics of ad hoc networks, such as its high mobility, fast topology changes, and battery power limitations, have made the routing protocols used in traditional networks inaccurate. This means that other routing protocols specially designed for ad hoc networks must be used.

Based on the route created, ad hoc routing protocols can be classified into two main categories: (a) proactive; and (b) reactive. In a proactive routing protocol, the nodes register the changes in the network topology and periodically update routing information. In contrast, in a reactive routing protocol, the routes are discovered on-demand, i.e. only when they are required.

1.2.1 Proactive Routing Protocols

A proactive or table-driven routing protocol seeks to maintain consistent up-to-date routing information from all the possible paths connecting each node to every other node in the network. Table-driven routing protocols require each node to create and maintain one or several routing tables to store routing information, even though some routes may never be used. The nodes propagate routing updates periodically to their neighbors, in order to react to changes in the network topology and keep up-to-date routing information related to the entire network.

Proactive routing protocols have low latencies, since routes are already available when they are required. But the periodic exchange of routing updates increases the routing traffic substantially, and can have a major impact on the bandwidth of the network, especially in networks with a high density and/or high mobility rate. In those kinds of networks, changes in the network topology constantly generate demands from the routing protocol to keep the routing information up to date.

Proactive routing protocols are basically used in networks with low mobility or supporting traffic requiring low latency (such as voice and other real-time services).

Two of the most popular proactive routing protocols in ad hoc networks are the Destination-Sequenced Distance-Vector (DSDV) and the Optimized Link State Routing (OLSR) Protocol.

Destination-Sequenced Distance-Vector (DSDV) Routing Protocol

The Destination-Sequenced Distance Vector (DSDV) routing protocol (Perkins and Bhagwat, 1994) is a table-driven routing protocol, where each node in the network maintains a routing table including the number of hops required to reach any destination and the next node in the path. The routing table also includes a sequence number generated by the destination node, which makes it possible to ensure the freshness of the route. When a new route update packet is created, the value of the sequence number is increased, guaranteeing that it is the most recent route update.

DSDV guarantees loop-free routes through the use of these sequence numbers. When a node receives new information involving an already active route, the node compares the sequence numbers of the update and the existing routes in the table. If the new sequence number is higher than the one in the table, the entry for that route is modified. Otherwise, the update is discarded and no modification is made.

Optimized Link-State Routing (OLSR) Protocol

Optimized Link-State Routing (OLSR) protocol (Jacquet et al., 2001) is a link state routing protocol adapted to ad hoc networks. In OLSR, nodes must transmit a periodic routing advertisement called link-state advertisement (LSA), which lists their direct neighbors. Those route advertisements must be flooded throughout the network efficiently (see Figure **1.3**).

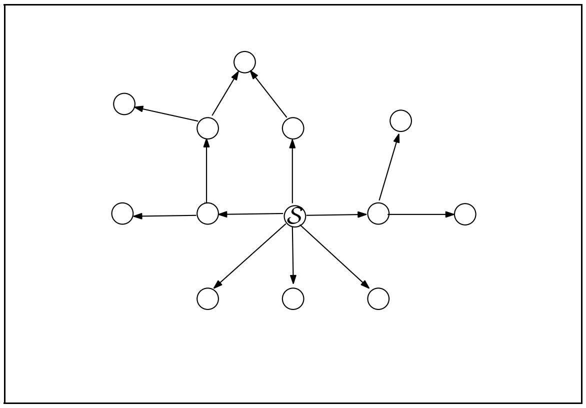


Figure 1.3 Flooding of route messages without optimization.

To reduce the number of LSAs traversing the network, the OLSR protocol introduces the concept of multipoint relay (MPR) nodes. The main purpose of MPR nodes is to allow an optimal transmission of LSAs through the network. Each node in the network must choose a minimal set of MPRs from its 1-hop neighbors. Those nodes are selected to ensure that the LSAs can reach all the 2-hop neighbors of the node.

In Figure 1.4, the node *S* broadcasts an LSA message containing its routing updates to their neighbors. Each of its 1-hop neighbors receives the message, but only those nodes forming part of the MPR set of node *S* are authorized to forward the LSA message, reducing the number of broadcasts and hence the overhead generated by the routing update process.

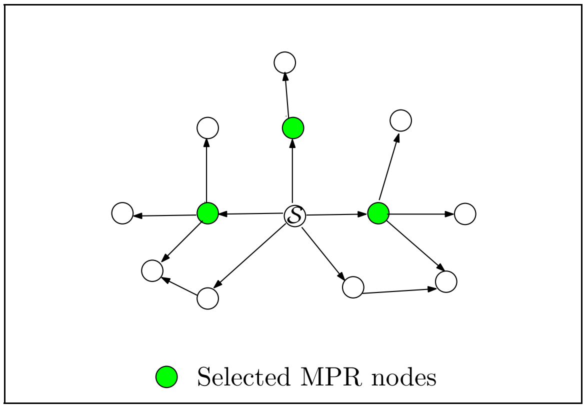


Figure 1.4 Selection of MPR nodes.

Two types of message are generally used in the OLSR protocol, which are (a) HELLO messages; and (b) Topology Control (TC) messages.

A HELLO message is used by the nodes to develop the neighbor discovery process and the MPR selection. In OLSR, each node periodically generates a HELLO message containing the list of its 1-hop neighbors and its own address. Exchanging HELLO messages frequently, each node in the network can build its list of 1-hop and 2-hop neighbors. HELLO messages are exchanged locally between the 1-hop neighbors and are not broadcast beyond that point. When a node has its list of 1-hop and 2-hop neighbors, it selects the minimum number of nodes from its 1-hop list, allowing it to reach all the nodes in its 2-hop list. Those nodes will form part of its MPR list.

When a node *A* has been selected as an MPR by another node *B,* node *A* must add node *B* to its MPR selector set list. This list contains all nodes that have selected node *A* as an MPR.

The Topology Control (TC) messages are the LSA messages of the OLSR protocol. TC messages are used to exchange route information in the network. TC messages are advertised periodically by MPR nodes, and they contain their MPR selector set list. Only MPR nodes can forward TC messages. When a node receives the TC messages from all its MPRs, it can recalculate its routing table with the information provided in those messages, building a route toward all the nodes in the network.

The OLSR protocol is one of the most important ad hoc routing protocols, and wide-ranging research is being developed to improve its performance and solve its security limitations.

1.2.2 Reactive Routing Protocols

Reactive or on-demand routing protocols create a route between two nodes only when a message needs to be exchanged between them. When a node needs to communicate with another and no route is available, the source node begins a route discovery process to find a route between the source and the destination, if such a route exists.

After establishing a route, the path is maintained by a route maintenance protocol until the end of the information exchange or until the destination becomes inaccessible.

The advantage of an on-demand routing protocol is the lower routing overhead generated when compared to that of proactive routing protocols. This is because no regular routing updates are exchanged by the nodes in the network, and routing traffic floods the network only when a route is required.

However, the delay generated by the route discovery process in finding a route is a concern for several applications (especially real-time applications, such as voice services).

Two of the most common on-demand routing protocols (Ad-hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR)) are described below.

Ad Hoc On-Demand Distance Vector (AODV)

The Ad Hoc On-demand Distance Vector (AODV) protocol (Perkins and Royer, 1999) is a reactive routing protocol developed specially for ad hoc networks. In AODV, when a node *S* wants to send a packet to a node *D* and there is no available route, node *S* uses a Route Request (RREQ) message to discover that route (see Figure 1.5). A RREQ message contains the IP addresses of the originating and target nodes.

The intermediate nodes that receive a RREQ message and do not have a fresh route to the destination rebroadcast the message to their neighbors, and so on, until the destination receives the RREQ message.

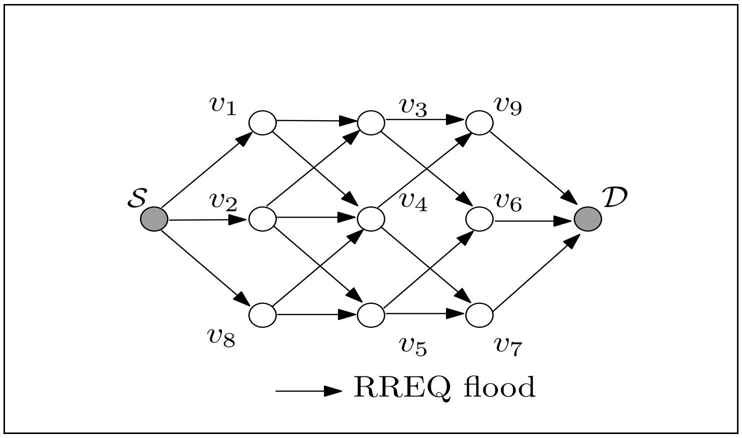


Figure 1.5 Flooding of RREQs in AODV.

After receiving the RREQ message, the destination uses the reverse path from which the original RREQ is received and forwards a unicast Route Reply (RREP) message to the source *S* (see Figure 1.6).

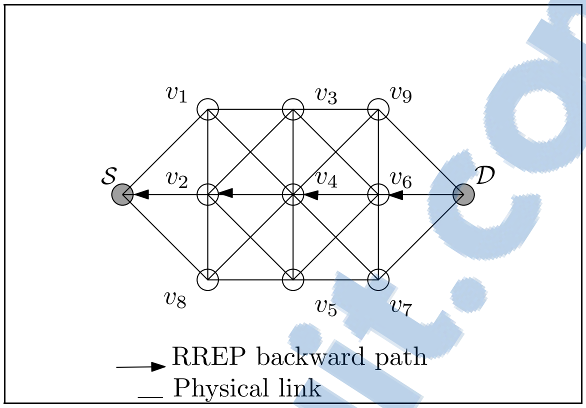


Figure 1.6 RREP backward path.

Other RREQ messages arriving at the destination later would be ignored. AODV allows intermediate nodes having a fresh enough route to the destination to respond and send a RREP message back containing the path to the destination node *D*.

AODV supports a route reparation mechanism using a Route Error (RERR) message. This message allows nodes that testify to the invalidation of a route to inform other nodes making use of that route about the link break. The RERR contains the list of destinations which have become unreachable as a consequence of the break.

Dynamic Source Routing (DSR) Protocol

The Dynamic Source Routing (DSR) protocol (Johnson et al., 1996) is an on-demand routing protocol that creates routes only when they are required, and does not exchange periodic routing traffic searching to create or maintain routes.

DSR is a source routing protocol. This means that, when a source node forwards a packet toward a destination, the source must add to the packet header the IP addresses of all intermediate nodes that the packet needs to traverse before arriving at the destination.

In DSR, when a node needs to send a packet to another node, it consults its route cache, searching for a valid route. If no valid route is found, the source begins the route discovery process.

In a route discovery process, the source node broadcasts a RREQ message containing its IP address, the IP address of the target node, and a unique identification number. Each intermediate node receiving the packet checks the unique identification number to confirm that this is the first time the message has been received, then the node searches its route cache for a route to the destination that is fresh enough. If no route is found, the intermediate node adds its own IP address and broadcasts the RREQ message, and so on, until the RREQ arrives at the destination (or an intermediate node with a fresh enough path to the destination) that responds to the source with a unicast RREP message using the reverse of the path obtained from the RREQ message.

Once the source receives the RREP message, it can start sending packets to the destination. Since the received message contains the IP addresses of all the intermediate hops in the path linking the two nodes, the message can be easily routed by an intermediate node, extracting the IP address of the next hop from the packet header. This means that intermediate nodes are not required to buffer the routing information for the traversing packets.

1.3 Major Routing Protocol Attacks in Ad Hoc Networks

An adversary can develop several attacks in ad hoc networks, taking advantage of constraints imposed by the nature of those networks, especially the lack of a fixed infrastructure and central control.

Examples of attacks in ad hoc networks include, but are not limited to:

- Data integrity attacks
- Illegitimate upgrading of privileges

- Denial of Service (DoS) attacks
- Passive eavesdropping
- Modification or injection of packets
- Signaling attacks
- Device stealing

We present below a review of the major attacks and threats in wireless ad hoc networks (Anjum and Mouchtaris, 2007).

1.3.1 Black Hole Attack

In a black hole attack, a malicious node broadcasts the network with forged routing information. The malicious node ensures the best route toward a part of the network, causing naïve nodes to route data packets through the malicious one.

After being selected as the proxy node, the attacker can drop the whole traffic performing a denial of service attack (DoS) or drop packets selectively, generating a grey hole attack (e.g. dropping only packets directed toward a specific network or *n* from each *m* packet, or a packet every *t* seconds).

An example of a black hole attack is illustrated in Figure 1.7, where malicious node *A* sends a fake RREP to the source node *S*, claiming that it has a fresh enough route toward the destination, the source node *S* will choose the route passing through node A, which, after finding a route to node *D* (e.g. sending a RREQ itself), will have access to the traffic exchanged between nodes *S* and *D*, and can begin to drop the packets that node *S* sends to node *D*.

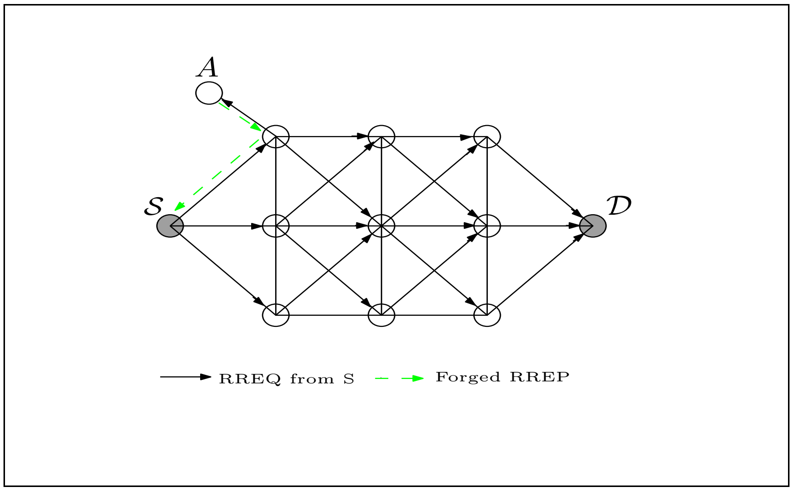


Figure 1.7 Black Hole attack.

1.3.2 Spoofing Attack

In a spoofing attack, an attacker attempts to impersonate a legitimate node by taking over its identity. With this attack, a malicious node can gain access to all the packets addressed to the victim.

An example of a spoofing attack is the man-in-the-middle attack (see Figure 1.8), where malicious node *B* spoofs node *A* into believing that malicious node *B* is node *C*, and spoofs node *C* into believing that malicious node *B* is node *A*. So, nodes *A* and *C* believe they are communicating with each other directly, but they are really communicating with the mediation of the malicious node *B* which controls all the exchanged traffic.

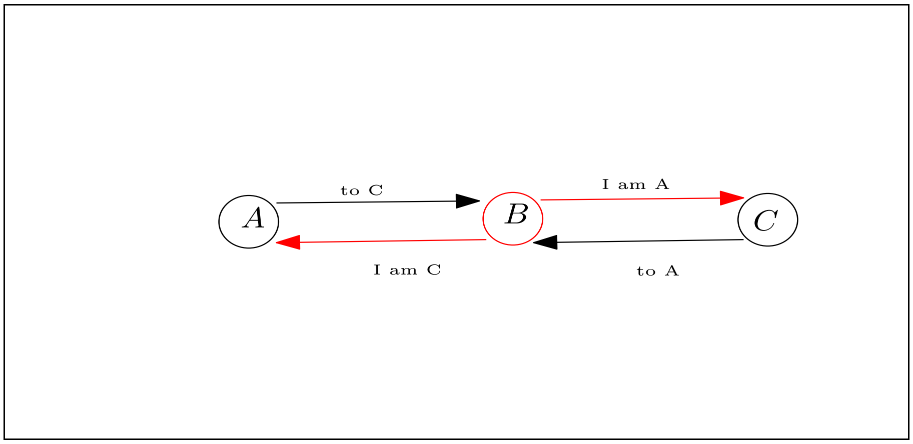


Figure 1.8 Man-in-the-Middle attack.

In OLSR (Jacquet et al., 2001), a link spoofing attack can be developed by a malicious node which advertises a fake link with 2-hop neighbors of a victim node, causing the victim node to select the attacker as its MPR. In the role of MPR of the victim node, the attacker can control the traffic going toward the target node (Kannhavong et al., 2007).

In Figure 1.9, malicious node *A* announces to node *S* that it shares links with nodes v_3 ,v_4, and v_5, and a fake node v_x. Thus, *A* is elected by *S* as the only valid MPR. After that, malicious node A controls all the traffic going toward *S*. If *A* throws out the packets related to *S*, the victim node can be isolated from the rest of the network.

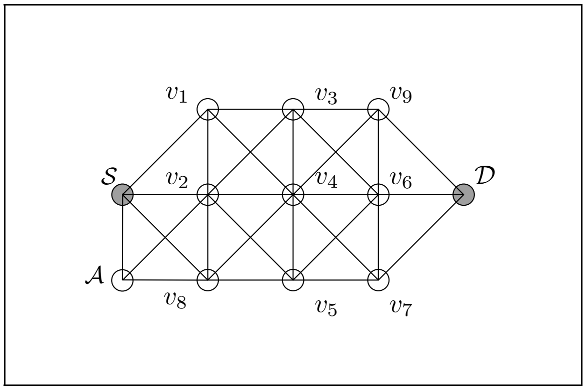


Figure 1.9 Link Spoofing attack

1.3.3 Sybil Attack

The Sybil attack incorporates a malicious device with the ability to illegitimately take on several identities in the same network. The forged identity from a malicious device is called a Sybil node. A malicious device can obtain an identity for a Sybil node in two different ways (Newsome et al., 2004); (a) generating a new identity; or (b) taking the identity from an existing node (with the cooperation of the node or by developing a spoofing attack).

Fabricated identity

In an ad hoc network where there is no central authority or predefined infrastructure, nodes must find a way to self-assign IP addresses. Taking advantage of this limitation, an attacker would create several addresses and allocate them to its Sybil nodes.

Existing identity

If there is a mechanism in the network limiting the free creation of new identities, an attacker can be obligated to steal the identity of an existing node (spoofing attack). The malicious node can either develop the impersonation far away from the legitimate node, preventing repair or temporarily disabling the victim node (e.g. through a DoS attack). In Figure 1.10, the malicious device A develops a Sybil attack creating a Sybil node with the identity of the legitimate node S, which is far away from the attack location.

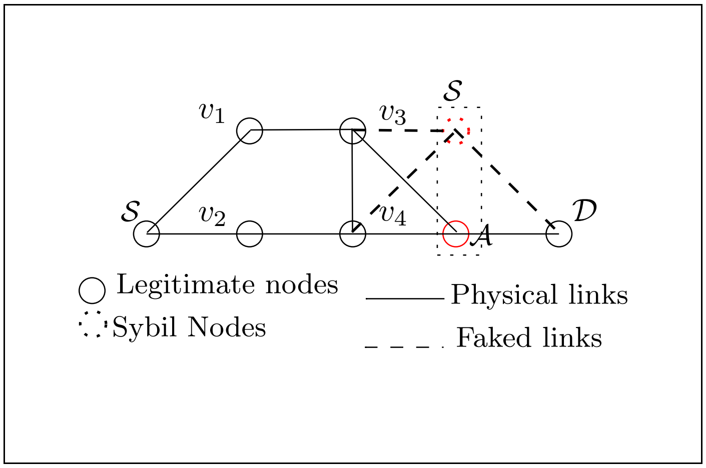


Figure 1.10 Sybil attack.

Moreover, two or more colluding nodes can participate in a coordinated Sybil attack by exchanging their IP address and cryptographic information (if there is any) and creating Sybil nodes with this information. In Figure 1.11, an attacker controlling malicious nodes *A* and *B* creates Sybil nodes *A* and *B* with the information from the original nodes.

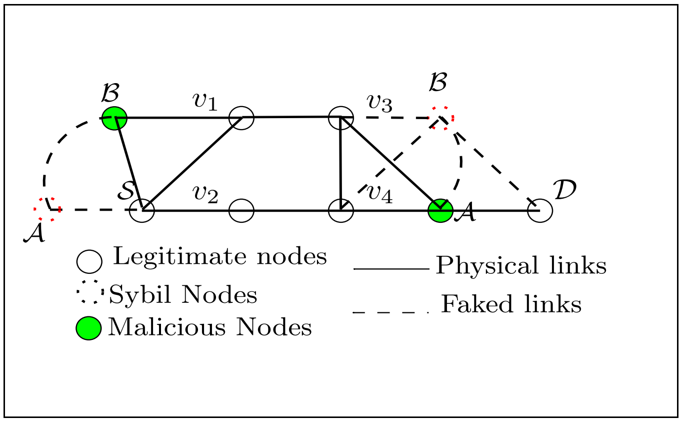


Figure 1.11 Sybil attack with colluding nodes.

1.3.4 Wormhole Attack

This is a serious attack developed by at least two colluding nodes sharing a private link or communicating via a tunnel that uses the existing resources of the network.

As the main objective of this thesis is to propose a method for detecting and preventing this attack, a detailed description of their characteristics is presented in the next chapter.

CHAPTER 2

WORMHOLE ATTACK

A wormhole attack is a serious attack developed on mobile ad hoc networks by at least two attackers sharing a private link and strategically located in the network. One of these attackers records the data obtained at one side of the network and tunnels it to the attacker at the other side, who replays the packets at that end. In Figure 2.1, colluding nodes M_1 and M_2 perform a wormhole attack tunneling the traffic sent by source node *S* toward destination node *D*.

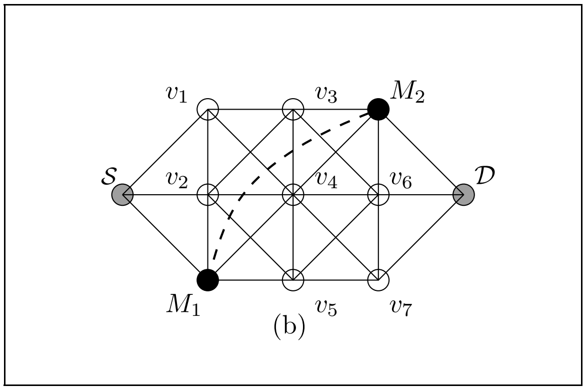


Figure 2.1 Wormhole attack.

Through replaying on one side of the network packets originating on the other side, the attackers search to manipulate the routing protocol, making other nodes believe they are closer, forcing all affected nodes to communicate through the malicious link. In this way, the wormhole attackers can have access to the packets exchanged by the victims and perform several other attacks, like a black hole attack (dropping all traffic), grey hole attacks (dropping a selected part of the traffic and forwarding the other part), or simply eavesdropping in the network, searching for sensitive information like cryptographic keys.

2.1 Classification of wormhole attacks

We present here several different models used to classify wormhole attacks. In section 2.1.1, we explain the two different classifications of wormhole attacks based on the communication technique used by the colluding nodes. Section 2.1.2 presents the classification of wormhole attacks based on the visibility of the attackers in the network. Section 2.1.3 introduces the Active-x-y wormhole attack model, and, finally, in section 2.1.4, we present the OSI Layer wormhole attack model.

2.1.1 Classification of wormhole attacks based on communication techniques

Wormhole attackers can use two different communication techniques to perform their attack (Wang et al., 2006): an encapsulation channel and an out-of-band channel.

Wormhole attack using an encapsulation channel

In this type of wormhole attack, two malicious nodes search to disrupt the normal operation of the routing protocol in the network, involving themselves in the route discovery process. To achieve this goal, the malicious nodes must give the false impression that the route through them is shorter than the other routes, even when they may be separated by a larger number of hops.

In Figure 2.2, if a reactive routing protocol (AODV, DSR, etc.) is used, when the node *S* seeks to establish a path toward the node *D* broadcasting a RREQ message, as soon as the malicious node M_1 hears the RREQ packet, it encapsulates the message into a new one intended for M_2 and tunnels it using the existing path between them (M_1-*A*-*B*-M_2).

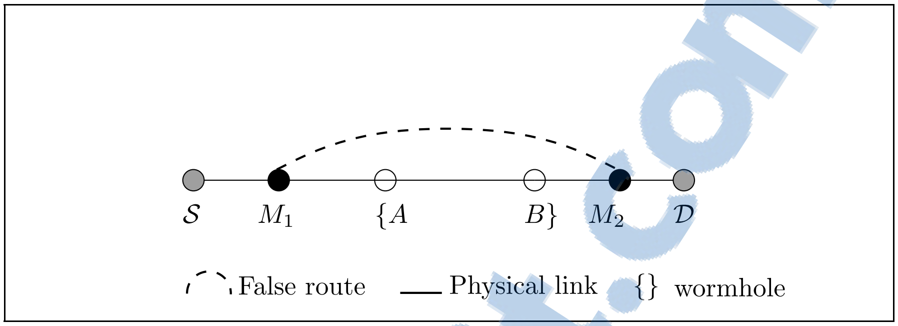


Figure 2.2 Wormhole using encapsulation.

Once node M_2 receives the message from node M_1, it decapsulates the packet recovering the original RREQ message and broadcasts it, seeking to reach node *D*. When the RREQ arrives at *D*, the number of hops presented in the packet will be fewer than the real number of hops traversed. In this way, in the presence of an algorithm using the number of hops as the metric of path selection, the malicious nodes will increase their possibility of being part of the chosen route.

In contrast, if a proactive routing protocol like OLSR is used, the malicious nodes M_1 and M_2 (Figure 2.2) will capsulate the HELLO messages from node *S* toward node *D*, and vice versa, through the path M_1-*A*-*B*-M_2 (see Figure 2.3). As a consequence, the neighbor discovery process is manipulated, giving nodes *S* and *D* false information about their neighbor nodes and malicious nodes M_1 and M_2 controlling the traffic shared between nodes *S* and *D*.

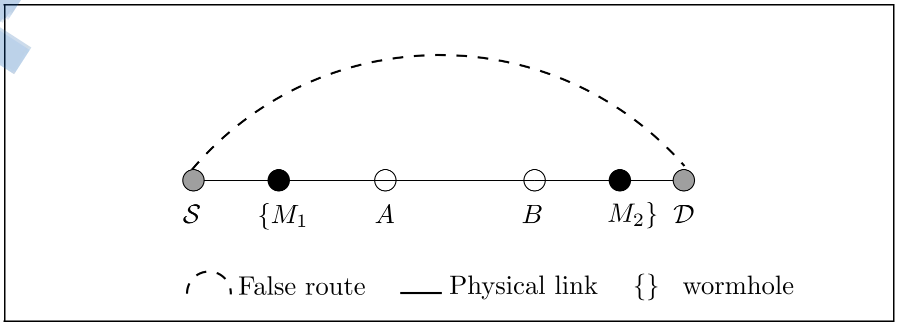


Figure 2.3 Encapsulation of HELLO messages.

This attack can be effortlessly deployed because it does not require additional hardware capability, but the delay caused by the process of tunneling messages between the two malicious nodes also makes it more easily detectable, and hence several measures can be implemented to prevent it. For this reason, this type of attack alternative is less efficient.

Wormhole attack using an out-of-band channel

This type of wormhole attack is launched by two malicious nodes strategically located in a network sharing an out-of-band high-bandwidth channel (wired or wireless). In Figure 2.4, when node *S* broadcasts a control message (RREQ, HELLO, etc.) directed to node *D*, the malicious node M_1 receives the message and tunnels it to malicious node M_2 using the shared out-of-band high-bandwidth channel. Node M_2 receives the control message and broadcasts it, seeking to reach node *D*. As the packets traveling through the wormhole link have the shortest delay and a smaller number of hops than the packets traveling by legitimates links, the wormhole link will have a higher probability of being chosen as part of the elected path connecting nodes *S* and *D*.

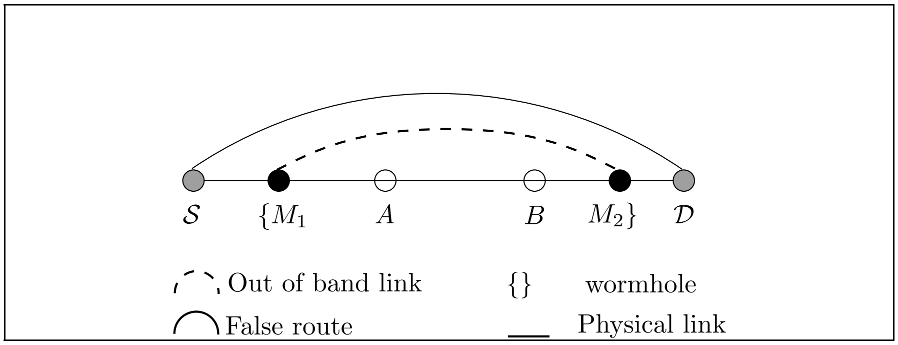


Figure 2.4 Wormhole attack using an out-of-band link.

2.1.2 Classification of wormhole attacks based on the visibility of malicious nodes

Based on the exposure of malicious nodes, wormhole attacks can be classified into three types: closed, half-open, and open (Wang et al., 2006).

Closed wormhole attacks

In closed wormhole attacks, two malicious nodes tunnel the packets from one end of the wormhole to the other and rebroadcast them without modifying them or adding any information, even when the packet is part of the route discovery process. In this way, the malicious nodes are invisible to the rest of the network. This attack can be developed by external agents (such as transceiver). In Figure 2.5 (a), the malicious nodes M_1 and M_2 develop a closed wormhole attack, making nodes S and D believe they are directly connected.

Half-open wormhole attacks

In a half-open wormhole attack, as shown in Figure 2.5 (b), only one of the malicious nodes adds its information to the control packet header according to the normal routing protocol, while the other malicious node tunnels and rebroadcasts the packets without adding any information about itself.

Open wormhole attacks

In this type of wormhole attack, the malicious nodes M_1 and M_2 are composed of internal nodes that participate actively in the routing protocol. The attackers include their own information in the control packet header according to the normal procedure. They do not hide their presence in the network, but lie about the distance that separates them, making honest nodes believe that the two malicious nodes are direct neighbors. In Figure 2.5 (c), both attackers (M_1 and M_2) are visible to nodes *S* and *D*, but the information about the distance between them is falsified.

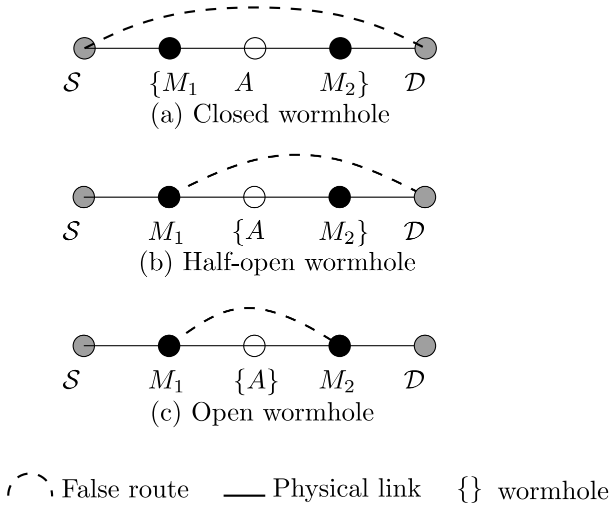


Figure 2.5 Classification of wormhole attacks.

2.1.3 *Active-x-y* wormhole attack model

Attackers in a wireless ad hoc network can be classified as passive or active, depending on their interaction with the network. A passive attacker does not create or modify packets, but only eavesdrops on the network. This type of attacker mainly represents a threat to privacy.

An active attacker can create and/or modify packets in a neighboring discovery, routing, or packet forwarding process, and also eavesdrop on communication between other nodes. This kind of attack is developed using compromised nodes. When an attacker compromises a node, it normally has access to the resources and information owned by the compromised node, including cryptographic keys.

In the attacker model proposed by Hu, Perrig, and Johnson (Hu, Perrig, and Johnson, 2005), an active attacker owning *n* nodes in a network of which *m* are compromised can be denoted an *Active-n-m* attacker.

A wormhole attack developed by an active attacker owning two colluding nodes can be modeled as an *Active-2-x* attack, where $0 \leq x \leq 2$. Hence, the possible wormhole attacks that can be developed by this attacker in the network can be described as *Active-2-0*, *Active-2-1*, or *Active-2-2* attacks.

An *Active-2-0* wormhole attack (closed wormhole) uses two external nodes, an *Active-2-1* attack (half-open) uses one external and one compromised node, and, finally, an *Active-2-2* attack (open wormhole) involves two compromised nodes.

2.1.4 OSI Layer wormhole attack model

Another characterization of a wormhole attack has been proposed by Buttyán and Hubaux (Buttyán and Hubaux, 2008). In this approach, the term is used exclusively for a layer-2 attack, where the malicious agents are simply transceivers exchanging messages from one end of the network to the other (closed or *Active-2-0* wormhole attacks). Those malicious agents remain invisible to the set of legitimate nodes.

The term *tunneling attack* is proposed for layer-3 attacks, where at least one of the colluding nodes is a compromised node participating actively in the routing protocol (open and half-open attacks).

2.1.5 Conclusion

In this thesis, we use the classification proposed by Wang et al. (open, half-open, and closed wormholes).

The malicious nodes can use either a physical out-of-band link (wired or wireless) or a logical encapsulated tunnel in the network itself. However, we assume the existence of the first alternative, since it is more realistic, because the second kind of wormhole attack can be easily detected due to the delay generated by the encapsulation of the message. This assumption will be made even when our approach can work efficiently in both.

CHAPTER 3

REVIEW AND ANALYSIS OF WORMHOLE COUNTERMEASURES

Wormhole attacks represent a serious threat to ad hoc networks, and the classical security methods are inappropriate to face this menace. As a result, a wide variety of new wormhole attack discovery and prevention techniques have been proposed. A review of the main approaches is presented in this chapter.

The rest of this chapter is organized as follows. Section 3.1 presents the most important solutions proposed to counteract closed wormhole attacks. Section 3.2 reviews the main solutions to open wormhole attacks. Finally, in section 3.3, we present a general summary of the current wormhole attack solutions.

3.1 Solutions for closed (Layer-2) wormhole attacks

As mentioned in section 2.1.2, closed wormhole attacks are developed by external agents seeking to threaten the security of the network by attacking the neighbor discovery (ND) protocol. The nature of wireless ad hoc networks, where the mobility of the nodes generates a constant modification of the network topology, makes attacks against ND relatively easy and securing ND a difficult problem (Poturalski, Papadimitratos and Hubaux, 2008).

In the closed wormhole attack, an external attacker falsely convinces a group of honest nodes that they are neighbors. The proposed solutions to thwart this attack are based on the deployment of a secure neighbor discovery protocol which allows an honest node to verify that the distance separating it from a remote node is shorter than the maximum allowed transmission distance in the network.

The solutions proposed for securing the ND protocol can be organized in several groups, according to their characteristics.

3.1.1 Distance Bounding Solutions

The distance bounding approaches establish an upper limit to the distance separating two nodes. This distance is calculated using the signal round-trip delay time of a challenge-response exchange or the received signal strength. If the resulting distance is larger than the maximum allowed distance in the network (which would be the case in a closed wormhole attack), the nodes estimate that a wormhole attack is being developed.

Temporal Leashes

An important solution proposed to the closed wormhole attack is the *packet leash* approach. The concept of using packet leashes for detecting wormhole attacks was introduced originally by Hu, Perrig, and Johnson (Hu, Perrig, and Johnson, 2003). This approach is based on the assumption that all the nodes in the network have location-aware mechanisms and/or synchronized clocks. Exchanging time and/or position information securely, two nodes can calculate an upper limit of the distance between them.

There exist two kinds of packet leashes: temporal and geographic. To use temporal leashes, all the nodes in the network must have tightly synchronized clocks. The maximum clock synchronization error Δ must be known by each node in the network. When node *A* forwards a packet to an assumed neighbor node *B*, it includes in the packet the time when the packet is sent, t_s. Node *B* will compare t_s with the time when the packet is received, t_r. An approximation of the distance between the two nodes can be calculated using t_r, t_s, and the signal propagation speed. If the packet has traveled too far, it must be eliminated.

Hu, Perrig, and Johnson also propose a modification to temporal leashes, in this case the sending node adding an expiration time t_e to the packet instead of t_s. If the expiration time has passed when the packet arrives at the next-hop node, it is discarded (Hu, Perrig, and Johnson, 2003). This approach does not offer significant advantages over the previous one, other than the lower calculation effort required by the receiver node which needs only to compare the expiration time t_e in the packet with its own reception time t_r, this gain in computation time being negligible. The receiving node must trust the sending node, because this latter is the one that defines the value of t_e.

The authenticity of messages in temporal leashes is ensured using the TIK (TESLA with Instant Key disclosure) protocol. TIK is an extension of the TESLA broadcast authentication protocol (Perrig et al., 2000). It offers efficient authentication for broadcast communication and a temporal leash based on a message authentication code (MAC).

The use of temporal leashes involves a tight synchronization between nodes. Special hardware must be added, hindering the deployment of this solution. Another difficulty lies in the fact that a sender does not know exactly when a packet will be sent, because of the contention-based characteristic in the MAC layer. This delay can add a non negligible error, making the solution non viable (Zhenning et al., 2004). In addition, the authors (Hu, Perrig, and Johnson) assume that the sending and receiving delay of the packet are negligible, but, in reality, especially when an RF signal is used, the error generated by the processing delay can be considerable.

The temporal leash approach was designed to secure the ND process and to prevent closed wormhole attacks. Therefore, it cannot detect open or half-open wormholes because of the hop-by-hop basis of this solution.

In Figure 3.1, nodes *S* and *D* are capable of verifying the distance of the links $S-M_1$ and M_2-D respectively using temporal leashes, but the distance of the link M_1-M_2 cannot be verified by any honest node, leaving open and half-open wormhole attacks undetected.

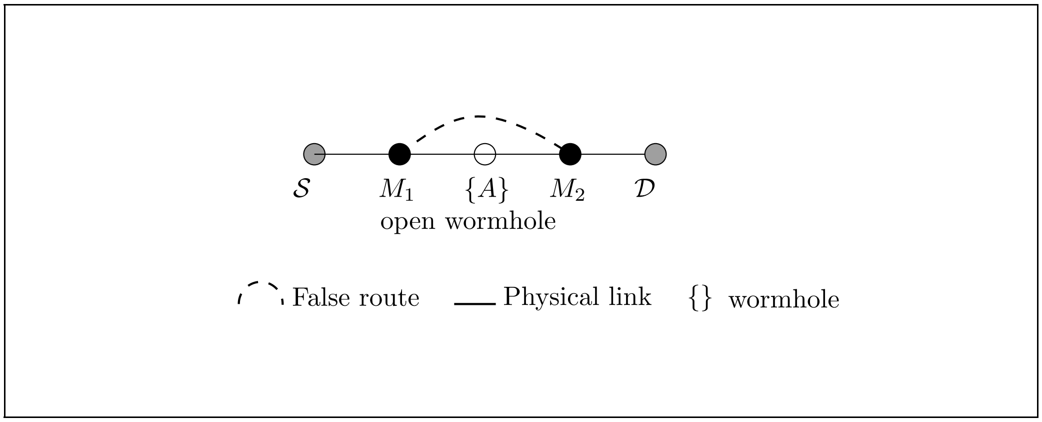


Figure 3.1 Open wormhole.

Time-of-Flight techniques

An approach comparable to the temporal packet leash approach is based on the time of flight of packets in the network. Essentially, the concept behind this technique is the following: there is a direct relationship between the distance that separates two points and the time it takes a signal to travel between them (Brands and Chaum, 1994; Capkun et al., 2003; Singelee and Preneel, 2005). If a signal travels a distance *d* in a time *t* at constant speed *v*, *d* can be given by:

$$d = v * t$$

Let D_{sr} be the distance between the nodes *s* and *r*, c the signal propagation speed, t_δ the total round-trip travel time measured between the two nodes, t_d the processing delay time at node *r*, and t_p the one-way propagation time between the two nodes. The distance D_{sr} can be approximated theoretically by:

$$D_{sr} = c * \frac{t_\delta}{2}$$

where:

$$t_\delta = 2 * t_p + t_d$$

A mechanism to decrease the value of t_d, which is a considerable source of error, is proposed by Capkun, Buttyan and Hubaux (Capkun, Buttyan and Hubaux, 2003). In this approach, each node must be equipped with a special hardware module allowing the response to a one-bit challenge with a one-bit response without involving any CPU participation and with a delay time t_d tending to zero. Replacing $t_d = 0$, then $t_\delta = 2 * t_p$, an approximate distance D_{sr} can be calculated as follows:

$$D_{sr} = c * t_p$$

This approach eliminates the need for tightly synchronized clocks. A node only requires the temporal information provided by its own clock, but any value of the processing delay time t_d different to zero adds a high error value to the distance calculated. In addition, the required use of esoteric hardware makes the cost of the solution prohibitive.

TrueLink is an approach proposed by Jakob et al. to deter wormhole attacks, which uses the store-and-forward technique (Jakob, Srikanth, and Michalis, 2006). It secures the ND process by verifying the adjacency of a pair of apparent neighbors using an exchange of cryptographically authenticated nonces in a tight timing scenario. This restricted time would not allow a malicious attacker to store and replay the packets at both ends of the wormhole link.

This approach fails to detect wormhole attacks when the malicious nodes replay the packet on a bit-by-bit basis instead of waiting for the whole packet to be collected at one end of the wormhole before forwarding it to the other end. In addition, this approach will leave open attacks undetected.

Received Signal Strength techniques

There are several proposals for calculating the distance between two points using the received signal strength (Meguerdichian et al., 2001; Patwari et al., 2003). The basic principle is that the strength of a signal decreases in proportion to the distance traveled.

If a node receives a signal and the value of the transmitted signal strengths is predefined or added by the source in the message, the receiving node can calculate an approximation of the distance using the difference between the transmitted and received signal strength values and a theoretical path loss model, as proposed in (Rappaport, 2002).

If the signals of several sources (like anchor nodes) are available, as well as their location information, the destination node can calculate an approximation of its location using empirical models, as proposed by Barbeau and Robert (Barbeau and Robert, 2006).

The advantage of this technique is that no additional hardware is required, and even resource-constrained nodes can use it. However, it requires that a trust relationship exist between the nodes. A strong assumption is also required: the nodes must not be able to modify their signal transmission power. Furthermore, the obstacles in the path modify the strength of the signal, adding an error to the calculated distance or location. Therefore, these approaches are very limited.

3.1.2 Location-aware approaches

There are several approaches using location-aware information for detecting and/or preventing wormhole attacks. The most important proposals are presented below.

Geographical Leashes

Hu, Perrig, and Johnson propose a variation of temporal leashes, where the nodes add self-location information to the packets, permitting neighbor nodes to calculate the distance separating them. With the distance obtained using the location information, the nodes can develop a secure ND process (Hu, Perrig, and Johnson, 2003).

To build a geographical leash, each node in the network must have the information concerning its own location. In addition, all nodes must have loosely synchronized clocks. When a node sends a packet, it includes in the packet the information concerning its own location p_s and the time when the packet is sent, t_s. The node receiving the packet compares the time and position values included in the packet with its own position, p_r, and the time when the packet is received, t_r.

Let Δ be the clock synchronization error, v the maximum node velocity, and σ the maximum possible location error. The receiving node can calculate the upper limit distance to the sending node d_{sr} using the equation:

$$d_{sr} \leq \left\| p_s - p_r \right\| + 2v(t_r - t_s + \Delta) + \sigma$$

In this way, a node can detect a wormhole attack if the calculated upper limit distance is not consistent with the information provided by the sending node.

Geographical leashes use a signature scheme to provide authentication and non repudiation. This approach, based on asymmetrical cryptography, can be computationally expensive, especially for the sender.

Geographical leashes require the existence of a mechanism providing correct location information to every node in the network. Global Positioning System (GPS) technology is generally proposed to provide this information.

Unfortunately, this technology has several limitations. The price of the devices remains high, limiting the use of geographical leashes in a mass application. The low location precision introduces a non negligible error into the estimate of the distance. Finally, the performance of the technology inside buildings is restricted. These constraints reduce the applicability of GPS in geographical leashes.

Geographical leashes is a protocol created to secure the ND process. It shares the same limitations as temporal leashes, relative to the detection of open and half-open wormhole attacks.

Directional antennas

Hu and Evans (Hu and Evans, 2004) propose the use of directional antennas to detect wormhole attacks. In this approach, each node communicates with the other nodes using a specific sector of its antenna. A node receiving a message has general information concerning the location of the other node based on the sector where the message has been received.

In Figure 3.2, we present a simplified circular model of a sector antenna where the range of the node transmission is divided into six equally sized zones.

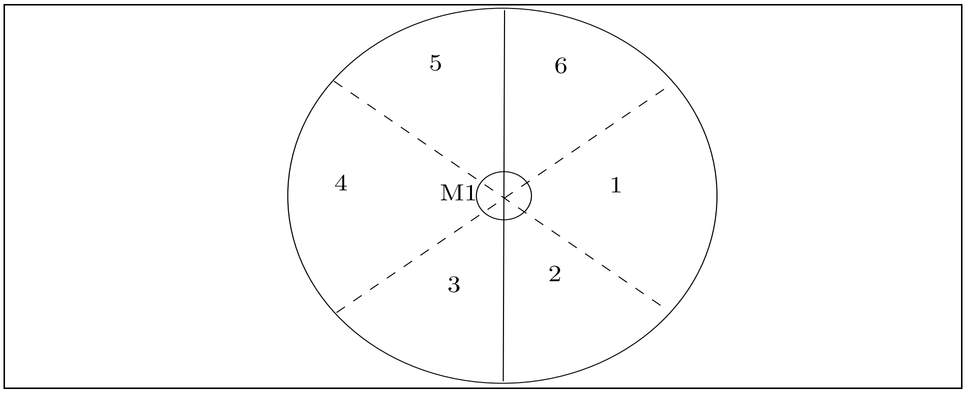


Figure 3.2 Sector antenna with 6 zones.

If a node *S* sends a (cryptographically protected) packet to its supposed neighbor node *D* using zone 1, node *D* should receive the packet in the opposite zone (4 in this case). If a wormhole attack exists, it would introduce inconsistencies in the sending and receiving zones that could be easily detected if honest nodes collaborate with each other to exchange accurate information.

Hu and Evans (Hu and Evans, 2004) introduce the concept of the link verifier node. A valid verifier *V* for the link formed by nodes *S* and *D* must fulfill two conditions:

1. Node *D* must hear nodes *S* and *V* in different zones.
2. Nodes *D* and V must hear node *S* in different zones.

According to this approach, if at least one verifier node *V* is found for the link formed by nodes *S* and *D,* then no wormhole attack is being developed over this link.

The solution presented by Hu and Evans (Hu and Evans, 2004) works well in terms of securing the ND process, thereby detecting closed wormhole attacks, but it is unable to detect open or half-open wormholes. This approach is based on the supposition that the communication, on a hop-by-hop basis, is established between honest nodes in an unsafe channel, which is not the case with open and half-open wormhole attacks.

In Figure 3.3, in the presence of a half-open wormhole attack formed by the malicious nodes M_1 and M_2, if the hidden wormhole node M_1 hears the packets sent by node *S* and tunnels them to malicious node M_2 with the information of the zone where it heard the packet, M_2 can establish a link with *S* through the wormhole link without being detected. In this case, node *V* fulfills the requirement to be the verifier node to the specified link.

Moreover, the cost of sector antennas is higher compared to omnidirectional antennas. This overcost can be a restraining factor in the massive application of this technique.

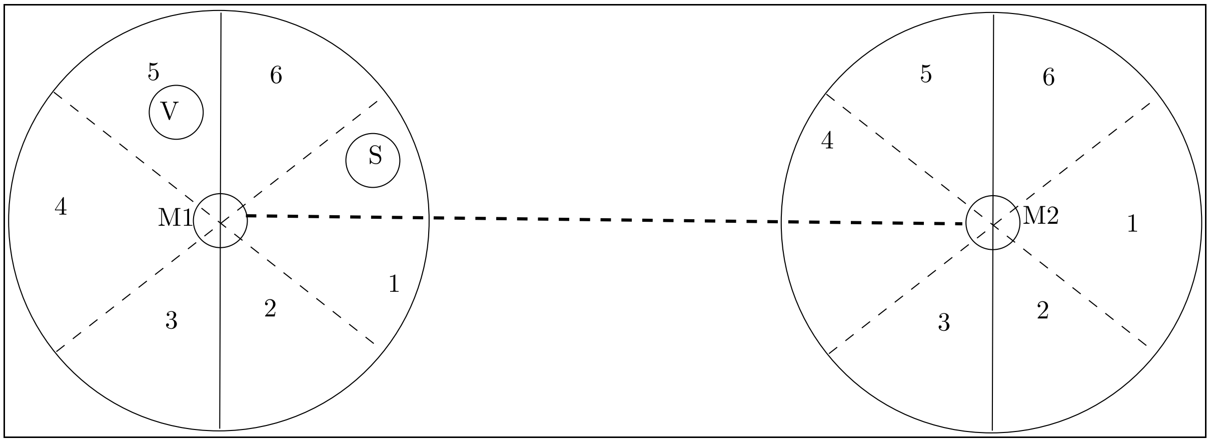


Figure 3.3 Wormhole attack against the sector antenna approach.

Lazos and Poovendran propose a modified approach which seeks to provide a secure localization method for honest nodes in networks with malicious adversaries (Lazos and Poovendran, 2004). This approach requires the presence in the network of a set of special nodes called *locators.* These nodes are equipped with directional antennas and self-location information. The rest of the nodes require no special hardware.

The nodes can calculate their location using beacons generated by the locator nodes. These beacons relay information about the locator's coordinates and the zone of the antenna from which the message was transmitted. With this information, a node picking up signals from beacons coming from several locators can approximate its location as the center of gravity (CoG) of the overlapping regions calculated with the information provided by the beacons.

This approach can detect a closed wormhole attack by combining the information of the beacon’s transmission zone and its estimated location; however, open and half-open wormhole attacks remain undetected. Also, even if the implementation cost of this approach decreases compared to that of the directional antennas (Hu and Evans, 2004), since most nodes in the network do not require them, the anchor nodes require additional access to up-to-date location information. Moreover, the computation of the location in the nodes can be resource-consuming.

3.1.3 Graphical and statistical techniques

Buttyán, Dóra, and Vajda propose an approach for detecting closed wormhole attacks in static sensor networks (Buttyán, Dóra, and Vajda, 2005). Their approach assumes a large number of sensor nodes and a small number of base stations placed uniformly on a flat two-dimensional surface. The base stations have unlimited resources and location-aware devices.

An undefined cryptographic protocol ensures the confidentiality, integrity, and authenticity of the messages exchanged between the nodes and the base stations.

The authors propose two wormhole detection mechanisms. The first is based on the fact that a closed wormhole link increases the number of 1-hop neighbors of the nodes within its radius. If the distribution of the placement of the nodes is known, the base stations can use the neighbor information collected by the nodes in the network to detect a significant variation of that distribution using statistical tools. If the distribution difference is larger than a predefined threshold, then a wormhole is detected.

The second detection mechanism is based on the fact that a wormhole attack modifies the distribution of the length of the shortest path between a pair of nodes affected by the wormhole link.

Using the distribution of the placement of the nodes, the base stations calculate a histogram of the lengths of the shortest paths between all possible pairs of nodes in the network. Then, they recalculate the same histogram based on the information provided by the nodes. If the difference between the two histograms is larger than a predefined threshold, then a wormhole is detected.

The approach is limited to a dense network with uniformly distributed sensors where the distribution of the nodes is known, limiting its applicability. In addition, node mobility is not supported. The presence of central base stations with unlimited resources is not suitable in ad hoc networks.

This approach does not detect open or half-open attacks. Also, if the protocol detects the presence of a wormhole attack, it is not capable of offering even an approximation of the location of the wormhole link, and therefore no countermeasures can be taken after an attack has been detected.

Lazos et al. propose a graph theory approach to prevent closed wormhole attacks in static networks (Lazos et al., 2005). Their approach is based on the use of *location-aware guard nodes* (LAGNs) equipped with self-location capability.

The approach uses the concept of *local broadcast keys* (LBKs). These are pair-wise keys shared between the 1-hop neighbors, making it possible to secure the ND process. If LBKs are used, a message encrypted with a local key on one side of the network cannot be decrypted on the other side of the network.

Nodes use the information broadcast by LAGNs to detect inconsistencies generated by the presence of wormhole attacks. In the absence of a wormhole attack, the nodes must receive just one copy of a message broadcast by a neighbor LAGN. If several copies of a message from a LAGN arrive at a node, then a replay message attack is being developed by one or several malicious nodes.

Moreover, the distance separating two LAGNs can be calculated by an intermediate node using the location information provided by each LAGN in its broadcast messages. If the node can hear both LAGNs, the distance separating them cannot be larger than 2R (where *R* is the maximum transmission range of the LAGNs). If the distance is larger than 2R, a wormhole attack is being developed.

The neighbor nodes obtain their LBKs from fractional keys securely distributed by the LAGNs. Two neighbor nodes having *w* fractional keys in common can establish an LBK, if *w* is larger than a predefined threshold. Therefore, if a node receives a message encrypted with a key that does not belong to its LBK set, then the message must have been generated by a non neighboring node on another side of the network and tunneled by a wormhole.

Even though this approach is of interest, it has several limitations. It is only applicable in static networks, and it assumes the presence of trustable LAGNs with self-location information. Those assumptions are generally hard to fulfill in ad hoc networks. Finally, since this approach is limited to providing a way of securing the neighbor discovery process, open and half-open attacks are not detected.

Maheshwari, Gao, and Das propose a distributed scheme based purely on local connectivity information (Maheshwari, Gao, and Das, 2007). This approach detects wormhole attacks by discovering *forbidden structures* introduced into the connectivity graph of the network by the presence of a wormhole link.

A forbidden structure is a sub-graph that cannot be present in a connectivity network without wormholes. These structures are detected using the notion of a *packing argument*, which stipulates that there is a limited number of nodes inside a fixed region which can be displayed without sharing edges. For example, a maximum of 5 nodes with pair-wise distances larger than 1 can be placed in a unit disk. A forbidden structure violates the packing argument.

The approach assumes that each node is capable of creating its list of *k*-hop neighbors. With this information, the node searches for a forbidden structure in its *k*-hop neighborhood, the algorithm for which has a computational complexity that depends on *k.* The authors develop simulations of the approach using a value of $k \leq 2$. Several simplified communication models (Unit-Disk model (UDG), Quasi-UDG model, and TOSSIM model) are used.

The approach yields no false alarms and a 100% detection rate when fully connected networks are simulated with the presence of a single wormhole link and a value of $k \leq 2$. The simulations do not analyze the consequences of mobility in the approach. The irregularity of antenna patterns can introduce a significant error and the computational resources necessary for the nodes to search for forbidden structures in its *k*-hop neighborhood list can be overwhelming, especially when a high value of *k* is chosen.

In order to detect open and half-open wormhole attacks, the nodes in the network must perform an analysis with values of *k* larger that 3.

Table 3-1 Summary of solutions for closed (Layer-2) wormhole attacks

Method	**Approach**	**Pros**	**Cons**
Distance Bounding solutions	***Temporal Leashes*** (Hu, Perrig, and Johnson, 2003)	Does not require a location mechanism.	Requires tight synchronization beyond the capability of existing devices.
	Time of Flight (Brands and Chaum, 1994; Capkun et al., 2003; Jakob, Srikanth, and Michalis, 2006; Singelee and Preneel, 2005). ***RSS*** (Barbeau and Robert, 2006; Meguerdichian et al., 2001; Patwari et al., 2003)	Does not require a location mechanism.	Requires additional hardware. Not secure. Easy to thwart. High error. Dependence on external variables (obstacles, environment, etc.).
Location-aware solutions	***Geographical Leashes*** (Hu, Perrig, and Johnson, 2003)	Very good solution, robust.	Requires additional hardware (GPS, loosely synchronized clocks). Expensive. Limitations associated with GPS technologies.
	Directional Antennas (Hu and Evans, 2004; Lazos et al., 2005)	Good solution.	Expensive, requires trustable nodes.
Graph and Statistical solutions	(Buttyán, Dóra, and Vajda, 2005)	Inexpensive, no additional hardware required.	Limited, does not support mobility. No wormhole location provided after detection.
	(Lazos et al., 2005)	Interesting concept of LBK.	Limited, does not support mobility, requires Central Authority.
	(Maheshwari, Gao, and Das, 2007)	Good simulation results (especially for dense networks).	Lack of results in a real deployment.

3.2 Solutions for open (Layer-3) wormhole attacks

As presented in section 2.1.2, the open wormhole attack involves at least two colluding internal nodes making honest nodes believe that they are neighbors, even when they are separated by a distance longer than the maximum distance allowed in the network.

Several approaches have been proposed to avoid the open wormhole attack. The major proposals are presented below.

3.2.1 Location-aware approaches

Wang and Wong propose the end-to-end detection of wormhole attacks in ad hoc networks using the concept of the *smallest hop count estimation between source and destination* (Wang and Wong, 2007).

In this approach, all nodes must have location-aware information and share a cryptographic authentication mechanism (pairwise secret or public keys). A destination node must respond to a RREQ with a modified RREP, including its current location. Based on this information, the source node calculates its distance to the destination node.

If a uniform distribution of nodes and a known density function are assumed, the smallest hop count separating a source and a destination node can be estimated using statistical tools. Comparing this value with the hop count of the RREP message, if the estimated smallest hop count is larger than the information obtained from the RREP message, the source can predict the presence of a wormhole link in the path.

If a wormhole attack is predicted, the source must begin a *tracing* procedure to detect the location of the wormhole link. The source forwards a tracing packet along the suspicious path. Each intermediate node receiving the message must reply with a tracing-response packet indicating its actual position, and so on, until the destination node receives the packet and sends out the response. With the location information of all the intermediate nodes, the source calculates the smallest hop count for every intermediate node. If the test fails for an intermediate node, the wormhole is located between this node and its previous hop. In this case, the path is not used.

The advantage of this approach is that, like other location-aware approaches (Hu, Perrig, and Johnson, 2003) (Wang et al., 2006), it does not require clock synchronization. The approach generates a cumulative estimation error, which means that the result can be inaccurate, especially when the distance between the source and the destination is long (several hops away). Also, it does not detect wormhole links where the distance separating two malicious nodes is short (a few hops away).

The authors assume a uniform distribution of the nodes and a known density function. Those assumptions do not regularly hold in ad hoc networks.

To conclude, this approach can detect a limited number of open and closed wormhole attacks (where the source-destination distance is not too large and the wormhole link is not too short), but cannot detect open wormhole attacks and is susceptible to the Sybil-Wormhole attack.

In Figure 3.4, the malicious nodes M_1 and M_2 develop a coordinated wormhole attack. If the malicious nodes have access to the cryptographic information of a third malicious node M_3, then they can create a Sybil node M_3 and place it in the middle of the wormhole link. In this way, the real path traversed by the packet (*S*, M_1, M_2, *D*) will be changed by the path (*S*, M_1, M_3, M_2, *D*).

When the destination node *D* performs the calculation of the distances and speeds for intermediate nodes, the wormhole link will not be detected if the colluding nodes choose the right position and time information for Sybil node M_3.

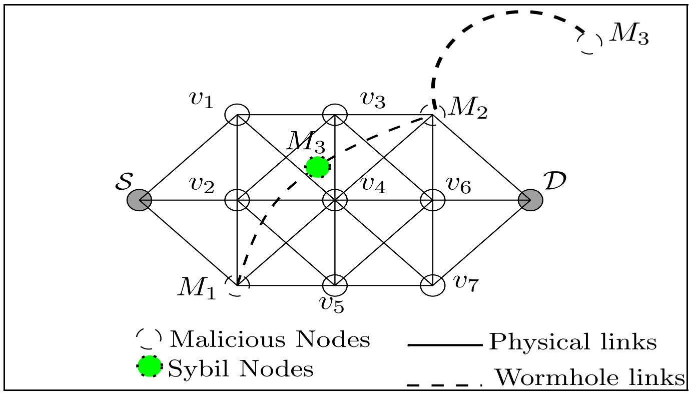


Figure 3.4 Sybil-Wormhole attack.

Wang et al. propose a modification to geographical packet leashes (Hu, Perrig, and Johnson, 2003), where instead of a hop-by-hop verification made by every single node in the path, an end-to-end verification is made by the destination node (Wang et al., 2006). This approach allows the destination node to detect open, closed, or half-open attacks.

Wang et al. assume the secrecy and authenticity of the communication between honest nodes through the use of pairwise secret keys shared between any pair of honest nodes in the network, digital signatures or a variant of TESLA (Perrig et al., 2000).

When a node *A* sends a packet toward a destination *D*, it adds to the packet the local time and position when the packet is sent $\langle t_{S\,send}, P_{S\,send}, \rangle$, besides the MAC of the packet.

Each intermediate node forwarding the packet must add two [time, position] pairs: $\langle t_{rec}, P_{rec} \rangle$ and $\langle t_{send}, P_{send} \rangle$ (the receiving and the sending time and position of the packet respectively) and a newly calculated MAC for the packet. When the destination node *D* receives the packet, it will contain the time and position information of each node in the path, as well as each MAC, ensuring that the information added by each node in the packet cannot be modified later by another node.

With the collected information, the destination can calculate the distances between intermediate nodes and their average moving speed. Let Δ be the maximum clock synchronization error, v the maximum node velocity, and σ the maximum possible location error in the network. Then, if, for an intermediate node *X*,

$$\frac{\left\| p_{X\,send} - p_{X\,rec} \right\| - \sigma}{\left\| t_{X_{Send}} - t_{X\,rec} \right\| + \Delta} > v$$

there is an incongruity in the position and/or time information given by node *X*, then this node is participating in a wormhole attack.

The disadvantage of this approach is the overhead added to the packet header with the inclusion of the pair position time and the MAC computation by each node. Also, the computation of the MAC code generates a processing load for the intermediate nodes, and, more especially, for the destination node, which will require a great deal of processing power and considerable storage capability for buffering the information coming into the packet header and the computation of the MAC codes of all intermediate nodes on a real-time basis.

In addition, as the approach does not assume the presence of trusted hardware such as tamperproof tokens, it allows colluding nodes to participate in a Sybil-Wormhole attack, circumventing the security mechanisms and remaining undetected.

Lee et al. propose another approach where an end-to-end verification is used, but here, instead of the destination node, it is the source node that performs the verification using the information in the RREP message (Lee, Jeon, and Kim, 2007).

After receiving the RREQ, the destination unicasts a RREP packet to the source using the path obtained from the RREQ message. When an intermediate node receives the RREP, it compares its actual location with the location in the RREP and forwards the packet only if the two locations coincide.

The source calculates the distance between the intermediate nodes using the location information in the RREP packet, verifying that each distance in the path is shorter than the maximum 1-hop distance allowed in the network. If any of the distances fail the test, the RREP is discarded and the nodes forming the link are put on a list of suspicious nodes.

The approach uses public key cryptography to ensure that a malicious intermediate node is not capable of modifying the location information in the packet. The destination signs the RREP packet before forwarding it, and the source confirms the signature of the destination in the packet before processing it. If the packet fails the verification, it is discarded.

The approach proposed by Lee, Jeon, and Kim (Lee, Jeon, and Kim, 2007) is able to detect open, half-open, and closed wormhole attacks, but requires each node in the network to have access to self-location information. This approach adds the danger of congestion at the source node, because of the assigned verification responsibilities, and is also susceptible to the Sybil-Wormhole attack (see Figure 3.4).

3.2.2 Statistical technique

Qian, Song, and Li propose a different approach for detecting wormhole attacks based on a statistical analysis of multipath routing protocols (Qian, Song, and Li, 2007). The authors observe that, when a wormhole attack is developed in a multipath environment, the malicious link appears with the greatest frequency in the route discovery process, because it is extremely attractive to routing requests.

The difference between the most frequent and the second most frequent links in a route discovery will be greater in the presence of a wormhole attack. This characteristic allows the detection of a link suspected of being part of a wormhole attack. The protocol does not perform any action after discovering a suspect link, because it is supposed to be part of an IDS system.

This approach does not require special hardware or changes to existing routing protocols, but is limited because it bases the wormhole attack confirmation on the assumption that the suspicious nodes will begin dropping packets right at the end of the routing process. This assumption is not accurate, because the goal of a wormhole attack can be different to generating a black/grey hole attack. It may simply want to eavesdrop on network communications. Furthermore, if the wormhole nodes wait for a time *t* after the end of the discovery route process before launching the black/grey hole attack, they will circumvent the security protocol and remain undetected.

Closed or half-open wormhole attacks cannot be detected with this approach. In the case of a closed wormhole attack, the malicious nodes will not appear in the information obtained in the multipath route discovery protocol, and no suspicious link with an abnormally high frequency will be found. In the case of a half-open wormhole attack, one of the malicious nodes will appear in most of the discovered paths, but the approach does not focus on single nodes appearing with abnormal frequency, but on frequently appearing links.

Finally, the authors present the results obtained using uniform static grid networks. It would be necessary to evaluate the result of the approach in randomly deployed networks to ensure the accuracy of the approach.

3.2.3 LiteWorp and MobiWorp

Khalil, Saurabh, and Shroff propose an approach called LiteWorp (Khalil, Saurabh, and Shroff, 2005). This protocol seeks to detect wormhole attacks in static networks. The protocol assumes that there are no compromised nodes at network deployment, and that an attacker will require at least a time T_{CT} (compromise threshold time) for compromising any node. In this interval of time, each node builds a secure list of 1-hop neighbor nodes R and shares this list securely with its neighbors.

At the end of T_{CT}, each node has a secure list of its 1-hop and 2-hop neighbors. With this information, a phase of local monitoring begins. The concept of the guard node is introduced. For a node *G* to be the guard node of the link formed by nodes *A* and *B*, *G* must be a neighbor of both, and monitors all the packets exchanged through the link *A-B*.

As seen in Figure 3.5, G is the guard node from link $A-B$ in the path $S-A-B-D$. The information exchanged between A and B is monitored by node G, which can discover whether or not abnormal behavior is developed by any of them.

In LiteWorp, additional information containing the address of the previous hop is added to each packet forwarded by a node. If node *A* wants to forward a packet received from node S (see Figure 3.5), it must add the address of node S as the previous hop of the message. In this way, node *G* can confirm that the packet was really received by node *A* from node *S*.

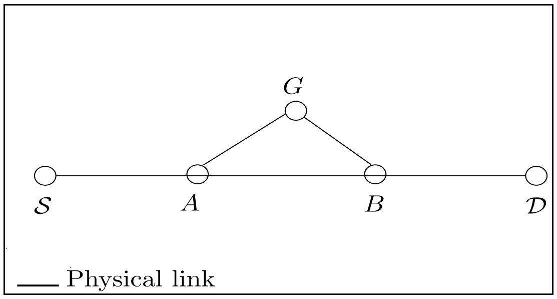


Figure 3.5 Node Guard concept.

In Figure 3.6, the malicious nodes M_1 and M_2 search to develop a wormhole attack. M_2 receives a control message (RREQ, HELLO, etc.) from M_1 directed from S to D.

After receiving the message, M_2 has two options: (1) identify node M_1 as the previous hop of the message; or (2) identify one of its own neighbors (v_4 in this case) as the previous hop. In the first case, the control message will be rejected by all the neighbors of M_2, because they know that node M_1 is not a valid 1-hop from node M_2.

In the second case, the packet will be rejected because the guard node from the link $v_4 - M_2$ (v_3 or v_6) knows that the message was not forwarded by v_4.

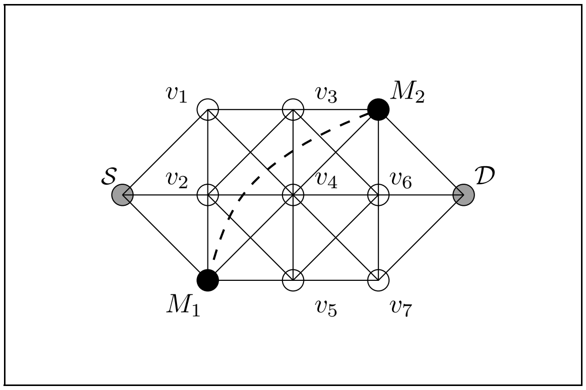


Figure 3.6 Wormhole attack.

This is an interesting approach, but its application is restricted to static networks, because the protocol does not allow modification of the neighbor list made during the initial setup time. Static routing would be a less complicated solution in static networks.

Khalil, Bagchi, and Shroff propose an approach called MobiWorp (Khalil, Bagchi, and Shroff, 2008), which seeks to overcome the mobility limitation of LiteWorp (Khalil, Saurabh, and Shroff, 2005), conserving its original structure. This protocol uses the same approach with guard nodes watching over the various links in the network.

To avoid the limitations caused by a static list of neighbors made at the time of deployment of the network, a central authority (CA) is assumed. When a node needs to change its position in the network, it must know its actual position, its final position, and its average speed of travel before moving. With this information, the CA will provide the node with an Authentication Neighbor Update Message (UNAM). The node must present this authenticated message to its new neighbors. They will add the mobile node to their neighbors list, allowing them to exchange packets with the new node.

This approach is aimed at overcoming the limitations of the original protocol (Khalil, Saurabh, and Shroff, 2005), but adds some requirements that can be hard of fulfill in ad hoc network environments. The need for a CA runs counter to the essence of the ad hoc network.

Also, the assumption that each node has previous knowledge of its present and future position, aside from the average travel speed, is unrealistic and difficult to apply in ad hoc networks. This requirement would demand that every movement in the network be previously planned. In addition, if all nodes have self-location information, then other, more efficient approaches, such as geographical packet leashes, should be used.

3.2.4 Trust-based solutions

Pirzada and Mcdonald propose a solution to encapsulation channel-based wormhole attacks using a trust-based model (Pirzada and Mcdonald, 2006). Each node in the network must build a trust table representing its level of trust with respect to one another node in the network. The value of the trust level in a specific node is the result of the combination of two singles variables, P_p and P_A, where P_p indicates the presence (with a value of 1) or the absence (with a value of 0) of a wormhole attack involving the node in question. P_A indicates the willingness of a node to participate in the routing process and seeks to detect selfish nodes.

A node willing to forward a packet to its neighbors buffers the packet header before the transmission. After forwarding the packet, the node switches its wireless interface to promiscuous mode, and overhears the channel during a predefined time interval TUI (Trust Update Interval), seeking to detect the retransmission of the message made by its neighbors.

Before the TUI timeout, if the node overhears the retransmission of the message made by one of its neighbors, the node compares the packet header of the retransmitted packet with the copy buffered initially. If no abnormal change has been made to the packet header, the node concludes that its neighbor is not a malicious node participating in a wormhole attack. The value of P_p for that node is set to false and P_A is increased. Otherwise, if the packet header has been abnormally modified, the value of P_p for that node is set to true and the node is declared as participating in a wormhole attack.

If, at the end of the TUI timeout, the node does not overhear the retransmission of the packet, its neighbor is declared to be selfish and its P_A value is decreased.

Links using nodes with a low P_A can potentially be used in a specific forwarding path if there is no other, more trustable link available. Otherwise, nodes cataloged as malicious cannot be included in any path and must be isolated from the network.

This proposal is very limited, because it can only be applied to wormholes based on encapsulation channels where malicious nodes modify a received message to tunnel it using the resources of the network. As mentioned before, this kind of wormhole attack has a very limited application spectrum, mainly due to the delay added to the packet in the encapsulation between the two malicious nodes. This delay allows the application of simple timing techniques to detect it (Xu and Boppana, 2007).

In the case of out-of-band wormhole attacks, the attacker can easily circumvent the security measures, forwarding to the network a valid copy of the message after sending the copy to the colluding node using the out-of-band channel. In this way, the victim will hear the transmission of the malicious node and will conclude (erroneously) that the attacker is an honest node.

Also, this approach allows malicious nodes to develop blackmail attacks, where legitimate nodes are qualified as malicious and isolated from the network.

No analysis of the number of false positives and false negatives generated by the approach has been performed. Also, the method for selecting the value of TUI is not presented. The choice of this value is critical, because it can generate a significant number of false positives and/or false negatives if it is not correctly chosen.

3.2.5 Network visualization

Weichao and Bharat propose a centralized approach using network visualization to detect wormhole attacks in static sensor networks (Weichao and Bharat, 2004). The existence of a centralized authority is assumed. This station is called the controller and does not have resource limitations.

In this proposal, each node must develop an ND process and calculate the distance to their neighbors using RSS. Each node sends a list with the calculated distance to each of its neighbors to the controller. With this information, the controller uses the Multidimensional Scaling (MDS) technique (Davidson, 1983) to reconstruct a layout of the network. The approach uses a smoothing function to minimize the impact of the error in the reconstructed network.

Since the assumed network is made up of static sensors deployed in a plane field and making up a connected graph, the reconstructed surface will be relatively flat in the absence of wormhole attacks. A wormhole attack introduces abnormal values into the connectivity graph, generating a bent surface around the two ends of the wormhole link.

In spite of seeming interesting, the approach has several major limitations. The assumption of a dense network deployed on a flat surface is very limiting. The presence of irregularities in the surface where the nodes are deployed, or a low sensor density, generates an error that makes the proposal inapplicable.

The requirement of a central authority is not suitable for ad hoc networks. Moreover, in the case of closed and half-open wormhole attacks, the approach can generate a blackmail attack after accusing nodes that establish network relationships through the wormhole link of being malicious.

Finally, the approach does not support node mobility, and the simulation does not show the performance of the protocol in the case of a large-scale network.

Wang and Lu propose a modification of the above solution to support node mobility (Wang and Lu, 2007). To handle the increment of traffic required to reconstruct the network structure in the presence of node mobility, the authors propose the use of an incremental MDS instead of the original MDS.

The use of the incremental MDS reduces the computational complexity of the reconstruction of a network with *n* nodes from $\mathrm{O}(n^3)$, when the original MDS is used, to $\mathrm{O}(n^2)$. In addition, the authors propose a system interface permitting a user to monitor the network and detect wormhole attacks by visualizing the resulting reconstructed network.

To reduce the complexity of the topology in the system interface, the authors add a self-adaptive tool which permits the system interface to adjust the resolution of the network obtained, thereby automatically redefining the sample points.

Table 3-2 Summary of solutions for open (Layer-3) wormhole attacks

Method	Approach	Pros	Cons
Location-aware solutions	***Geographical Information*** (Lee, Jeon, and Kim, 2007; Wang and Wong, 2007)	Robust, can be trusted.	Generates overhead in the packet header and a high processing overload for intermediate nodes. Requires GPS. Susceptible to the Sybil-Wormhole attack.
	Smallest hop count estimation. (Wang and Wong, 2007)	No clock synchronization required	Cumulative estimation error. Does not detect short wormholes. Does not perform well when the distance between the source and the destination is long. Requires additional hardware.
Statistical solutions	(Qian, Song, and Li, 2007)	No additional hardware required. No protocol modification required.	Assumptions not accurate. Not results in randomly deployed networks.
LiteWorp and MobiWorp	(Khalil, Bagchi, and Shroff, 2008; Khalil, Saurabh, and Shroff, 2005)	Relatively simple.	Limited, does not support mobility (LiteWorp) or requires a central authority and location information (MobiWorp).
Trust-based solutions	(Pirzada and Mcdonald, 2006)	No additional hardware required	Limited to encapsulation channels. Easy to circumvent. Susceptible to blackmail attack.
Network visualization	(Wang and Lu, 2007; Weichao and Bharat, 2004)	Innovative.	Centralized solution.

3.3 General summary of existing wormhole attack solutions

Sections 3.1 and 3.2 have presented the most important existing approaches for detecting and/or preventing open and closed wormhole attacks. Since half-open wormhole attacks are a combination of both of these, there are no specific solutions for this threat, but several solutions used either for open or closed attacks, or both, are also effective for detecting and/or preventing half-open attacks.

In Table 3-3, we present a general summary of the various approaches and their applicability to detecting and/or preventing open, half-open, and/or closed wormhole attacks.

Table 3-3 General summary of the various approaches

Method	**Approach**	**Open**	**Half-open**	**Closed**
Distance Bounding solutions	***Temporal Leashes*** (Hu, Perrig, and Johnson, 2003)			X
	Time of Flight (Brands and Chaum, 1994; Capkun et al., 2003; Jakob, Srikanth, and Michalis, 2006; Singelee and Preneel, 2005)			X
	RSS (Barbeau and Robert, 2006; Meguerdichian et al., 2001; Patwari et al., 2003)			X
Location-aware solutions	***Geographical Leashes*** (Hu, Perrig, and Johnson, 2003)			X
	Geographical Information (Lee, Jeon, and Kim, 2007; Wang and Wong, 2007)	X	X	X
	Smallest hop count estimation. (Wang and Wong, 2007)	X	X	X
	Directional Antennas (Hu and Evans, 2004; Lazos et al., 2005)			X
Graph and Statistical solutions	(Buttyán, Dóra, and Vajda, 2005)			X
	(Lazos et al., 2005)			X
	(Maheshwari, Gao, and Das, 2007)			X
	Qian, Song et Li, 2007)	X		
	Network visualization (Wang and Lu, 2007; Weichao and Bharat, 2004)	X		
Trust-based solutions	(Pirzada and Mcdonald, 2006)	X		
LiteWorp and MobiWorp	(Khalil, Bagchi, and Shroff, 2008; Khalil, Saurabh, and Shroff, 2005)	X		

Equation Section (Next)

CHAPTER 4

WIM-DSR

This chapter presents Witness Integration Multipath DSR (WIM-DSR), which is a protocol offering a new mechanism for detecting and preventing open wormhole attacks in ad hoc networks using the information provided by a variant of the Split Multipath Routing (SMR) protocol (Lee and Gerla, 2001).

WIM-DSR finds suspicious patterns related to the presence of wormholes in the network without any hardware requirement or protocol modification.

The rest of the chapter is organized as follows. Section 4.1 presents an introduction to Multipath Source Routing protocols. Section 4.2 introduces the notion of strong and weak wormhole models. Section 4.3 presents the assumptions sustaining our proposal. Section 4.4 explains the cornerstone concept of Edge witnesses. Finally, the WIM-DSR route discovery process is presented in section 4.5 and the analysis of this approach is given in section 4.6.

4.1 Multipath Source Routing

The Dynamic Source Routing (DSR) protocol (Johnson et al., 1996) is an on-demand source routing protocol for mobile ad hoc networks allowing a node in the ad hoc network to dynamically discover and maintain a valid route to any other node in the network.

When a source node wants to set up a communication with any destination node and no valid route is found in its route cache, the sender may attempt to discover one using the *route discovery protocol*. The source node broadcasts a RREQ message in the network, identifying the target node. As the RREQ message is forwarded by intermediate nodes, they insert their respective addresses in the packet header. When the destination receives the RREQ, it replies to the source with a unicast RREP message, reversing the route found in the route record from the RREQ header.

The original DSR protocol allows the discovery of a unique path linking two nodes in a network. Since mobility is a primary characteristic of ad hoc networks, paths are constantly being broken. Nodes are forced to restart the route discovery process, bringing down the performance of the network by the bandwidth-consuming broadcast associated with route discovery.

Using multiple paths can improve the quality of service, as well as the fault resilience of a network. Several adaptations of the DSR protocol seeking to discover disjoint multipaths between a source and a destination have been proposed (Lee and Gerla, 2001; Lijun, Ning, and Xiangfang, 2007; Xuefei and Cuthbert, 2004; Zafar et al., 2007).

The Split Multipath Routing protocol (SMR) proposed by Lee and Gerla seeks to build maximally disjoint paths between any pair of nodes in an ad hoc network (Lee and Gerla, 2001). When a node wants to set up a communication with another node and no valid route is found in its route cache, the source node floods the network with a RREQ message.

As the message is broadcast in the network, several copies traversing different routes arrive at the destination. The destination node selects several edge-disjoint paths and sends back RREP messages to the source along each of the selected paths.

Unlike DSR, where duplicate RREQs arriving at intermediate nodes are dropped, SMR allows intermediate nodes to rebroadcast a duplicate RREQ message if the packet traverses a different incoming link from the link from which the first RREQ was received, and if the hop count of the incoming packet is no larger than the hop count of the first RREQ received. In this way, SMR can select edge-disjoint paths from the routes available.

Lijun, Ning, and Xiangfang propose a modification to SMR (Lijun, Ning, and Xiangfang, 2007). The modified protocol allows intermediate nodes to forward a repeated copy of an RREQ message, as long as its hop count is no larger than the hop counts of copies already received, even if the packets share the same incoming link. With this modification, the destination node will receive a larger number of RREQ messages and will have more available paths linking it to the source node, even when those resulting paths are not edge or node disjoints. This modified protocol will be used in our approach.

4.2 Strong and Weak Open Wormholes

Two new types of open wormholes are considered in our approach: *weak* and *strong*. In the former, the tunnel connects two malicious nodes, M_1 and M_2, located at *d* and (*d* + 1) hops from the source respectively (see Figure 4.1 (a)). In the latter, the two nodes are located at *d* and at least (*d* + 2) hops from the source respectively (see Figure 4.1 (b)).

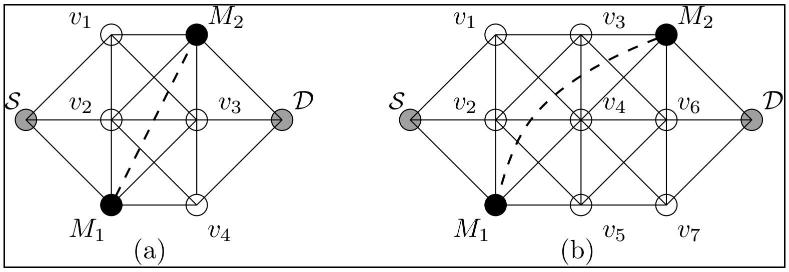


Figure 4.1 Open wormholes: (a) weak; and (b) strong.

The specific objective of the attacker in Figure 4.1 (a) would be to substantially increase the possibility of having access to the data exchanged by nodes *S* and *D*.

In the absence of the wormhole link $M_1 - M_2$ and if the modification of SMR proposed by (Lijun, Ning, and Xiangfang, 2007) is used, the attacker is present with any of its two controlled nodes (M_1 and M_2) in 4 of 7 paths (57.1%) (see Table 4-1). If the wormhole link $M_1 - M_2$ is activated, then the path $S - M_1 - M_2 - D$ is added to Table 4-1 and the attacker will have five over eight opportunities of having at least one node in the chosen path (62.5%), raising his chances by only 5.4%. But, most importantly, the wormhole path does not have a shorter hop count than the other paths, and therefore having a weak wormhole in a network does not give a substantial advantage to an attacker.

Table 4-1 Paths found by the routing
protocol (weak wormhole attack)

No.	**Path**
1	$S - v_1 - M_2 - D$
2	$S - v_1 - v_3 - D$
3	$S - v_2 - M_2 - D$
4	$S - v_2 - v_3 - D$
5	$S - v_2 - v_4 - D$
6	$S - M_1 - v_3 - D$
7	$S - M_1 - v_4 - D$

If weak open wormholes do not necessarily represent shorter paths to the destinations, strong open wormholes do. In fact, strong open wormholes represent the real threat and would always be chosen by any routing protocol choosing the shortest paths. The paths obtained by node D (see Figure 4.1 (b)) in the absence of the strong wormhole link $M_1 - M_2$ are presented in Table 4-2.

The attacker is present with at least one of his two malicious nodes in 9 of 17 paths (52.9%). If the wormhole link $M_1 - M_2$ is activated, the links $S - M_1 - M_2 - D$ and $S - M_1 - M_2 - v_6 - D$ are added to Table 4-2, but the paths 1, 3, 6, 8 and 13 would be erased from Table 4-2, because, when the RREQs traversing legal links arrive at node M_2, they will have one hop more in their hop counter than the $S-M_1-M_2$ RREQ already received and, according to the routing protocol, the newly arrived RREQs must be dropped by node M_2.

Table 4-2 Paths found by the routing protocol (strong wormhole attack)

No.	Path
1	$S-v_1-v_3-M_2-D$
2	$S-v_1-v_3-v_6-D$
3	$S-v_1-v_4-M_2-D$
4	$S-v_1-v_4-v_6-D$
5	$S-v_1-v_4-v_7-D$
6	$S-v_2-v_3-M_2-D$
7	$S-v_2-v_3-v_6-D$
8	$S-v_2-v_4-M_2-D$
9	$S-v_2-v_4-v_6-D$
10	$S-v_2-v_4-v_7-D$
11	$S-v_2-v_5-v_6-D$
12	$S-v_2-v_5-v_7-D$
13	$S-M_1-v_4-M_2-D$
14	$S-M_1-v_4-v_6-D$
15	$S-M_1-v_4-v_7-D$
16	$S-M_1-v_5-v_6-D$
17	$S-M_1-v_5-v_7-D$

The attacker's presence in the remaining paths as a percentage is now 50% (7 out of 14 paths). Even if the attacker decreases his presence in the paths by 2.9%, the truly important point is that the shortest path in the whole set is the path containing the malicious link $M_1 - M_2$. In this way, the attacker succeeds in short-circuiting the network and is sure to be part of the chosen path.

Strong wormholes give an attacker a real advantage in the network, becoming a real threat to security. Because of this, our approach focuses on detecting and avoiding those attacks.

4.3 Assumptions and Treat Model

The main objective of the WIM-DSR protocol proposed in this document is to gather information during the route discovery phase and use this information to find possible anomalies generated by open wormhole attacks. Several assumptions are made to define the operation of WIM-DSR in a wireless ad hoc network:

Assumption I The number of malicious nodes coordinating a wormhole attack is restricted to two colluding nodes. Even when several pairs of nodes can develop a wormhole attack simultaneously in the network at a specific time, each of those attacks will be limited to an *Active-2-2* attack.

Assumption II Each legitimate node can carry out its neighbor discovery process securely (using temporal leashes, RSS, RTT, etc.).

Assumption III Each legitimate node has a unique cryptographic identity, which can be used to sign control messages. All legitimate nodes can verify these signatures.

Assumption IV The cryptographic identity is implemented in a secure tamperproof token (e.g. a smart card).

The second assumption has two important benefits: the nodes know the identity of their neighbors, and a malicious node cannot use another remote malicious node as an oracle, limiting its chance of performing Sybil attacks (Newsome et al., 2004)

The last two assumptions limit the capacity of a compromised node to impersonate another compromised node by using multiple wireless interfaces and multiple cryptographic identities, locally limiting the chance of Sybil attacks. Even where these assumptions are not stated in the preceding approaches, most of those approaches rely on them.

Finally, the first assumption ensures that a wormhole attack cannot be coordinated by more than two nodes, because, if that occurs, the malicious nodes can find a way of circumventing our approach. This possibility will be evaluated later.

4.4 Edge Witnesses

The WIM-DSR protocol determines whether or not the information gathered by the modified SMR protocol during the route discovery phase reveals anomalies which are typical of wormhole attacks.

We introduce some notations here. Let *G(r) = (N,E(r))* be the geometric graph defined by the set of nodes *N*, and *r* the transmission range of the set of nodes *N* (Penrose 2003). In this graph, two nodes are connected if and only if their distance is less than or equal to *r*. This graph represents the topology of the network. Finally, let $L_G(X,d)$ define the set of nodes in *G* at *d* hops of the node *X* (see Figure 4.3 (b)). This can be obtained by a breadth-first search from *X*.

An open wormhole attack between two malicious nodes should simply add an edge between two distant nodes in *G(r)*. This new edge usually shortens the distance between some pair of nodes. In Figure 4.1, the edge (M_1, M_2) is added to the underlying graph representing the topology of the network. The resulting graph is denoted $G^w(r)$.

During route discovery, the destination receives multiple copies of a RREQ message, each of which represents a distinct path between the source and the destination. Thus, the destination can reconstruct a subset of *G(r)* (or $G^{w}(r)$, if there is a wormhole attack). This subset is denoted $G_{RREQ}(r)$.

The WIM-DSR protocol presented in the next section relies on the following concept of edge witnesses:

Definition 1 Let $e=(v_i,v_j)$ be an edge of a path from *S* to *D* in $G_{RREQ}(r)$. A node v_w is a weakly forward witness for *e* iff

i. $(\exists d)\,[\,v_i, v_w \in L_{G_{RREQ}}(S,d)\,]$;

ii. edge (v_w, v_j) is in $G_{RREQ}(r)$.

Definition 2 Let $e_1=(v_i,v_j)$ and $e_2=(v_j,v_k)$ be two consecutive edges of a path from *S* to *D* in $G_{RREQ}(r)$. A node v_w is a strong witness for the subchain $((v_i,v_j,v_k))$ iff

i. $(\exists d)\,[\,v_j, v_w \in L_{G_{RREQ}}(S,d)\,]$;

ii. edges (v_i, v_w) and (v_w, v_k) are in $G_{RREQ}(r)$.

The second definition gives a strong witness and should be preferred. It simply states that there are two distinct paths joining two nodes in the network. Since there are only two malicious nodes (Assumption 1), these strongly witnessed nodes cannot form a wormhole.

Intuitively, an edge witness gives more evidence that two nodes are really neighbors and are not part of a wormhole. For example, in Figure 4.1 (b), M_1 is the only node at 1 hop from *S* claiming to be adjacent to M_2. Hence, no witness can be found for the edge (M_1, M_2).

The aim of WIM-DSR is to find fully witnessed paths, i.e. paths with only witnessed edges between the source and the destination. The algorithm constructs the fully forward witnessed path inductively. Assuming that the source S is not compromised, it proceeds forward hop by hop, constructing the set of nodes $L_{G_{RREQ}}(S,i)$ for $i \geq 1$.

Unfortunately, such a forward path does not always exist. In such a case, assuming that the destination *D* is not compromised, it proceeds backward hop by hop. Definitions 1 and 2 have to be adapted accordingly to define backward witnesses.

A fully witnessed path should not contain any open wormholes. Strongly witnessed paths should be preferred. However, weakly witnessed paths should also be considered, since the strongly witnessed condition is very restrictive and can generate numerous false positive alarms.

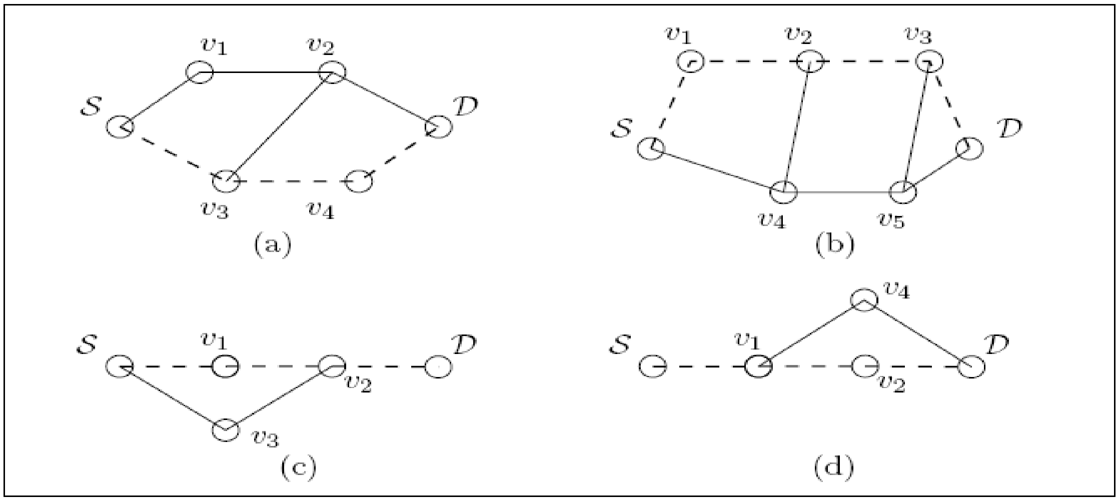


Figure 4.2 Witnessed paths: (a) weakly backward; (b) weakly forward; (c) strongly forward; (d) strongly backward.

Hence, in Figure 4.2 (a), node v_2 is a backward witness for the edge (v_3, v_4). Similarly, in Figure 4.2 (b), nodes v_4 and v_5 are weakly forward witnesses for the edges (v_1, v_2) and

(v_2, v_3) respectively. In Figure 4.2 (c), node v_3 is a strongly forward witness for (S, v_1, v_2). Finally, in Figure 4.2 (d), node v_4 is a strongly backward witness for (v_1, v_2, D).

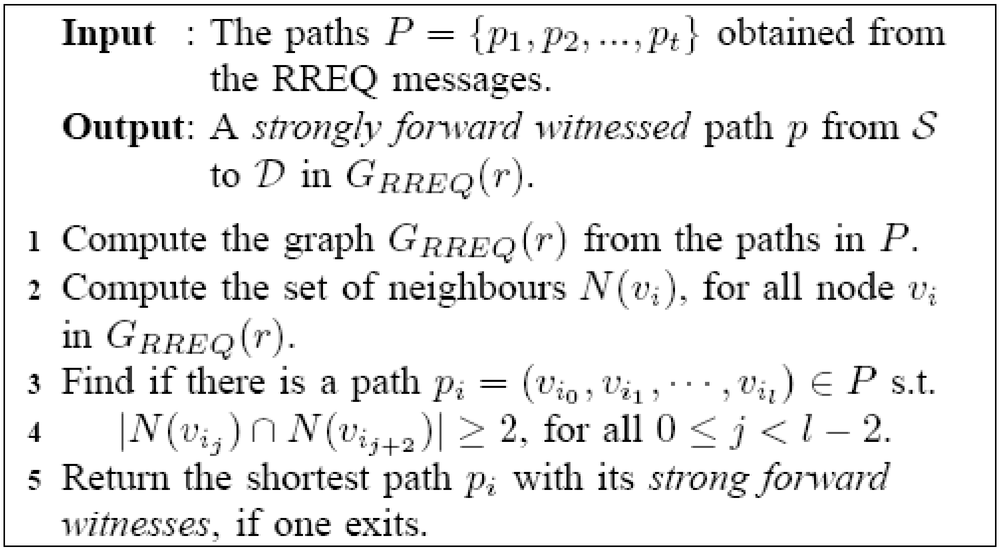
4.5 WIM-DSR Route Discovery

When a source *S* wants to discover a valid route to a destination *D*, it broadcasts a RREQ message to the network. Each intermediate node rebroadcasts copies of the RREQ message, as long as their hop counts are no larger than the hop counts of copies already received. Since this is a source routing protocol, the nodes add their identification to the RREQ messages. Thus, several copies of the original RREQ message should arrive at *D*.

With all the RREQ messages received within a given time interval, destination *D* is able to build a partial representation of the network topology. This is the graph $G_{RREQ}(r)$. The next step is to determine whether or not there exists a fully witnessed path $p = (S = v_{i0}, v_{i1}, \ldots\ldots, v_{il-1}, v_{il} = D)$ in this graph s.t.

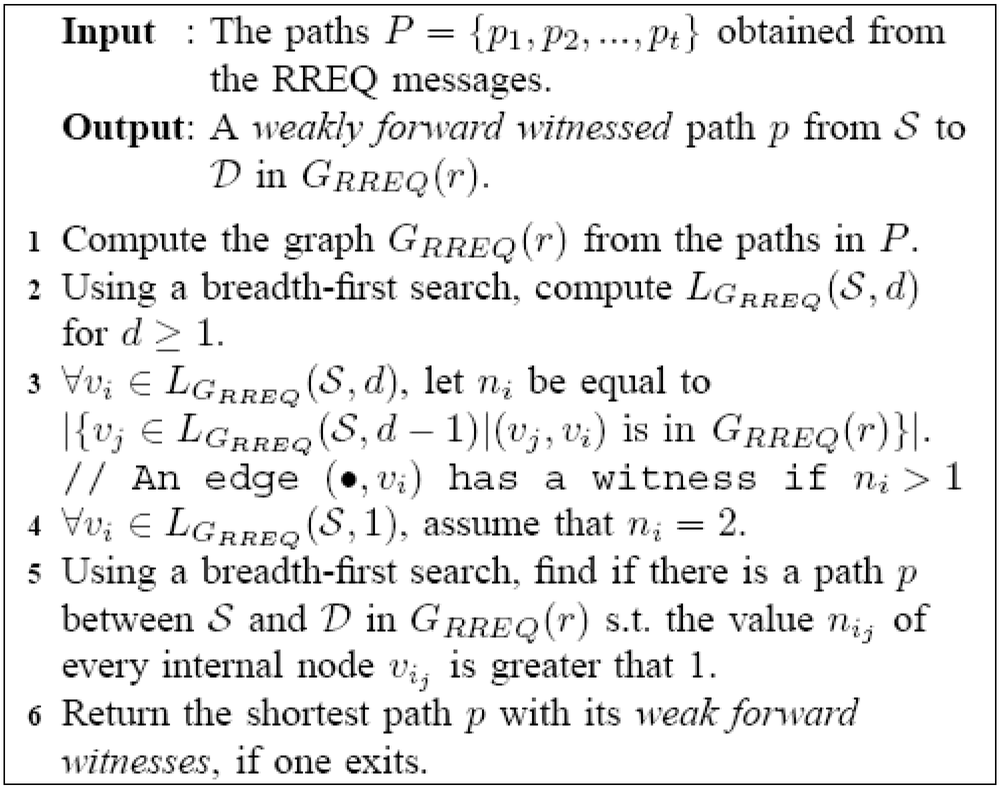
- ($\forall k$ s.t. 0≤k < l− 2) [$\exists$ *strongly forward witness* w_k for ($v_{ik}, v_{ik+1}, v_{ik+2}$)], or
- ($\forall k$ s.t. 1< k ≤l− 1) [$\exists$ *weakly forward witness* w_k for (v_{ik}, v_{ik+1})].

The first alternative is presented in **Algorithm 4-1** and finds strongly forward witnessed paths. The second alternative is presented in **Algorithm 4-2** and finds weakly forward witnessed paths. Both algorithms can be easily adapted to determine whether fully backward witnessed paths exist or not.



```
Input : The paths $P = \{p_1, p_2, ..., p_t\}$ obtained from
        the RREQ messages.
Output: A *strongly forward witnessed* path $p$ from $\mathcal{S}$
        to $\mathcal{D}$ in $G_{RREQ}(r)$.
Compute the graph $G_{RREQ}(r)$ from the paths in $P$.
Compute the set of neighbours $N(v_i)$, for all node $v_i$
in $G_{RREQ}(r)$.
Find if there is a path $p_i = (v_{i_0}, v_{i_1}, \cdots, v_{i_l}) \in P$ s.t.
    $|N(v_{i_j}) \cap N(v_{i_{j+2}})| \geq 2$, for all $0 \leq j < l - 2$.
Return the shortest path $p_i$ with its *strong forward
witnesses*, if one exits.
```

Algorithm 4-1 Strongly forward witnessed path selection algorithm.



```
Input : The paths $P = \{p_1, p_2, ..., p_t\}$ obtained from
        the RREQ messages.
Output: A *weakly forward witnessed* path $p$ from $\mathcal{S}$ to
        $\mathcal{D}$ in $G_{RREQ}(r)$.
Compute the graph $G_{RREQ}(r)$ from the paths in $P$.
Using a breadth-first search, compute $L_{G_{RREQ}}(\mathcal{S}, d)$
for $d \geq 1$.
$\forall v_i \in L_{G_{RREQ}}(\mathcal{S}, d)$, let $n_i$ be equal to
$|\{v_j \in L_{G_{RREQ}}(\mathcal{S}, d-1) | (v_j, v_i) \text{ is in } G_{RREQ}(r)\}|$.
// An edge $(\bullet, v_i)$ has a witness if $n_i > 1$
$\forall v_i \in L_{G_{RREQ}}(\mathcal{S}, 1)$, assume that $n_i = 2$.
Using a breadth-first search, find if there is a path $p$
between $\mathcal{S}$ and $\mathcal{D}$ in $G_{RREQ}(r)$ s.t. the value $n_{i_j}$ of
every internal node $v_{i_j}$ is greater that 1.
Return the shortest path $p$ with its *weak forward
witnesses*, if one exits.
```

Algorithm 4-2 Weakly forward witnessed path selection algorithm.

Once a fully witnessed path has been found, the destination signs its RREP message and broadcasts it towards the source.

For a strongly witnessed path, the destination broadcasts a unique RREP message. This message is rebroadcast by all the nodes of the path. The other nodes simply overhear it. This allows each witness to receive the RREP message from at least two sources. For a path of length *l*, only *l*-1 RREP messages are sent overall.

For a weakly witnessed path, the destination unicasts a signed RREP message along the path. Moreover, for each witness, the destination also unicasts a signed confirmation RREP message along a path going through that witness. For a path of length *l*, $l-1$ messages are sent by the nodes in $L_{G_{RREQ}}(S,i)$, $1 \leq i < l$. Therefore, $(l-1)^2$ messages are sent in total.

Strongly witnessed paths should be preferred. The answering process is more efficient. Moreover, these paths offer some alternatives. The subchain (v_{i-1}, w_i, v_{i+1}) witnessed by v_i can replace (v_{i-1}, v, v_{i+1}) witnessed by w_i.

4.6 Analysis

The path selection algorithms used by WIM-DSR cannot discover a weak open wormhole attack. In Figure 4.1 (a), the malicious nodes M_1 and M_2 have established a wormhole between them. The algorithms should find two strongly (v_1 and v_2), two weakly forward (v_1 and v_2), and two strongly backward (v_3 and v_4) witnesses for this wormhole.

Although this represents false negative detection for WIM-DSR, it does not increase the security risk significantly. As seen previously, even without this tunnel between M_1 and M_2, they would have belonged to four of the seven shortest paths discovered by WIM-DSR (see Table 4-1). Thus, the weak open wormhole just adds one new shortest path between the source and the destination.

The real gain for the malicious nodes is the strong open wormhole attack (see Figure 4.1 (b)). In such a case, the malicious node would be selected by any protocol having the shortest path as the selection measure. Such a wormhole represents a shortcut in the network.

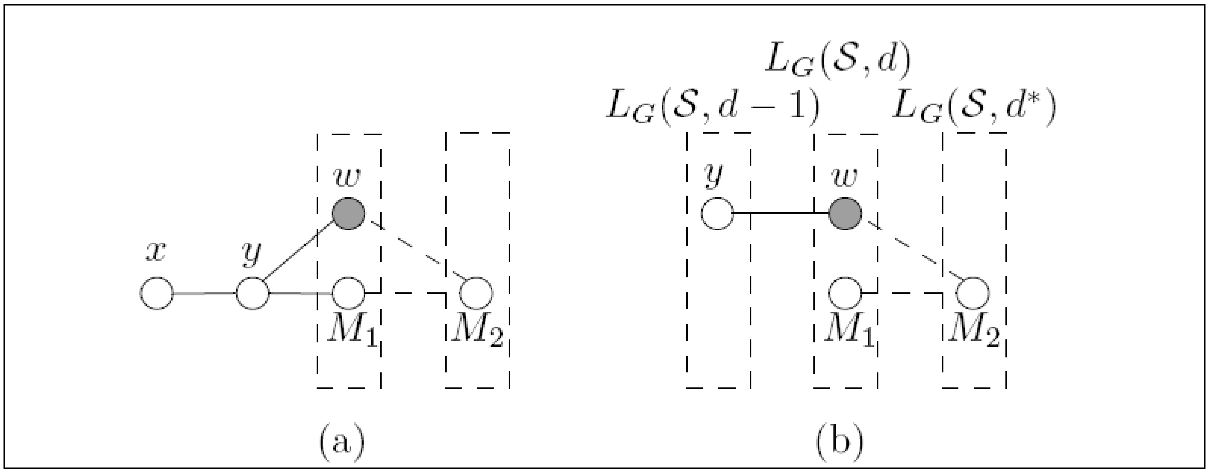


Figure 4.3 Witness: (a) strong; (b) weak forward, $d^* > d+1$**.**

The effectiveness of WIM-DSR in detecting open wormhole attacks is proven in the following lemmas, which show that the path selection algorithms cannot find false witnesses for strong open wormholes.

Lemma 4.1 Let p be a path containing a strong open wormhole. **Algorithm 4-1** cannot find any false strong witnesses for the wormhole connecting the two malicious nodes without being detected.

Proof Let (x, y, M_1, M_2) be the subchain of p containing the strong open wormhole between the only malicious nodes M_1 and M_2 (Assumption I). The simplest case, where y = S, is omitted. By definition, there is no strong witness connecting y and M_2. Thus, M_1 with the help of M_2 would have to forge a fake RREQ message including the partial path (· · · , y, w, M_2) to force **Algorithm 4-1** to find the fake witness w.

First, suppose that y receives the RREP message from M_1 (see Figure 4.3 (a)). Either w is a legitimate neighbor of y or it is not. In the former case, y accepts the message and rebroadcasts it. Then, w verifies the signature, rejects the message and broadcasts a signed warning message. It is important here that M_1 cannot impersonate M_2 for w during the neighbor discovery process (Assumptions II to IV). In the latter case, y verifies the signature, rejects the message, and broadcasts a warning message.

Now suppose that M_1 tries to impersonate y and unicasts the RREP message directly to x. When x rebroadcasts the message, y detects an anomaly, verifies the signature, and broadcasts a warning message. This also applies if M_1 tries to connect to any node in the path between S and M_1. Thus, any fake witness can be fought back from a legitimate node. ■

Lemma 4.2 Let p be a path containing a strong open wormhole. **Algorithm 4-2** cannot find any forward witness for the wormhole connecting the two malicious nodes without being detected.

Proof Let (M_1, M_2) be the subchain of p containing the strong open wormhole between the only malicious nodes M_1 and M_2 (Assumption I) and let (· · · , M_2 ,w, y, · · ·) be the confirmation RREP message for the d-level witness, i.e. $w \in L_{G_{RREQ}}(S,d)$, and any $y \in L_{G_{RREQ}}(S,d-1)$ (see Figure 4.3 (b)). By definition, there is no weakly forward witness connecting y and M_2. Thus, M_1 with the help of M_2 would have to forge a fake RREQ message including the partial path (· · · , y,w, M_2) to force **Algorithm 4-2** to find the fake witness w.

First suppose that w receives the confirmation RREP message from M_2 through M_1 (see Figure 4.3 (b)). w then verifies the signature, rejects the message, and broadcasts a signed warning message. It is important here that M_1 cannot impersonate M_2 for w during the neighbor discovery process (Assumptions II to IV).

Now suppose that M_1 tries to impersonate w and unicasts the confirmation RREP message directly to y. This can be discarded either by extending Assumption II to this phase or by asking y to unicast an acknowledgement to w, if w is the d-level witness. The latter option seems more appropriate, since it avoids having stronger assumptions. Thus, any fake witness can be fought back from a legitimate node. ■

For efficiency, an intermediate node validates the signature of a message only if it claims falsely that the node is connected to another node, or if it is a warning message. Thus, if there is no attack, the intermediate nodes do not have to do any cryptographic processing. This is a major improvement over several previous solutions, such as that proposed by Wang et al. (Wang et al., 2006).

Finally, the malicious nodes M_1 and M_2 can cause denials of service by altering the information provided during the source routing protocol. They can add or remove nodes in the messages. This can be prevented by secure source routing protocols, like Ariadne (Hu, Perrig, and Johnson, 2005) or SECTOR (Capkun et al., 2003). However, these protocols are quite demanding, and do not prevent wormhole attacks.

CHAPTER 5

SIMULATION RESULTS

In a network security system, false negatives and false positives are used to describe possible errors developed in an attack detection process. In the first case, the term *false negative* describes the error that occurs when the security system fails to detect an active attacker assailing the network. In the second case, the term *false positive* refers to an error which occurs when the security system in a network erroneously concludes that a legitimate node forms part of a network attack.

In Figure 5.1, a false negative error is generated if an existing network security system concludes that the link connecting nodes M_1 and M_2 is legitimate.

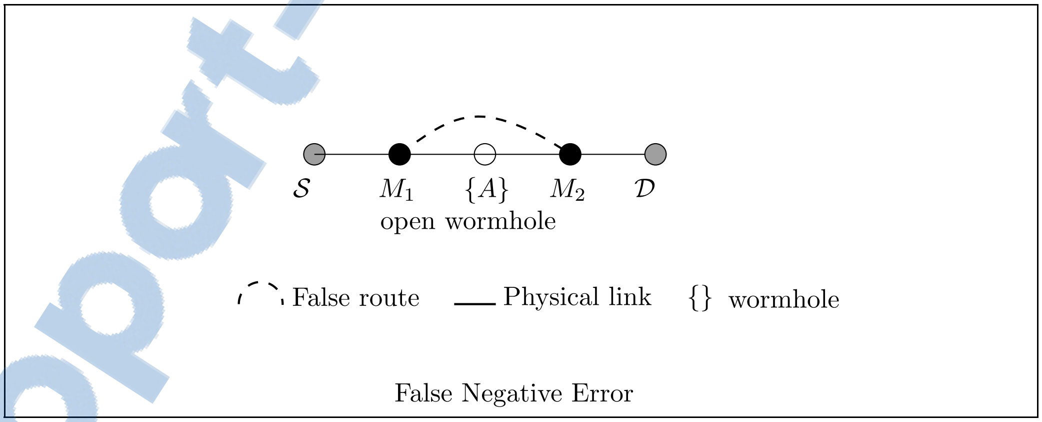


Figure 5.1 False negative error.

In Figure 5.2, a false positive error is generated if an existing network security system accuses legitimate nodes M_1 and M_2 of acting maliciously.

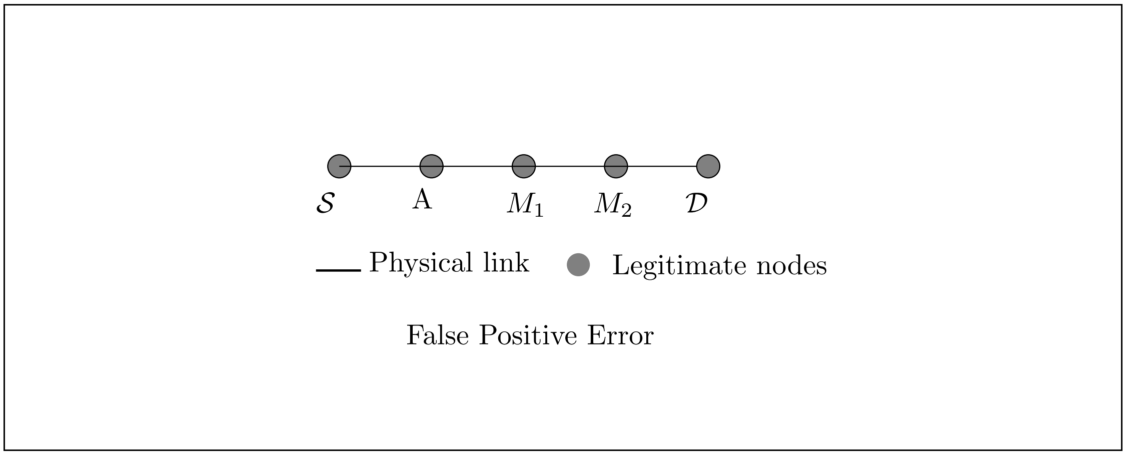


Figure 5.2 False positive error.

As demonstrated formally in **Lemma 4.1** and **Lemma 4.2** (sections 4.5 and 4.6), it is not possible to find any witness node for a strong open wormhole link connecting two malicious nodes. Therefore, the rate of false negatives for our approach is zero; thus, no false negative alarm is possible. An attacker developing a strong open wormhole attack is always detected by our protocol. As there are no possible false negatives, no wormhole attack is simulated.

The final important point to verify in terms of the simulations is whether or not the WIM-DSR protocol is too restrictive, leading to the detection of numerous false positive attacks.

We have developed a program simulating 100 static nodes randomly distributed in a 1000m×1000m square. Each simulation analyzes the paths connecting all the possible pairs of source and destination nodes. Each experiment presented here corresponds to the average results of 1000 simulations on independent sets. The objective is to determine how many pairs of source and destination nodes do not have fully witnessed paths in a given set of points. These pairs would represent the false positive alarms for WIM-DSR, because they would be declared as malicious, even in the absence of a real attack.

Network density is important in ad hoc networks. For a given region, there are two ways to increase it: (1) increase the number of nodes; or (2) increase the transmission range of the nodes. Since the complexity of our simulation program depends on the number of nodes, that number is fixed and different range values are used.

It is essential to find a lower bound on the transmission range to ensure that G(r) would be connected, or at least would have a large connected component containing most of its nodes. It has been proven that such a graph with n nodes is connected with a high probability if the transmission range is at least = $\sqrt{\frac{\ln n \pm O(1)}{\pi n}}$ (see (Penrose, 1999) and (Diaz, Mitsche, and Perez-Gimenez, 2008)). Such a bound was observed empirically during the simulation process. The graph G(r) was not connected in only 128, 44, and 17 cases out of 1000, for 190m, 210m, and 230m respectively.

We are now ready to present the results of our simulations.

Table 5-1 Number of pairs of nodes (average on 1000 simulations)

Paths	**190m**	**210m**	**230m**
Paths with *length* = 1	473.0 (9.6%)	567.1 (11.5%)	677.8 (13.5%)
Paths with *length* = 2			
Strongly witnessed paths	456.9 (9.2%)	616.5 (12.5%)	787.3 (15.9%)
No witnessed paths	245.7 (5.0%)	256.8 (5.2%)	263.6 (5.3%)
Paths with *length* > 2			
Strongly witnessed paths	2794.0 (56.4%)	3114.0 (62.9%)	3105.5 (62.7%)
If not, forward witnessed paths	302.8 (6.1%)	174.2 (3.5%)	69.5 (1.4%)
No path between the pair of nodes	22.7 (0.4%)	6.7 (0.1%)	2.4 (0.05%)
No witnessed path (false positive)	655.0 (13.2%)	214.7 (4.3%)	54.0 (1.1%)

In a randomly deployed network with 100 nodes, there are 4950 possible combinations of pairs of source and destination nodes. Those pairs have been classified based of the length of the paths connecting them.

Classified in the first group are the pairs of nodes connected by a path of length = 1 (1-hop neighbors). According to Table 5-1, the 11.5% of the pairs of source and destination nodes obtained in the simulation belong to this group (when a transmission range of 210 m is used). For those pairs, no wormhole detection process is required.

In Figure 5.3, if nodes S and v_1 set up a communication, assuming that the source and destination nodes are trustable (if the attacker has not compromised either node), then the attacker has access to the data exchanged and no wormhole attack is required. If the source and destination nodes are directly connected, then the wormhole attack is discarded.

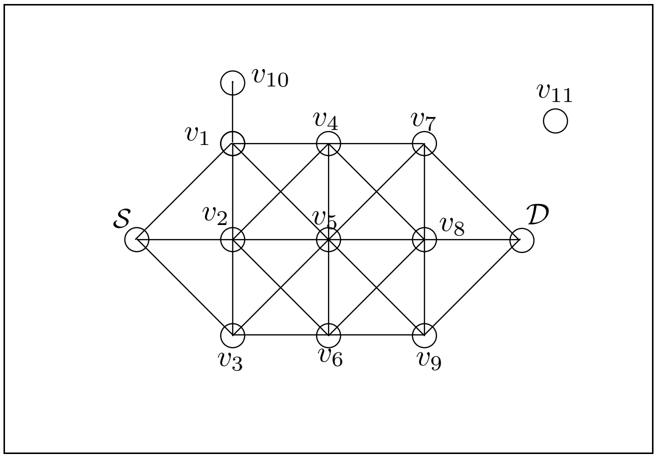


Figure 5.3 Pairs of source-destination nodes.

The second group is formed by the pairs of source and destination nodes connected by a path of length = 2 (2-hop neighbors). This group represents 17.7% of the network (when a transmission range of 210 m is used). The 70% of pairs in this group are connected by a fully witnessed path and 30% share a path that is not fully witnessed.

In Figure 5.3, the pair formed by the nodes S and v_5 is an example of this group. In this case, nodes v_1 and v_3 are strong witnesses of the path S-v_2-v_5. In contrast, the path between nodes S and v_{10} is not fully witnessed, since no witness node can be found for the link v_1-v_{10}.

Even though the rate of false positives seems to be high in this group, the general performance of the network is not affected, since no wormhole attack can be developed in this set of pairs. If nodes S and v_{10} set up a communication using path S-v_1-v_{10}, assuming that the source and destination nodes are trustable, as nodes S and v_{10} can ensure that node v_1 is a mutually valid 1-hop neighbor (Assumption II, chapter 4.3), then no wormhole attack can be developed.

The third group is formed by the pairs of source and destination nodes connected by a path with length *l,* where $3 \leq l$. This group represents 75.7% of the network (56.4% sharing strongly witnessed paths, 6.1% sharing forward witnessed paths and 13.2% with no fully witnessed paths representing the false positives) when a transmission range of 210 m is used.

In Figure 5.3, the pair of source and destination nodes formed by nodes S and *D* is an example of this group. In this case, nodes v_1, v_4, and v_7 are strong witnesses in the path S-v_2-v_5-v_8-*D*.

In the fourth group are the pairs of source and destination nodes where no connecting path was found. This group represents 0.1% of the network when a transmission range of 210 m is used. In Figure 5.3, the pair of source and destination nodes formed by nodes S and v_{11} is an example of this group.

In conclusion, for an appropriate network density (210 m or 230 m), only a few pairs of nodes (false positive alarms) cannot find fully witnessed paths (4.3% and 1.1% respectively), showing that the rate of false positives for this approach is low. In cases where a fully witnessed path cannot be found between two nodes, pairs of nodes should pay closer attention to their communications.

CONCLUSION

The special characteristics of ad hoc networks make them a preferred target from some of the most elaborate attacks developed against wireless networks. A specific example of one of those attacks is the wormhole attack. This type of attack can pose a severe threat to the proper functioning of the network, and its detection and prevention are proving to be a difficult task.

In this dissertation, we have focused on analyzing the wormhole attack, reviewing the previous approaches and proposing a new efficient solution aimed at counteracting it. The WIM-DSR protocol is able to detect all strong open wormhole attacks with a very low rate of false positive alarms. This solution does not require any cryptographic processing by the intermediate nodes in the absence of an attack, nor does it involve any packet overhead. This represents an improvement over the solution presented by Wang et al. (Wang et al., 2006).

Furthermore, unlike most of the previous approaches (Hu and Evans, 2004; Hu, Perrig and Johnson, 2003; Jakob, Srikanth and Michalis, 2006; Khalil, Bagchi and Shroff, 2008; Khalil, Saurabh and Shroff, 2005; Lijun, Ning and Xiangfang, 2007; Xu and Boppana, 2007), WIM-DSR offers a verifiable solution to the Sybil-Wormhole attack. The main contributions of this dissertation are listed below.

- A general introduction to the principal types of attacks that can be launched on ad hoc networks, in particular a detailed review and classification of wormhole attacks, including the definition of the Sybil-Wormhole attack.

- A complete survey of the most important approaches proposed to detect and prevent open, half-open, and/or closed wormhole attacks in ad hoc networks.

- An analysis of the Sybil-Wormhole attack and the accuracy of the existing solutions for preventing it.

- A new mechanism preventing strong open wormhole attacks, including the Sybil-Wormhole attack. The WIM-DSR protocol uses the information collected by the destination node during the route discovery process of a multipath routing protocol to detect suspicious behavior.

- Simulation results which probe the functioning of WIM-DSR in static ad hoc networks and its performance in false positive detection.

In this dissertation, a new mechanism for detecting open wormhole attacks in ad hoc networks has been proposed. This mechanism can be improved in several ways. A list of possible future works related to this subject is presented and includes the following:

- Development of simulations for studying the behavior of WIM-DSR in an environment with mobile nodes.

- Addition of modifications to the protocol with the aim of detecting and preventing coordinated wormhole attacks developed by more than two colluding nodes.

BIBLIOGRAPHY

Anjum, Farooq and Petros Mouchtaris. 2007. *Security for wireless ad hoc networks*. John Wiley & Sons, Inc.

Barbeau, Michel, and Jean-Marc Robert. 2006. “Rogue-base station detection in WiMax/802.16 wireless access networks.” *Annales des Telecommunications/Annals of Telecommunications,* vol. 61, no. 11-12, pp. 1300-1313.

Brands, S. and D. Chaum. 1994. “Distance-bounding protocols.” In *Coll. Advances in Cryptology – EUROCRYPT '93. Proceedings of the Workshop on the Theory and Application of Cryptographic Technique*s, pp. 344-59. Berlin, Germany: Springer-Verlag.

Buttyán, L., L. Dóra, and L. Vajda. 2005. “Statistical Wormhole Detection in Sensor Networks.” *Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS 2005).*

Buttyán, L. and J.-P. Hubaux. 2008. *Security and Cooperation in Wireless Networks*. Cambridge University Press.

Capkun, Srdjan, Buttyan, Levente, and Jean-Pierre Hubaux. 2003. “SECTOR: Secure tracking of node encounters in multi-hop wireless networks.” In *Coll. Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (in association with the 10th ACM Conference on Computer and Communications Security)*, pp. 21-32. Fairfax, VA, United States: Association for Computing Machinery.

Davidson, M. 1983. *Multidimensional Scaling*. John Wiley & Sons, Inc.

Diaz, Josep, Dieter Mitsche, and Xavier Perez-Gimenez. 2008. “On the connectivity of dynamic random geometric graphs.” In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. San Francisco, California: Society for Industrial and Applied Mathematics.

Eriksson, Jakob, Krishnamurthy Srikanth, and Michalis Faloutsos. 2006. “TrueLink: A Practical Countermeasure to the Wormhole Attack in Wireless Networks.” In *Proceedings of the 2006 IEEE International Conference on Network Protocols*. IEEE Computer Society.

Hu, L. and D. Evans. 2004. “Using Directional Antennas to Prevent Wormhole Attacks.” In *Proc. of the Network and Distributed System Security Symposium.*

Hu, Y. C., A. Perrig, and D. B. Johnson. 2003. “Packet leashes: a defense against wormhole attacks in wireless networks.” In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies.* IEEE, vol. 3, pp. 1976-1986.

Hu, Yih-Chun, Adrian Perrig, and David Johnson. 2005. “Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks.” *Wireless Networks,* vol. 11, no. 1-2, pp. 21-38.

Jacquet, P., P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. 2001. “Optimized link state routing protocol for ad hoc networks.” In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings.* IEEE International, pp. 62-68.

Johnson, David, David Maltz, Imielinski, T. and H. Korth. 1996. “Dynamic Source Routing in Ad Hoc Wireless Networks.” In *Mobile Computing*, vol. 353. Kluwer Academic Publishers.

Kannhavong, Bounpadith, Hidehisa Nakayama, Nei Kato, Abbas Jamalipour, and Yoshiaki Nemoto. 2007. “A study of a routing attack in OLSR-based mobile ad hoc networks.” *International Journal of Communication Systems.*

Khalil, I., S. Bagchi, and N. B. Shroff. 2008. “MobiWorp: mitigation of the wormhole attack in mobile multihop wireless networks.” *Ad Hoc Networks,* vol. 6, no. 3, pp. 344-62.

Khalil, I., Bagchi Saurabh, and N. B. Shroff. 2005. “LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks.” In *Coll. Proceedings of the 2005 International Conference on Dependable Systems and Networks,* pp. 612-621. Los Alamitos, CA, USA: IEEE Comput. Soc.

Lazos, L., R. Poovendran, C. Meadows, P. Syverson, and L. Chang. 2005. “Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach.” *Wireless Communications and Networking Conference,* IEE.

Lazos, Loukas, and Radha Poovendran. 2004. “SeRLoc: Secure range-independent localization for wireless sensor networks.” In Coll. Proceedings of the 2004 ACM Workshop on Wireless Security, WiSe, pp. 21-30. New York, NY 10036-5701, United States: Association for Computing Machinery.

Lee, Kyuho, Hyojin Jeon, and DongKyoo Kim. 2007. “Wormhole Detection Method based on Location in Wireless Ad-Hoc Networks.” In *New Technologies, Mobility and Security*, pp. 361-372.

Lee, S. J. and M. Gerla. 2001. “Split multipath routing with maximally disjoint paths in ad hoc networks.” In Coll. IEEE International Conference on Communications. Helsinki, Finland: Institute of Electrical and Electronics Engineers Inc. Vol. 10, pp. 3201-3205.

Lijun, Qian, Song Ning, and Li Xiangfang. 2007. “Detection of wormhole attacks in multipath routed wireless ad hoc networks: A statistical analysis approach.” *Journal of Network and Computer Applications,* vol. 30, no. 1, pp. 308-330.

Maheshwari, R., J. Gao, and S. Das. 2007. “Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information.” *INFOCOM 2007. 26th IEEE International Conference on Computer Communications.* IEE.

Meguerdichian, S., S. Slijepcevic, V. Karayan, and M. Potkonjak. 2001. “Localized algorithms in wireless ad hoc networks: location discovery and sensor exposure.” In *Coll. MOBIHOC 2001. Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 106-16. New York, NY, USA: ACM.

Newsome, J., E. Shi, D. Song, and A. Perrig. 2004. “The Sybil attack in sensor networks: analysis & defenses.” In *Third International Symposium on Information Processing in Sensor Networks, 2004.* IPSN *2004*, pp. 259-268.

Patwari, N., Hero, M. Perkins, N. S. Correal, and R. J. O'Dea. 2003. “Relative location estimation in wireless sensor networks.” *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137-2148.

Penrose, Mathew. 2003. *Random Geometric Graphs (Oxford Studies in Probability)*. Oxford University Press.

Penrose, Mathew D. 1999. “On k-connectivity for a geometric random graph.” *Random Struct. Algorithms,* vol. 15, no. 2, pp. 145-164.

Perkins, C. and E. Royer. 1999. “Ad-hoc on-demand distance vector routing.” In *Mobile Computing Systems and Applications, 1999. Proceedings of WMCSA '99. Second IEEE Workshop*, pp. 90-100.

Perkins, Charles and Pravin Bhagwat. 1994. “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers.” In *ACM {SIGCOMM}'94 Conference on Communications Architectures, Protocols and Applications*, pp. 234-244.

Perrig, Adrian, Ran Canetti, J. D. Tygar, and Dawn Song. 2000. “Efficient authentication and signing of multicast streams over lossy channels.” In *Coll. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 56-73. Berkeley, CA, USA: IEEE.

Pirzada, Asad Amir and Chris Mcdonald. 2006. "Detecting and evading wormholes in mobile ad-hoc wireless networks." *International Journal of Network Security Vol.3, No.2, PP.191–202, Sept. 2006 (http://ijns.nchu.edu.tw/).*

Poturalski, Marcin, Panos Papadimitratos, and Jean-Pierre Hubaux. 2008. "Secure neighbor discovery in wireless networks: formal investigation of possibility." In *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security*. Tokyo, Japan: ACM.

Qian, Lijun, Ning Song, and Xiangfang Li. 2007. "Detection of wormhole attacks in multipath routed wireless ad hoc networks: A statistical analysis approach." *Journal of Network and Computer Applications,* vol. 30, no. 1, pp. 308-330.

Rappaport, T. S. 2002. *Wireless Communications: Principles and Practice*, 2nd Edition.

Singelee, D. and B. Preneel. 2005. "Location verification using secure distance bounding protocols." In *Coll. 2005 IEEE International Conference on Mobile Adhoc and Sensor Systems*," p. 7. Piscataway, NJ, USA: IEEE.

Wang, W., B. Bhargava, Y. Lu, and X. Wu. 2006. "Defending against wormhole attacks in mobile ad hoc networks." *Wireless Communications and Mobile Computing,* vol. 6, no. 4, pp. 483-502.

Wang, Weichao and Aidong Lu. 2007. "Interactive wormhole detection and evaluation." *Information Visualization,* vol. 6, no. 1, pp. 3-17.

Wang, Xia and Johnny Wong. 2007. "An End-to-end Detection of Wormhole Attack in Wireless Ad-hoc Networks." In *Proceedings of the 31st Annual International Computer Software and Applications Conference*. IEEE Computer Society.

Weichao, Wang and Bhargava Bharat. 2004. "Visualization of wormholes in sensor networks." In *Proceedings of the 3rd ACM workshop on Wireless security*. Philadelphia, PA, USA: ACM.

Xu, Su and R. V. Boppana. 2007. "On Mitigating In-band Wormhole Attacks in Mobile Ad Hoc Networks." In *IEEE International Conference on Communications, 2007 (ICC '07)*, pp. 1136-1141.

Xuefei, Li and L. Cuthbert. 2004. "On-demand node-disjoint multipath routing in wireless ad hoc networks." In *Coll. Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN 2004)*, pp. 419-420. Los Alamitos, CA, USA: IEEE (Comput. Soc.).

Zafar, H., D. Harle, I. Andonovic, and M. Ashraf. 2007. “Partial-disjoint multipath routing for wireless ad-hoc networks.” In *Coll. 2007 32nd IEEE Conference on Local Computer Networks*, pp. 258-259. Piscataway, NJ, USA: IEEE.

Zhen-ning, Kong, D. H. K. Tsang, B. Bensaou, and Gao Deyun. 2004. “Performance analysis of IEEE 802.11e contention-based channel access.” *IEEE Journal on Selected Areas in Communications,* vol. 22, no. 10, pp. 2095-2106.