國立交通大學

資訊科學與工程研究所

碩士論文

在無線隨意網路中抵抗蟲洞攻擊之
動態來源路由協定

A Wormhole-Proof Dynamic Source Routing Protocol for Wireless Ad-hoc Networks

研 究 生：吳奕叡

指導教授：簡榮宏　教授

中華民國 九十六 年六月

在無線隨意網路中
抵抗蟲洞攻擊之動態來源路由協定

研究生：吳奕叡　　　　指導教授：簡榮宏 博士

國立交通大學資訊科學與工程研究所

摘　　要

近年來，無線行動隨意網路無論在研究上或實務上都越來越受受重視。由於無線網路的傳輸訊號暴露在空氣中，使得惡意攻擊者可以輕易地干擾、竊聽甚至入侵無線行動隨意網路；然而，現存許多無線行動隨意網路方面的研究，大都假設無線網路環境是安全且可信賴，這與實際狀況有所不符。在本篇論文中，我們主要針對蟲洞攻擊，進行研究與探討，提出一個在網路結點上，不需要時間同步或特殊硬體的動態來源路由協定，所提出的協定可抵抗蟲洞攻擊。藉由模擬的結果顯示，我們提出的方法可有效地偵測蟲洞攻擊，且所提出的協定耗費的網路資源相當微小。

A Wormhole-Proof Dynamic Source Routing Protocol for Wireless Ad-hoc Networks

Student：Yi-Jui Wu Advisor：Dr. Rong-Hong Jan

INSTITUTE OF COMPUTER SCIENCE AND ENGINERRING

NATIONAL CHIAO TUNG UNIVERSITY

Abstract

In recent years, wireless mobile ad-hoc networks (MANet) have becoming attractive and important in both research and practice. However, many previous works on MANet assume a trusty network environment, while malicious adversaries can easily disrupt, eavesdrop and intrude because of the nature of wireless communication. In this thesis, we study a particular attack called wormhole attacks, and develop a wormhole-proof dynamic source routing protocol, which requires neither time synchronization nor specialized hardware, to counter wormhole attacks. The results show that the proposed protocol can effectively detect wormhole attacks and involve less computation overhead while compared to the previous works.

Contents

List of Figures

Chapter 1

Introduction

Wireless Mobile Ad-hoc network (MANet) is becoming increasingly attractive and important in both research and practice. The main characteristic of MANet is that it requires absolutely no pre-established fixed infrastructures such as access points or base stations, thus it can be deployed more easily and ubiquitously. MANet is also a self-organized network; it needs no central administration, while every MANet node has to perform routing of network traffic. Besides, every MANet node is free to move arbitrarily. As a result, more and more practical and valuable applications, not only for military but also for civilian, choose MANet as the platform due to its mobility, feasibility and flexibility.

To make MANet function properly and perfectly, many issues about MANet should be investigated and considered carefully and thoroughly. Moreover, among all research topics relating to MANet, security is one of the most crucial but usually disregarded subjects. Most previous research on MANet assumed a trusted environment. Unfortunately, because of the nature of wireless communication, all data signals are exposed for anyone to capture. If a wireless network does not have sufficient protection mechanisms, a malicious attacker can eavesdrop on privacy data, or even intrude into a network freely.

Over the past few years, several studies have been made on MANet security: some researches pointed out potential vulnerabilities of wireless networks and proposed possible solutions [1][2]; some others utilized cryptographic techniques on routing protocol for security and privacy [3][4]; and still others composed concise and elegant survey papers about MANet security[5][6][7]. It seems more than ample research on MANet security has been done.

1.1 Wormhole Attacks

However, little research about MANet security has focused on an extraordinary severe, devastating and powerful attack called "wormhole attacks", which maybe the one of the easiest launching MANet attacks and one of the most serious MANet security problems.

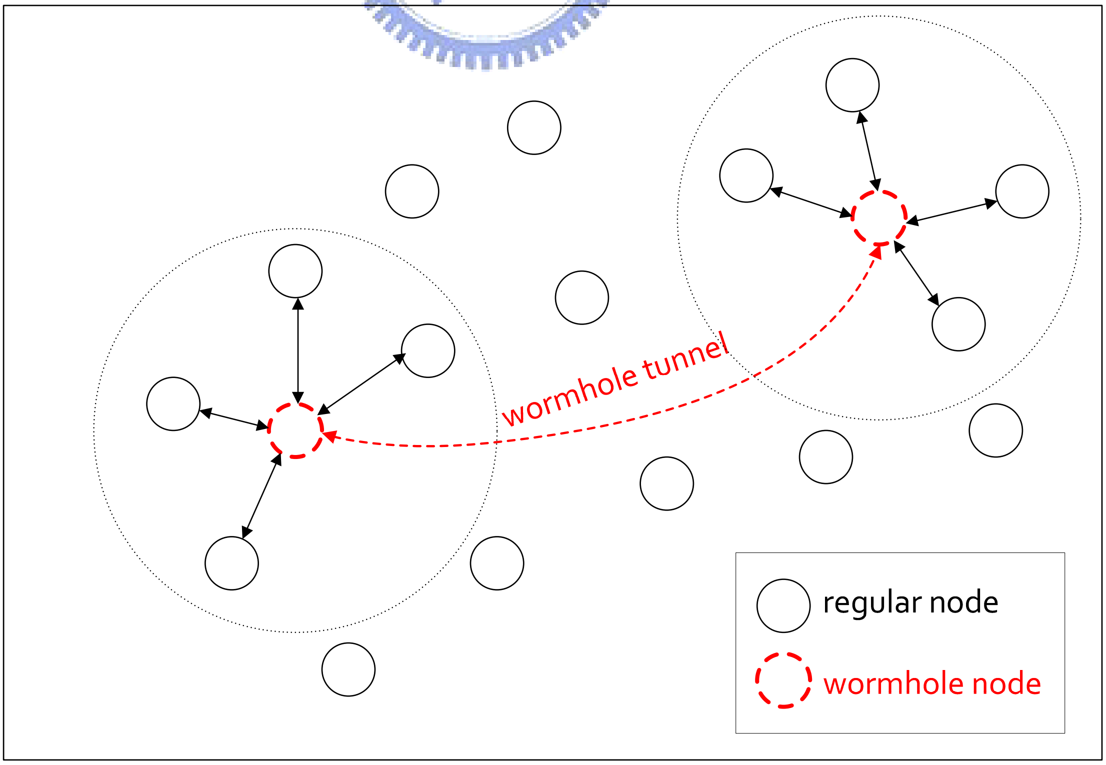


Figure 1.1: A wormhole attack scenario

Wormhole attacks happen when one wormhole node eavesdrops and records packets at one location, and then tunnels the eavesdropped packets to a certain faraway collusive wormhole node. After receiving the tunneled packets, the faraway collusive wormhole node replays these packets (figure 1.1). The tunnel between collusive wormhole nodes can be established in various ways, such as direct wire connection, high power transmission, and out-of-band hidden channel.

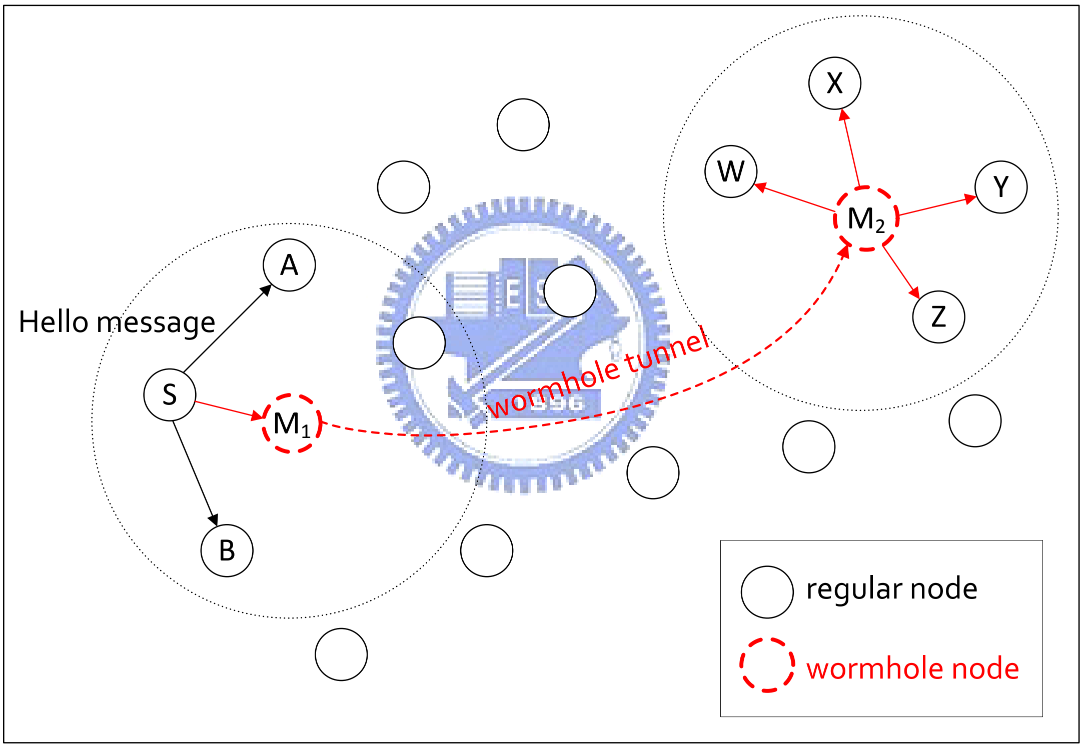


Figure 1.2: A wormhole attack scenario on proactive routing protocol

Wormhole attacks affect a network most significantly while nodes are establishing route to another node. Wormhole attacks can make particular nodes in a network generate improper routing tables for themselves. For example, if a wormhole attack is launched on a periodic routing protocol, such as optimized link state routing protocol (OLSR) [8], the collusive wormhole nodes can let a regular node trust some other nodes are its neighbor nodes. In figure 1.2, node S broadcast a ***Hello*** message periodically. If no wormhole nodes exist, only node A and B learn that S is their neighbor nodes; however, if wormhole attacks exist, M_1 can tunnel the ***Hello*** message to $M_{2,}$ and M_2 replays the Hello message. As a result, node W, X, Y and Z also believe that S is their neighbor nodes.

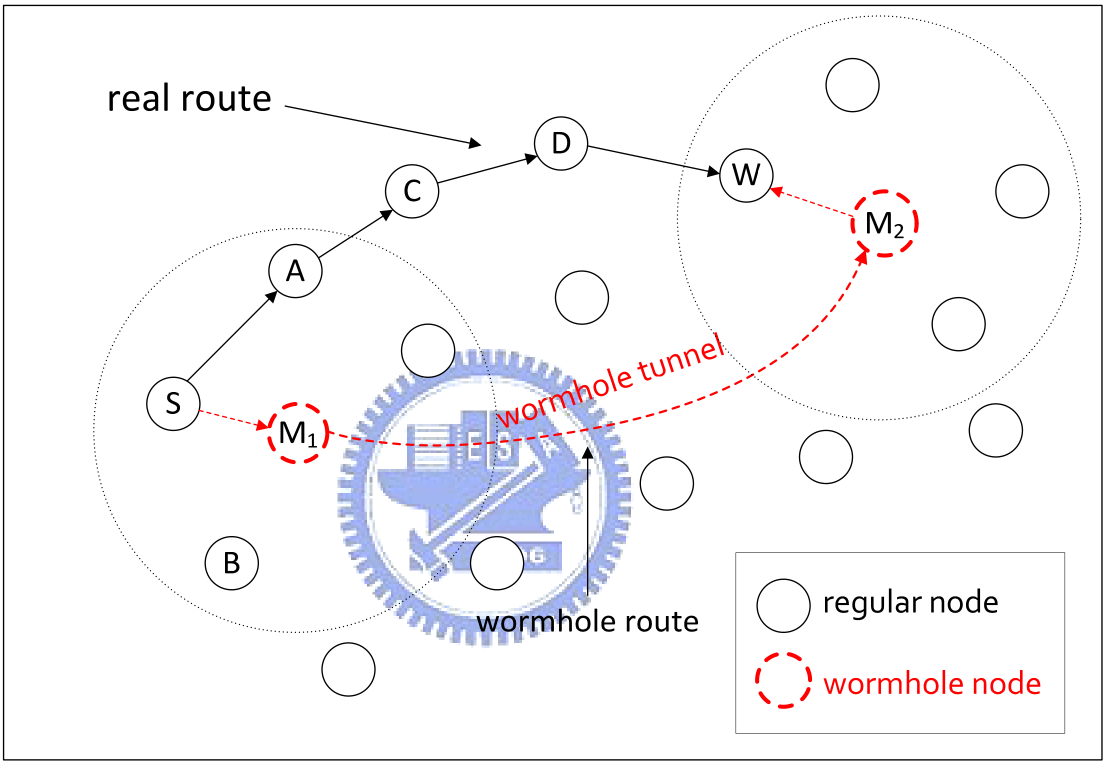


Figure 1.3: A wormhole attack scenario on demand routing protocol

Besides periodic routing protocol, wormhole attacks can also cause great disruption on on-deman routing protocol, such as DSR [9] and AODV [10]. Firstly, we suppose no wormhole attacks happen in figure 1.3. We assume that S is attempting to establish a new route to W. S can establish a good route to W by sending a route request (RREQ) through node A, C, D and W respectively, thus S knows a route of S -> A -> C -> D -> W . On the other hand, we assume that wormhole attacks exist, where there are two collusive wormhole nodes M_1 and M_2. Neither M1 nor M2 needs to alter the contents of any packets, but merely tunnel and relay radio signals, frames or packets in the network. Still, we assume that S is attempting to establish a new route to W. In this case, M_1 can capture a RREQ packet from S and tunnel the RREQ packet to M_2; next, M_2 replays this RREQ locally; after that, W receives the RREQ and believes that S is its direct neighbor node. Hence, the route from S to W is merely S -> W, and the network is disrupted by simply two adversary nodes.

It is worth noting that in the network, although both M_1 and M_2 physically exist, they virtually vanish. M_1 and M_2 have no network identities but are repeaters in; i.e., none of the good nodes in the network is able to aware of the existence of M_1 and M_2. This is the major reason why wormhole attacks are difficult to deal with. As it is not able to easily find out wormhole nodes in a network, a specific mechanism to prevent and detect wormhole attacks is necessary.

Wormhole attacks have very different features from other attacks on MANet: an adversary does not have to breach the cryptography nor compromise any nodes for launching wormhole attacks, because the adversary nodes need not decapsulate any frames or packets during attacks. What an adversary needs to do is to setup collusive wormhole nodes in wireless networks, to capture radio signals and to build tunnels between the collusive nodes. Then the wormhole nodes can eavesdrop, capture and tunnel radio signals or steal private information that flows via collusive wormhole nodes, although they have no identities or cryptography keys required in the network. Additionally, since wormhole nodes replay signals at a place and can attract network traffic, wormhole attacks are a combination of replay attacks, black hole attacks or grey holes attacks [5][6][7]. Hence, it is much more difficult to detect and prevent wormhole attacks.

In fact, if the wormhole nodes conduct no mal-behaviors in the network or are configured by network administrators, wormhole tunnel may be a very pleasing feature. They may provide alternate and faster routes, and even reduce the use of wireless bandwidth and save energy of mobile nodes. But quite often the wormhole nodes may be laid down by malicious adversaries.

In this thesis, we propose a new ad-hoc routing protocol for defending against wormhole attacks called “Wormhole-Proof Dynamic Source Routing Protocol (WP-DSR)” for wireless mobile ad-hoc networks. This protocol is an on-demand source routing protocol, which needs neither time synchronization nor specific hardware on network nodes to detect wormhole attacks. Our protocol is based on the combination of time limitation and watchdog strategies. It approximates the hop-to-hop transmission time between two nodes during route discovery phase. After receiving route replies (RREP), the source node can obtain all hop-to-hop transmission time of the candidate route, and only selects routes containing no contaminated link. If the transmission time is over a certain reasonable upper bound, that link is said to be a contaminated link.

The rest of this thesis is organized as follows: In Chapter 2, we discuss the related works and compare their advantages and disadvantages. In Chapter 3, we present our Wormhole-Proof Source Routing Protocol for Wireless Ad-Hoc Networks. In Chapter 4, we discuss our evaluation methods and the results. Finally, we conclude this thesis in Chapter 5.

Chapter 2

Related Works

In this chapter, several research works related to wormhole attacks will be introduced. These works include theoretic analysis on wormhole attacks and detecting mechanisms to wormhole attacks. Moreover, the detecting mechanisms can be classified into distance or time limiting detection approaches, false geometry or topology detection approaches, and neighbor nodes monitoring approaches. We will discuss the theoretic characters of wormhole attacks and these three types of detecting approaches below.

2.1 Theoretic Analysis

These works [11][12] apply theoretic analysis on wormhole attacks. They define wormhole attacks in graph theory and show wormhole attacks will disrupt a network to what degree.

Lazos et al. [11] proposed a graph theoretic framework for modeling wormhole attacks and stated the necessary and sufficient conditions for any candidate solutions to such attacks. They demonstrated that an Ad-hoc network can be modeled as a geometric random graph whose connectivity matrix displays whether the distance between any two nodes of the network is less than or equal to the transmission range of the network nodes. Subsequently, they provided a theorem to show that a candidate solution is able to prevent wormhole attacks if and only if it can construct a communication graph that is a subgraph of the geometric random graph of the networks.

Khabbazian et al. [12] analyzed the effect of wormhole attacks in shortest path routing. If wormholes exist in a network, a regular node has to communicate with nodes inside a certain region via wormhole repeaters, while a shortest path routing is used. This region is called unreachable, and Khabbazian et al. measure the unreachable region to indicate how many communications are disrupted. Their results showed that two colluding wormhole repeaters with a strategic placement can disrupt on average 32% of the communications across the network. Moreover, (n≧2) repeaters can disrupt on the average at most (1-1/n) of all communications.

2.2 Distance or Time Limiting Detection Approaches

The idea behind this group of countermeasures is intuitive: limit the distance a packet can traverse between nodes [13][14][15][16][17][18]. If a packet traverses more than a reasonable distance, usually within the transmission range of nodes, this packet is considered being affected by wormhole attacks. Besides, it is also possible to limit traverse distance by limiting the packet traverse time, since (time = distance/speed), where speed is the speed of wireless radio signal. The advantages of these mechanisms are their ideas’ simplicity; however, they usually require time synchronization or location information on each node to calculate the distance between nodes.

Hu et al. [13] introduced a general mechanism called “Packet Leashes”, which was the first mechanism to detect and defend against wormhole attacks. A packet leash is any information that is added to a packet designed to restrict the packet’s maximum allowed transmission distance. Two types of packet leashes were presented: Geographic Leashes and Temporal Leashes. In Geographic Leashes, each node has to know its precise location and all nodes have loose time synchronization. Before sending a packet, each node attaches its current location and time to the packet. When the receiving node receives the packet, it computes the distance to the sending node and the transmission time of the traverse path. The receiving node can use this distance and time information to decide whether the packet was transmitted through wormhole repeaters. In Temporal Leashes, all nodes require very tight time synchronization. Before sending a packet, each node attaches its current time to the packet. When receiving the packet, the receiving node compares the temporal leash of the packet to its time, and computes the distance to the sending node by assuming the propagation speed is equal to the light speed. Consequently, it can determine if the packet traveled an overlong distance caused by wormhole attacks. The drawbacks of Packet Leashes are that both leashes need time synchronization; and Geographic Leashes require some other methods, such as GPS or location service, to let each node get its precise location.

Capkan et al. [14] proposed the Secure Tracking of Node Encounters in Multi-hop Wireless Networks (SECTOR), which is a set of mechanisms for the secure verification of the time of encounters between nodes, and can be used to detect wormhole attacks without requiring any time synchronization or location information. MAD (Mutual Authentication with Distance-bounding) is one protocol in SECTOR for determining the distance between any two communicating nodes. In MAD, each node is equipped with a specialized hardware module that can temporarily take over the control of the radio transceiver unit of the node, and can receive a single bit, perform a XOR operation on two bits, and then transmit a single bit without involving the CPU of the node. MAD uses this module to perform a series of bit exchanges between two nodes to calculate the propagation time needed, thus obtain the transmission distance for confirming if these two nodes are communicating via wormholes. The main disadvantage of SECTOR is that requiring specialized hardware may make it impractical to apply SECTOR in MANet.

Sastry et al. [15] did not focus on wormhole attacks, but address on a problem called location verification, which is somewhat similar to wormhole attacks. This kind of mechanisms verifies if malicious nodes send factitious information about their location information. Sastry et al. proposed the Echo protocol, which use two signals of different speed, e.g. the radio frequency and ultrasonic, to obtain the distance of two nodes from the time delay. This protocol is lightweight, and does not require time synchronization; however, the requirement of two different signals on a node makes it impractical to apply the Echo protocol in MANet.

Eriksson et al. [16] presented Truelink, a time limiting method to counter wormhole attacks. Truelink does not rely on time synchronization or location information, but require only minor modification to IEEE 802.11 standards without changing any frame format. Truelink is also backwards compatible and able to cooperate with IEEE 802.11 standard hardware. Truelink is not a general solution to wormhole attacks, but considers IEEE 802.11 MAC and use IEEE 802.11 RTS/CTS exchange to verify if the transmission time between two nodes is under the time constraints. In IEEE 802.11 standards, the functions to deal with receiving CTS and data packets both time out after one SIFS interval, while the shortest frame of IEEE 802.11 takes much more time than a SIFS interval to be fully transmitted. Therefore, if the wormhole repeaters decapsulate the eavesdropped frames before tunneling out, Truelink is able to prevent wormhole attacks. However, Truelink may fail if the wormhole repeaters do not decode messages, but immediately retransmit radio signals.

Chiu et al. [17] proposed another detection mechanism called Delay Per Hop Indication (DelPHI), which is also based on time bounding approach and does not depend on time synchronization or location information. DelPHI is a routing protocol similar to AODV. DelPHI collects the hop count and round trip time of a route, and calculates the delay/hop value as the indicator of detecting wormhole attacks. Under normal circumstance, this value should be relatively low, whereas under wormhole attacks it may be unreasonably high. However, they do not consider any packet processing delay or queuing delay, which may lead to a high delay/hop value, too.

Nait-Abdesselam et al. [18] considered the wormhole attacks in Optimized Link State Routing Protocol (OLSR), and developed a detecting mechanism for it. In OLSR, if wormhole attack exists, it may cause nodes to choose false multipoint relays (MPRs) and result all nodes to have incorrect topology information. Their method consists of two phases: the suspicious links detection and wormhole verification. In the former, Nait-Abdesselam et al. define two new HELLO messages, the $\text{HELLO}_{\text{req}}$ and $\text{HELLO}_{\text{rep}}$, which supersedes the standard HELLO message in OLSR. These new messages are used to detect if a link is suspicious of containing a wormhole tunnel. A node can achieved this by handshaking messages and checking whether the $\text{HELLO}_{\text{rep}}$ is arrived within a reasonable timeout. If it is not, the node ranks the link suspicious and originates the wormhole verification. The node sends a probing message to another endpoint of the suspicious link to ask its opinion about the link. If it is also suspicious, the originated node concludes the suspicious link contains a wormhole tunnel. The improved protocol does not require any time synchronization, location information, complex computation or special hardware, and mostly be the same as the original OLSR; however, the use of an approximated timeout may lead it fail if the packet processing time plus queuing delays is variable.

2.3 False Geometry or Topology Detection Approaches

These methods apply false geometric or topology information to detect wormhole attacks. If the analyzed results of the collected information violate the definition of uncontaminated situation, wormhole attacks may exist in the network. These methods do not require time synchronization, but need more complicated processes and message exchanges to achieve the goal.

Besides theoretic analysis, Lazos et al. [11] proposed a cryptographic mechanism, called local broadcast keys (LBK), based on keys only known within each real neighbor nodes to prevent wormhole attacks. LBK does not need any time synchronization, but require a small fraction of network nodes, the guard nodes, which know their location and own broader transmission range than the regular nodes. While establishing LBKs, all guard nodes broadcast their fractional keys and location information to the network; and then regular nodes collect every fractional key they received. If two regular nodes share more than a threshold number of fractional keys, they use these keys to generate a pairwise key. Finally, every node generates an LBK and unicasts it to the nodes which it shares a pairwise key with. After establishing the LBKs, each node can only communicate with their real nodes. In addition, Lazos et al. also provide a simple mechanism, called closet guard algorithm (CGA), which adopts the observation that a regular node should not receive fractional keys from guard nodes that are at a distance of more than two times of the transmission range of guard nodes, to distinguish which guards are infected by wormhole attacks.

Wang et al. [19] presented Multi-Dimensional Scaling – Visualization of Wormhole (MDS-VOW), a mechanism that reconstructs the network using multi-dimensional scaling and detects the wormhole by visualizing the anomalies introduced by wormhole attacks for wireless sensor networks. Multi-dimensional scaling (MDS) was originally a technique developed in the behavioral and social science for studying the structures of objects, and now has been applied to solve the localization and positioning problems in wireless networks. MDS-VOW does not require any special hardware or time synchronization; it requires only connectivity information, the received signal strength of each node and a centralized controller. In MDS-VOW, each node estimates the distance to its neighbor nodes by the received signal strength, and sends this information to the centralized controller. The centralized controller uses Dijkstra algorithm to calculate the distance between all nodes, and uses MDS to find the virtual position for each node. Then a surface smoothing scheme is used to compensate the distortions caused by distance measurement errors. If the reconstruction result between two nodes is bent toward each other, these two nodes may connect through a wormhole. The centralized controller gathers the information, computes wormhole indicators and distributes them to all nodes. However, the performance of MDS-VOW in a large scale environment may not be efficient. And MDS-VOW assumes that all nodes are static; therefore it may not work in wireless mobile networks.

Maheshwari et al. [20] proposed a localized algorithm that requires only connectivity information to detect wormhole attacks. Their algorithm searches for forbidden substructures in the connectivity graphs that should not be preset in a legal connectivity graph, while it needs no any additional hardware and time synchronization. They proved that inside a unit disk region, the maximum number of points that every pair of points which is strictly more than the unit distance away from each other is two. If this number in some network region is more than two, this region consists of a forbidden structure, where a wormhole may exist. They made use of this observation to develop a k-hop detection algorithm to look for forbidden structures to detect wormhole attacks.

2.4 Neighbor Nodes Monitoring Approaches

The ideas behind these methods are to monitor false behaviors of neighbor nodes to detect wormhole attacks. If a node displays false behavior, it means that wormholes are in the network. These methods do not require time synchronization, neither; but they need some special equipment such as directional antennas or central authority.

Hu et al. [21] proposed a mechanism using directional antennas to detect wormhole attacks. The basic idea behind their mechanism is to check if packets arrival direction is logical. They noticed that when under normal circumstances, if a sender uses a certain direction to transmit packets, the receiver must receive these packets from the opposite direction. But under wormhole attacks, the receiver may receive these packets from an illogical direction. Based on this observation, they developed verified neighbor discovery protocol and strict verified neighbor discovery protocol, which introduce verifiers to make their mechanism more accurate. However, their mechanism may not ascertain some wormhole attacks, and use of directional antennas is not practical in some scenarios.

Khalil et al. [22] presented MOBIWORP, a mechanism using local monitoring of neighborhood communication by each node as the basis to detect wormhole attacks. MOBIWORP composes of two protocols: the Selfish Move Protocol (SMP) and the Connectivity Aided Protocol with Constant Velocity (CAP-CV). The former protocol is for local detection, where the malicious nodes can be detected by the guards in its current neighborhood in a distributed approach. SMP assumes that a good node can generate, send and receive its own packets but cannot forward packets. A node can forward packets only if it is a wormhole node. However, SMP was proposed for static networks at first [23], and may cause a network to be disconnected if many nodes in it are mobile; hence, Khalil et al. developed the later protocol CAP-CV for global detection, where the adversary is detected on a global network scale by a secure central authority (CA) collecting report from guards at multiple locations. Besides detection, MOBIWORP also provides a mechanism to isolate malicious nodes from the network by removing the connectivity to malicious nodes locally and globally. MOBIWORP do not require any time synchronization or specialized hardware at the network nodes, but it requires a secure central authority (CA) for position tracking, which makes MOBIWORP unfeasible while constructing mobile ad-hoc networks.

Chapter 3

The Proposed Wormhole-Proof Protocol: WP-DSR

We proposed an on-demand ad-hoc routing protocol for protecting from wormhole attacks, called WP-DSR (Wormhole-Proof Dynamic Source Routing Protocol). This protocol is based on the DSR (Dynamic Source Routing) [9] protocol. WP-DSR uses time-limiting approach to detect wormhole attacks. It estimates all hop-to-hop transmission durations during the route discovery phase, and uses them to judge if the discovered route is contaminated by wormhole attacks. In this chapter, first we will review the DSR protocol and discuss our observation that can be used for detecting wormhole attacks. And then we will describe the proposed WP-DSR in detail.

3.1 Dynamic Source Routing Protocol

Dynamic Source Routing Protocol (DSR) is a routing protocol based on source routing technique designed for mobile ad-hoc networks. DSR is a source-initiated on-demand routing protocol, which creates routes only when required by the source node. DSR protocol consists of two major phases: route discovery and route maintenance, where the former is the main technique for a node to learn new routes in DSR.

In DSR, when a node wants to send packet to some other node, it first checks its route cache table to find if a route has already existed. If not, this node initiates a route request (RREQ) packet with its network ID, a unique sequence number and the network ID of the destination node, and then broadcasts it to the network.

When a node receives an RREQ packet and it is the destination node of the RREQ packet, it first checks if it has processed an RREQ with the same requestor ID and sequence number already. If it has, it then drops the RREQ packet directly. If not, it appends its network ID into the route path, and keeps the requestor ID and the sequence number of the RREQ packet. Afterwards, the neighbor node rebroadcasts the RREQ packet to the network.

If a node receives an RREQ packet, and it is the destination node of the RREQ packet, it initiates a route reply (RREP) packet with all network IDs in the route path and the same sequence number of the RREQ packet. After that, the destination node unicasts the RREP packet to the requestor along the reverse route path. Subsequently, the requestor node obtains a route to the destination node.

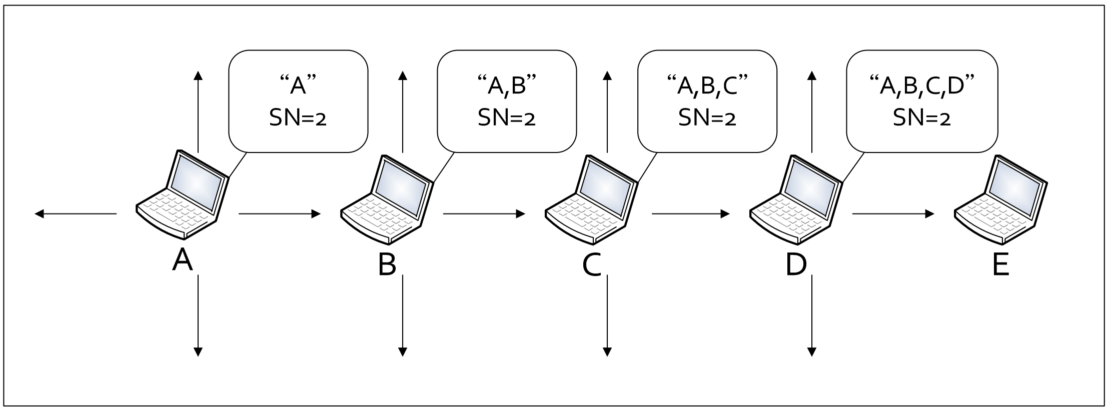


Figure 3.1: An example of route discovery in DSR

For example, in figure 3.1, we assume that node A wants to establish a route to node E. Node A initiates an RREQ packet with its network ID “A”, a unique sequence number “2” and destination node ID “E”. Next, A broadcasts this RREQ packet to the network, and node B, a neighbor node of node A, receives the RREQ packet. Since it has not processed an RREQ packet with network ID “A” and sequence number “2”, it appends its network ID “B” into the route path of the RREQ packet, and rebroadcasts it to the network. So do node C and node D. When the destination node, node E, receives this RREQ packet, it initiates an RREP packet with route path “A, B, C, D, E” and sequence number “2”, and then unicast it to A through node D, C, B, respectively.

3.2 The Proposed Protocol: WP-DSR

3.2.1 An observation on DSR

First we will introduce the basic idea of the proposed protocol. In the original DSR protocol, while a node receives an RREQ packet with a certain requestor ID and sequence number, and if the node has processed an RREQ packet with the same requestor ID and sequence number pair, it shall not process it anymore but silently drop it. But we observe that if the receiving node checks one more thing to this kind of RREQ packet, this feature can be used to calculating the traverse time between two nodes and detecting wormhole attacks: if a certain node is the second last hop in the route path of an RREQ packet, it means that the node broadcasting this RREQ received an RREQ packet with same requestor ID and sequence number from this certain node. Thus, if this certain node records the timestamp of broadcasting the RREQ packet and that of receiving an RREQ of previous description, it can obtain the round trip time between it and the rebroadcasting node without an acknowledgement packet.

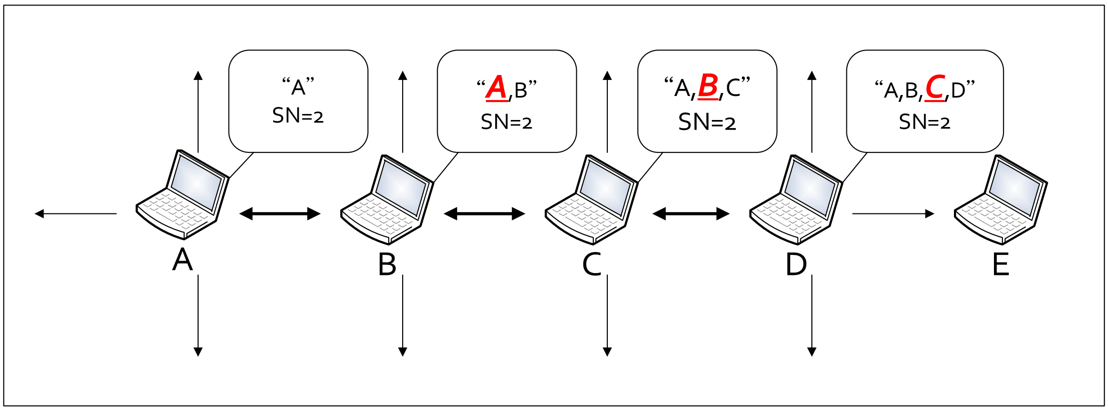


Figure 3.2: The observation

For instance, as illustrated in figure 3.2, node B receives an RREQ packet from node A and rebroadcasts an RREQ with requestor "A" and sequence number "2" to the network. Since node A, the node sending this RREQ packet to B, is a neighbor node of B, it shall receive this RREQ packet as other neighbor nodes. In the original DSR, node A finds that it has processed an RREQ with the same requestor ID and sequence number; therefore, node A drops it silently. But if node A checks the route path in the RREQ packet and notices that itself is the second last hop in it, and also A has the timestamp of broadcasting the RREQ packet, node A can record when it receives this RREQ packet to get the round trip time between node A and node B. So do node B to node C, and node C to node D.

In next section, we will narrate the proposed protocol and how the proposed protocol applies this feature for detecting wormhole attacks in detail.

3.2.2 The Proposed Protocol

The route discovery of WP-DSR is based on that of DSR, while with only a few changes.

In WP-DSR, when a node initiates or rebroadcasts an RREQ packet, it records the timestamp of when the packet is sending out in $T^{node}Send_{(requestor,\ SN)}$, where ***node*** is the network ID of the sender, ***requestor*** is the network ID of the requestor of the RREQ packet, and *SN* is the sequence number of the RREQ packet.

When a node receives an RREQ packet from some node, or an RREP packet that this node is the second last hop in the route path of this RREP, it records the timestamp of when the packet is receiving in $T^{node}Recv_{(requestor,\ SN,\ from)}$, where ***node*** is the network ID of the receiver, ***requestor*** is the network ID of the requestor of the RREQ packet, ***SN*** is the sequence number of the RREQ packet, ***from*** is the network ID of the previous hop of the RREQ packet.

We can know that on a certain node, if some $T^{node}Send_{(requestor,SN)}$ is greater than a $T^{node}Recv_{(requestor,SN,from)}$ (with the same ***requestor*** and ***SN***), ($T^{node}Send_{(requestor,SN)}$ - $T^{node}Recv_{(requestor,SN,from)}$) is the time duration that ***node*** cost for processing RREQ packet of network ID ***requestor*** and sequence number ***SN***. Elsewhere, if some $T^{node}Recv_{(requestor,SN,from)}$ is greater than a $T^{node}Send_{(requestor,SN)}$ (with same ***requestor*** and ***SN***), ($T^{node}Recv_{(requestor,SN,from)}$ - $T^{node}Send_{(requestor,SN)}$) is the round trip time between node ***node*** and node ***from***.

Therefore, WP-DSR detects wormhole attacks by gathering all $T^{node}Send_{(requestor,SN)}$ and $T^{node}Recv_{(requestor,SN,from)}$ during the route discovery phase, and makes use of them to compute the traverse time between two neighbor nodes. For any two neighbor nodes X and node Y (Y receives an RREQ from X), the traverse time between X and Y, named $Duration_{(X,Y)}$, can be calculated by:

$$Duration_{(X,Y)} = \frac{1}{2}(T^{X}Recv_{(requestor,SN,Y)} - T^{X}Send_{(requestor,SN)} - (T^{Y}Send_{(requestor,SN)} - T^{Y}Recv_{(requestor,SN,X)})) \qquad (3\text{-}1)$$

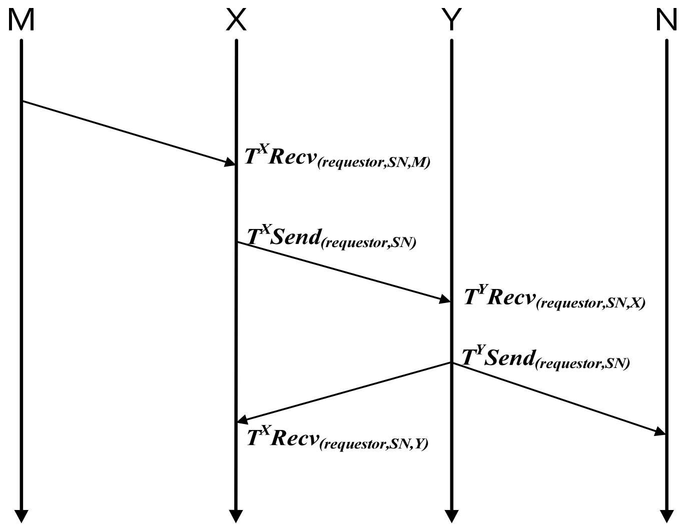

M
X
Y
N
$T^{X}Recv_{(requestor,SN,M)}$
$T^{X}Send_{(requestor,SN)}$
$T^{Y}Recv_{(requestor,SN,X)}$
$T^{Y}Send_{(requestor,SN)}$
$T^{X}Recv_{(requestor,SN,Y)}$

Figure 3.3: Illustration of $T^{node}Send_{(requestor,SN)}$ and $T^{node}Recv_{(requestor, SN, from)}$

Now, one problem is remained: how can node X obtain $T^{Y}Recv_{(requestor,SN,X)}$ and $T^{Y}Send_{(requestor,SN)}$ to know how long Y takes to process this RREQ? For this problem, WP-DSR chooses to embed ($T^{Y}Recv_{(requestor,SN,X)}$ - $T^{Y}Send_{(requestor,SN)}$) into the RREP packet; i.e., when Y transmits an RREP packet to X, Y will embed this value into the RREP packet to inform X the RREQ processing duration on Y.

But another problem emerges: Since the destination node of an RREQ packet does not rebroadcast it, it does not have a $T^{node}Send_{(requestor,SN)}$. Hence, on the destination node of an RREQ packet, $T^{node}Send_{(requestor,SN)}$ is replaced to $T^{node}Send_{(requestor,SN,\#)}$, which is the timestamp of when the number #st RREP packet with ***requestor*** and ***SN***. And because an RREP cannot embed its $T^{node}Send_{(requestor,SN,\#)}$ when it is sent out, a redundant RREP is necessary for bring the ($T^{node}Send_{(requestor,SN,\#)}$ - $T^{node}Recv_{(requestor,SN,from)}$) of the first RREP to node ***from***.

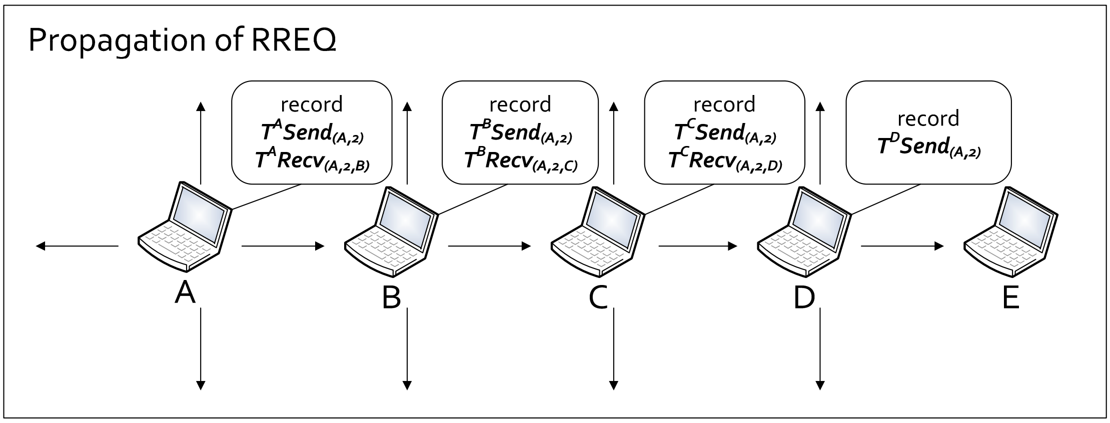
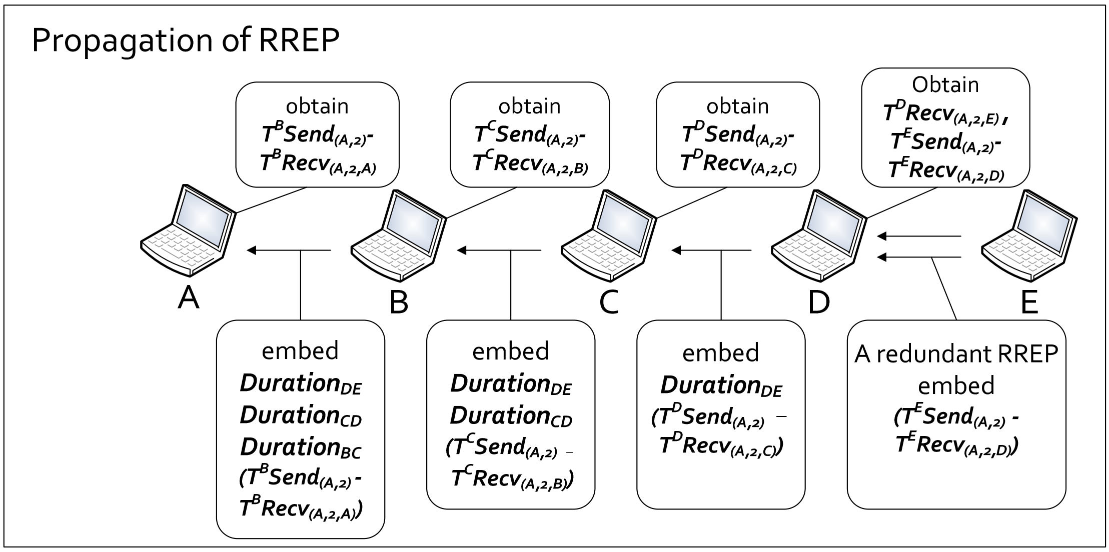


Figure 3.4: Example of the WP-DSR RREQ propagation

Now we demonstrate an entire route discovery procedure of WP-DSR. As illustrated in figure 3.4, we assume that node A wants to establish a route to node E, and the propagation of RREQ packets is as illustrated in figure 3.4. First node A initiates a new RREQ packet with requestor network ID “A”, the sequence number “2” and the destination network ID “E”. While it is broadcasting this RREQ packet, it records the timestamp of when the packet is sending out in $\boldsymbol{T^A Send_{(A,2)}}$. Afterwards, node B, a neighbor node of A, receives the RREQ packet, and records the timestamp of when it received the packet in $\boldsymbol{T^B Recv_{(A,2,A)}}$. After processing the RREQ packet as what is done in DSR, B rebroadcasts the RREQ packet and records the timestamp of when it sends out in $\boldsymbol{T^B Send_{(A,2)}}$. Later on, node A receives the RREQ and records the timestamp of when it received the packet in $\boldsymbol{T^B Recv_{(A,2,B)}}$. Although it has processed an RREQ packet with the same requestor and sequence number, it finds that it is the second last hop in the route path of the RREQ packet. Thus node A drops the RREQ packet, but keeps $\boldsymbol{T^B Recv_{(A,2,B)}}$ in its memory. And nodes B, C, D do the similar process, until the RREQ packet reaches its destination, node E.



Propagation of RREP
obtain $T^BSend_{(A,2)}$- $T^BRecv_{(A,2,A)}$
obtain $T^CSend_{(A,2)}$- $T^CRecv_{(A,2,B)}$
obtain $T^DSend_{(A,2)}$- $T^DRecv_{(A,2,C)}$
Obtain $T^DRecv_{(A,2,E)}$, $T^ESend_{(A,2)}$- $T^ERecv_{(A,2,D)}$
A
B
C
D
E
embed $Duration_{DE}$ $Duration_{CD}$ $Duration_{BC}$ ($T^BSend_{(A,2)}$ - $T^BRecv_{(A,2,A)}$)
embed $Duration_{DE}$ $Duration_{CD}$ ($T^CSend_{(A,2)}$ - $T^CRecv_{(A,2,B)}$)
embed $Duration_{DE}$ ($T^DSend_{(A,2)}$ - $T^DRecv_{(A,2,C)}$)
A redundant RREP embed ($T^ESend_{(A,2)}$ - $T^ERecv_{(A,2,D)}$)

Figure 3.5: Example of the WP-DSR RREP propagation

In figure 3.5, when node E receives the RREQ packet and finds itself is the destination node, it first initiates an RREP packet following the DSR, and then transmits the RREP packet to node D and records the timestamp when it is sent out in ***$T^{E}Send_{(A,2,1)}$***. Thereupon, node E sends a redundant RREP packet embedded with (***$T^{E}Send_{(A,2,1)}$ – $T^{E}Recv_{(A,2,D)}$***), the processing time of the first RREP packet, to node D. Node D records the timestamp of when the first RREP packet is received in ***$T^{D}Recv_{(A,2,D)}$***, and obtains (***$T^{E}Send_{(A,2,1)}$ – $T^{E}Recv_{(A,2,D)}$***) from the second RREP packet. Node D then can computes the traverse time, ***$Duration_{DE}$***, by $\frac{1}{2}$(***$T^{D}Recv_{(A,2,E)}$*** – ***$T^{D}Send_{(A,2)}$*** – (***$T^{E}Send_{(A,2,1)}$*** – ***$T^{E}Recv_{(A,2,D)}$***)). Next, D embeds $\mathrm{Duration}_{\mathrm{DE}}$ and (***$T^{D}Send_{(A,2)}$ – $T^{D}Recv_{(A,2,C)}$***), the processing time of the RREQ packet on node E, into the RREP packet and sends it to node C. So do node C and node E. Eventually, the RREP arrives at the requestor, node A. Node A checks if all the values of ***$Duration_{XY}$*** along the route path are less than a reasonable threshold. If yes, this route is a good route which does not pass through wormholes. But if any single value is larger than the threshold, this route is said to be contaminated by wormhole attacks and should not to be used.

Chapter 4

Evaluation and Simulation Results

This chapter is composes of three parts: Firstly, we will show the control packet overheads of WP-DSR protocol to that of the original DSR protocol. And then we will simulate the impact of wormhole attacks on DSR protocol on ns-2 [24]. At last, we will demonstrate the performance of WP-DSR, also on ns-2.

4.1 Overheads of WP-DSR

Having no modification on any DSR RREQ packets, the WP-DSR protocol has no overheads comparing to DSR protocol during the route request stage. In route reply stage, since the destination node of WP-DSR protocol requires a redundant RRER packet for informing the processing duration of destination node, the destination node creates one packet overhead comparing to the DSR protocol; i.e., if an RREQ packet is initiated, the overhead of packet numbers of WP-DSR protocol is only ***P***, where ***P*** is the number of disjoint paths found in this route discovery phase.

Besides, every node on the reverse route of RRER packet of WP-DSR protocol needs to carry the duration (4 bytes) of the RREQ processing delay on this node, and traverse durations (4 bytes for each) of all previous links; therefore, the amount of overhead bytes of each RREQ packet initiated in WP-DSR is:

$$\sum_{i=1}^{p}\left(\left(\left(\sum_{j=1}^{h_i-2}4j\right)+4(h_i-1)\right)+8+4h_i\right)=7p+2\sum_{i=1}^{p}h_i(h_i+1) \qquad \textbf{(4-1)}$$

where $\boldsymbol{h_i}$ is the number of hops of $\boldsymbol{i}$ th discovered path.

Figure 4.1 and figure 4.2 show the comparisons in overheads between DSR and WP-DSR protocols. The network size is 1000m*1000m, node transmission range is 150m, and the tunnel length is 4 hops. It is worth to notice that WP-DSR protocol, comparing to DSR, produces only a few overheads.

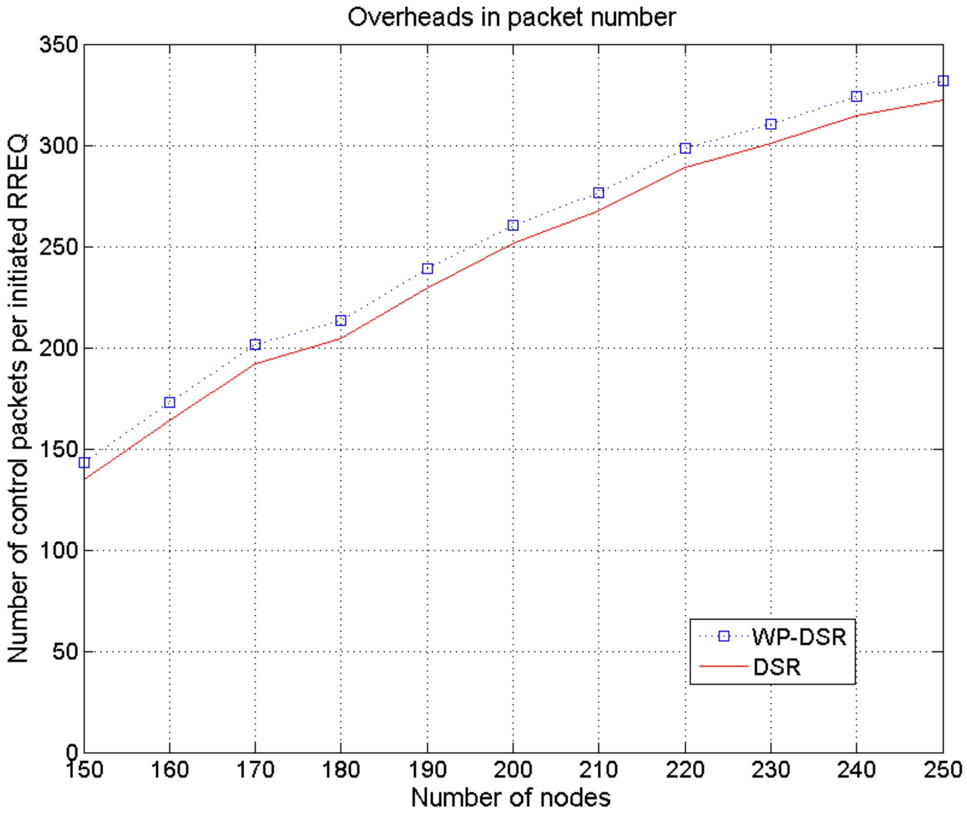


Figure 4.1: Overheads in packet number

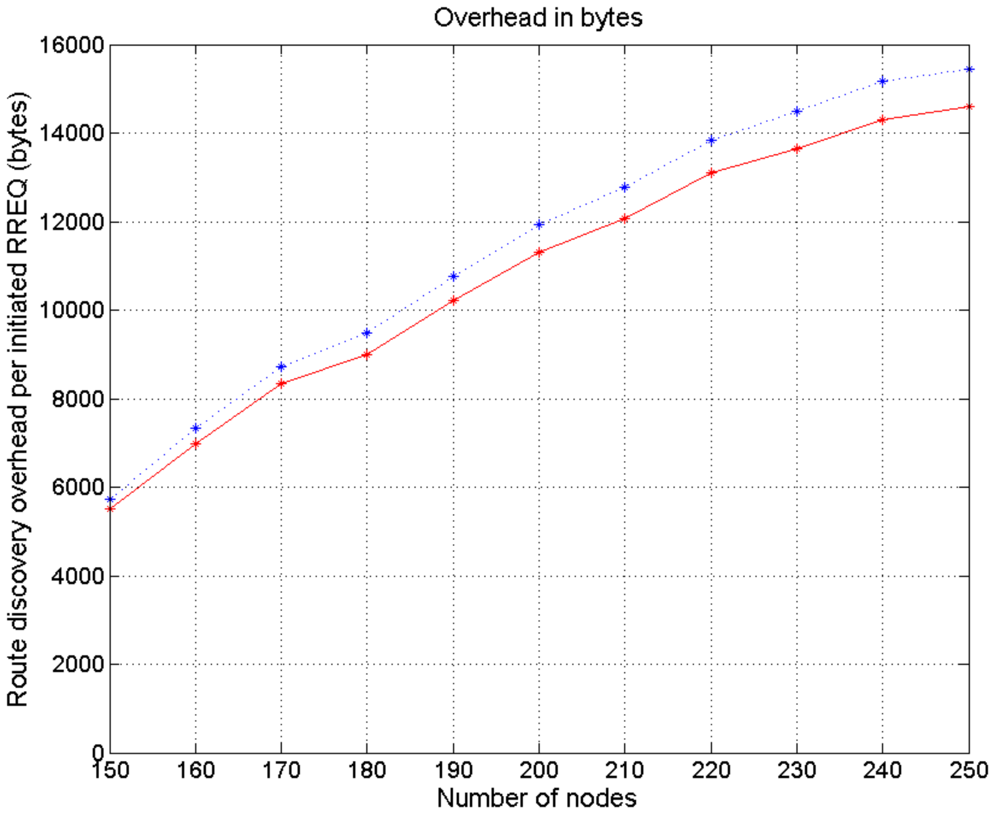


Figure 4.2: Overheads in bytes

4.2 Wormhole attacks impact on DSR

In this section, we demonstrate how many links are contaminated in DSR under wormhole attacks of only two wormhole nodes existing. We evaluate the impact rate by ns-2, where the network size is 1000m*1000m, node transmission range is 150m, and the tunnel length is 2 to 9 hops.

In figure 4.3, we can observe that if DSR has no any wormhole protection mechanisms, on average over 30% links can be spoiled; i.e., if the malicious attacks well lay down two collusive wormhole nodes, they can eavesdrop over 30% links of a network with ease.

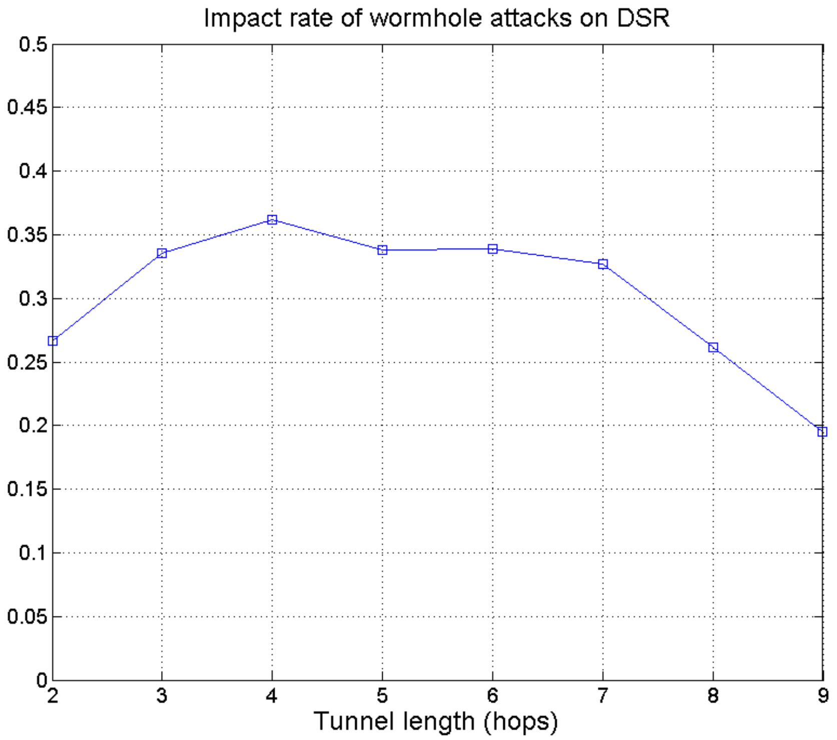


Figure 4.3: Wormhole attacks impact on DSR

4.3 Performance of WP-DSR

In this section, we evaluate the performance WP-DSR by ns-2. We assume random topologies of square size from 450*450m, 900*900m and 1500*1500m, and with 15, 30 and 50 regular nodes respectively. The transmission rages are 75m, 150m and 250m. The locations of two collusive wormhole nodes are randomly selected. Each variable is performed 100 times and averaged to avoid statistical bias.

Figure 4.4 shows the wormhole detection rate under different topologies. We can observe that WP-DSR can achieve on average over 90% detection rate under almost all scenarios. The detection rate of tunnel length 2 hops is inferior to other tunnel length. This can be explained by that short tunnel length means less traverse time, and leads more contaminated links to pass the detection mechanism.

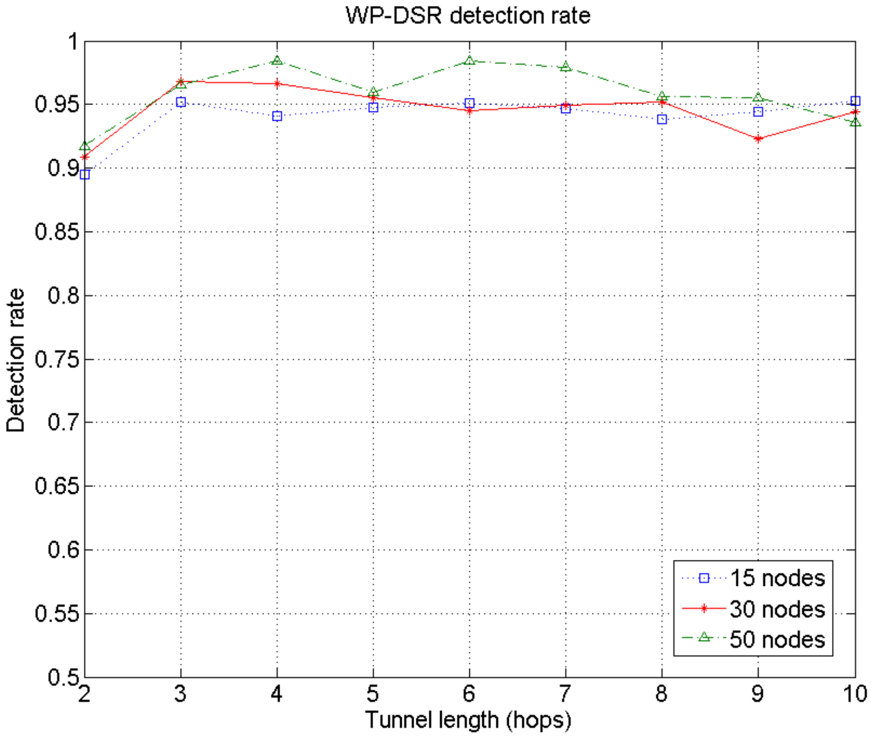


Figure 4.4: WP-DSR detection rate

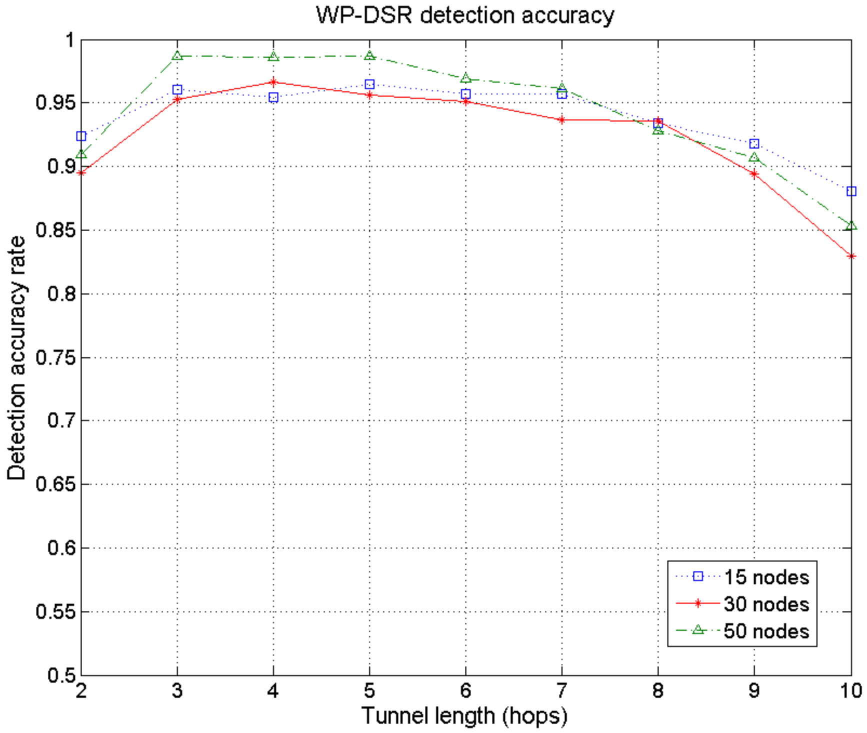


Figure 4.5: WP-DSR detection accuracy

Figure 4.5 displays the detection rate accuracy of WP-DSR. It shows that WP-DSR can attain over 90% detection accuracy rate under most situation. The reason of that tunnel length of 2 hops has little accuracy rate is similar to that of the detection rate. In addition, the reason of tunnel length over 8 hops shows little accuracy is that a longer tunnel length can spoil only fewer contaminated links than a reasonable shorter tunnel length. Thus, a single false positive may cause more impact on detection accuracy rate while the tunnel length is longer.

Chapter 5

Conclusion

In this thesis, we proposed a new on-demand routing protocol, Wormhole-Proof Dynamic Source Routing Protocol (WP-DSR), to detect wormhole attacks for wireless mobile ad-hoc networks. WP-DSR uses time limiting approach to detect wormhole attacks. It measures every hop-to-hop traverse time during route discovery phase. If any of these is larger than a reasonable threshold, the discovered route is contaminated by wormhole attacks. We have compared the routing overhead to original DSR, and simulated the detecting performance of our method. The results show that WP-DSR has a high detection rate and a low false positive rate, while requiring only a few overhead to DSR.

References

[1] A. D. Wood and J.A. Stankovic, “Denial of service in sensor networks,” *IEEE Computer*, vol. 35, no.10, pp. 54-62, 2002.

[2] J. F. Raymond, “Traffic Analysis: Protocols, Attacks, Design Issues and Open Problems,” *Proceedings of Design Issues in Anonymity and Unobservability Workshop*, pp. 7-26, Berkeley, California, USA, 2000.

[3] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, “A secure routing protocol for ad hoc networks,” *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pp. 78-87, Paris, France, 2002.

[4] Y. C. Hu and A. Perrig, and D. B. Johnson, “Ariadne: a secure on-demand routing protocol for ad hoc networks,” *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, pp. 12-23, Atlanta, Georgia, USA, 2002.

[5] Y. C. Hu and A. Perrig, “A survey of secure wireless ad hoc routing,” *IEEE Security & Privacy*, vol. 2, issue 3, pp. 28-39, 2004.

[6] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, “Security in mobile ad hoc networks: challenges and solutions,” *IEEE Wireless Communications*, vol.1, issue 1, pp. 38-47, 2004.

[7] P. G. Argyroudis, and D. O'Mahony, “Secure routing for mobile ad hoc networks,”

IEEE Communications Surveys & Tutorials,” vol.7, no.3, pp. 2 – 21, 2005.

[8] T. Clausen and P. Jacquet, “Optimized Link State Routing Protocol (OLSR), ” *IETF MANET Working Group*, Internet Draft 2003.

[9] D. Johnson, D. Maltz, and U. Hu, “The dynamic source routing protocol for mobile ad hoc networks,” *IETF MANET Working Group*, Internet Draft 2003.

[10] C. Perkins, E. Belding-Royer and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” *IETF MANET Working Group*, Internet Draft 2003.

[11] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang, “Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach,” *Proceedings of IEEE Wireless Communications & Networking Conference (WCNC)*, vol. 2, pp. 1193-1199, New Orleans, USA, 2005.

[12] M. Khabbazian, H. Mercier, and V. K. Bhargava, “Wormhole attack in wireless ad hoc networks: analysis and countermeasure,” *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, San Francisco, California, 2006.

[13]Y. C. Hu, A. Perrig, and D. B. Johnson, “Wormhole attacks in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 370-380, 2006.

[14] S. Capkun, L. Buttyan, and J-P Hubaux, “SECTOR: secure tracking of node encounters in multi-hop wireless networks,” *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 21-32, 2003.

[15] N. Sastry, U. Shankar, and D. Wagner, “Secure verification of location claims,” *Proceedings of ACM Workshop on Wireless Security (WiSe)*, pp. 1-10, San Diego, California ,USA, 2003.

[16] J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos, “Truelink: a practical countermeasure to the wormhole attack in wireless networks,” *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pp. 75-84, Santa

Barbara, California, USA, 2006.

[17] H. S. Chiu and K. S. Lui, “DelPHI: wormhole detection mechanism for ad hoc wireless networks,” *Proceedings of International Symposium on Wireless Pervasive Computing*, Phuket, Thailand, 2006.

[18] F. Na¨ıt-Abdesselam, B. Bensaou, and J.Yoo, “Detecting and avoiding wormhole attacks in optimized link state routing protocol,” *Proceedings of IEEE Wireless Communications & Networking Conference (WCNC)*, Hong Kong, 2007.

[19] W. Wang and B. Bhargava, “Visualization of wormholes in sensor networks,” *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, pp. 51–60, Philadelphia Pennsylvania, 2004.

[20] R. Maheshwari, J. Gao, and Samir R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," *Proceedings of IEEE Infocom*, Anchorage, Alaska, USA, 2007.

[21] L. Hu and D. Evans, “Using directional antennas to prevent wormhole attacks,” *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp. 131-141, San Diego, California, Baltimore, Maryland. 2004.

[22] I. Khalil, S. Bagchi, and N. B. Shroff, “MOBIWORP: mitigation of the wormhole attack in mobile multihop wireless networks,” *Proceedings of the Second International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2006.

[23] I. Khalil, S. Bagchi, and N. B. Shroff, “LITEWORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks,” *Proccedings of the International Conference on Dependable Systems and Networks (DSN)*, Yokohama, Japan, pp. 612-621, 2005.

[24] ns-2. [Online]. Available: http://www.isi.edu/nsnam/ns/ .