The Hong Kong Polytechnic University

Department of Computing

Secure Routing in Multi-hop Wireless Networks

by

Jie ZHOU

A thesis submitted in partial fulfillment of the requirements for

the Degree of Master of Philosophy

March 2012

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signature)

Jie ZHOU_____(Name of Student)

Abstract

Secure routing protocols play an essential role for ensuring security in multi-hop wireless networks. Specifically speaking, the entire network could be paralyzed by misdirecting routing control messages, which could lead to lower network throughput, frequent packet loss and eavesdropping. Thus routing protocols should be secure enough to defend from attack, yet optimal enough to ensure routing performance.

Most existing work on secure routing does not consider routing performance, nor does it adequately address the issues of providing users with information integrity and confidentiality. Moreover, current attack detection approaches make strong assumptions and require extra hardware support. In this research work, we study and propose solutions to address these challenging issues. We make the following original and significant contributions.

Firstly, we propose a Security Extended Optimised Link State Routing protocol (SE-OLSR) to guarantee the integrity, confidentiality and freshness of current OLSR. Previous routing protocols focus on improving performance with the assumption the wireless environment is friendly and trustworthy. However, the multi-hop wireless network is vulnerable to numerous attackers. Thus, we adopt basic security techniques to encrypt the routing packets, in order to ensure the packets received by the destination node are the original ones sent by the source node. At the same time, a digital signature and hash values are used to guarantee the packets are the latest ones

to prevent replay attacks. We implement the SE-OLSR on the Linux platform to identify its accuracy, and then transplant this secure routing protocol to mesh routers T902 and laptops to establish a Wireless Mesh Network (WMN) testbed.

Secondly, we analyse the impact of wormhole attacks and develop a countermeasure for attack detection based on a real testbed. Although many works have been done on detecting wormhole attacks, few of them actually evaluated their solutions on a testbed to consider real network conditions. In order to fill this gap, we set up a WMN testbed for studying wormhole attacks through comprehensive experiments. Some existing approaches used RTT to detect wormhole attacks. However, from both theoretical analysis and experimental results, we observed that the standard deviation of round trip time (stdev(RTT)) is a more efficient metric than RTT to identify wormhole attacks. Accordingly, we propose a new algorithm called Neighbour-Probe-Acknowledge (NPA) to detect wormhole attacks. Compared with existing works, NPA does not need time synchronisation or extra hardware support. Moreover, it achieves a higher detection rate and a lower false alarm rate than the methods using RTT under different background traffic load conditions.

Finally, we propose an Optimal Secure Routing (OSR) protocol to find a secure path resilient to active attack with the best routing performance. Traditional routing protocols are designed to efficiently find paths containing high quality links in assumed trust environments. Although several routing schemes have recently been proposed as defence from attack, with increasing attention on security issues in the application of multi-hop wireless networks, only a few of these have considered

routing performance. To fill this gap, we have designed a new secure routing protocol OSR taking into consideration routing performance optimisation. OSR relies on a trusted third party, Trust Clearance Center (TCC), which utilises game theory to calculate and assign a trust value for each node according to its utility report behaviour. We prove that this TCC is able to detect malicious nodes and segregate them from the network when they try to launch attacks. Therefore, optimal paths can be discovered by OSR without any utility cheating. Through extensive simulations, we demonstrate that OSR can effectively discover optimal paths with a high detection rate and a low false alarm rate. Furthermore, we observe that the behaviour of active attacks can be comprehensively formulated by using game theory. To the best of our knowledge, this is the first piece of work that adopts game theory to deal with problems that jointly consider security and routing performance.

Publications

Journal Papers

1. **Jie Zhou**, Jiannong Cao, Analysis and Countermeasure for Wormhole Attacks in Multi-hop Wireless Networks, invited to *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, a selected paper of *AINA-2012*.

Conference Papers

1. **Jie Zhou,** Jiannong Cao, Jun Zhang, Chisheng Zhang, and Yao Yu, Analysis and Countermeasure for Wormhole Attacks in Wireless Mesh Networks on a Real Testbed, *26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012)*, March, 2012, Fukuoka, Japan.

2. **Jie Zhou,** Jiannong Cao, Tao Li, OSR: Optimal and Secure Routing Protocol in Multi-hop Wireless Networks, *Ninth Workshop on Wireless Ad hoc and Sensor Networks (WWASN2012)*, June, 2012, Macau, China, submitted.

3. Jiannong Cao, Chisheng Zhang, Jun Zhang, Yueming Deng, Xin Xiao, Miao Xiong, **Jie Zhou**, Yang Zou, Gang Yao, Wei Feng, Liang Yang, and Yao Yu, SHAWK: Platform for Secure Integration of Heterogeneous Advanced Wireless Networks, *Eighth International Symposium on Frontiers of Information Systems and Network Applications (FINA-2012)*, March, 2012, Fukuoka, Japan.

4. Chisheng Zhang, Jiannong Cao, Jun Zhang, and **Jie Zhou**, Cost-constrained Incremental Network Planning in Multi-hop Wireless

Networks, *Global Communications Conference (GLOBECOM-2011)*, December, 2011, Houston, USA.

Acknowledgements

First and foremost, I would like to express my immense gratitude to my supervisor, Dr. CAO Jiannong, for his patience, systematic guidance, illuminating advice and encouragement in my research and project work and the writing of this thesis. Especially, Dr. Cao enlightened me on personal capabilities as well, which has helped me to improve myself in becoming not only a good research student, but also an outstanding individual.

I also appreciate Dr. ZHANG Jun, Mr. ZHANG Chisheng, and Ms. YU Yao for their insightful suggestions on my research. Moreover, I want to thank them for their kind support and help in my project work. Especially for Mr. ZHANG Chisheng, wish your baby boy a healthy body, a happy life and a bright future!

I would also like to thank Dr. XIE Kun and her husband Mr. WEN Jigang. I am heartily grateful for their help when I first arrived in Hong Kong. I would also like to thank my friends Mr. LI Tao, Ms. LI Jingjing, Ms. ZHOU Binbin, Ms. XIONG Miao, Ms. FEI Ning, Dr. FENG Wei, Mr. XIAO Xin, Mr. YANG Liang, Mr. YAO Gang, Mr. YANG Lei, Mr. MA Jun, and all other members of our research group that I cannot elaborate on here. I would like to thank them for their help with my research work. In addition, I want to thank them for their kindness, which helped me a great deal in satisfactorily handling problems in my life.

Another person I would like to thank is my previous roommate, Ms. LIU Xuan. I owe you a debt of gratitude for your help when I was sick. Thanks for your intensive care when I suffered from acute bronchitis. Staying with you made me feel like I was back in my home town. It was an irreplaceable time that I spent together with you in the student hall.

Finally and most importantly, I would like to thank my parents. Their endless love, constant support, and unwavering confidence in me is what has driven me forward to make this far, and to continue on the road ahead. I am proud of my parents and will love them forever.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

MANET: Mobile Ad-hoc Network

WMN: Wireless Mesh Network

WSN: Wireless Sensor Network

SE-OLSR: Secure Extended OLSR

OLSR: Optimised Link State Routing

AODV: Ad hoc On-Demand Distance Vector

OSR: Optimal and Secure Routing

Chapter 1. Introduction

In this Chapter, we provide an introduction to our research, discuss the characteristics of multi-hop wireless networks and secure routing, especially in Wireless Mesh Networks, as well as security goals and open research issues. In Section 1.1, we give an overview of the multi-hop wireless network and then study the security requirements and goals for secure routing issues in Section 1.2. We also point out the open research issues on secure routing in Section 1.3, to emphasise the importance of this topic. We illustrate our contributions and highlight their significance in Section 1.4. Finally, a brief outline of the thesis is given in Section 1.5.

1.1. Overview

A multi-hop wireless network - a network composed of numerous mobile nodes - can be deployed instantly, and provides network communications for highly mobile users, without a need for pre-established infrastructure. The typical technologies of multi-hop wireless networks range from Mobile Ad-hoc NETwork (MANET), Wireless Mesh Network (WMN) to Wireless Sensor Network (WSN), which have already received wide publicity in the past few years, because of their irreplaceable application values in the areas of the military, research, and the commercial market, etc. WMN has emerged as a key technology for next-generation wireless communication [SV05]. Contributing to the increasing use of WMN, security turns into an urgent issue and attracts a great deal of attention. Furthermore, routing protocol plays an important role in wireless networks to establish network topology.

The existing secure routing protocols proposed for ad hoc networks can be adopted for WMN, but most are not mature enough to fit within its features. The differences between these two kinds of networks usually render new secure routing solutions specifically for WMN. Therefore, the main objective of this research is to investigate the issues of secure routing in multi-hop wireless networks, analyse existing and new attacks according to their characteristics, and design novel algorithms, techniques and protocols for secure routing in multi-hop wireless networks.

Wireless mesh networks (WMNs) are dynamically self-organised and self-configured, with the nodes in the network automatically establishing an ad hoc network and maintaining mesh connectivity. WMNs are comprised of two types of nodes: mesh routers and mesh clients. All the mesh routers form the mesh backbones. Other than the routing capability for gateway/bridge functions as in a conventional wireless router, a mesh router contains additional routing functions to support mesh networking. Through multi-hop communications, a mesh router with much lower transmission power can achieve the same coverage. To further improve the flexibility of mesh networking, a mesh router is usually equipped with multiple wireless interfaces built on either the same or different wireless access technologies. In spite of all these differences, mesh and conventional wireless routers are usually built based on a similar hardware platform. This feature brings many advantages to WMNs, such as easy network maintenance, robustness, reliable service coverage, etc.

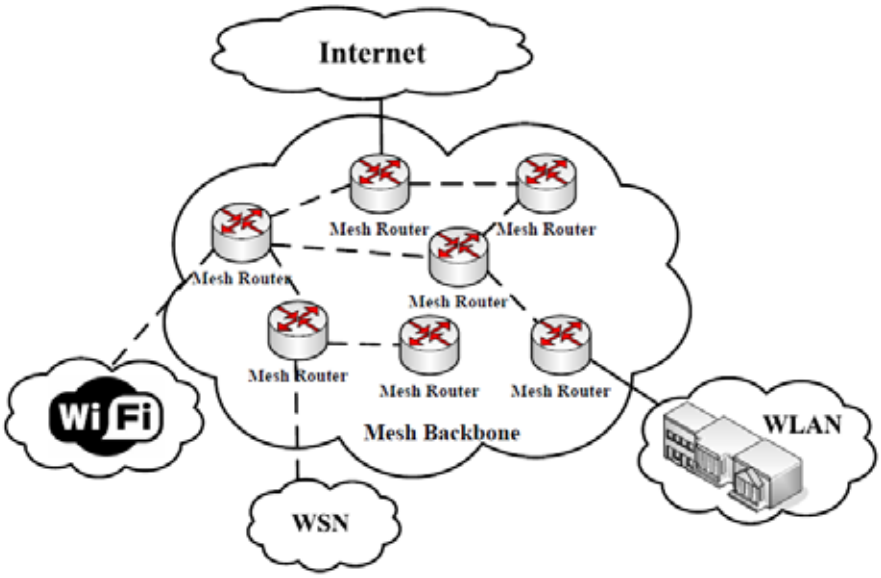

Figure 1.1: The wireless mesh network

WMN has been accepted in the traditional application sectors of ad hoc networks according to its characteristics. More importantly, WMNs are undergoing rapid commercialisation in many other application scenarios, such as broadband home networking, community networking, building automation, high-speed metropolitan area networks, and enterprise networking. However, there are some challenging issues hindering the development of a WMN which makes it difficult to be widely applied. One of the major issues is that it lacks a comprehensive scheme to guarantee its security. Therefore, it is obvious that WMN is vulnerable to many attacks from both inside and outside of the network. This creates considerable challenges, and requires novel solutions to address it.

1.2. Secure Routing in Wireless Mesh Networks

As is known to all, nodes can only communicate with one another and receive services by being equipped with a high quality routing protocol to establish and maintain accurate paths. Routing protocols should be robust not only against dynamic changing topology, but also malicious attackers, in order to achieve their security goals. The primary goal of a WMN routing protocol is to establish a correct and efficient route between a pair of nodes. The entire network topology will be paralysed by misdirecting routing control messages. Therefore, routing security plays an important role in the security of the multi-hop communication network. Traditionally, security implies detecting potential attacks and defenses against them. Such functionality can always be associated with encryption, digital signatures and key management. However, they are only the basic techniques for security. Thus, in this subsection, we illustrate the security goals first and then study the attacks in multi-hop wireless networks.

Not just in WMN but also in other applications, it is desirable to ensure network security as well as to achieve the following goals. Confidentiality, integrity and availability, also called C.I.A., are the three most fundamental aspects of the information system security process.

Confidentiality: The concept of confidentiality is the assurance that sensitive data is being accessed and viewed only by those who are authorised to see it.

Integrity: Ensuring the data content or correspondences are preserved intact, through the transfer from sender to receiver. Integrity embodies the guarantee that a message sent is the message received, that is, it was not altered either intentionally or unintentionally during transmission.

Availability: In networking, availability refers to those elements that create reliability and stability in networks and systems, assuring that connectivity is functioning as needed so that authorised users have access to the network systems, such as fault tolerance.

To make information secure, we must protect information and information systems from unauthorised access, use, disclosure, disruption and modification. In wireless networks, especially the multi-hop wireless networks, the disclosure of system and dynamic topology and so on will make the network unsafe. However, the widespread use of wireless ad hoc networks and wireless mesh networks make confidential information held by governments, military, corporations, hospitals and private businesses very vulnerable. That is the primary motivation of our research on the security aspect for the preparation and development of these secure approaches.

In addition, a new requirement called Authenticity becomes more and more important. It enables a node to ensure the identity of the peer node it is communicating with. Without authenticity, an adversary could masquerade as a node, thus gaining unauthorised access to resources and sensitive information, and interfering with the operation of other nodes. With the implementation of the concepts such as an

ubiquitous system, the abundance of networking nodes is reasonable. All of these nodes should have authentic communication within the network. The usual authentication mechanisms involve a centralised system which administers restrictions on the basis of an access list or capability certificates. In a mesh network, the presence of such a server is sometimes not possible.

The objective of designing routing protocols includes self-configuration of routing tables, self-adaptation on the wireless link diversity, and the realisation of maximising each performance metric. However, the requirement of self-configuration and self-adaptation asks for co-operation among all network nodes. In addition, each node should run the protocol continuously at the same time. Therefore, a lack of efficient and effective protection will make the protocols work improperly. In multi-hop wireless networks, routing attacks can be categorised into passive and active attacks [RS02] in multi-hop wireless communication networks. A passive attack only attempts to discover valuable information by monitoring routing traffic, which makes it very difficult to detect, while an active attack intends to improperly modify data, gain authentication by injecting false packets, or modify packets transition through the network. The active attacks can be further divided into outside and inside attacks. The first one, generated from outside of the network, threatens the network by injecting erroneous routing control message, replaying old routing information, or distorting routing tables. The latter is much more severe because the misbehaviour nodes (also called compromised nodes) are already part of the network; they are able to generate valid signatures using their private keys. Therefore, simply using cryptography methods cannot defend against such an attack. Once this kind of node

distributes an incorrect routing message, it is more difficult to detect. Since this thesis will mainly focus on dealing with inside attacks, we will next illustrate the major kinds of inside attacks and provide some explanations.

Blackhole attack: This kind of attack breaks services by dropping packets. It can drop all of the packets, or selectively forward packets (grayhole attack). Traditionally, malicious nodes advocate that they have the ideal path to a destination, such as the shortest path. Therefore, they can attract flows and then intensively drop packets. As shown in Figure 1.2, originally node S is four hops away from node D and S can communicate with D successfully. However, once a malicious node X exists, it will continuously send messages to all the nodes within its transmission range, and tell them it has the shortest path to node D. Consequently, both nodes S and A will believe that X is the next hop on the path towards node D. Then, X could gain control of all of the packets that S wants to send to D. If it chooses to drop all of them, it becomes a blackhole attack. Correspondingly, if it decides to drop the packets selectively, it is a grayhole attack.

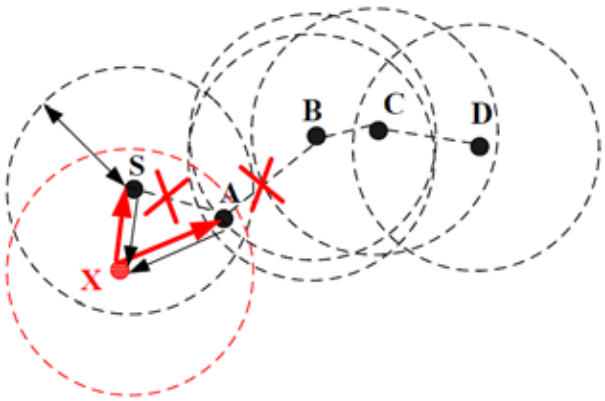

Figure1.2: An example of blackhole/grayhole attack

Grayhole attack: This attack is a kind of subset of a blackhole attack. Malicious nodes only send packets partially and drop others - named selective forwarding. If it drops all the packets, it will become the blackhole attack previously mentioned. For example [GRAY08], a malicious node can selectively forward packets specifically originating from a single source or a range of IP addresses. To state this further, nodes which perform as a gray hole can set up a so-called customised filter to drop whatever packets they want. Karlof et al. [SR03] first studied the selective forwarding attack and suggested that this can be countered by using multi-path forwarding.

Wormhole attack: The wormhole attack is one of the most powerful and severe attacks in multi-hop wireless networks. Although traditional methods such as cryptography and digital signatures can prevent malicious nodes from compromising the integrity and confidentiality of routing packets, a wormhole attack is transparent to these methods. Malicious nodes can work together to undermine the routing procedure. For example, as shown in Figure 1.3, a wormhole attack can be launched by two colluding malicious nodes X and Y. They first establish a secret tunnel called a wormhole link by using out-of-band channel, packet relay or other similar methods, and become endpoints of the wormhole link. Then, each overhears and captures packets in its transmission range and sends the packets to its counterpart, which replays the packets to another distant place in the network. As a result, two faraway nodes S_1 and S_2 will believe that they are one-hop neighbours. Consequently, X and Y can control all of the traffic passing through the wormhole link. Wormhole attacks can lead to other attacks such as packet dropping, eavesdropping, and DoS. When malicious nodes cooperate with each other to launch wormhole attacks in a more

complex network, they will become increasingly severe [WRS10]. Wormhole attacks can be classified from the perspective of what technology they have adopted: wormhole using encapsulation, wormhole using out-of-band channel, wormhole with high power transmission, wormhole using packet relay, and wormhole using protocol deviations. Based on the visibility of a wormhole in the network, we can categorise it as: open wormhole attack, closed wormhole attack, and half open wormhole attack [FIW09]. In this thesis, we are dedicated to studying closed wormhole attacks using out-of-band channel and packet relay. Yih-Chun Hu et al. [PL03] first discussed wormhole attacks and proposed a packet leashes approach, which could detect and then defend against this attack. However, since this is the first work to be proposed at the very beginning of research into wormholes, it requires strong assumptions, such as time synchronisation and location information.

It is a challenging task to design solutions for detecting wormhole attacks. Existing works [PL03] [DEL06] [DAW08] proposed techniques to detect wormhole attacks from the perspective of time mismatch, geographic mismatch, statistical mismatch, etc. Most of the works evaluated their methods by simulations rather than experiments on real testbeds. Few of these studied the impact of wormhole attacks in a real WMN environment, taking complicated network conditions into consideration. This fact motivates us to build a real testbed and closely analyse wormhole attacks in WMNs. This attack will be studied in Chapter 4.

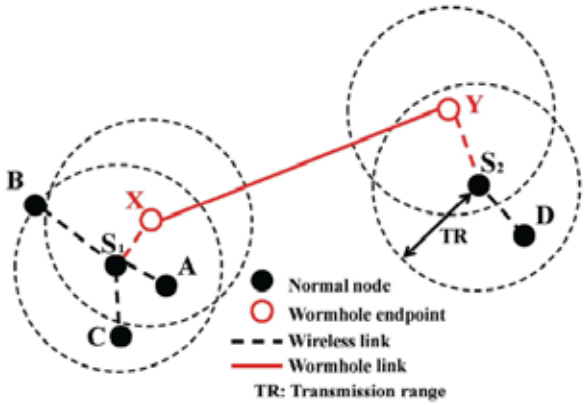

Figure 1.3: An example of wormhole attack

1.3. Open Research Issues

Except for the discussed issues, there are still numerous challenges remaining in terms of security in multi-hop wireless networks. First, current secure routing protocols are well-developed for defending outside attacks. A more comprehensive protocol should be taken into consideration against both inside and outside attacks. In addition, such a routing protocol only concerns the security aspects that ignore routing performance. Therefore, another problem is how to design strong routing which also has high routing performance.

1.4. Contribution of the Thesis

The contributions of this thesis mainly lie in designing novel algorithms, techniques, mathematical models, and practical testbed for secure routing in multi-hop wireless networks. As illustrated in Figure 1.4, our contributions are listed as follows.

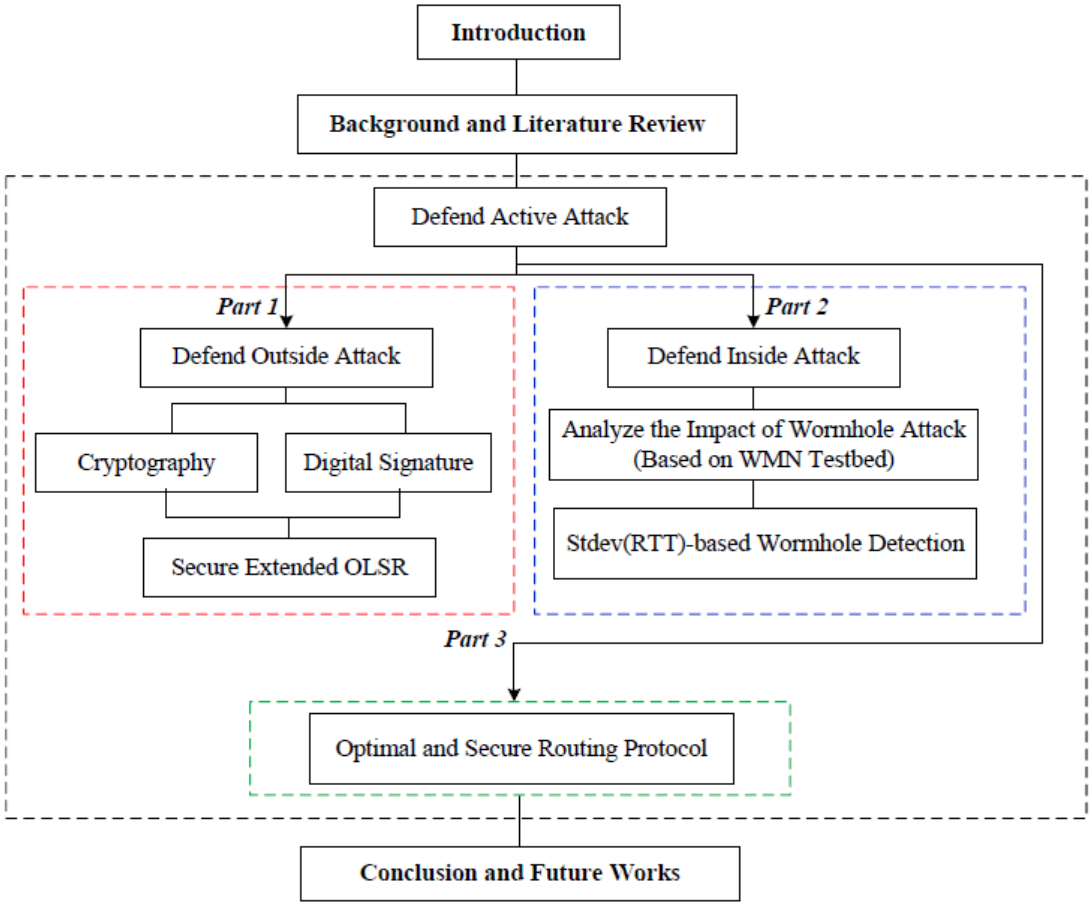

Figure 1.4: The Outline of the contributions in this thesis

1.4.1. Secure Extended OLSR

With the respect to the integrity and confidentiality of the routing messages, we propose a secure scheme based on OLSR to guarantee the security of routing information.

Our SE-OLSR can defend outside attacks while guaranteeing the confidentiality and integrity of messages by using cryptography and hash methods. Then, we run it on our mesh network testbed. The strategy of defending outside attacks is almost mature.

However, few current works have been implemented on a real testbed. Therefore, our contribution in both design and implementation of our SE-OLSR is significant.

1.4.2. Analysis and Countermeasure for Wormhole Attack

The wormhole attack is a severe attack in Wireless Mesh Networks (WMNs). It involves two or more wormhole endpoints colluding to capture traffic from one place in the network and replaying it to another faraway place through a secret tunnel, so as to distort network routing. It may lead to even more serious threats such as packet dropping and denial of service (DoS). Although a great deal of work has been done on detecting wormhole attacks, few of these have evaluated their solutions on a testbed to consider actual network conditions. In this thesis, we set up a WMN testbed for studying wormhole attacks in order to fill this gap. Some existing approaches have used RTT to detect wormhole attacks. However, from both theoretical analysis and experimental results, we observed that the standard deviation of round trip time (stdev(RTT)) is a more efficient metric than RTT to identify wormhole attacks. Accordingly, we propose a new algorithm called Neighbour-Probe-Acknowledge (NPA) to detect wormhole attacks. Compared with existing works, NPA does not need time synchronisation or extra hardware support. Moreover, it achieves a higher detection rate and a lower false alarm rate than the methods using RTT under different background traffic load conditions.

1.4.3. Optimal and Secure Routing Protocol in Multi-hop Wireless Network

Most existing works did not consider routing performance at the time of designing secure routing protocols. We analyse the characteristics of attacks and utilise the idea of utility-based routing protocols to satisfy both security and routing performance. Thus, we propose an Optimal and Secure Routing (OSR) protocol to find a secure path that is resilient to active attacks with the best routing performance. OSR is a utility-based routing protocol that uses ***Trust Value*** to prevent attacks. Moreover, previous works adopted game theory to investigate selfish nodes, but few utilised this mathematical tool to formulate malicious behaviour. We use game theory to prove that the path found by OSR is secure and optimal. To the best of our knowledge, we are the first to study this problem from the perspective of game theory method. The extensive simulation results demonstrate that OSR is able to successfully find an optimal routing path that contains no attackers.

1.5. Outline of the Thesis

The remainder of this thesis is organised as follows. Chapter 1 is the introduction to this thesis. Chapter 2 briefly presents the literature review of relevant research topics. The main body of this thesis is divided into three parts, from Chapter 3 to Chapter 6. Finally, we conclude the thesis with a discussion of the direction of our future work in Chapter 7. The details of Chapter 3 to Chapter 6 are presented as follows.

In the first part, we mainly discuss the secure scheme to guarantee the integrity and confidentiality of routing messages based on Optimised Link State Routing protocol. In Chapter 3, we propose our own security strategy that adopted both encryption and a digital signature to satisfy the requirements of integrity and confidentiality.

In the second part, we thoroughly analyse the wormhole attack. This part is composed of two chapters. In Chapter 4, we establish a real WMN testbed to observe the impact of wormhole attacks through a number of experiments. Based on our observations, in Chapter 5, we propose a NPA algorithm to detect wormhole attacks. The evaluation results demonstrate that our NPA can detect wormhole attacks effectively and efficiently.

Finally, we design a comprehensive routing protocol resilient to attacks in the third part. In Chapter 6, we propose an optimal and secure routing (OSR) protocol for a multi-hop wireless network to segregate malicious nodes, as well as to find the paths for corresponding source-destination node pairs with maximum routing utility value. We utilise game theory methods to investigate the behaviour of active attacks, and develop a centralised method to issue the node's trust value to guarantee the security of utility-based routing selection procedure. Based on both analytical and experimental results, the OSR can achieve a nearly 100% attack detection rate and satisfy the best-utility selection requirement.

Chapter 2. Background and Literature Review

In this Chapter, we provide a literature review and some background knowledge related to the research in this thesis. The organisation of this Chapter is as follows. First, Section 2.1 presents an overview of security issues in routing in WMN. We discuss the classification of current approaches in Section 2.2. Section 2.3 presents secure solutions based on existing routing protocols. We then make a brief introduction of attack detection and defense in Section 2.4. Finally, Section 2.5 gives an investigation of current research that takes into consideration both routing security and performance.

2.1. Security Issues in Routing in Multi-hop Wireless Networks

Multi-hop wireless networks are vulnerable to numerous threats due to their shared wireless media and multi-hop communication features. Therefore, we intend to discuss security issues from the following four aspects: Security Goals, Existing Attacks, Basic Techniques, and Possible Solutions. Speak of Security Goals, traditionally, they can be defined as follows [SA99]. Authentication ensures that the other ends of a connection or the originator of a packet is the node that is claimed. Access control prevents unauthorised access to a resource. Confidentiality protects overall content or a field in a message (or prevents an adversary from undertaking traffic analysis). Privacy prevents adversaries from obtaining information that may have private content. Integrity ensures that a packet is not modified during

transmission. Authorisation authorises another node to update information or to receive information. Anonymity hides the source of a packet or frame. Non-repudiation proves the source of a packet. Freshness ensures that a malicious node does not resend previously captured packets. Availability mainly targets DoS attacks, and is the ability to sustain the networking functionalities without any interruption due to security threats. Resiliency from attacks is required to sustain a network's functionalities when a portion of its nodes are compromised or destroyed.

Existing attacks can be categorised from the perspective of their attack purposes. Physical attack aims to disrupt the physical devices, targeting them by making them fail to work appropriately. Physical destruction and electromagnetic pulse are common attacks. Attack against availability is a most significant attack, which should be highly concerning. For a deep investigation, this attack can be further discussed from the perspective of five network layers. The first one is DoS in the physical layer. As is known to all, Wireless IEEE 802.11 protocol is vulnerable to many over-the-air attacks, such as Physical layer Radio Frequency Jamming. For supporting continuous transmission in the IEEE 802.11 wireless spectrum, all legitimate devices will always believe the medium is occupied and then reply, thus launching a denial of service attack. The second attack is DoS in the data link layer. CTS Jamming, False RTS/CTS, and ACK spoofing are the most well known attacks. The first two attacks are another kind of RF Jamming called Virtual Attacks. The malicious nodes send legitimate packets with multiple CTS replies after getting RTSs (CTS Jamming) or obeying CSMA/CA with a high duration value (False RTS/CTS). Based on the same idea, ACK spoofing stands for reply numerous ACK messages to occur DoS attack. The

third one is about attacks against routing protocols. Currently, this kind of attack aims to disrupt routing tables, distort routing information, or insert outdated routing entries. For example, spoofed, altered or replayed routing information, Hello flood attack, Wormhole attack[Worm06], Sybil attack [SA04], Black hole attack [SUV04], Gray hole attack [SUV04], Rushing attack[RSH03], etc.

In this research, we aim to study attacks against routing protocols. Therefore, we would like to introduce the classification of existing attacks first. As shown in Figure 2.1, attacks can be classified as passive and active. A passive attack does not disrupt the operation of a routing protocol, but only attempts to discover valuable information by listening to routing traffic, which makes it very difficult to detect. An active attack is an attempt to improperly modify data, gain authentication, or procure authorisation by inserting false packets into the data stream or modifying packets transition through the network.

Active attacks can be further divided two kinds: outside attack and inside attack. The first comes from external attackers. By injecting erroneous routing information, replaying old routing information, or distorting routing information, an attacker could successfully partition a network or introduce excessive traffic load into the network by causing retransmission and inefficient routing. The inside attack is a severe threat coming from compromised nodes, which could advertise incorrect routing information to other nodes. Detection of such incorrect information is difficult, as merely requiring routing information to be signed by each node would not work, because compromised nodes are able to generate valid signatures using their private keys.

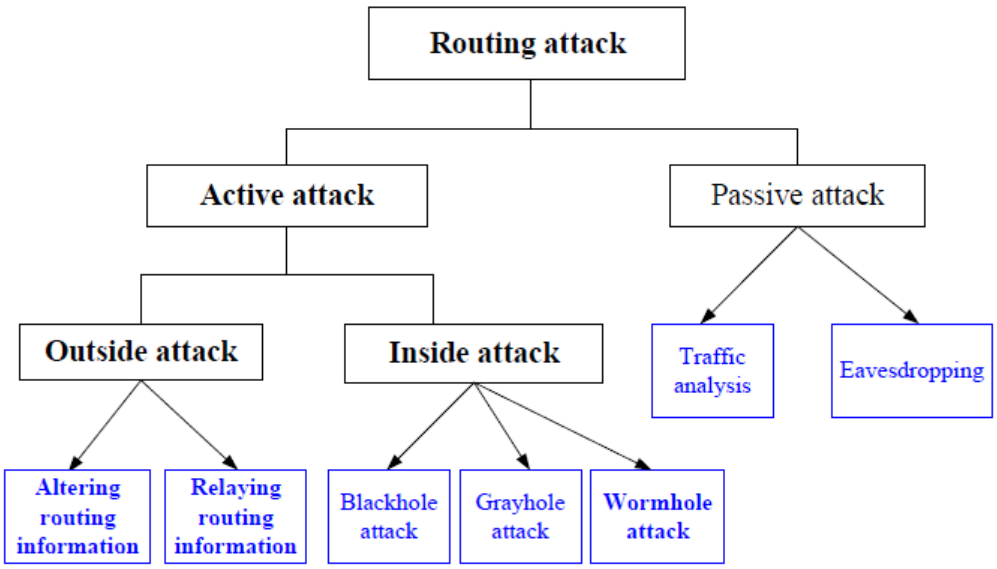

Figure 2.1: Classification of routing attacks

After introducing the classification of current attacks, we would like to briefly discuss current approaches to solve the security problems.

Basic Techniques for Security are the general ones adopted and well-developed. Symmetric encryption, asymmetric encryption, hash functions, digital signature, and message authentication can be utilised in both wired and wireless networks to encrypt messages and verify message accuracy. Due to the fact the former has been well studied, key distribution, exchange and management becomes a hot topic. Intrusion detection is a technique to defend outside attacks. Works done by [IDS08], [OPE09], and [LID08] are the most recent ones, which have a profound effect on this field.

Possible Solutions Reviewing the literature, there has been a substantial amount of work utilising basic techniques for routing security to design secure routing protocols. [SEAD03], [SAR02] adopt a symmetric cryptography approach, asymmetric cryptography approach and hybrid approach respectively, while [SOLSR04] applies a digital signature. Apart from that, Watchdog and CONFIDANT are reputation-based solutions and [IDW00] is a kind of intrusion detection. However, all the methods discussed aim to defend against outside attacks. As these techniques become increasingly mature, how to defend against inside attack becomes more challenging, Packet Leashes[PL03] first introduced a solution against wormhole attacks using time-synchronisation and GPS positioning to identify wormhole links. Another widely used solution is to monitor neighbours to discover abnormal behaviours. Byzantine failure resilience [BSMR09] focuses on defending attacks, which does not crack all suspicious links, but avoids them after finding them out. As for anti-jamming, the most recent work is studied by [WAJ06]. It is inspired by the impact and characteristic of wormhole attacks to defend against jamming attacks. Onion routing [ONION] is the most traditional solution to achieve anonymous routing.

2.2. Classification of Secure Routing Approaches

Existing secure routing approaches can generally be classified based on different kinds of attacks and corresponding techniques. In Figure 2.2, we give a classification of existing secure routing approaches.

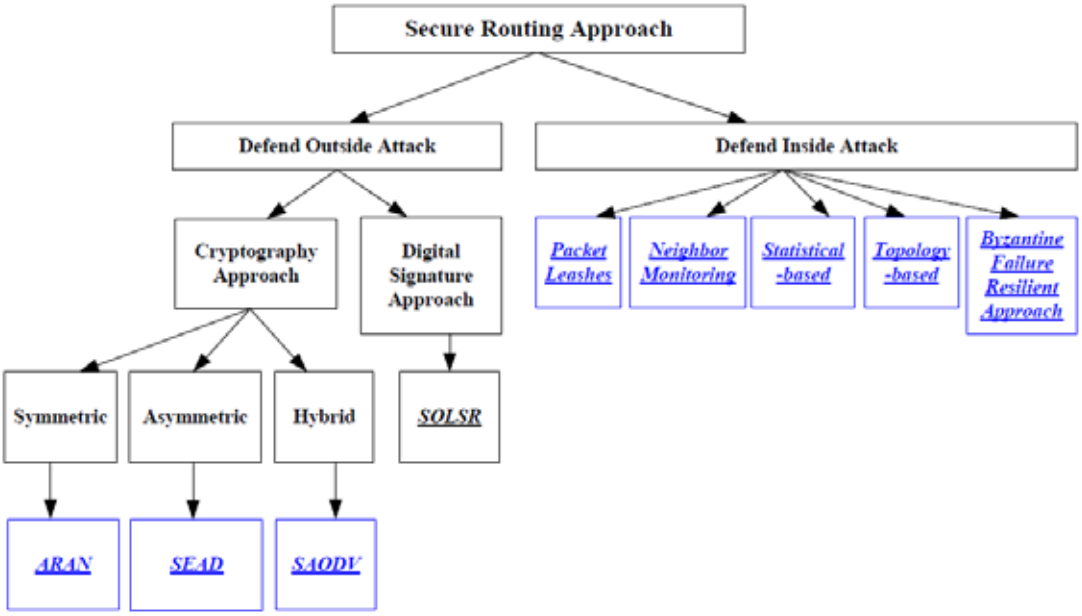

Figure 2.2: Classification of secure routing approaches

2.3. Security Extension on Current Routing Protocol

In wireless multi-hop networks, routing protocols should be robust not only against dynamic changing topology and malicious attackers, but also compromised nodes in order to achieve the C.I.A. requirement. The primary goal of designing a secure routing protocol is to establish a correct and efficient route between a pair of nodes, so that transformed messages will be delivered safely. If routing can be misdirected by external attackers, the entire network can be paralysed. We now discuss related work on security solutions.

2.3.1. Secure Ad hoc On-demand Distance Vector Routing (SAODV)

Secure Ad hoc On-demand Distance Vector (SAODV) is a proposal for security extensions to the AODV protocol [SAODV02]. The proposed extensions utilise

digital signatures and hash chains in order to secure AODV packets. In order to facilitate the transmission of the information required for the security mechanisms, SAODV defines extensions to the standard AODV message format. These SAODV extensions consist of the following fields. The hash function field identifies the one-way hash function that is used. The field max hop count is a counter that specifies the maximum number of nodes a packet is allowed to go through. The top hash field is the result of the application of the hash function max hop count times to a randomly generated number, and finally the field hash is this random number.

The authors mention that the main problem with securing an on-demand protocol like AODV is that it allows intermediate nodes with fresh routes to reply to a route query, since the reply has to be signed on behalf of the destination node. In order to overcome this problem, the authors suggest two solutions. The first one is to forbid intermediate nodes to respond to route request messages, since they cannot sign the message on behalf of the final destination. The second solution involves the addition of the signature that can be used by intermediate nodes to reply to a route request by the node that originally created the route request.

2.3.2. Secure Efficient Ad hoc Distance Vector Routing (SEAD)

The Secure Efficient Ad hoc Distance vector (SEAD) [SEAD03] is a secure ad hoc network routing protocol based on the design of the Destination-Sequenced Distance-Vector (DSDV) [DSDV94] algorithm. In order to find the shortest path between two nodes, the distance vector routing protocols utilise a distributed version of the Bellman-Ford algorithm. The SEAD routing protocol employs the use of hash chains

to authenticate hop counts and sequence numbers. In SEAD, every node that participates in the ad hoc network has a hash chain. The elements of the hash chain are used in succession to authenticate the entries in the transmitted routing messages, given that an initial authenticated element exists. The hash chains have a finite size and must be generated again when all their elements have been used.

2.3.3. Byzantine Failure Resilience

In [SR03], an on-demand secure routing protocol was proposed that is resilient to Byzantine failures caused by Byzantine behavior, which is defined as “any action by an authenticated node that results in disruption or degradation of the routing service”. The failure refers to “any disruption that causes significant loss or delay in the network”. The detection of such failures is based on acknowledgements (acks). The destination node sends an ack back to the source node when receiving a packet. If an ack is not received after a certain time, the source node assumes it has been lost. The number of lost (to the same destination) exceeding a threshold triggers the Byzantine fault detection.

The source node launches a binary search of all the links along the path by probing the intermediate nodes. The nodes that are behaving normally send acks back to the source when receiving the probe. Half of the links are excluded from the suspects of failure for each probe. The faulty link will be identified after log n probes, where n is the number of hops between the source and destination. After the failure is located, the source node will start a new route discovery process and try to bypass the faulty link.

2.3.4. Secure OLSR (SOLSR)

Secure OLSR (SOLSR) [SOLSR04] is proposed to handle with the replay attack on the classic OLSR but without a synchronised clock. The author proposes a solution which utilises signatures to ensure integrity of OLSR control-traffic data is presented. A digest of the packet and a secret shared key are attached to all OLSR packets. Only a node with access to the secret key can produce such a signature.

The most valuable contribution of this Secure OLSR (SOLSR) is the new scheme, which attaches the signature to the OLSR packet and uses timestamp exchange to keep its freshness and robustness. This secure OLSR proposal is implemented as an OLSRD plug-in. Such implementation includes the message signature and the timestamp exchange. Apart from that, the open source code of SOLSR is available on the Internet, which means it can still be improved based on the former version. (For the newest version, they use HD-5 as a new encryption algorithm, and refresh the plug-in to fulfill the latest requirement, in order to get a higher quality of security.)

2.4. Wormhole Attack Detection and Evaluation Methods

2.4.1. Wormhole Attack Detection Methods

The methods against wormhole attacks can be categorised according to the following aspects: time and location based methods, topology based methods, and statistic based methods.

Time and location-based methods [PL03] [DEL06] [TL06] [LB06] make use of the fact that the propagation of wireless signals cannot be faster than the speed of light. The first work of studying wormhole attacks, Packet Leashes in [PL03], utilised geographical and temporal leashes to detect wormhole links. It required a synchronised clock or special hardware to identify a time or location abnormality in order to detect wormhole attacks. Improvements to the method have been proposed in recent works [DEL06] [TL06] [LB06], Most of them require fine-grained time synchronisation, extra special hardware (like GPS), or modifications to current MAC protocols.

Topology based methods [SUV04] [WCL09] aim to detect wormholes by observing abnormal network topologies. In general, these kinds of methods require dense node distribution, and are not applicable to WMNs, which are usually sparsely distributed. Statistics-based methods [DAW08] [DLW05] usually obtain normal statistics from theoretical analysis and detect wormhole attacks by identifying the difference between the current situation and the statistics. They always assume there is no wormhole attack at the initial stage of network establishment.

2.4.2. Evaluation of Wormhole Attack Detection Methods

The evaluation of existing works on detecting wormhole attacks can be divided into two classes: simulation and experiment over real testbed.

Most wormhole attack detection methods are evaluated by simulators [DAW08] [TL06] [VS04] [WAJ06]. Some are written in programming languages, and some

adopt complex network simulators. Simulators can easily simulate a large scale network with a huge number of nodes in it, and generate ideal network flows as well. After giving specific settings, simulators usually give a clear picture of how different parameters influence the detection ratio and false alarm ratio of these methods. Even if the simulation results can demonstrate the accuracy of algorithms and protocols, we may still wonder what will occur in a real wireless network.

As far as we know, only one work [BAN08] is found which established a real sensor network testbed for wormhole attacks. The authors first introduced the hardware and software components of their sensor testbed, named BANAID. They continually described the methodology of launching a wormhole attack on BANAID. Based on the testbed, they implemented a classic detection method, Packet Leashes [PL03], and tested its accuracy. Different from our work, this piece of work did not utilise the testbed to further study the impact of wormhole attacks. In addition, it did not propose any new methods, but just used Packet Leashes to detect wormhole attacks. Compared with this work, we not only set up a real world testbed but also investigated the impact of wormhole attacks on our testbed. Based on our fundamental theoretical analysis and experimental results, we propose NPA to detect wormhole attacks. The NPA does not need time synchronisation or extra hardware, and is applicable to all kinds of multi-hop wireless network systems.

2.5. Routing considering Security and Performance

Existing routing protocols can be divided into two categories. Traditionally, a routing protocol aims to achieve a high level of network performance, assuming the network is trustworthy and friendly. Another kind of routing protocol focuses on detecting or defending against attacks to exclude compromised nodes from routing paths.

Hop count is the most commonly adopted routing metric to quickly choose the shortest path between source node and destination node by existing ad-hoc routing protocols such as AODV [AODV03], OLSR [OLSR], etc. However, it cannot guarantee finding a high link quality path, because of the characteristics of wireless networks. Expected Transmission Count (ETX) [ETX07] is the first work that proposed finding high-throughput paths on multi-hop wireless networks. In addition, a certain number of studies (e.g., ETT , and WCETT [RMM04]) try to incorporate much more comprehensive factors, such as data rate, packet size, and so on to find the path with high quality links. Admittedly, these high performance approaches implicitly assume that the wireless network is friendly or trustworthy.

Due to the vulnerability of the multi-hop wireless network, numerous significant works are dedicated to address this problem through identifying attacks. Cryptography [SEAD03][SAODV02], digital signature approaches, neighbour monitoring [GRAY07}, and statistic differentiation [SWD05], etc., are utilised in methods that are proposed to detect active attacks with a high detection rate and a low

false alarm rate. Nevertheless, the security schemes may ignore the performance of secure routing and choose paths with sub-optimal utility value.

As far as we know, only one work [SH11] presents a secure high-throughput multicast routing in wireless mesh network. They assume that the multicast routing can only be attacked when malicious nodes drop packets to affect the packet delivery ratio. However, due to the complexity of malicious behaviour, we intend to utilise game theory method to comprehensively analyse the impact of attackers. Compared with existing works, we propose OSR by considering not only complex attack behavior but also routing performance. Based on theoretical proof and extensive simulation, OSR is able to find secure paths that are resilient to active attacks and with maximum path utility value. Thus, the joint consideration of both routing security and performance makes our work significant and irreplaceable.

Chapter 3. Security Extended OLSR Protocol

In this Chapter, we introduce the proposed Security Extended OLSR protocol (SE-OLSR) with its security analysis. Currently, existing routing protocols designed for Mobile Ad hoc NETwork (MANET) are widely used. However, their lack of security causes a deep concern while using them, since all the routing information is transmited from the entire network in plain text. Therefore, how to protect the integrity of routing information becomes a challenge and an important problem. This Chapter is organised as follows: Section 3.1 is the overview to this work. Section 3.2 introduces classical routing protocol OLSR, and shows the basic idea of symmetric encryption. In Section 3.3, we present the design of SE-OLSR and then give the security analysis in Section 3.4. Finally, Section 3.5 concludes this Chapter.

3.1. Overview

As mentioned before, existing routing protocols are vulnerable to attacks because the routing messages are transmitted by plaintext. Thus, the outside attacks will try to undermine the authenticity, confidentiality and integrity of routing information. Here, confidentiality stands for protecting overall content or a field in a message (or prevent an adversary from undertaking traffic analysis) while integrity represents the guarantee of a packet that cannot be modified during transmission. As shown in Table 3.1, the basic layout of any packet in OLSR (omitting IP and UDP headers) was given by RFC 3626 [OLSR]. All these packets are disseminated in plaintext which may be

distorted by malicious nodes. Thus, our objective is to design an encryption method to defend against outside attacks.

3.2. Existing OLSR Protocol

OLSR is a classical routing protocol for mobile ad hoc networks. It is an optimisation of the concept used in the protocol is that of multipoint relays (MPRs). MPRs are selected nodes which forward broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared to a classical flooding mechanism, where every node retransmits each message when it receives the first copy of the message. In OLSR, link state information is generated only by nodes elected as MPRs. Thus, a second optimisation is achieved by minimising the number of control messages flooded in the network. As a third optimisation, an MPR node may choose to report only links between itself and its MPR selectors.

Hence, as contrary to the classic link state algorithm, partial link state information is distributed in the network. This information is then used for route calculation. OLSR provides optimal routes (in terms of number of hops). The protocol is particularly suitable for large and dense networks, as the technique of MPRs works well in this context. The original OLSR packet format is illustrated in Table 3.1.

Table 3.1: OLSR Packet Format

packet length		sequence number
Message Type	VTime	Message Size
Originator Address		
Time To Live	Hop Count	Message Sequence Number
MESSAGE		

3.3. Security Extended OLSR (SE-OLSR) Protocol Design

Before introducing the SE-OLSR, we intend to give some cryptography preliminaries first to show the principle of the technology that we adopted in design. Symmetric cryptography is a classical encryption method in which both transmitter and receiver can share the same key directly, or by calculating relevant preliminary acknowledgement. The symmetric encryption algorithm is characterised by open source algorithms, less requirement for computation capability, fast encryption probability and high efficiency. However, it is not prevalent in distributed network systems since it is difficult to apply and difficult to do key management at low cost. The modern study of symmetric-key algorithms relates mainly to the study of block ciphers and cryptography hash functions. A block of plaintext and a key is the input of block ciphers, whereas the output is a block of cipher text with the same size. Cryptographic hash functions take a message of any length as its input, and output a short, fixed length hash value as cipher text. As good hash functions, an attacker cannot deduce the plaintext from the hashed values. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are a widely used application of block ciphers in computer networks. Currently, the DES has been identified as not secure enough, thus it is rarely be used after the generation of AES. As for the second algorithm, MD-5 and SHA-1 are widely deployed. Advanced Encryption Standard (AES) is a widely used Symmetric Cryptography Solution. The block size of AES was fixed by 128 bits, while the length of key can be 128, 192 or 256 bits. Most AES calculations are done in a special finite field. AES operates on a 4×4 array of bytes, termed the state, which is initialised with a plaintext block. The AES cipher is

specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds is applied to transform ciphertext back into the original plaintext using the same encryption key. And the format of original OLSR packet will be changed into that shown in Table 3.2.

After gaining knowledge of AES, the detailed design of EEOLSR will be given as follows. First of all, encrypt the routing packet by using the AES algorithm, with a rough description of the AES algorithm given as follows:

A à B: Encrypt routing message with A's private key and attach a hash value.

B à A: First, check the attached hash value and then decrypt the routing message.

Table 3.2: SE-OLSR Packet Format

packet length		sequence number
Enc_Alg	Hash_Alg	Message Size
Encrypted Message		
HASH		

The Enc_Alg and Hash_Alg - two new fields in the EEOLSR packet - aim to describe the advanced encryption algorithm and hash function. By using this preliminary encryption algorithm, once a node A intends to send the OLSR packet to node B, it will encrypt the packet using its private key. Then, A will utilise the hash function to deduce a value and bind a random value denoted by nonce. After doing so, the revised

packets will be transmitted to other nodes such as node B. If B is a common node, it can first decrypt the routing packet using A's public key, which is pre-shared with common nodes. The hash function will help verify whether or not this packet was revised by misbehaviour during transmission. The detailed procedure is illustrated as the following flow chart in Figure 3.1.

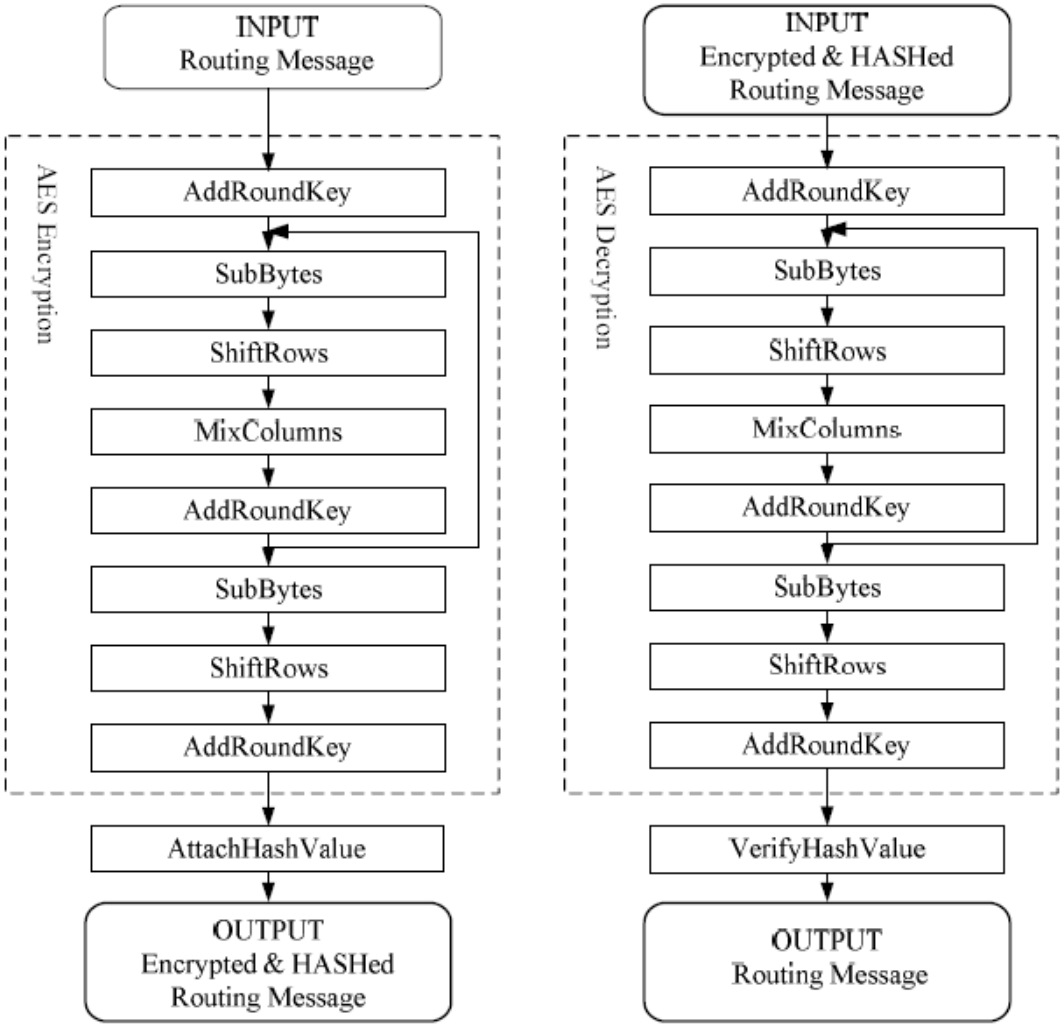

Figure 3.1: Security Extended OLSR(SE-OLSR)

3.4. Security Analysis

Guaranteed Confidentiality. Outside attack is not part of the current network which intends to participate in the network communications and do malicious things. The SE-OLSR encrypt the routing messages. As is known to all, routing protocol is designed for finding paths from a specific source to its communication destination. Therefore, outside attacks cannot decrypt them and fail to join in the network. Therefore, the SE-OLSR can guarantee routing messages confidentiality.

Guaranteed Integrity. The advantage of hash function is that misbehaviour cannot deduce the original content from the hashed value. Nevertheless, honest users use pre-negotiated hash function to verify whether the original message has been revised. Finally, the results will help to discard the messages that were manipulated by attackers.

3.5. Summary

EEOLSR can defend outside attacks while guaranteeing messages' confidentiality and integrity, by using cryptography and hash methods. It developed a well-known routing protocol and made it more secure for multi-hop wireless networks. It is implemented on in our current mesh network testbed. However, the strategy of defending outside attacks is almost mature. But few current works implement a real testbed. So, in the following section, we will focus on dealing with a more challenging attack, classified as inside attack.

Chapter 4. Analysis of Wormhole Attacks

In this Chapter, we introduce our WMN testbed and describe how to launch a wormhole attack on the testbed for analysis in real world scenarios. Then, we will study the feature of wormhole attacks through both theoretical analysis and experimental results. This Chapter is arranged as follows. First, Section 4.1 describes the methodology of establishing the Wireless Mesh Network Testbed for wormhole attack with the demonstration of its correctness. In Section 4.2, we present both theoretical and experimental results to analyse the impact of wormhole attacks. Finally, Section 4.3 concludes this Chapter.

4.1. Wireless Mesh Network Testbed for Wormhole Attack

In this section, we first introduce the design of our WMN testbed including hardware and software components. Then, the evaluation results will be given.

4.1.1. Testbed Design

Hardware Component. After considering the characteristics of both wormhole attacks and mesh network backbone, we set our WMN testbed in the following way. This WMN testbed consists of seven mesh routers developed by our lab, called T902 [HAWK09] shown in Fig. 2. The system specification of T902 is listed in Table 4.1. Each T902 is equipped with three IEEE 802.11a/b/g [IE802] wireless network interface cards (NIC). The available channels are ranging from channel 1 to channel 11. All the testbed components are running on the Linux operating system. The

routing algorithm used on the WMN testbed is our security extended Optimised Link State Routing (OLSR) protocol which is modified based on OLSR [OLSR]. In order to preserve the integrity and confidentiality of packets, the new security extended OLSR routing packets are encrypted and attached by the timestamped digital signature.

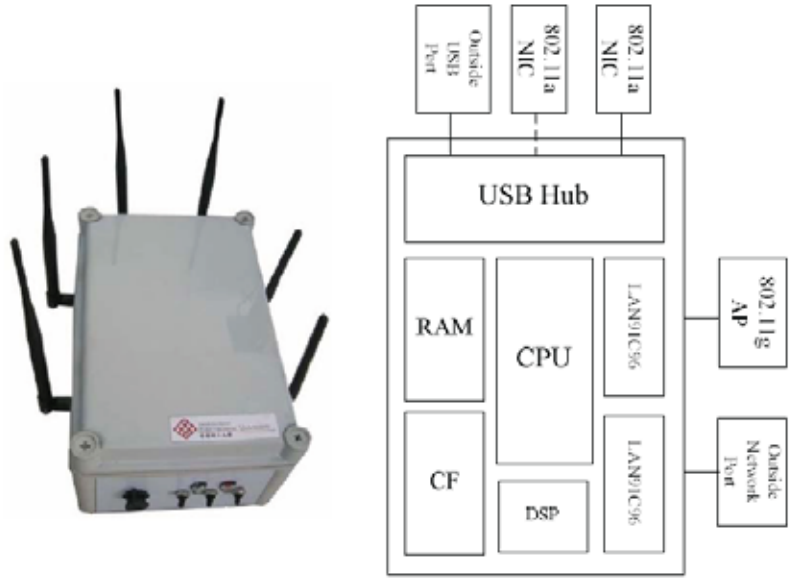

Figure 4.1: T902 and its hardware design structure

Software Component. In order to launch a wormhole attack, we modified two components of Aircrack-ng [AIR], which is a tool for monitoring wireless networks, involving Airodump-ng and Aireplay-ng. The original Airodump-ng can capture packets from assigned port, channel and BSSID, while Aireplay-ng can replay packets and inject them into the network. Based on existing functionalities, we combined these two separate components and made them work together. We established a UDP tunnel which performed as a wormhole link to connect Airodump-ng and Aireplay-ng. Both modified Airodump-ng and Aireplay-ng run on two

wormhole endpoints, i.e., T902 nodes. The modified Airodump-ng can dump routing packets from neighbors of one wormhole endpoint, and send these packets to another endpoint through the tunnel. The modified Aireplay-ng will replay all the routing packets from one specific endpoint of the wormhole link to neighbours of another endpoint.

Table 4.1: Mesh router system specification

CPU	AMD Geode LX800 processor 500MHz
Compact Flash	2G
Serial port	Three RS-232 and one RS-485 serial ports
Wireless network	IEEE802.11 a/b/g
Ethernet	Support for 2-way 10/100M Ethernet interface
Operating System	Embedded Linux
Size	280mm × 190mm × 130mm

4.1.2. Experiment Scenario

After introducing the implementation details, we intend to show you our experiment deployment. It is a practical and typical scenario that can occur in a real wireless network environment. As shown in Figure 4.2, we placed seven T902 mesh routers in one floor of our lab building on campus. Five mesh routers (denoted as nodes A to E) were placed at locations A to E performing as normal backbone nodes. They communicated with each other through wireless links and formed a U shape deployment. As the nodes were placed in classrooms or at the corridor, we took advantage of obstacles like doors and walls between them, and adjusted both the location and the transmission power of each node to guarantee that the nearby nodes are one-hop neighbours. Two T902 mesh routers (denoted as node X and Y) were placed at locations X and Y , which are close to nodes A and E respectively. These

two routers acted as the endpoints of a wormhole attack. We connected them directly by cable to set up a secret tunnel as a wormhole link. All the nodes used the same channel, and adopted ESSID “wh”. Actually, node A was four-hop away from node E. Once the wormhole attack was enabled, the routing table of the node E revealed that node A became one-hop neighbour of node E. This shows that the wormhole attack launched by our mechanism really works.

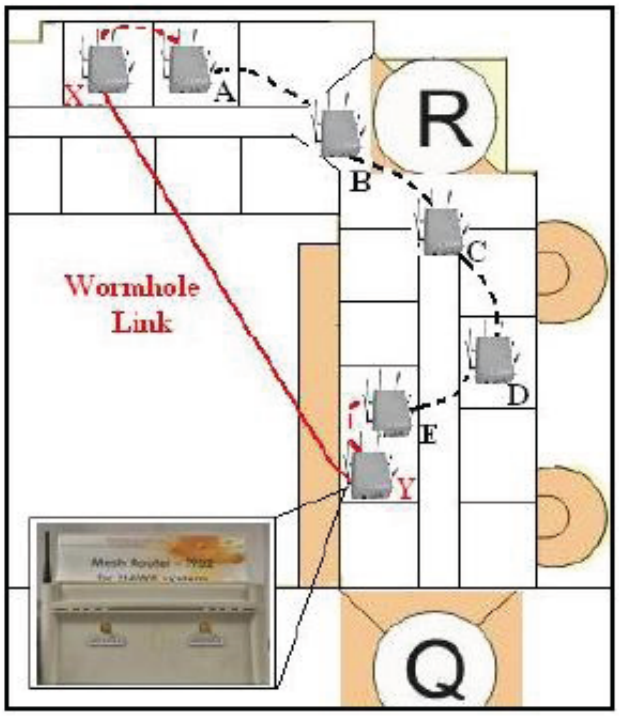

Figure 4.2: Wormhole testbed evaluation scenario.

4.2. Analysis of Impact of Wormhole Attacks

In this section, we first propose a theoretical analysis of the impact of wormhole attacks by using mathematical investigations. Then, we set up different experiments

to evaluate the impact of wormhole attacks. Finally, our novel observation will be given.

4.2.1. Theoretical Analysis of Impact of Wormhole Attacks

Previous works, such as [DEL06], used a large average or instant per-hop delay value (i.e., RTT) to detect wormhole attacks. The Round Trip Time (RTT) is the time for a probe to be sent plus the time it takes for a corresponding acknowledgement to be received. They proposed that large per-hop delay is caused by wormhole attacks. In other words, once the RTT is larger than a threshold, the detection can report there is a wormhole attack. However, the methods will not work properly because the per-hop delay is also large when the network has congestion or is in a complicated wireless network environment. Therefore, we propose using the standard deviation of RTT (stdev(RTT)) to study this problem. Stdev(RTT) is defined as the variance of the value of RTT. We prove that stdev(RTT) can identify wormhole attacks accurately and effectively in both theoretical and experimental ways.

First, we illustrate the theoretical analysis on the effectiveness of stdev(RTT) on identifying wormhole attacks. The RTT in a wireless environment under normal conditions (RTTnorm) consists of the delay on wireless links (Tw) and packet processing time (Tproc, including queuing delay). We have the following expression.

$$\mathrm{RTT}_{norm} = T_w + T_{proc} \tag{4.1}$$

Normally, $\mathrm{stdev}(\mathrm{RTT}_{\mathrm{norm}})$ between one-hop neighbours can be calculated as follows,

$$\begin{aligned} &\text{stdev}(RTT_{norm}) \\ &= stdev(T_w + T_{proc}) \\ &= \sqrt{\text{var}(T_w) + \text{var}(T_{proc}) + 2\,\text{cov}(T_w, T_{proc})} \\ &= \sqrt{\text{var}(T_w) + \text{var}(T_{proc})} \end{aligned} \tag{4.2}$$

where $\text{var}(\text{X}) = \text{stdev}^2(\text{X})$ stands for the variance of X and $\text{cov}(\text{X}, \text{Y})$ means the coherence between X and Y . Since T_w is independent of T_{proc} , the value of $\text{cov}(T_w, T_{proc})$ equals 0.

In the environment with wormhole attacks, the value of stdev(RTT) will be changed because of the variation of T_w and T_{proc}. The wormhole attacks usually contain two wireless links refer to the link S1-X and Y -S2 in Figure 1.3. Wormhole endpoints will co-operate to capture numerous packets and send them through the wormhole link. All the packets should be scheduled for replaying. Due to the large amount of packets, all the packets should wait in the queue to be sent out. This will certainly cause a lot of queuing delay. In addition, attackers capturing and replaying the bypassing packets may generate much more processing time on the wireless links. Therefore, we can conclude that the RTT under wormhole impact (RTT_{worm}) equals $2T_w + T¢_{proc}$ where $T¢_{proc} > T_{proc}$. Here we ignore the processing time caused by wormhole link due to its high link quality. The $\text{stdev}(\text{RTT})$ under a wormhole attack's impact can be calculated as follows,

$$\begin{aligned} &\text{stdev}(RTT_{worm}) \\ &= stdev(T_w + T'_{proc}) \\ &= \sqrt{\text{var}(2T_w) + \text{var}(T'_{proc}) + 2\,\text{cov}(2T_w, T'_{proc})} \\ &= \sqrt{4\,\text{var}(T_w) + \text{var}(T'_{proc})} \end{aligned} \tag{4.3}$$

From equation (4.2) and equation (4.3), we can easily deduce equation (4.4) which represents that the value of stdev(RTT) under the impact of wormhole attacks is much larger than that in normal environment. From equation (4.4), we can see that

the stdev(RTT) could be a new effective and efficient metric to identify the wormhole attack.

$$\text{stdev}(\text{RTT}_{\text{worm}}) > \text{stdev}(\text{RTT}_{\text{norm}}) \tag{4.4}$$

4.2.2. Experiments Design for Evaluating Impact of Wormhole Attack

After introducing the theoretical analysis, we set up experiments to evaluate the impact of wormhole attacks on RTT and stdev(RTT). In our experiment setting, referring to Figure 4.2, link AB is chosen as the representative normal link, because nodes A and B are always one-hop neighbours, no matter whether there is a wormhole attack or not. Correspondingly, the link AE under wormhole attack is chosen as the representative wormhole influenced link because nodes A and E will become one-hop neighbours once a wormhole attack exists. Each data is the average value collected every 50 times independent experiments aim to mitigate the randomness.

To study the impact of wormhole attacks on RTT and stdev(RTT), we first conduct experiments with different settings, which separately consider the following three traditional factors in a wireless environment.

4.2.2.1. Impact of Channel Conditions

We firstly study the impact of communication channel on RTT and stdev(RTT) without wormhole attack (w/o wormhole) or with wormhole attack (w/ wormhole). As mentioned in Section 4.2.1, our WMN routers use the IEEE 802.11b/g standard. Thus, in order to analyse the completely possible situations, we should investigate the conditions of all 11 channels. The packet size is fixed to 32 bytes with reference to the size of OLSR packet header. There is no background traffic.

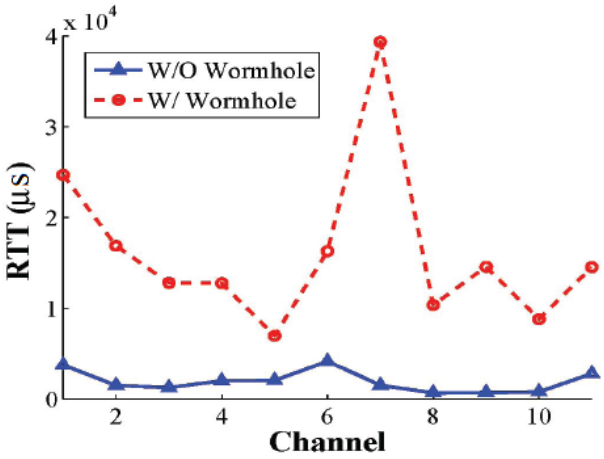

Figure 4.3: RTT vs. channel

From the results shown in Figure 4.3 and Figure 4.4 , we can see that, the RTT statistic is not constant because the interference level and occupancy rate are totally different from channel to channel. When there is a wormhole attack, RTT and

stdev(RTT) of a wormhole influenced link are consistently larger than those of a normal link. This means that the extra processing and queuing delay through a wormhole link lead to a larger delay, and make the link unstable.

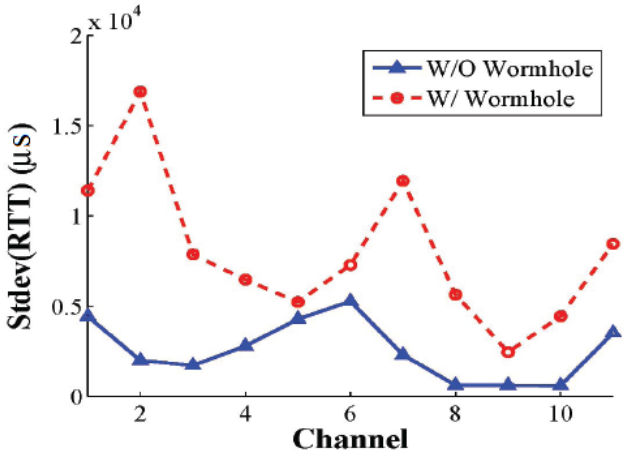

Figure 4.4: Stdev(RTT) vs. channel

4.2.2.2. Impact of Background Traffic Load

We secondly consider the impact of background traffic load on RTT and stdev(RTT) with or without a wormhole attack. We generate the background traffic between each nearby neighbour by using Iperf [IPERF]. As we can see from Figure 4.3, the minimum RTT value occurs in the 8th channel. This means that it has the minimum channel occupancy rate. To avoid channel interference, we use the 8th channel as the communication channel and still set the packet size to 32 bytes. Based on our knowledge of IEEE 802.11 standard, the maximum available bandwidth can reach 20 Mbps without interference, while the bandwidth can still reach 5 Mbps in a normal lab environment. Thus, we examine all the cases from no background traffic load to 50 Mbit per second that includes these two special values.

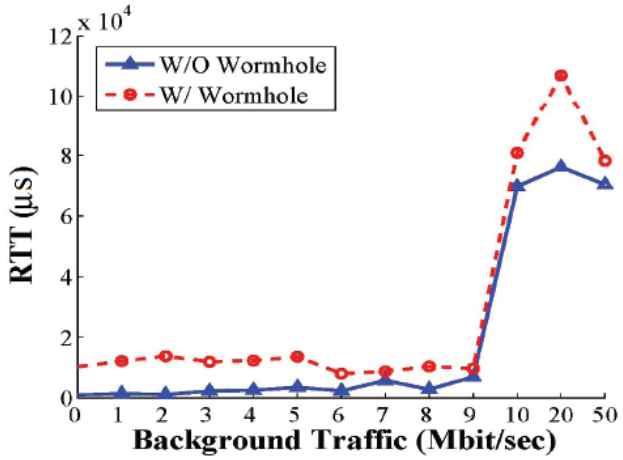

Figure 4.5: RTT vs. background traffic

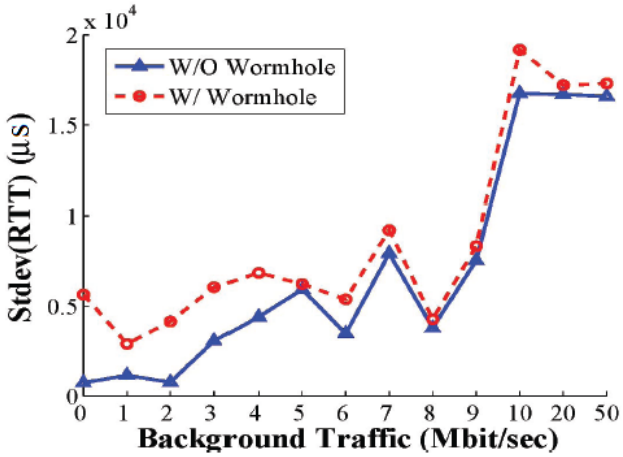

Figure 4.6: Stdev(RTT) vs. background traffic

From the results shown in Figure 4.5and Figure 4.6, we can observe that RTT and its deviation of a wormhole influenced link is consistently larger than that of a normal link when there is a wormhole attack, but the performance gap decreases as the increase of background traffic load. When the background traffic load increases beyond 10 Mbps, there is a sudden increase in both RTT and stdev(RTT). It is due to a large queuing delay and the limitation of available bandwidth.

4.2.2.3. Impact of Packet Size

Third, we study the impact of packet size on RTT and stdev(RTT), with or without a wormhole attack. Referring to the routing packet size in OLSR, the total size of the routing packet depends on the types of different messages. Thus, in this condition, the packet size is changed from 32 bytes to 1024 bytes. The communication channel is fixed as the 8th channel, and there is no background traffic. From Figure. 4.7 and Figure 4.8, we can observe that both RTT and stdev(RTT) have non-monotonic relationships with the packet size. Similar to the previous two cases, we still observe an obvious increase in RTT and stdev(RTT) when there is a wormhole attack.

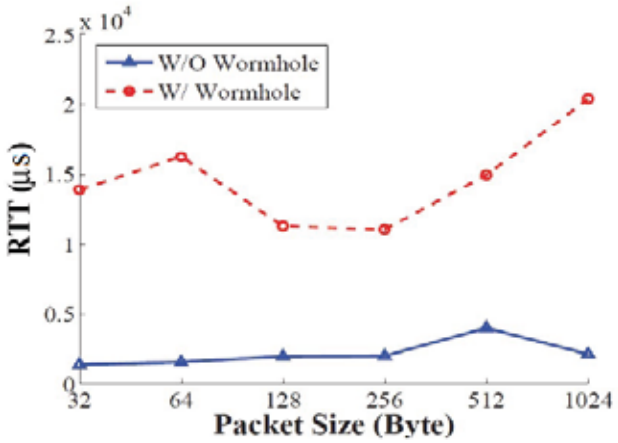

Figure 4.7: RTT vs. packet size

From all of the above experimental results, we can see that not only RTT but also stdev(RTT) will increase under the impact of wormhole attacks. To the best of our knowledge, this is the first work to describe that wormhole attacks can lead to the occurrence of large stdev(RTT) value. Previous works only use RTT to identify wormhole attacks once its value is larger than a threshold. However, referring to

Figure 4.7, the RTT will become even larger under heavy background traffic load and this may lead to false alarms. Therefore, we believe stdev(RTT) could be a more efficient and effective metric to identify wormhole attacks.

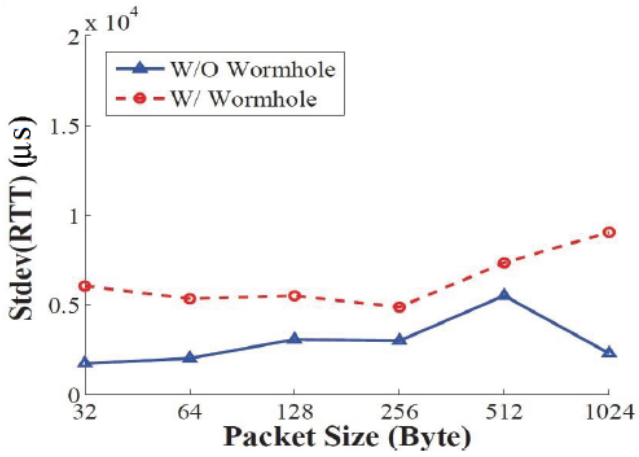

Figure 4.8: Stdev(RTT) vs. packet size

4.2.2.4. A Case Study on stdev(RTT) with/without WormholeAttacks

In order to directly examine the differences between these two metrics, RTT and stdev(RTT), we set up a general scenario to observe variations when the network is with or without wormhole attacks. Furthermore, from this general case, we will explain why we can utilise the characteristics of stdev(RTT) to detect wormhole attacks. In this case, we look into a more detailed scenario which fully shows every value of each 50-time experiment round. The communication channel is fixed to the 8th channel, the packet size is fixed to 32 bytes, and there is no background traffic load. Figure 4.9 illustrates the value of RTTs with or without the wormhole attack in 50 experiments. We can see that there is a large stdev(RTT) when there is a wormhole attack. In contrast, the RTT is stable and relatively small when there is no

wormhole attack. Especially, the maximum RTT is still less than two times of the minimum RTT. Therefore, the stdev(RTT) can be used as a more accurate and efficient metric to decide whether there is a wormhole attack in the network.

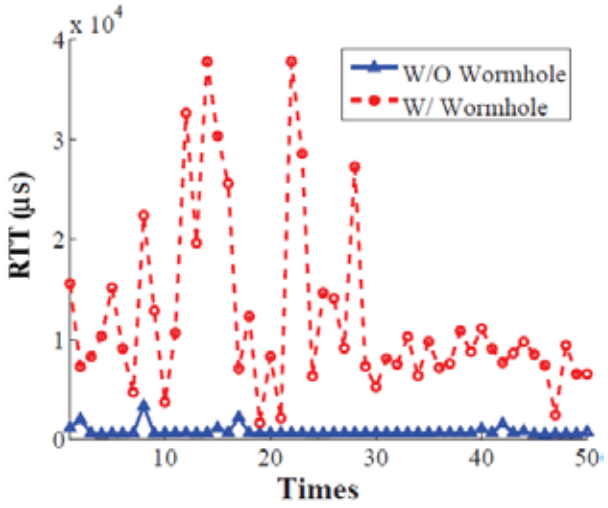

Figure 4.9: RTT of 50 times experiments

4.3. Summary

In general, wormhole attacks can cause large stdev(RTT). This large stdev(RTT) can be interpreted as the occurrence of frequent large RTTs.

To the best of our knowledge, this is the first work that has obtained such an interesting observation. This observation will be used to design our NPA algorithm as shown in the next section.

Chapter 5. Neighbour-Probe-Acknowledge Wormhole Detection Algorithm

In this Chapter, we introduce the proposed Neighbour-Prove-Acknoledge (NPA) wormhole detection algorithm. This Chapter is organised as follows. First, Section 5.1 describes the NPA algorithm. In Section 5.2, we give the security analysis of NPA. Simulation results are reported in Section 5.3. We also compare NPA with other related works and discuss its significant contributions in Section 5.4. Finally, Section 5.5 concludes this Chapter.

5.1. The NPA Algorithm

Based on the analysis mentioned in Chapter 4, we find that the stdev(RTT) can effectively identify wormhole attacks. In this section, we propose a new wormhole detection algorithm, named as Neighbour-Probe-Acknowledge (NPA). NPA is used to detect wormhole attacks, based on observing the occurrence of large stdev(RTT). We make the following assumptions. The network consists of N nodes. The number of normal nodes is much larger than wormhole endpoints. The wormhole attacks are traditional but not intelligent, which means that the wormhole attacks cannot influence the procedure of calculating RTTs.

NPA will be triggered when the network topology changes. To obtain RTTs, each node sends probe messages locally for T times to all its neighbours when its routing table changes and sees T acknowledging messages from each neighbour. In order to

prevent the interference of outside attacks, NPA needs to preserve the confidentiality and the integrity of both routing messages and probe/acknowledge messages. Thus, we utilise the AES encryption algorithm [18] and timestamped digital signature to guarantee the security of routing packets. Then, for each node i, it calculates RTT for T times between itself and its neighbour j, denoted as $R_{i,j,t}(t=1,2,...,T)$. Then it calculates the maximal and minimal values of all $R_{i,j,t}$ for each neighbour j, denoted as $RTT_{max}^{i,j}$ and $RTT_{min}^{i,j}$ min respectively. As inspired by the general case studied in section 4.2.2.4, if $RTT_{max}^{i,j}$ is less than 2 times of $RTT_{min}^{i,j}$, as shown in equation (5.1), we conclude there is no wormhole attack between this pair of nodes since this shows a very slight variation of RTTs.

$$RTT_{max}^{i,j} < 2 \times RTT_{min}^{i,j}, (0 < i, j < N) \quad (5.1)$$

Otherwise, the algorithm will continue to calculate the $s_{i,j,t}$, which stands for the situation between node i and node j at time t, as the following way shown in equation (5.2). If $s_{i,j,t}$ equals to 1, it describes that the $RTT_{i,j}$ is larger than $a RTT_{max}^{i,j}$ at time t. Here a a given parameter which is between 0 and 1.

$$a_{i,j,t} = \begin{cases} 1, RTT_{i,j,t} \geq a \times RTT_{max}^{i,j} (0<i,j<N) \\ 0, else \end{cases} \quad (5.3)$$

$R_{i,j}$ is given by

$$R_{i,j} = \sum_{t=1}^{T} s_{i,j,t} \tag{5.4}$$

which records the number of times that RTT between node i and its neighbour j is over a threshold, $a\,\mathrm{RTT}_{\max}^{i,j}$. When $R_{i,j}$ is over a threshold b, the situation becomes:

$$R_{i,j} > b, (1 \le b \le T) \tag{5.5}$$

we claim there is a wormhole attack between node i and its neighbour j because of the occurrence of large stdev(RTT). In short, the main idea of NPA is presented in Algorithm 1.

Algorithm 1 The NPA Algorithm

```
for all Node i do
  for all Neighbor j do
    t = T;
    while t ≥ 1 do
      RTT_{i,j,t} = current RTT;
      t = t − 1;
    end while
    Calculate RTT^{i,j}_{max} and RTT^{i,j}_{min};
    Calculate R_{i,j};
    if RTT^{i,j}_{max} > 2 × RTT^{i,j}_{min} then
      if R_{i,j} > β then
        There exists a wormhole attack !!
      end if
    end if
  end for
end for
```

The complexity of NPA can be calculated as follows. The extreme case is that each node has N neighbours, and needs to send probe/acknowledgment messages to all its neighbour nodes for T times. Therefore, the complexity of the whole NPA wormhole

detection is $O(N^2T)$, where N is the number of nodes in the network. It is easy to see that the complexity is low. In addition, the overhead of NPA only depends on the times of sending/receiving probe/acknowledgement messages, which is also small. So, we can say that the NPA is an effective and efficient algorithm.

5.2. Security Analysis

NPA is a security enhancement in WMNs to detect wormhole attacks. Thus, it is necessary for us to study its robustness carefully to avoid introducing new potential vulnerabilities. Below we discuss possible attacks which may be introduced while using NPA and their corresponding countermeasures.

5.2.1. Packet Modification Attack

Traditionally, wireless networks are assumed to operate in a friendly and trustworthy environment in terms of no malicious nodes. Since NPA needs to send probe messages and receive corresponding acknowledgement to get neighbour information, this makes it possible for malicious nodes to manipulate these two kinds of messages to distort the accuracy of NPA. In NPA, we encrypt every message by using mature cryptography technologies in order to preserve the confidentiality and integrity of packets. In this way, attackers cannot recognise and control probe/acknowledge messages since they don’t have any key information to decrypt the messages.

5.2.2. Replay Attack

Due to the dynamic features of wireless networks, the network topology may change from time by time. Thus, malicious nodes can perform replay attack to record out-dated but still valid routing control messages. Then, such attackers can distort the procedure of updating the network topology. However, after attaching the timestamped digital signature on each routing control message, this kind of attack can be successfully prevented.

5.2.3. RTT Modification Attack

Before introducing NPA, we assume that the attacker cannot interfere with the RTT calculation procedure. However, once the wormhole attack technology becomes increasingly mature, an intelligent attacker may come into being. Then, it can realise the mechanism of NPA and try to provide a stable RTT for its influenced link by adjusting the time to transmit the packet. But due to the contentious nature of the wireless channel, it is hard for an attacker to occupy the channel and send the packet to its intended receiver at its preferred time slot.

5.3. Performance Evaluation

We evaluate the performance of NPA in the scenario shown in Figure 4.2. The nodes adopt the same configuration in Chapter 4. They work on the 8^{th} channel. The packet size is set as 32 bytes in all experiments. The number of probe/acknowledgement times T is set as 50, and the frequency of sending probe packets is 10 times per

second. Firstly, we investigate the effect of parameters a and b to the NPA when there is no background traffic. As shown in Figure 5.1 and Figure 5.2, the wormhole detection rate is insensitive to the change of parameters, while the false alarm rate decreases as the increase of a and b. The reason is that both RTT and stdev(RTT) collected from a normal link is less than that from a wormhole-influenced link according to our experiments shown in section 4.2.2. When a and b increase, the probability that a normal link has a large stdev(RTT) becomes negligible, which leads to fewer false alarms in normal links. However, when a and b become too large, the detection rate decreases. Only the wormhole links with an extremely large stdev(RTT) are detected. As observed in Figure 5.1 and Figure 5.2, the combination of a = 0.5, b = 7 achieves a good tradeoff between detection rate and false alarm rate.

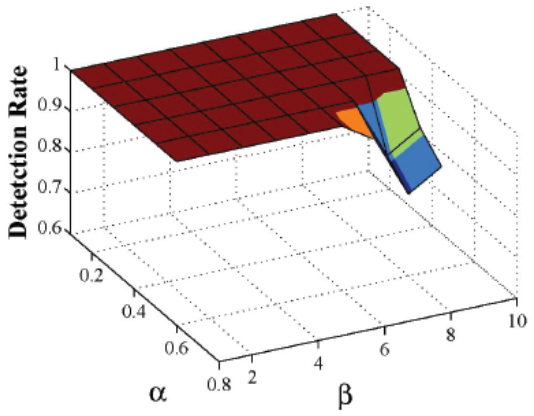

Figure 5.1: Wormhole detection rate with no background traffic

Secondly, we investigate the performance of NPA when background traffic load varies by setting a = 0.5, b = 7. As shown in Figure 5.3, NPA achieves a satisfactory performance (nearly 100% detection rate and a zero false alarm rate) when traffic

load is less than 10 Mbps. When the traffic load increases further, the wormhole detection rate decreases a little bit, but is still above 80% with a negligible false alarm rate. Thus our wormhole detection algorithm is robust and applicable to even high background traffic load cases.

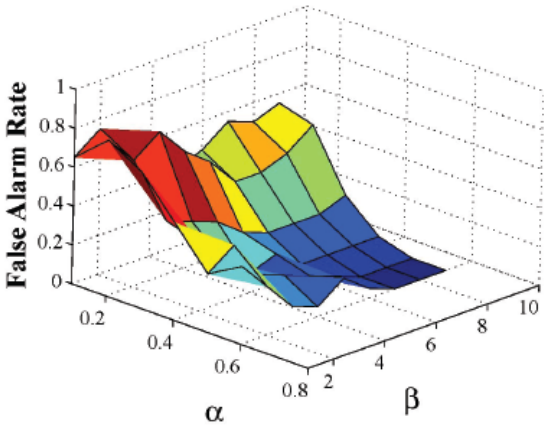

Figure 5.2: False alarm rate with no background traffic

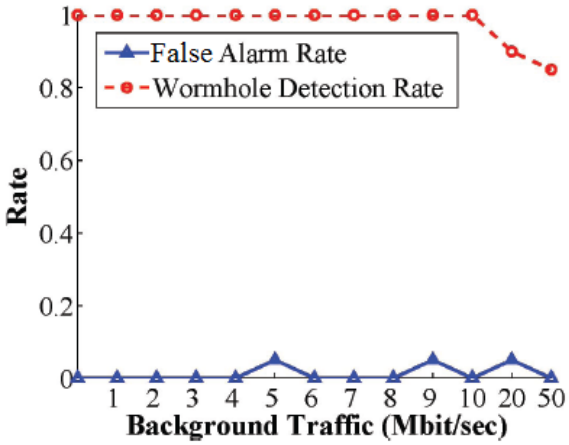

Figure 5.3: False alarm rate and wormhole detection rate

5.4. Discussion

In order to emphasise our contribution, we want to compare our NPA with other related algorithms and discuss the differences among them. As shown in Table 5.1, the NPA can detect wormhole attacks without strong assumptions, such as time synchronisation and extra hardware support or complicated modification of existing protocols. Moreover, our work is one of limited works which established a real testbed to study wormhole attacks. As mentioned in part B of Section II, our work is more significant than BANAID [BAN08] because we analyse the impact of wormhole attacks on top of our testbed and find interesting observations. It is the first work to find the observation of the occurrence of large stdev(RTT) under the impact of wormhole attacks. Furthermore, we have designed a practical approach to defend against wormhole attacks and evaluate its performance. Another fundamental contribution we want to clarify, is that our NPA can work efficiently and effectively under both light and heavy background traffic load. The performance evaluation demonstrates that the stdev(RTT) is a more efficient metric than RTT to detect wormhole attacks. DelPhi [DEL06] utilised delay/hop value ($\mathrm{DPH_i}$, $\mathrm{DPH}_i = \frac{\mathrm{RTT}_i}{2h_i}$) to identify wormhole attacks, where node i initiated DelPhi and sent a request at time t_s and then received the reply at time t_i where the hop count filed in the reply message was h_i. Once $\mathrm{DPH_i}$ was larger than a threshold, they claimed there was a wormhole attack. However, the value of DPH will increase under more complicated traffic load conditions and become larger than a certain threshold. This could lead to false alarms. Thus, the impact of background traffic load needs to be taken into

account at the time of design. Different from DelPhi, our NPA is designed based on the experimental results. NPA utilises the characteristics of stdev(RTT) to take the background traffic load into consideration. The performance evaluation scenarios for DelPhi and NPA are designed according to similar background traffic load conditions (no background traffic load, light background traffic load and heavy background traffic load). Therefore, we can refer to their evaluation results to make a comparison. We find that the average detection rate of DelPhi is around 78%, which is worse than our NPA, i.e., around 90% when the wormhole tunnel is four-hop long. In addition, the average false alarm rate of DelPhi is 15% which is worse than that of NPA (i.e., around 5%).

Table 5.1: Comparison between NPA and related works

Method	Packet Leashes [PL03]	DelPhi [DEL06]	TrueLink [TL06]	BANAID [BAN08]	NPA
Extra Hardware	YES	NO	NO	NO	NO
Time Synchronisation	YES	NO	NO	YES	NO
Modification of MAC Protocol	NO	NO	YES	NO	NO
Concern of Background Traffic	NO	NO	NO	NO	YES
Evaluation Method	Simulation	Simulation	Simulation	Testbed	Testbed

5.5. Summary

Based on the observation, we propose a neighbour-probe-acknowledge algorithm (NPA) to detect wormhole attacks by identifying the occurrence of large stdev(RTT). The evaluation results on testbed show that the proposed algorithm can achieve near 100% wormhole detection rate and zero false alarm rate in both light and heavy background traffic load scenarios.

Chapter 6. Optimal and Secure Routing Protocol in Multi-hop Wireless Networks

In this Chapter, we propose an optimal and secure routing (OSR) protocol in multi-hop wireless networks. This Chapter is organised as follows. Firstly, Section 6.1 gives an overview of this work. Section 6.2 shows the preliminary including network model and attack model. The OSR is described in detail in Section 6.3. Extensive evaluation results are reported in Section 6.4. Finally, Section 6.5 summarises this Chapter.

6.1. Overview

In recent years, there have been increasing concerns about ensuring the security of multi-hop wireless network, especially security-sensitive applications. It is a challenging task to sustain the communications between wireless nodes in insecure environments due to the lack of pre-established infrastructures. Therefore, one of the main challenges of assuring the wireless network, is to develop secure routing protocols. There has been a lot of security enhancement for routing protocols. However, how to find an optimal path between source node and destination node with security assurance becomes an open issue [SUV04].

Previous routing protocols focus on improving their performance by choosing a path with optimal performance metrics, such as throughput and average packet delay [DN92] that can substantially affect the routing algorithm. These routing protocols assume the wireless environment is friendly and trustworthy. However, the multi-hop

wireless network is vulnerable to numerous attacks. A number of works [SEAD03][SAODV02][SE02] propose their security enhancement methods based on current routing protocols to guarantee the integrity of routing packets by using cryptography and digital signatures. Another category of methods utilised time differentiation, topology abnormality, and neighbour monitoring to identify the attackers. Since these technologies are becoming more and more mature, they can work without strong assumptions or extra hardware support, yet can still detect or defend against attack with high accuracy. However, due to the introduction of security concerns, the paths which are resilient to attacks may not satisfy the performance requirements of traditional routing protocols. Therefore, how to find an optimal and secure path at the same time becomes an open issue. This surely requires our attention and motivates us to develop a new routing protocol.

6.2. Preliminary

In this section, we first introduce some important definitions and then discuss our system architecture, including network model and attack model.

We use (i, j) to denote the link between node i and j. The quality of link (i, j), denoted as $Q(i, j)$, can refer to any metric that is of our interest, such as ETX [ETX07].

Definition 1: For each node $n_{0,}$ suppose P_{n_0} is a path from node n_0 to destination n_p, $P_{n_0} = \{n_0, n_1, .., n_k, .., n_p\}$, (0<k<p). Then ***path utility***, $U_{path}(P_{n_0})$, stands for the accumulated link quality values of all the links belonging to path P_{n_0} .

$U_{path}(P_{n_0}) = \sum_{0 \le k < |P_{n_0}|} Q(n_k, n_{k+1})$, where $|P_{n_0}|$ represents the length of path P_{n_0} .

Note that the accumulation of link quality values can also be reflected by other operations in addition to summation, such as multiplication [HT06]. Here, summation is used for ease of presentation. Also there may exist multiple paths from n_0 to destination n_p . All these paths constitute a set, denoted by $P(n_0, n_p)$.

Definition 2: ***Node utility*** of node n_0 , denoted by $U_{node}(n_0, n_p)$, equals to the maximum value of all the path utility values between n_0 and n_p .

$$U_{node}(n_0, n_p) = \max_{P_i \in P(n_0, n_p)} (U_{path}(P_i)) \tag{6.1}$$

Node utility reflects a node's capability of routing data to the destination. In utility-based routing protocols, the node with largest node utility will be included in the routing path.

6.2.1. Network model

We characterise our network model as a multi-hop wireless network. Figure 6.1 shows the overall architecture of our network, which consists of the Trust Clearance Centre (TCC) and a number of wireless nodes. Basically, TCC is a trusted third party. In different phases of the routing protocols, each wireless node needs to report the required data to get a trust value, which is issued by TCC. It can work either on-line or off-line. For simplicity, we assume TCC works on-line in what follows in work.

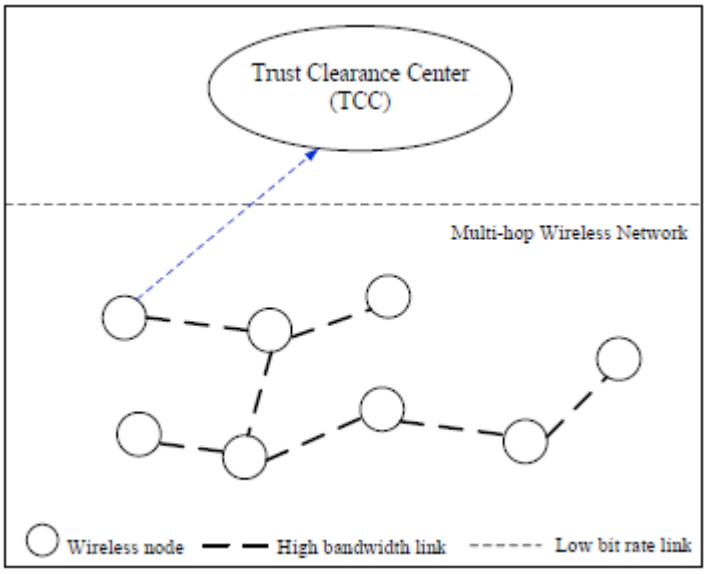

Figure 6.1: Network model

In addition, there are two types of links in the network. On the one hand, short-range and high-bandwidth links are used for inter-node communication. On the other hand, the links between nodes and TCC are long-range and low bit-rate in order to minimise the communication cost between them [MC10].

6.2.2. Attack model

Attacks can be categorised into passive attacks and active attacks [RS02]. Passive attackers do not disrupt the operation of routing protocols, and thus we will not study them further here. An active attacker intends to improperly modify data, gain authentication by injecting false or fake packets or modifying packets transitioning through the network. Our attack model assumes that the malicious nodes may cheat normal nodes by manipulating routing messages or claiming fake node utility to launch active attacks, which may conduct injecting, dropping, replaying, or relaying packets attacks in order to disrupt the routing procedure. For example, malicious nodes can cheat during link quality measurement, or report false node utility to attract its neighbour's routing messages or data packets. Furthermore, we assume multiple nodes cannot collude to launch more sophisticated attacks.

6.3. The Optimal and Secure Routing Protocol

This part describes the Optimal and Secure Routing (OSR) protocol in detail, and proves that OSR is able to defend against various active attacks as well as discover optimal routing paths.

6.3.1. OSR protocol design

There are four main phases in OSR. The first phase is to the authentication part, which aims to protect the integrity of the routing packets and prevent the interference of outside attacks. Based on the characteristics of an active attack, an attacker may try

to fake the link quality value or report false node utility. Then, we explain how to find paths including the nodes with both maximum trust value and utility in phases two and three from these two aspects. We utilise game theory method to investigate the behaviour of attackers and adopt the trust value to set the node's security level at the same time. Finally, we describe the routing discovery procedure of OSR in the fourth phase.

Phase I: Authentication. TCC authenticates all the nodes that want to join the network. The authentication method is done based on PKI. Each node *i* is assigned with a pair of public key PK_i and private key SK_i. Public key is known to all the other nodes in the network, whereas private key is secret. After TCC issues keys to node *i* it will immediately notify all the other nodes in the network of *i*'s public key PK_i. Even though non-authenticated nodes can launch many attacks, such as flooding false node utility values to attract data packets, they will be identified immediately by other nodes or TCC because of the lack of their public keys. In other words, in this work we only consider inside attacks launched by authenticated nodes. Correspondingly, attacks launched by non-authenticated nodes outside the network can easily be prevented.

Phase II: Secure measurement of link quality. In the work, we focus on securing utility-based routing protocols. To calculate the path and node utilities, at the beginning each node has to measure the quality of all its links. In this phase, malicious nodes can manipulate the messages exchanged so as to affect the link quality measurement of its neighbours.

Suppose (i,j) is an arbitrary link, and thus node i and j are arbitrary nodes in the network. We define three types of link quality values:

Real link quality: It is the real metric that characterises the quality of the link, denoted by Q(i,j).

Measured link quality: Either node i or j can be a malicious node, and can cheat during the link quality measurement process. Therefore, it is possible the measured link quality could be different from the real one. The link quality measured by node i and j are written as $Q_{i(i,j)}$ and $Q_{j(i,j)}$ respectively.

Reported link quality: After link quality measurement is done, node i and j are required to report their measured link quality to TCC. Again in this process there may exist malicious behaviours. We assume that the link quality values reported to TCC by node i and j are $Q'_{i(i,j)}$ and $Q'_{j(i,j)}$, respectively.

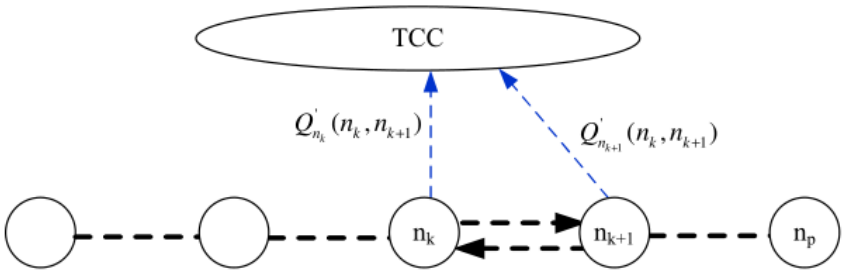

Figure 6.2: Secure measurement of link quality

Let us take ETX as an example of link quality metric. ETX value characterises the bidirectional transmission success ratios [ETX07], i.e., . $ETX(i,j)=\frac{1}{d_f \times d_r}$

So if one node of link (i,j) tells the other a fake transmission success ratio, then the determined $Q_i(i,j)$ and $Q_j(i,j)$ are not the same as the real link quality Q(i,j).

Now we use game theory to analyse the security problems in the link quality measurement process and describe the method adopted by TCC to determine the trust value for each node. Obviously, any node i has two measuring strategies:

a_1: Honestly measure the link quality;

a_2: (malicious behaviour): Cheat during measuring link quality.

In order to prevent a malicious node from choosing strategy a_2, in OSR we require nodes i and j to report their measured link quality values to TCC. Figure 6.2 depicts the link quality measurement phase. If TCC receives link quality reports from both i and j, and meanwhile $Q_i(i,j)=Q_j(i,j)$, then TCC believes they are normal nodes.In addition, TCC will record the quality of (i,j) for future use. Afterwards, TCC will set the trust value of nodes i and j to 1. Otherwise, both nodes i and j will be viewed as malicious nodes by TCC, and their trust values will be set to -1. Thus they will be segregated from the network later on if their trust values are below a certain threshold q_t. In addition, malicious nodes can launch attacks by no report or reporting the fake link quality values to TCC. We define node i's reporting strategy space as follows.

b_1: Report the measured link quality to TCC, i.e., $Q_i(i, j) = Q_i(i, j)$;

b_2: Report a fake link quality to TCC, i.e., $Q_i(i, j) \neq Q_i(i, j)$;

b_3: Do not report any link quality to TCC.

Finally, we define the {combined strategy} as the combination of a measuring strategy and a reporting strategy. Therefore, there will be $2 \times 3 = 6$ combined strategies for each node:

$s_1 = (a_1, b_1)$; $s_2 = (a_1, b_2)$; $s_3 = (a_1, b_3)$; $s_4 = (a_2, b_1)$; $s_5 = (a_2, b_2)$; $s_6 = (a_2, b_3)$.

In this strategy space, only s_1 is the normal or honest behaviour, and all the rest of the strategies are malicious. For link (i,j), nodes i and j choose their combined strategies s^i and s^j respectively in the game. Accordingly, TCC will determine the payoff, denoted by $F_Q(s^i, s^j)$, that nodes i and j will receive. Here, payoff corresponds to the node's trust value, and nodes i and j receive the same payoff. We now prove that the honest strategy $s_1 = (a_1, b_1)$ is the dominant strategy for node i. We assume that without knowing the counterpart's measured link quality value, nodes i and j coincidentally report the same link quality to TCC with probability e. Since a node's link quality value is within a continuous interval, e should be a very small value within (0,1).

Theorem 1: Strategy s_1 is a strictly dominant strategy for each node. Moreover, $q_t > 2e - 1$ is a sufficient condition to make sure all the nodes choose s_1 in order to remain in the network, where q_t is the threshold of trust value to detect a malicious node.

Proof: Following the traditional notation style, we use s_{-k} to denote the strategy set except s_k. TCC determines the payoff value according to node i and j's combined strategies. As node i and j do not collude, the payoff function is depicted in Table 6.1. Since nodes i and j have the same payoff, only one payoff value is listed in the table to reduce redundancy. Due to space limitations, we take two cases as an example to illustrate how to calculate payoff values.

Case 1. If either node chooses strategy s_3 or s_6, i.e., node i or j refuses to report link quality to TCC, then TCC will find link quality report missed and thus will set the payoff value to -1.

Case 2. If both i and j choose strategy s_2, they will report fake link quality values to TCC, even though they accurately measure the quality of the link (i,j). Since the probability that nodes i and j report the same link quality value is $Pro\left(Q_i(i,j) = Q_j(i,j)\right) = \theta$, the expected payoff is

$$F_Q(i,j) = 1 \times \theta + (-1) \times (1-e) = 2e - 1.$$

We have $2e - 1 \approx -1$, since $e \approx 0$. Thus, $-1 < 2e - 1 < 1$. According to Table 6.1,

apparently $F_Q(s_1,*) > F_Q(s_{-1},*)$. Therefore, s_1 is a strictly dominant strategy for node i. This conclusion also applies to node j. The strategy set (s_1, s_1) is a Nash Equilibrium. As one round of this game finishes, TCC determines whether a node should be segregated from the network by comparing its trust value (payoff) with threshold q_t. As long as $2e-1 < q_t$, a node choosing strategy s_{-1} will be segregated from the network ($F_Q(s_{-1},*) < q_t$), and thus will not be able to launch more attacks on the routing protocol. In other words, all the nodes have to choose s_1 in order to remain in the network.

Table 6.1 Payoff matrix for the game in link quality measurement phase

		node j					
		s_1	s_2	s_3	s_4	s_5	s_6
node i	s_1	1	-1	-1	-1	$2\varepsilon-1$	-1
	s_2	-1	$2\varepsilon-1$	-1	$2\varepsilon-1$	$2\varepsilon-1$	-1
	s_3	-1	-1	-1	-1	-1	-1
	s_4	-1	$2\varepsilon-1$	-1	$2\varepsilon-1$	$2\varepsilon-1$	-1
	s_5	$2\varepsilon-1$	$2\varepsilon-1$	-1	$2\varepsilon-1$	$2\varepsilon-1$	-1
	s_6	-1	-1	-1	-1	-1	-1

Because of Theorem 1, even for a malicious node, it must accurately measure link quality and report to TCC so as to remain in the network. Otherwise, the whole network is aware of the existence of the malicious node, which will not be able to launch more attacks in subsequent phases.

Phase III: Secure utility calculation. Now that the secure link quality measurement phase finishes, TCC should collect all the links' quality values. Meanwhile, existing

malicious nodes are already segregated from the network by TCC, and the links associated with the malicious nodes are removed from the link set. In addition, each node knows the real link quality values for its associated links. Now we need to design a security mechanism for nodes to calculate their real node utility values. This is because the routing protocols are based on node's utility value; fake node utility values will negatively affect the routing protocols, resulting in degradation of routing performance, for example, a sub-optimal routing path is chosen.

We look at a specific routing path $P_{n_0} = \{n_0, n_1, .., n_k, .., n_p\}$, where n_0 and n_p are the source and destination, respectively. To determine the path utility of $P_{n_k} = \{n_k, n_{k+1}, .., n_p\}$, node n_k needs to obtain from node n_{k+1} the path utility of $P_{n_{k+1}} = \{n_{k+1}, .., n_p\}$. However, node n_{k+1} can refuse to tell its path utility value or send the false path utility value to n_k, if it is a malicious node. In this case, we should propose certain mechanisms to enforce malicious nodes to honestly behave during the utility calculation process. Otherwise, they will be caught and expelled from the network.

While receiving a request for path utility value from node n_k, node n_{k+1} should first sign the path utility $U_{path}(P_{n_{k+1}})$ by using its private key $SK_{n_{k+1}}$. Then the signed utility value, denoted by $SIG_{n_{k+1}}(U_{path}(P_{n_{k+1}}))$, will be sent to node n_k. By using node n_{k+1}'s public key $PK_{n_{k+1}}$, node n_k is able to successfully decode and obtain $U_{path}(P_{n_{k+1}})$, based on which n_k can determine $U_{path}(P_{n_k})$ by the following equation.

$$U_{path}(P_{n_k}) = U_{path}(P_{n_{k+1}}) + Q(n_k, n_{k+1}) \quad (6.2)$$

Afterwards, n_k is supposed to report two values to TCC: its own path utility value determined by (1.1) and n_{k+1}'s signature of path utility value received from n_{k+1}.

However, it is possible that both n_{k+1} and n_k cheat to disrupt the utility calculation if they are compromised. During this process, n_{k+1} has the following strategies.

c_1: Apply its signature on the real path utility $U_{path}(P_{n_{k+1}})$, and send $SIG_{n_{k+1}}(U_{path}(P_{n_{k+1}}))$ to node n_k;

c_2: Apply its signature on a fake path utility $U\text{¢}_{path}(P_{n_{k+1}})$, and send $SIG_{n_{k+1}}(U\text{¢}_{path}(P_{n_{k+1}}))$ to node n_k;

c_3: Do not sign the path utility $U\text{¢}_{path}(P_{n_{k+1}})$ but send $U\text{¢}_{path}(P_{n_{k+1}})$ directly to n_k, or send nothing.

After sending a request for n_{k+1}'s path utility to n_{k+1}, node n_k will view any response from n_{k+1} as n_{k+1}'s signed path utility value. If node n_{k+1} chooses strategy c_3, which is a malicious action, then it will be caught by TCC at the end of the utility calculation phase. This is because node n_k needs to report the received data as n_{k+1}'s signature to TCC. Moreover, node n_k cannot create n_{k+1}'s signature due to lack of n_{k+1}'s private key $SK_{n_{k+1}}$. Therefore, TCC is able to conclude that n_k or n_{k+1} is a malicious node and

they will be segregated from the network. In this way, malicious action c_3 can be prevented. Later on, we will only consider two strategies, c_1 and c_2 for node n_{k+1}.

In terms of node n_k's strategy space, it can report the received signature of node n_{k+1}'s path utility (e_1), report a fake signature of node n_{k+1}'s path utility (e_2), or report nothing to TCC (e_3). Meanwhile, it can report its calculated path utility (d_1), report a fake path utility (d_2), or report nothing to TCC (d_3). Since node n_k does not have n_{k+1}'s private key, it is impossible for n_k to create the right signature of n_{k+1}. So if n_k is compromised and chooses action e_2, it will immediately be identified by TCC, which then sets n_k's trust value to -1. In addition, if node n_k chooses d_3 or e_3 to deny reporting to TCC, TCC will set its trust value to -1 as well. Finally, n_k will be segregated from the network if its trust value falls below threshold q_t in the utility calculation phase. Therefore, malicious strategies including d_3, e_2 and e_3 can easily be prevented. So we will not consider those strategies in the proof later on. Now we could shrink n_k's strategy space as follows.

$f_1 = (d_1, e_1)$: Report the calculated path utility $U_{path}(P_{n_k})$ and n_{k+1}'s signature to TCC;

$f_2 = (d_2, e_1)$: Report a fake path utility $U_{path}(P_{n_k})$ and n_{k+1}'s signature to TCC.

According to node n_{k+1} and n_k's strategies, TCC will calculate trust values (payoffs) for n_k and n_{k+1} respectively. The payoff function is denoted as $F_{path}(f^{n_k}, c^{n_{k+1}})$.

Theorem 2: All the nodes in the network will honestly calculate their path utility values in OSR protocol.

Proof: Let us look at an arbitrary link (n_k, n_{k+1}) in path $P_{n_0} = \{n_0, .., n_k, .., n_{p-1}, n_p\}$. Due to the above explanation, we are safe to only consider strategies: c_1, c_2, f_1 and f_2. Since TCC knows the global network topology and link quality information after the first phase, it is able to compute the path utility for all the paths in the network. Once the reported path utility is different from the real one, TCC will treat the node as malicious and set its corresponding trust value to -1. Therefore, we have the payoff matrix listed in Table 6.2.

Obviously, $F_{path}(f_1, *) \geq F_{path}(f_{-1}, *)$ and $F_{path}(*, c_1) \geq F_{path}(*, c_{-1})$. Therefore, strategies f_1 and c_1 are the weakly dominant strategies for node n_k and n_{k+1}, respectively. Moreover, once a node chooses to cheat, it receives a trust value of -1 and this malicious behaviour will be caught by TCC due to its trust value being lower than q_t. Only if nodes n_k and n_{k+1} are honest in calculating path utility can they remain in the network. Otherwise, they will be segregated from the network while the utility calculation phase ends.

Table 6.2: Payoff matrix for the game in path utility calculation phase

		node n_{k+1}		
		g_1	g_2	g_3
node n_k	h_1	$(1,1)$	$(1,-1)$	$(-1,-1)$
	h_2	$(-1,0)$	$(-1,0)$	$(-1,0)$
	h_3	$(-1,0)$	$(-1,0)$	$(-1,0)$

The above theorem guarantees that in OSR a node can correctly calculate the path utility value if it honestly complies with the routing protocol. Meanwhile, malicious behaviours can be prevented. For an arbitrary node n_0, after calculating the path utilities for all the possible paths from n_0 to destination n_p, it is able to determine its node utility $U_{node}(n_0, n_p)$ according to (6.1). However, even though n_0 knows the exact node utility value, it still can launch attacks by broadcasting false node utility value to yield sub-optimal routing paths. This sort of malicious action will be discussed in the next subsection.

Phase IV: Secure route discovery. In this phase, the source node n_0 starts to discover the optimal routing path to destination n_p. To achieve the best possible routing performance, we should guarantee that the neighbouring node with the highest node utility is selected into the routing path. However, during the route discovery or establishment process, malicious nodes can launch attacks by broadcasting false node utility to disturb this process. For example, an attacker tells its neighbours a fairly high node utility value, so that all its neighbours select it as the forwarding node in the routing path. In this way, the attacker is able to fuse all the packets from its neighbours, and then tamper with or drop the packets.

Again we focus on a specific link (n_k, n_{k+1}), and suppose node n_k is discovering a routing path to destination n_p. The real node utility of n_{k+1} is $U_{node}(n_{k+1}, n_p)$, whereas the node utility that is broadcasted to n_k, denoted by $U\text{¢}_{node}(n_{k+1}, n_p)$, may not equal the real one. Node n_k is supposed to receive node utility values from all of its neighbours,

and then chooses the neighbour with the largest node utility as the next hop to destination n_p.

In the OSR protocol, node n_{k+1} is required to perform digital signature over the broadcasted node utility value, denoted by $SIG_{n_{k+1}}\left(U_{node}^{¢}(n_{k+1}, n_p)\right)$. Node n_k is required to report $SIG_{n_{k+1}}\left(U_{node}^{¢}(n_{k+1}, n_p)\right)$ to TCC once this signature is received from n_{k+1}. The communications in the secure route discovery phase are shown in Figure 6.2, where node n_{k+1} and $n_{k+1}^{¢}$ which are neighboring to node n_k may cheat n_k by broadcasting fake node utility values.

Now we start constructing the game. Node n_{k+1} has three strategies:

g_1: Use $SK_{n_{k+1}}$ to sign the real node utility and broadcast to its neighbours, i.e., $U_{node}^{¢}(n_{k+1}, n_p) = U_{node}(n_{k+1}, n_p)$;

g_2: Use $SK_{n_{k+1}}$ to sign a fake node utility and broadcast to its neighbours, i.e., $U_{node}^{¢}(n_{k+1}, n_p) \neq U_{node}(n_{k+1}, n_p)$;

g_3: Do not sign the node utility $U_{node}^{¢}(n_{k+1}, n_p)$ but broadcast $U_{node}^{¢}(n_{k+1}, n_p)$ directly, or do not broadcast any message.

Node n_k's strategy space is as follows.

h_1: Report the received n_{k+1}'s signature of node utility to TCC;

h_2: Report a fake signature of n_{k+1}'s node utility to TCC;

h_3: Do not report to TCC.

Lemma 1: In OSR protocol, each node will honestly broadcast its node utility to its neighbours in order to remain in the network.

Proof: As node n_k is not able to create n_{k+1}'s signature, forging n_{k+1}'s signature (h_2) will be detected by TCC. Meanwhile, missing report from n_k (strategy h_3) will be easily detected by TCC as well. So in these two cases, node n_k will be viewed as a malicious node and its trust value will be set to -1. Another trivial case is that node n_{k+1} chooses g_3. Since TCC cannot receive n_{k+1}'s signature, it will be labeled as a malicious node and receive a trust value of -1 as well. The detailed payoff matrix is listed in Table 6.3. Note that if TCC cannot determine whether a node is malicious or normal, it assigns a trust value of 0 to the node.

Table 6.3: Payoff matrix for the game in route discovery phase

		node n_{k+1}		
	g_1	g_2	g_3
node n_k	h_1	$(1,1)$	$(1,-1)$	$(-1,-1)$
	h_2	$(-1,0)$	$(-1,0)$	$(-1,0)$
	h_3	$(-1,0)$	$(-1,0)$	$(-1,0)$

Table 6.3 shows that h_1 and g_1 are the weakly dominant strategies for node n_k and n_{k+1}, respectively. Moreover, only if the strategy set (h_1, g_1) is chosen, will node n_k and n_{k+1} not be segregated from the network. Using other strategies will result in n_k or n_{k+1}'s trust value lower than q_t, since $q_t > -1$ and $q_t > 0$. Therefore, all the malicious nodes will be caught and eventually segregated from the network. In other words, in order to remain in the network, each node has to honestly broadcast its node utility to its neighbours.

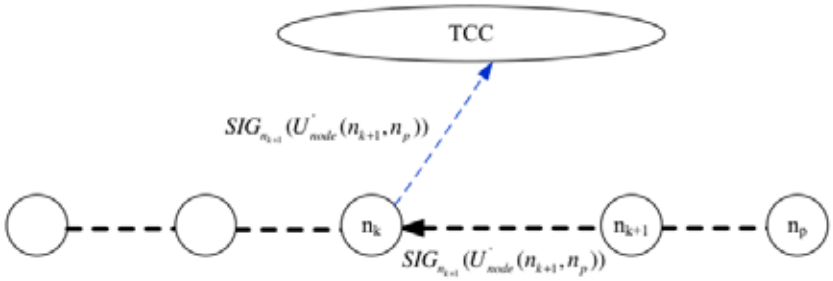

Figure 6.3: Secure route discovery

Theorem 3: In OSR protocol, the neighbour which has the largest node utility among the neighbour list will be selected as the next hop in the routing path. Eventually, the optimal route will be found.

Proof: For an arbitrary node n_k, due to Lemma 1, it will receive the real node utility values from all its neighbours. Then, n_k is able to select the neighbour which has the largest node utility as the next hop for the route to the destination. In fact, node utility reflects a node's capability of successfully routing data to the destination. Starting from the source node, all nodes select the neighbour with the highest node utility as

the next hop. Finally, it guarantees that the selected routing path has the optimal performance.

After the route discovery phase finishes, the optimal routing path should be constructed. What's more, all the malicious nodes or attackers in the routing process can be detected and segregated from the network so that they are not allowed to launch more attacks.

6.3.2. Discussion

As mentioned in section III-A, each node in the network needs to periodically send reports to TCC for identifying malicious nodes. Shorter period D causes nodes to report to TCC more frequently, and thus malicious nodes can be caught more quickly, and vice versa. Therefore, we will examine the impact of D on the detection rate in the following evaluation part.

A path that does not contain any malicious nodes is called a secure path. The optimal routing path stands for a secure path that has the largest path utility among all the secure paths from source to destination. Even if we give efficient proofs on the behaviour of demonstrating the optimality and security of OSR, there is still a special case that may cause its failure. In OSR protocol, some normal nodes may be treated as malicious by mistake and excluded from the network, we call this phenomenon a false alarm. Therefore, if normal nodes are segregated from the network, the discovered routing path may become suboptimal. Thus, both the detection rate and false alarm rate will be studied in the next section.

6.4. Evaluation

We carried out an evaluation of OSR by using simulations written in Java. We construct a 1500m $\times$1500m area, where 100 nodes are randomly deployed. Among those nodes, d percent of them are malicious. The communication range of each node is 100 m, namely, there is a link between two nodes if their distance is shorter than or equal to 100 m. However, those links are associated with different link quality values. Refer to Theorem 1, we set the threshold of detecting malicious nodes to 0:1 for this given scenario, i.e., $q_t = 0.1$. For each pair of normal nodes, they try to discover the optimal routing path between them. In terms of a digital signature, we use the well-known RSA algorithm with 1024-bit keys. The discovered routing path by OSR protocol is denoted by P_{osr}. In the simulation, we are concerned about the following performance metrics.

Successful detection rate (SDR): The percentage of malicious nodes that have been successfully detected.

False alarm rate (FAR): The percentage of normal nodes that have been caught and treated as malicious nodes.

Normalised quality difference between the discovered and the optimal routing path (PQD): Suppose the path utility of the optimal path which is defined in Section IV-B is $U_{path}^{\max}$. Then, $PQD = \frac{U_{path}^{\max} - U_{path}(P_{osr})}{U_{path}^{\max}}$. In fact, PQD characterises the optimality of the discovered routing path by OSR protocol.

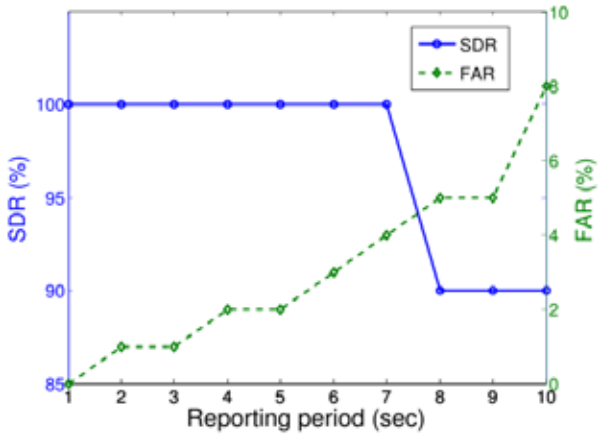

Figure 6.4: Successful detection rate and false alarm rate

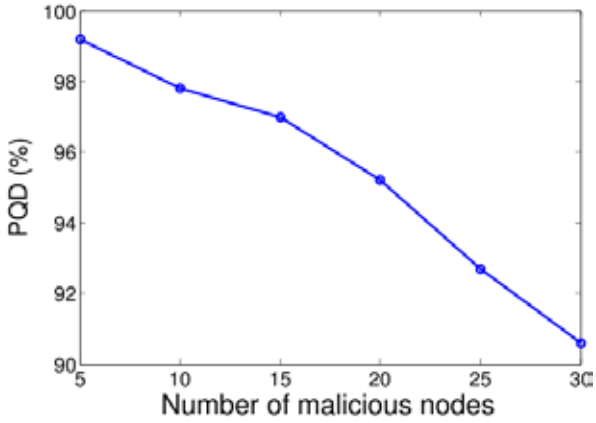

Figure 6.5: Optimal paths discovery rate

First, we set the number of malicious nodes to 10, and vary the reporting period D from 1 sec to 10 sec [TO06]. The simulation runs for 300 sec, i.e., the routing procedure will be finished in 300 sec. Thus, we can find the impact of D on the value of SDR and FAR, as well as testify the detection accuracy of OSR protocol after 300 sec. As shown in Figure 6.4, we observe that the malicious nodes will be detected with a 100% detection rate if each node sends reports to TCC when the value of reporting period is less than 7 sec. In addition, the evaluation results show that OSR

can successfully detect almost all the malicious nodes (SDR = 100%) with a low false alarm rate (FAR < 10% for all the cases).

Now we fix the reporting period and vary the number of malicious nodes. We randomly choose the value of D in the interval of (1, 7), here we set D = 6 sec. Since normal nodes may be treated as malicious when a false alarm occurs, then they will be segregated from the network. This may lead OSR to choose sub-optimal paths. Figure 6.4 demonstrates that OSR protocol has a low false alarm rate, which implies that OSR protocol is able to discover the optimal routing path with a high probability. In Figure 6.5, the statistics for PQD show that more than 90% of the discovered routing paths are optimal when the number of malicious node varies from 5 to 30. From this simulation study, we demonstrate that OSR protocol is secure and resilient to attacks, i.e., it can detect malicious nodes with high accuracy. Moreover, OSR protocol can achieve high routing performance, i.e., the optimal routing path can be discovered with high probability.

6.5. Summary

In this work, we propose an optimal and secure routing (OSR) protocol to segregate malicious nodes as well as find the paths for corresponding source-destination node pairs with maximum routing utility value. We utilise game theory methods to investigate the behaviour of active attacks and develop a centralised method to issue the node’s trust value to guarantee the security of utility-based routing selection procedure. Based on both analytical and experimental results, the OSR can achieve

more than a 90% attack detection rate and satisfy the best-utility selection requirement.

Currently, the OSR has a Trust Clearance Centre (TCC) to manage the trust value. Therefore, this may introduce potential security issues and extra communication overhead. So, in the future, we will further improve our ideas and work on the distributed algorithm for calculating the trust value of each node. Furthermore, in this work, we assume that the attackers cannot co-operate with each other to launch collusion attacks. Thus, we will take this kind of attack into consideration.

Chapter 7. Conclusions and Future Works

In this Chapter, we summarise our original contributions in Section 7.1 and outline the directions for future research in Section 7.2

7.1. Conclusions

As the prevalence of multi-hop wireless networks applications, security issues have received significant attention in recent years. Routing protocols play an important role in establishing and maintaining the topology of the entire network. However, it is vulnerable to all kinds of attacks due to a lack of fixed architecture. Therefore, how to prevent the routing procedure from the impact of malicious nodes is a challenging research topic. Attackers which actively drop packets, modify and manipulate routing messages can launch active attacks. Correspondingly, as the name suggests, passive attacks do not initiate malicious behaviour but are just monitoring the vulnerable packets. In this thesis, we focus on dealing with active attacks. One of the major security concerns is how to guarantee the integrity and confidentiality of routing packets. Based on OLSR routing protocol, we propose Security Extended OLSR (SE-OLSR) protocol, which utilises cryptography and a digital signature to secure the routing procedure. Then, we implement it on a Linux platform to identify its accuracy and then transplant it to our mesh routers T902 and IBM laptops to set up a Wireless Mesh Network (WMN) testbed for the convenience of the following study.

Wormhole attack is one of the most well-known severe attacks in multi-hop wireless networks. It is powerful and hard to detect. Thus, it inspires researchers to defend against wormhole attacks. Our research into wormhole attack detection is based on experimental analysis through our WMN testbed. This differentiates our work from others, because existing works only evaluate their methods by evaluation; few consider the actual network conditions. In order to fill this gap, we first investigate the characteristics of wormhole attack by theoretical analysis, and then utilise our WMN testbed to study the impact of wormhole attacks through comprehensive experiments. Some existing works use RTT to identify wormhole attacks. However, based on our fundamental analysis, we demonstrate that the standard deviation of RTT (stdev(RTT)) could be a more efficient and effective metric to distinguish wormhole attack. Accordingly, we design a Neighbour-Probe-Acknowledge (NPA) algorithm to detect wormhole attacks. We have evaluated the NPA by extensive experiments on the real testbed. Compared with existing works, NPA does not need time synchronisation or extra hardware support. More importantly, it can still achieve a higher detection rate and a lower false alarm rate than those methods using RTT under different background traffic load conditions.

An open research topic motivates us to not only consider the security issues in multi-hop wireless networks, but also to take routing performance into account. Thus, we design and propose an Optimal and Secure Routing (OSR) protocol which is able to find secure paths resilient to active attacks with the best routing performance at the same time. Traditionally, routing protocols are designed for discovering paths efficiently and with high-quality links under the assumption of a trustworthy and

friendly environment. Moreover, due to security concerns, a large number of works propose methods to detect specific attacks. But only a few consider routing performance while also ensuring routing security. Therefore, we aim to design a new secure routing protocol OSR, which takes routing performance optimisation into consideration. OSR is able to find the secure path with maximum utility value as well as resilience to active attacks. We use game theory method to identify the behaviour of malicious nodes and adopt trust values to segregate them from the network. We also prove that OSR can discover optimal paths that are resilient to active attacks through both theoretical analysis and extensive simulations.

In summary, our algorithms and protocols are theoretically and experimentally proven to provide routing security and performance in multi-hop wireless networks. So, we can claim that we have achieved the objectives stated at the beginning of this thesis.

7.2. Future Research Work

We will end this thesis with our perspective on the ways in which our current research can be advanced.

Our NPA has considered real conditions, such as channel condition and background traffic load, the network at the time of designing. This approach lacks the concern of different types of traffic such as video flow. In the future, we will design a more comprehensive experimental scenario to consider this pattern. We have realised that diverse kinds of traffic flow may lead to unpredictable impacts of wormhole attacks.

So far, our OSR protocol is able to find secure paths with maximum routing utility. However, we introduce a third trust party named Trust Clearance Centre (TCC) to manage the trust value of each node. In future work, we want to improve this centralised algorithm and try to design a distributed method - because the TCC may bring about potential new security issues to multi-hop wireless networks. In addition, in this work, even though we consider a complex attack model, co-operative attacks also need to be taken into consideration, since this kind of attack is more powerful and disrupts the detection procedure.

Finally, we would like to investigate the overhead by using attack detection or the methods concerning security issues. Security enhancement and attack detection need to create a certain amount of messages for frequent communication between nodes. Thus, this may introduce extra costs and lower network throughput.

Bibliography

Bibliography

[SV05] Akyildiz, I.F., Xudong Wang. A survey on wireless mesh networks. in *IEEE Communications Magazine*, Vol. 43, pp. S23 - S30, Sep. 2005

[OLSR] Olsr daemon. [Online]. Available: http://www.olsr.org/

[HAWK09] J. Cao, K. Xie, W. Wu, C. Liu, G. Yao, W. Feng, Y. Zou, J. Wen,C. Zhang, X. Xiao, X. Liu, and Y. Yan, Hawk: Real-world implementation of high-performance heterogeneous wireless network for internet access, in *IEEE International Conference on Distributed Computing Systems Workshops*, 2009.

[DEL06] H. Chiu and K.-S. Lui, Delphi: Wormhole detection mechanism for ad hoc wireless networks, in *International Symposium on Wireless Pervasive Computing (ISWPC)*, 2006.

[IE802] IEEE 802.11 standards. [Online]. Available: http://www.ieee802.org/11/

[AIR] Aircrack-ng. [Online]. Available: http://www.aircrack-ng.org/

[IPERF] Iperf. [Online]. Available: http://sourceforge.net/projects/iperf/

[AES] Advanced encryption standard (aes). [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[WORM06] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Wormhole attacks in wireless networks, in *IEEE Journal on Selected Areas in Communications*, Vol. 24, pp. 370-380, 2006.

[SA04] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig, The sybil attack in sensor networks: analysis and defenses, in *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2004

[WRS10] R. Zhang, and Y. Zhang, Wormhole-resilient secure neighbor discovery in

underwater acoustic networks, in *IEEE International Conference on Computer Communications (INFOCOM)*, 2010.

[RS02] H. Deng,W. Li, and D. P. Agrawal, Routing security in wireless ad hoc networks, in *IEEE Communications Magazine*, vol. 40, pp. 70 – 75, 2002.

[GRAY08] D. M. Shila and T. Anjali, A game theoretic approach to gray hole attacks in wireless mesh networks, in *Military Communications Conference (MILCOM)*, 2008.

[SR03] C. Karlof and D.Wagner, Secure routing in wireless sensor networks: Attacks and countermeasures, in *IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[FIW09] M. Azer, S. Kassas, and M. E. Soudani, A full image of the wormhole attacks towards introducing complex wormhole attacks in wireless ad hoc networks, in *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 1, no. 1, pp. 41 – 52, 2009.

[PL03] Y.-C. Hu, A. Perrig, and D. B. Johnson, Packet leashes: A defense against wormhole attacks in wireless networks, in *IEEE International Conference on Computer Communications (INFOCOM)*, 2003.

[SA99] L. Zhou and Z. J. Haas, Securing ad hoc networks, in *IEEE Network*, vol. 13, pp. 24–30, 1999.

[SYA02] J. R. Douceur, The sybil attack. In *International workshop on Peer-To-Peer Systems (IPTPS)*, 2002.

[SUV04] Y.-C. Hu and A. Perrig, A survey of secure wireless ad hoc routing, in *IEEE Security and Privacy*, vol. 2, pp. 28–39, 2004.

[RSH03] Y.-C. Hu, A. Perrig, and D. B. Johnson, Rushing attacks and defense in wireless ad hoc network routing protocols, in *ACM Workshop on Wireless Security*, 2003.

[IDS08] G. Li, J. He, and Y. Fu, A distributed intrusion detection scheme for wireless sensor networks, in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2008.

[OPE09] F. Hugelshofer and P. Smith, Openlids–a lightweight intrusion detection system for wireless mesh networks, in *ACM Internation Conference on Mobile Computing and Networking (MobiCom)*, 2009.

[LID08] I. Krontiris, T. Giannetsos, and T. Dimitriou, Lidea– a distributed lightweight intrusion detection architecture for sensor networks, in *International Conference on Security and Privacy in Communication Networks (SECURECOMM)*, 2008.

[SEAD03] Y.-C. Hu, D. B. Johnson, and A. Perrig, Sead: secure efficient distance vector routing for mobile wireless ad hoc networks, in *Ad Hoc Networks*, vol. 1, p. 175192, 2003.

[SAR02] M. G. Zapata and N. Asokan, Securing ad hoc routing protocols. In *International Conference on Web Information System Engineering (WiSe)*, 2002.

[SOLSR04] A. Hafslund, A. Tnnesen, R. B. Rotvik, J. Andersson, and . Kure, Secure extension to the olsr protocol, in *OLSR Interop/Workshop*, 2004.

[IDW00] Y.-G. Zhang and W.-K. Lee, Intrusion detection in wireless ad-hoc networks., in *ACM Internation Conference on Mobile Computing and Networking (MobiCom)*, 2000.

[BSMR09] R. Curtmola and C. Nita-Rotaru, Bsmr: Byzantine-resilient secure

multicast routing in multihop wireless networks, in *Mobile Computing*, vol. 8, pp. 445–459, 2009.

[WAJ06] M. Cagalj, S. Capkun, and J.-P. Hubaux, Wormhole-based antijamming techniques in sensor networks, in *IEEE Transaction on Mobile Computing (TMC)*, vol. 6, pp. 100 – 114, 2006.

[ONION] Onion routing. [Online]. Available: http://www.onion-router.net/.

[DAW08] F. Abdesselam, B. Bensaou, and T. Taleb, Detecting and avoiding wormhole attacks in wireless ad hoc networks, in *IEEE Communications Magazine*, vol. 46, no. 4, pp. 127 –133, 2008.

[DEL06] H. Chiu and K.-S. Lui, Delphi: Wormhole detection mechanism for ad hoc wireless networks, in *International Symposium on Wireless Pervasive Computing (ISWPC)*, 2006.

[TL06] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, Truelink: A practical countermeasure to the wormhole attack, in *IEEE International Conference on Network Protocols (ICNP)*, 2006, pp. 75 – 84.

[LB06] W. L. Y. Zhang, W. Liu and W. Fang, Location-based compromise tolerant security mechanisms for wireless sensor networks, in *IEEE IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 24, pp. 247–260, 2006.

[WCL09] D. Dong, M. Li, Y. Liu, and X. Liao, Wormcircle: Connectivity-based wormhole detection in wireless ad hoc and sensor networks, in *IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, 2009.

[DLW05] L. Qian, N. Song, and X.-F. Li, Detecting and locating wormhole attacks in wireless ad hoc networks through statistical analysis of multipath, in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2005.

[VS04] B. B. Weichao Wang, Visualization of wormholes in sensor networks, in *International Conference on Web Information System Engineering (WiSe)*, 2004.

[BAN08] A. Hani, A. Suhail, K. Salil, and C. Tung, Banaid: A sensor network test-bed for wormhole attacks. in *Australia's Leading Computer Emergency Response Team (AusCERT)*, 2008.

[DSDV94] C.E. Perkins, and P. Bhagwat, Highly dynamic Destination-Sequenced Distance-Vector (DSDV) for mobile computers, in *ACM Special Interest Group on Data Communication (SIGCOMM)*, London, UK, August 1994, pp. 234-244.

[SAODV02] M.G. Zapata, and N. Asokan, Secure ad hoc on-demand distance vector routing, in *ACM Mobile Computing and Communications Review*, vol. 3, no. 6, July 2002, pp. 106-107.

[ETX07] J. B. Douglas S. J. De Couto, Daniel Aguayo and R. Morris, A high-throughput path metric for multi-hop wireless routing, in *Wireless Networks*, vol. 11, pp. 419–434, 2007.

[DN92] D. Bertsekas and R. Gallager, Data Networks. Prentice Hall, 1992.

[SE02] K. Sanzgiri and B. Dahill, A secure routing protocol for ad hoc network, in *IEEE International Conference on Network Protocols (ICNP)*, 2002.

[HT06] S. Roy, D. Koutsonikolas, S. Das, and Y. Hu, High-throughput multicast routing metrics in wireless mesh networks, in *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2006.

[TO06] Y. C. Huang, S. Bhatti and D. Parker, Tuning OLSR, in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2006.

[MC10] B. Chen and M. C. Chan, Mobicent: A credit-based incentive system for disruption tolerant network, in *IEEE International Conference on Computer Communications (INFOCOM)*, 2010.

[AODV] Perkins, C., Belding-Royer, E., and Das, S. (July 2003). Ad hoc On-Demand Distance Vector (AODV) routing. in Internet Engineering Task Force (IETF) *RFC 3561*. Retrieved 2010-06-18.

[SH11] Jing Dong, Reza Curtmola and Cristina Nita-Rotaru, Secure high throughput multicast routing in wireless mesh network, in *IEEE Transactions On Mobile Computing*, vol. 10, pp. 653 – 668, 2011.

[RMM04] R. Draves, J. Padhye and B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in *ACM Internation Conference on Mobile Computing and Networking (MobiCom)*, 2004.

[GRAY07] B. Xiao, B. Yu and C. Gao, CHEMAS: Identify suspect nodes in selective forwarding attacks, in *Journal of Parallel and Distributed Computing*, vol. 67, pp. 1218-1230, 2007.

[SWD05]L. Buttyan and L. Dora and I. Vajda, Statistical wormhole detection in sensor networks, in *IEEE International Workshop on Engineering Semantic Agent Systems (ESAS)*, 2005.