

# University of Wollongong - Research Online

## Thesis Collection

Title: Secure communication over mobile ad-hoc network

Author: Zhenfei Zhang

Year: 2009

Repository DOI:

### Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following: This work is copyright. Apart from any use permitted under the Copyright Act 1968, no part of this work may be reproduced by any process, nor may any other exclusive right be exercised, without the permission of the author. Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material.

Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

**Unless otherwise indicated, the views expressed in this thesis are those of the author and do not necessarily represent the views of the University of Wollongong.**

Research Online is the open access repository for the University of Wollongong. For further information contact the UOW Library: [research-pubs@uow.edu.au](mailto:research-pubs@uow.edu.au)

*University of Wollongong Thesis Collections*

*University of Wollongong Thesis Collection*

---

*University of Wollongong*

*Year 2009*

---

Secure communication over mobile  
ad-hoc network

Zhenfei Zhang  
University of Wollongong

Zhang, Zhenfei, Secure communication over mobile ad-hoc network, ME-Res thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, 2009.  
<http://ro.uow.edu.au/theses/839>

This paper is posted at Research Online.  
<http://ro.uow.edu.au/theses/839>

## **NOTE**

This online version of the thesis may have different page formatting and pagination from the paper copy held in the University of Wollongong Library.

## **UNIVERSITY OF WOLLONGONG**

### **COPYRIGHT WARNING**

You may print or download ONE copy of this document for the purpose of your own research or study. The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site. You are reminded of the following:

Copyright owners are entitled to take legal action against persons who infringe their copyright. A reproduction of material that is protected by copyright may be a copyright infringement. A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

# Secure Communication over Mobile Ad-hoc Network

A Thesis submitted in partial fulfilment of  
the requirements for the award of the degree

Master of Engineering by Research

from

University of Wollongong

by

Zhenfei Zhang

School of Electrical, Computer and Telecommunications  
Engineering

March 2009

# Declaration

I, Zhenfei Zhang, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Master of Engineering by research, in School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualification at any other academic institution.

# Acknowledgements

I would like to thank Dr. Raad Raad and A.Prof Willy Susilo, my supervisors, for their patient guidance and constant support. I admire Dr. Raad's knowledge in telecommunication and appreciate him taking me into the area of mobile ad hoc networking. I also respect A.P. Susilo's knowledge in cryptography field, and appreciate him guiding me in the right direction.

I would also like to thank my parents, who support me constantly with their love, without which I would never be able to have all my achievements.

# Publications

## **Conference Paper**

Zhenfei Zhang, Willy Susilo and Raad Raad, "Mobile Ad-hoc Network Key Management with Certificateless Cryptography", In Proceeding of the International Conference on Signal Processing and Communication Systems, 2008.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	The Challenges and Solutions . . . . .	2
1.3	Contribution of Thesis . . . . .	4
1.4	Thesis Structure . . . . .	4
<b>2</b>	<b>Key Management Schemes Review</b>	<b>6</b>
2.1	Pre-established Key Management . . . . .	6
2.1.1	Key Distribution . . . . .	6
2.1.2	Key Exchange . . . . .	8
2.2	Distributed System . . . . .	10
2.2.1	Public Key Cryptography with Distributed System . . . . .	11
2.2.2	ID-based Cryptography with Distributed System . . . . .	14
2.2.3	Drawbacks of Distributed Models . . . . .	16
2.3	Transitive Trust Model . . . . .	16
2.3.1	Mobility-based Key Management . . . . .	16
2.3.2	Certificate Chain-based Key Management . . . . .	17
2.4	Conclusion . . . . .	21
<b>3</b>	<b>Secure Routing Protocols Review</b>	<b>23</b>



3.1	ARAN . . . . .	23
3.2	SAODV . . . . .	26
3.3	Ariadne . . . . .	28
3.4	SLSP . . . . .	30
3.5	SEAD . . . . .	32
3.6	Conclusion . . . . .	34
<b>4</b>	<b>Certificateless Cryptography over MANET</b>	<b>35</b>
4.1	Certificateless Cryptography . . . . .	36
4.2	Generic Model . . . . .	36
4.2.1	Fully Distributed System . . . . .	38
4.2.2	Partially Distributed System . . . . .	39
4.3	Detailed Scheme . . . . .	40
4.4	Issues and Design Principles . . . . .	44
4.5	Conclusion . . . . .	45
<b>5</b>	<b>Secure Routing Protocol</b>	<b>46</b>
5.1	OLSR . . . . .	46
5.1.1	Basic Mechanism of OLSR . . . . .	46
5.1.2	Remaining Problems of OLSR . . . . .	47
5.2	Reputation System . . . . .	47
5.3	Reputed-OLSR . . . . .	49
5.3.1	Assumptions . . . . .	49
5.3.2	Generic Model . . . . .	49
5.3.3	Detailed Protocol . . . . .	50
5.4	Conclusion . . . . .	56
<b>6</b>	<b>Security Issues</b>	<b>58</b>
6.1	Fake Route Information . . . . .	59

6.2	Man in the Middle . . . . .	59
6.3	ID Stealth . . . . .	60
6.4	Pseudo Spoofing . . . . .	60
6.5	Whitewashing . . . . .	61
6.6	Shilling . . . . .	61
<b>7</b>	<b>Simulation and Results</b>	<b>63</b>
7.1	CL-PKE Over MANET . . . . .	63
7.1.1	Simulation with C . . . . .	63
7.1.2	Simulation with OPNET . . . . .	65
7.2	Rep-OLSR . . . . .	71
7.2.1	Simulation against Whitewashing Attacks . . . . .	71
7.2.2	Simulation with OPNET . . . . .	72
<b>8</b>	<b>Conclusions</b>	<b>78</b>
<b>A</b>	<b>Results of Whitewashing attacks</b>	<b>84</b>
<b>B</b>	<b>Glossary</b>	<b>91</b>

# List of Figures

2.1	Key Distribution in Partially Distributed Model . . . . .	13
2.2	Certificate Graph $G$ [16] . . . . .	18
2.3	3 Steps to Establish a Certificate Chain [16] . . . . .	19
5.1	Network Setup . . . . .	51
5.2	Neighbour Discovery . . . . .	52
5.3	Polling . . . . .	54
5.4	TC Message Forwarding . . . . .	57
7.1	Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Packet Dropped . . . . .	68
7.2	Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Route Discovery Time . . . . .	68
7.3	Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Packet Send . . . . .	69
7.4	Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Packet Received . . . . .	69
7.5	Simulation Results of CL-PKE over MANET with OPNET: 20 Users, Packet Dropped . . . . .	70
7.6	Simulation Results of CL-PKE over MANET with OPNET: 20 Users, Route Discovery Time . . . . .	71

7.7	Simulation Results of CL-PKE over MANET with OPNET:	
	20 Users, Packet Send . . . . .	72
7.8	Simulation Results of CL-PKE over MANET with OPNET:	
	20 Users, Packet Received . . . . .	73
7.9	Simulation Results of Rep-OLSR over MANET with OPNET:	
	20 Users, Routing Delay . . . . .	73
7.10	Simulation Results of Rep-OLSR over MANET with OPNET:	
	20 Users, Hello Message . . . . .	74
7.11	Simulation Results of Rep-OLSR over MANET with OPNET:	
	20 Users, MPR Count . . . . .	75
7.12	Simulation Results of Rep-OLSR over MANET with OPNET:	
	20 Users, Traffic Dropped . . . . .	75
7.13	Simulation Results of Rep-OLSR over MANET with OPNET:	
	20 Users, Traffic Received . . . . .	76
7.14	Simulation Results of Rep-OLSR over MANET with OPNET:	
	20 Users, TC message forward . . . . .	77

# List of Tables

3.1	RREQ and RREP in ARAN . . . . .	25
5.1	MRP Reputation Table, Stage 1 . . . . .	53
5.2	MRP Reputation Table, Stage 2 . . . . .	55
5.3	MRP Reputation Table, Stage 3 . . . . .	56
6.1	Six Types of Attacks . . . . .	58
7.1	Programming Environment for Key Generation . . . . .	63
7.2	Simulation Results of CL-PKE over MANET with C . . . . .	64
7.3	The AODV Parameters . . . . .	67
7.4	Programming Environment for Simulation against Whitewash- ing Attack . . . . .	71

## **Abstract**

A Mobile Ad-hoc Network (MANET) is an ideal network that merely consists of mobile devices without any pre-established infrastructure. However, the secure communication over a MANET is not straightforward. In this thesis, we present a solution for MANET secure communication. Generally speaking, it covers two main areas, namely key management and secure routing.

In the key management area, we present an idea of adopting certificateless public key encryption (CL-PKE) schemes over mobile ad hoc network (MANET), which have not been explored before. In the current literature, there exist two main approaches, namely public key cryptography and identity-based (ID-based) cryptography. Unfortunately, they both have some inherent drawbacks. To avoid these obstacles, Al-Riyami and Paterson proposed certificateless cryptography systems. In this thesis, we adopt Al-Riyami's advantage over MANET. To implement CL-PKE over MANET and to make it practical, we incorporate the idea of Shamir's secret sharing scheme. The master secret keys are shared among some or all the MANET nodes. This makes the system self-organized once the network has been initiated. In order to provide more flexibility, we consider both a full distribution system and a partial distribution system.

In the secure routing area, we present the idea of adopting a reputation system over the optimized link state routing (OLSR) protocol. In the literature, there exist two main routing approaches, namely proactive routing and reactive routing. Several secure reactive routing protocols have been

proposed. However, as far as proactive routing is concerned, few secure protocols are presented, yet they all possess different drawbacks that make them only practical on certain routing protocols. One of major problems is how to determine whether a node is worthy of trust or not. In other networks, for example, peer-to-peer sharing networks, reputation systems are designed to judge users. Unfortunately, they are designed specifically for peer-to-peer systems, while the adoption to MANET is not very straightforward. To this end, we present our Rep-OLSR, which selects routes wisely based on users' former performance by periodically collecting polling results from neighbour nodes.

Finally, we demonstrate that our solution is robust against several types of attacks. We also test our solution with several simulations. The results of the simulations indicate that our solution efficiently secure the communication with little extra traffic compared with pure MANET routing protocols.

# Chapter 1

## Introduction

### 1.1 Motivation

The mobile ad hoc network (MANET) is a network that merely consist of mobiles devices without any pre-established infrastructure. Many applications have been presented on MANET, for instance, peer-to-peer communications and group broadcast. However, in this type of networks, finding paths (routing) is not very straightforward, since unlike traditional network, MANET has no access point for the nodes to connect to and communicate with. Each users should be able to perform routing regardless of the neighboring users. The routing problem is even more troublesome when the users are freely joining and leaving the MANET, which consequently makes routes that are valid one second before unavailable. Fortunately, several routing protocols are able to deal with these problems. Theoretically they can be categorized into two types, proactive routing protocols and reactive routing protocols.

With the fast development of MANET technology, those protocols alone cannot meet the requirements of using MANET, especially in the area where security is concerned. Furthermore, security aspects exert more and more



influence over the design of protocols. With those applications developed over MANET devices, the need for MANET security increased significantly during the last several years.

To this end, we suggest that the old requirements of MANET, availability, scalability as well as mobility, cannot fully satisfy the applications. On top of that, we argue that a fully functional MANET routing solution should provide the following features:

- *Security*; the protocol is safe from malicious attacks.
- *Availability*; users can connect to any number of other users using this protocol.
- *Scalability*; the protocol should be efficient not only with a small number of users but also with a large number of users, and allow instant change of user numbers.
- *Mobility*; the protocol should allow the users change their location and online/offline status while providing acceptable connectivity.

In this thesis we propose a solution which meets all the above requirements. It consists of a key management scheme and a secure routing protocol.

## 1.2 The Challenges and Solutions

From a security point of view, the problem of secure communication can be divided into two sub-questions:

1. how to distribute keys?
2. how to perform routing?

The reason why scholars research those two topics separately and independently is to increase the research efficiency and simplify the problem, given that the relationship between a key management scheme and a secure routing protocol is considerably weak. Although few secure routing protocols are suitable for specific key management schemes, nonetheless, most key management schemes can be applied to several routing protocols without too much modification, and vice versa. Hence, in this thesis we adopt the advantage of this research method.

On one hand, for key management schemes, the basic problem is how to bind the keys to the node IDs. In public key cryptography, key binding is not an issue as a key generation center (KGC) is responsible for generating keys and certificates. However, in a MANET, the key distribution procedure is problematic, as a consequence of its non-centralized structure. Obviously, no user is always available, consequently, no user alone can be used as a KGC. Thus, we need to find another method to distribute keys.

On the other hand, for a secure routing, because MANET makes use of wireless signals which every user in the transmitting range is capable of receiving, routing information may be leaking, misleading, or maliciously dropped. To this end, to transmit the correct packet to the right node is challenging.

In this thesis, we present a solution to the problem of secure communication over a MANET. We firstly propose a distributed system with certificateless key cryptography and threshold secret sharing scheme to manage keys in Chapter 4. Instead of using one single server, which might be unavailable from time to time, we use several users collectively to form the key generation center. Then we adopt the reputation system and implement it to the Optimized Link State Routing (OLSR) protocol in Chapter 5. Unlike

XRep, X<sup>2</sup>Rep and X<sup>2BT</sup>Rep systems [21, 22, 30] that are designed specifically for peer to peer systems, our system is implemented uniquely over the optimized link state routing protocol [5]. Finally we carried our schemes out with several simulations. The results indicate that our schemes ensure users communicate effectively and securely.

### 1.3 Contribution of Thesis

Firstly, we presented a key management scheme for the MANET. We adopted the certificateless key cryptography to avoid the drawbacks of public key cryptography and ID-based cryptography. Moreover, in order to make the scheme functional on a MANET, we proposed a distributed system with threshold secret sharing scheme to make the MANET fully self-organized.

Secondly, we presented a secure routing protocol for the MANET. We adopted the reputation system that has been widely used over peer to peer file sharing systems and implement it to the Optimized Link State Routing (OLSR) protocol. The reputation system effects the multi-point relay (MPR) nodes selections of OLSR and eventually secures the routing.

Finally, we carried out our scheme and protocol with several simulations in C++ and OPNET to prove the efficiency and security. We also carried out our solution against several specific attacks to prove the correctness.

### 1.4 Thesis Structure

The rest of the thesis is organized as follows: in Chapter 2 we review three types of MANET key management schemes; in Chapter 3 we discuss five of most popular routing protocols; in Chapter 4 we present our key management

scheme; Chapter 5 presents our routing protocols; Chapter 6 discusses the security issues related to MANET and how we solve them; Chapter 7 presents the results of simulation of our schemes and finally, Chapter 8 concludes the thesis. In the next chapter, we will start with a review of existing key management schemes.

# Chapter 2

## Key Management Schemes Review

In the literature, there exist three ways to manage keys, namely, pre-established key management schemes, key management schemes with distributed system and key management schemes with a trust model.

### 2.1 Pre-established Key Management

In the literature, there exist two approaches to pre-establish keys; to distribute keys by a distributor and to exchange keys by users [28]. In MANET, both approaches are adopted.

#### 2.1.1 Key Distribution

One of the common methods to pre-establish keys is to generate and store the keys in a trusted third party and then distribute them directly to the users. This method is simple and easy to achieve. However, it faces several problems when it is brought into a MANET. Theoretically, there is no fixed

infrastructure in a MANET, thus no single user or server can be trusted or accessible constantly to provide authorization. Furthermore, the scalability in this model is another problem. Two facts make these schemes impractical in this situation:

1. every new user must be authorized prior to joining the network;
2. the certificate of the new user must be updated by every online users.

The first aspect prevents unregistered users from entering the network. Meanwhile, the second aspect ensures that the network can expand. However, in real scenarios the drawbacks are predictable. Since every user must register before the initialization, scalability of the network will be affected. Moreover, the frequent updating is time and resources consuming. This will also reduce the scalability.

Despite the inherent drawbacks of pre-established key management schemes, several approaches are presented to maximize the scalability, among which, Kaman [1] is one of the most efficient schemes.

**Kaman** Kaman is an example of pre-establishment key management scheme based on Kerberos Authentication [1]. It is an adoption of Kerberos system from traditional network. The authors also made some modification to adapt to MANET. In Kaman it is assumed that there are multiple Kerberos servers sharing a master secret key. It is also assumed that all the client nodes have their secret keys while the hash values of these keys are stored and duplicated in every Kerberos server. Every node needs to talk to at least one of the servers before setting up a connection.

In wired networks, although every Kerberos Server is a single point of failure, the performance of the network is acceptable because cable networks

are not easily to fail, and the servers can service heavy loads. Unfortunately, in MANET, any node is more vulnerable as they can freely join and leave the network. Moreover, most MANET devices are less efficient than the typical workstation and desktop, and hence, may not be able to handle as heavy load as servers from cable networks. To this end, Kaman proposed a multi-server model to strengthen the availability of Kerberos servers. The servers are elected based on the coverage and range of the nodes. Those nodes with the highest coverage and lifetime are selected to be the servers automatically.

However, the disadvantages of this model are noticeable. The first issue is security. The server election method is problematic as servers are elected based on their coverage and their lifetime. Thus, there is no method to prevent a malicious node from being elected as long as it has acceptable coverage and lifetime. Secondly, scalability is a concern. The servers need to exchange data frequently to synchronize data. Meanwhile, Kaman does not provide an efficient mechanism to deal with any newly joined nodes, for instance, how does a user who is not on the server's list obtain its secret key and certificate.

### **2.1.2 Key Exchange**

In the public key cryptography area, an alternative way to pre-establish keys is to share keys between every pair of nodes. This model is simple, easy to establish, and flexible, and most of all, can be implemented over MANET without any modification. Moreover, since it is applicable with either symmetric key cryptography or asymmetric key cryptography, it can work alone with some restrict routing protocols.

Nonetheless its drawback is also significant. Firstly, users in this model consume lots of resources due to its key sharing strategy. With the increasing

number of the users, this model becomes impractical, considering the limited computing power of the mobile devices. Secondly, authenticity is not ensured in this scenario. All the users are free to share the keys with others. Any malicious user is capable of obtaining the keys of other users.

To this end, Montenegro and Castelluccia [12] proposed a method called SUCV addresses to provide authenticity.

**SUCV Addresses** The statistically unique cryptographically verifiable (SUCV) address is proposed by Montenegro and Castelluccia [12]. In their model, each user will have a pair of public/private keys where the public key can be hashed into a unique IPv6 address.

A hash function is an one way function where you can map an arbitrary length of data to a fixed length of string. Theoretically, for a hash function  $h$ , it should at least provide the following feature [29]:

- *mixing-transformational*; for any arbitrary length of input  $x$ , the length of output  $h(x)$  should be fixed.
- *collision resistant*; for any given value  $x$ , it is computationally infeasible to find out another value  $y$  so that  $h(x) = h(y)$ .
- *pre-image resistant*; for any hash result  $h(x)$ , it is computationally infeasible to find out the input value  $x$ .
- *practical efficient*; for any given value  $x$ , it is relatively easy to compute  $h(x)$ .

The hash mechanism provides automatic binding between user's public key and user's IP address. Thus the public key is unforgeable. Unfortunately, the authenticity still remains unsolved as in this method malicious users can



forge their IP addresses. It also requires assistance from a trusted third party to verify the IP addresses. The difference is that in this scheme a trusted third party is used to store the binding of public keys and IP addresses whereas in other methods it is used to store the binding of public keys and private keys (certificates). To this end, the SUCV address alone cannot provide authenticity.

Although it does not provide an entire solution for MANET key management, the SUCV address successfully increases the key distribution efficiency and reduces the computing cost, because instead of verifying keys with certificates, users only need to verify the addresses with keys. To this end, the SUCV is adopted by several other schemes. For example, in ID-based cryptography [9], the SUCV addresses are used as IDs; in the mobility-based key management scheme [17], SUCV is optional to strengthen the authentication, while in the secure link state protocol (SLSP) for mobile ad hoc network [20], the SUCV model is recommended as the key management scheme.

## **2.2 Distributed System**

Although the pre-established key management schemes are simple, their drawbacks we discussed in the last chapter make them inefficient and impractical over a MANET. Fortunately, there is another model called distributed system. The intention of establishing a distributed system is to replace the trusted third party with a distributed system, which only consists of on-line users, so that the network can be fully self-organized. In the following sections, two different cryptography schemes are discussed over this system.

### 2.2.1 Public Key Cryptography with Distributed System

**Public Key Cryptography** The concept of the public key cryptography scheme was put forth by Diffie and Hellman in their seminal paper in [10] and the first realization of the public key cryptography was proposed by Rivest, Shamir and Adleman in 1978 [3].

In a public key cryptography, two separate keys are involved, namely the public key and the private key. In an encryption scheme scenario, the public key is used for encrypting the message and the private key is used to decrypt the message. The main idea of this system relies on the fact that if the private key is known, then it is easy to compute the public key, but not vice versa. Therefore, the public key can be made public and known by anyone. This method makes it possible for a user to deliver some messages without any pre-established shared keys.

Nonetheless, the key management is the main stumbling block in the public key scenario, since it is not possible for anyone who obtains someone's public key from a public place, such as the Internet, to verify the authenticity of this public key. Therefore, there is a necessity to authenticate this public key and hence, an adversary cannot replace a genuine public key with any other public key of its choice. Henceforth, a trusted third party called the certification authority (CA) is required. The role of the CA is to issue certificates on public keys for users. Then, anyone who obtains any user's public key can verify its authenticity by verifying whether the certificate attached is indeed valid.

However, to bring the public key cryptography in to the MANET is not straightforward. The CA plays a key role in key distribution, while the MANET does not trust any single node because it is so easily to be unavail-

able, thus, it is infeasible to select a CA from any single node. This is the main drawback of this system. Fortunately, Zhou and Haas [18] presented a distributed model to replace the CA.

**Partially Distributed Model** Zhou and Haas [18] adopt the idea of public key cryptography and implement it into MANET with a secret sharing method [23] over a partially distributed authority scheme. In their scheme it is assumed that there is an Offline Trusted Third Party (OTTP) constructing and distributing keys for all the nodes. As indicated in Figure 2.1, firstly, this OTTP generates a pair of master public/secret keys. The master public key (mpk) is known by every node in the MANET, while the master secret key (msk) is divided into  $n$  parts, where each part is represented by  $S_i$  ( $i = 0, 1, 2 \dots n$ ). Then OTTP picks  $n$  arbitrary nodes, randomly distributed with msk parts. These  $n$  nodes collectively form the Distributed Certificate Authority (DCA).

The OTTP then generates all the certificates and distributes them to the corresponding node. In Zhou and Haas' scheme, those certificates are fully stored in each DCA node as well. Any unauthorized node does not have a valid certificate, hence will not get key shares from DCA nodes.

Assuming the threshold of the system is  $t$ , any node, namely  $i$ , needs to obtain at least  $t + 1$  msk shares to retrieve the msk. Node  $i$  will send out requests to  $t$  DCA nodes, with a certificate of its own. Once the certificate is verified to be valid by those DCA nodes, which is achieved by comparing with DCA's certificate database, the DCA node will reply with a share of the msk. After successfully receiving  $t$  valid key shares, node  $i$  will retrieve the msk.

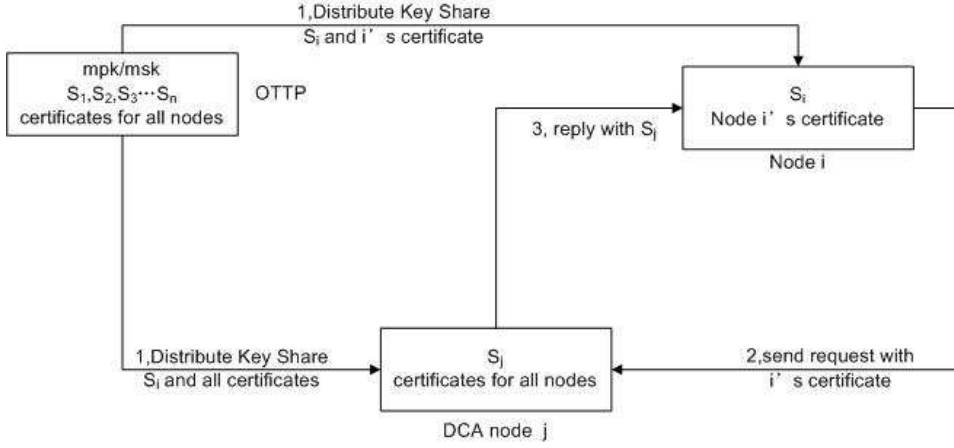


Figure 2.1: Key Distribution in Partially Distributed Model

This authentication scheme with the partially distributed system is similar to Kaman. The difference between them lies in the fact that in Kaman the msk is fully stored in every server, while in Zhou and Haas' scheme, the DCA node only knows a share of the msk.

The imbalanced load to the DCA nodes remains problematic because those DCA nodes issue keys for the whole network. Another drawback of this scheme is its necessity of pre-establishment before the initiation, as the certificates of each node are pre-stored in the DCA nodes.

In order to balance the loads, Yi and Kravers proposed a modified model [31]. Their scheme makes use of the broadcast certification request (CREQ) and the certification reply (CREP) packets. It allows nodes to broadcast the certification request (CREQ) packets using a flooding method. Any DCA node which gets this packet answers with a certification reply (CREP). If the node successfully collects  $t + 1$  CREPs, it will be able to reconstruct the full certificate. If the certificate is valid, the certification is successful; otherwise, the node will generate another CREQ packet.

**Fully Distributed Model** A fully distributed authority scheme is a modification of partially distributed certificate authority scheme, firstly proposed by Luo et al. [26]. This scheme also adopts the  $(n, t)$  threshold secret sharing scheme [23]. The difference between Luo et al.'s model and Zhou and Haas' model lies in the following facts: In Zhou and Haas' model, the DCA nodes are randomly selected from all the nodes while Luo et al.'s model picks all of the nodes in the MANET to form the DCA. The msk is shared among all the nodes and for this reason, this scheme is called "fully distributed".

Firstly, an offline trusted third party (OTTP) generates a key pair mpk/msk. The mpk is shared in the MANET. The msk is divided into  $n$  shares; each part is a Secret Key (sk) for every node. The nodes' Public Keys (pk) are created from those sks.

Then the OTTP creates certificates signed with the msk for each node, in order to bind nodes' unique ID with nodes' public key. These certificates are unforgeable and are stored in every node in the network.

When a node, namely, node A, needs to get the msk, it sends out requests to all of its one hop neighbor nodes. If one of the neighbor nodes, namely, node B, receives the request, it compares node A's ID and certificate with the information B stored in its database. If the result is positive, node B will send back its own share of the msk, as well as the certificate of itself. If the number of the nodes which replied with the valid certificates and the key shares is more than  $t$ , the node A obtains the msk.

## 2.2.2 ID-based Cryptography with Distributed System

**ID-based Cryptography** The concept of the identity-based (ID-based) cryptography was introduced by Shamir in [24] to solve the main drawback of public key cryptography by removing the necessity of the certificates. In

an ID-based system, the identity of users are used as their public keys and therefore there is no need to have these public keys (i.e. the users' identity) certified. The secret key is derived from the user's identity together with the trusted authority, called the Private Key Generator (PKG)'s secret key.

Nonetheless, this makes the system impractical since the PKG will know all the secret keys that the users have and therefore, the PKG can impersonate any user. This inherent problem in ID-based cryptography is known as the key escrow problem, which makes the ID-based system only practical in a closed organization. An unconditional trust to the PKG is required and it is assumed that the PKG will not be malicious.

**Distributed Model** One of the Identity-based authority schemes was proposed by Boneh and Franklin [9], which is a modified solution to Zhou and Haas' scheme. It replaced the DCA with a threshold private key generator (PKG).

Initially, users in the network will collectively form the PKG. This PKG will generate a pair of  $mpk/msk$ , and the  $msk$  is divided and shared among all the initial nodes. It is not stated in [9] how this PKG is formed nor how the  $msk$  is distributed. In [7], Van Der Merwe, Dawoud and McDonald designed an OTTP party which is called centralized PKG to generate and distribute keys. After the initiation, the user's identity is used as the user's public key, while each PKG node will generate a part of this user's private key, which is based on the user's identity. In this way, each user needs to obtain  $t + 1$  parts of private key to retrieve the private key.

### **2.2.3 Drawbacks of Distributed Models**

Both fully distributed model and partially distributed model can successfully issue keys for MANET users. However, both public key cryptography or ID-based cryptography have got inherent problems when they are applied over MANET. In the public key cryptography system, a certificate authority (CA) is required to issue certificates between users' public keys and private keys to ensure their authenticity, whilst in an ID-based cryptography system, users' private keys are generated by a key generation center (KGC), which means the KGC knows every users' keys (the key escrow problem).

To this end, in Chapter 4, we considered the certificateless cryptography with the distributed model.

## **2.3 Transitive Trust Model**

Another approach to manage keys other than distributed system is to use a trust model. This model requires transitive trust, which means if A trust B and B trust C, then A must trust C. This model requires no OTTP or any initial phases. The problem of authorization over the network becomes the problem of peer to peer authentications. In current literature, there exist two key management schemes, called Mobility-based key management scheme and certificate Chain-based key management scheme.

### **2.3.1 Mobility-based Key Management**

In [17], Capkun, Hubaux and Buttyan proposed a mobility-based key management. Unlike the schemes we have discussed before, it relies on nodes' mobility to solve the peer to peer authentication problem. They use an integrated side channel to establish a session key. Ideally, this side channel could

be an infrared interface or a Bluetooth interface, which can be established when the two users are physically close to each other. The physical encounter of two users can be considered as a visual authentication, and allows users to exchange communication keys and bind the counterpart's ID with those keys. In addition, the communication keys could be either asymmetric, which means both users generate their own public/private keys and exchange public keys, or symmetric, which means two users will agree on a common key using some existing key exchanging scheme.

Compared with distributed models, the avoidance of any OTTP is the major contribution of this scheme. Nodes are free from any type of third parties, and meanwhile purely self-organized. Another advantage is it is adaptable to both asymmetric key cryptography and symmetric key cryptography. This advantage brings some flexibility, since some routing protocols only allow asymmetric keys or symmetric keys.

However, the necessity of a secure side channel brings limitation. The users should be close to each other, or at least, in the transmission range of the side channel. The other disadvantage lies on the efficiency and scalability, as the system is much like the key exchange system where every two users establish a set of keys. As the authors pointed out, the efficiency of the scheme could be low, since the initiation time increases exponentially with the raise of the nodes' density and mobility.

### **2.3.2 Certificate Chain-based Key Management**

In addition to the mobility-based key management scheme, Capkun, Hubaux and Buttyan proposed certificate chain-based key management scheme. In this scheme, the public keys and certificates are generated by users themselves and shared with other nodes the same way as the Pretty Good Privacy



(PGP) [15]. The difference is, in PGP model the certificates are stored in a centralized infrastructure while in the Certificate Chain-based Key Management scheme the certificates are stored by the nodes in a self-organized way.

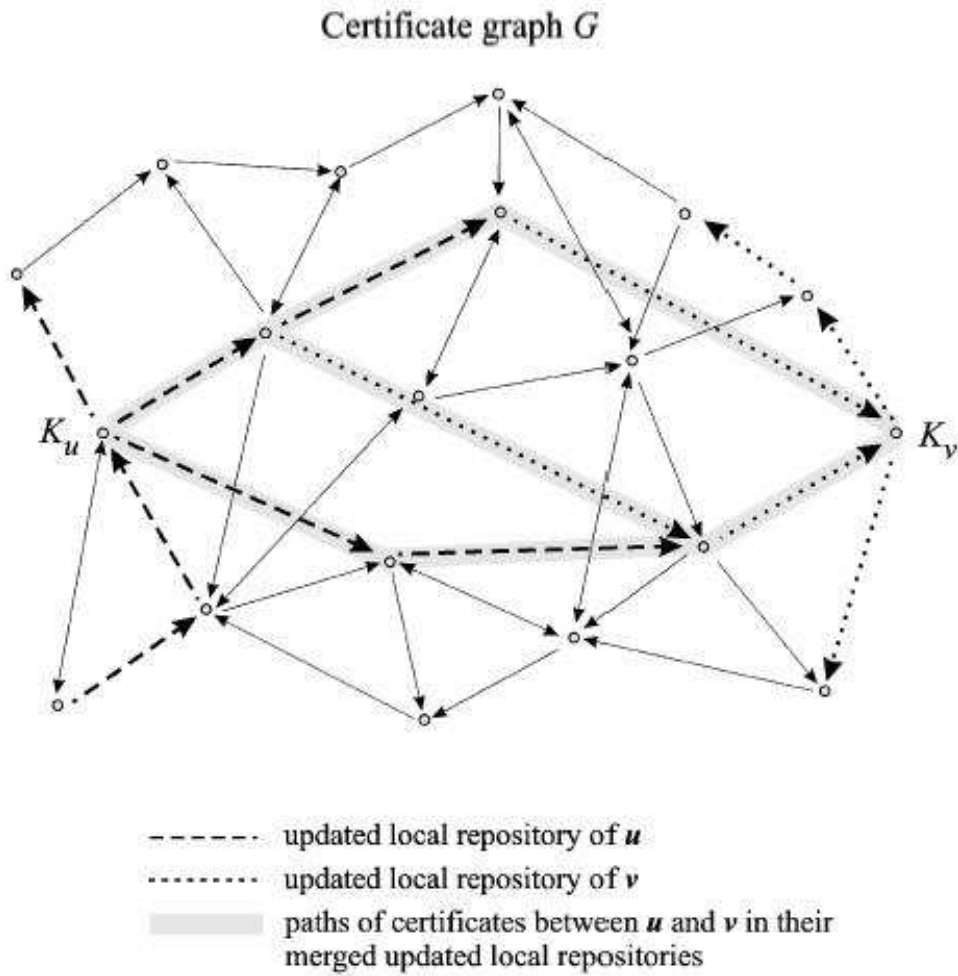


Figure 2.2: Certificate Graph  $G$  [16]

In their scheme, the certificates are represented as a directed graph  $G(V, E)$  as shown in Figure 2.2, where  $V$  and  $E$  stands for vertices and edges respectively. In this graph,  $V$  represents the public keys and  $E$  represents certificates. For example, an edge from node A to node B means a certificate signed by A's private key that binds B's public key with B's ID. Thus, a certificate chain from node A to node C means C is reachable from A in  $G$ . Thus the problem of certification becomes the problem of updating and exchanging  $G$ .

For any user  $U$ , two graphs are involved in this scheme, the updated graph  $G_U$  and the non-updated graph  $G_U^N$ . Four operations are involved on those graphs during the process.

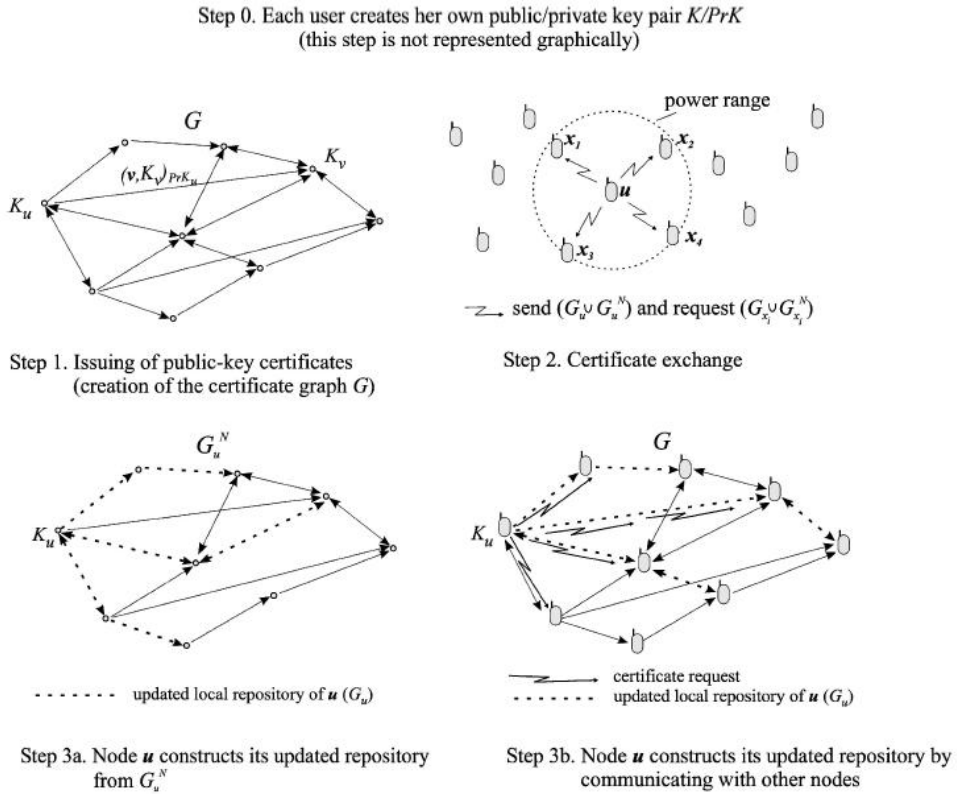


Figure 2.3: 3 Steps to Establish a Certificate Chain [16]

**Creation of Public Key and Certificates:** As shown in step 1 of Figure 2.3, the public/private key pairs are created by the users themselves. Node A will issue a certificate to Node B (with a timestamp  $T$ ), if it believes that  $pk_B$  belongs to B. Moreover, the certificate is duplicated so that both A and B will have a copy.

**Certificate Exchange:** As shown in step 2 of Figure 2.3, the certificates in  $G$  are shared and distributed in this operation. User A will send both  $G_A$  and  $G_A^N$  updating packets to its neighbors. Node B, for example, on receiving these graphs, will expand  $G_B^N$  based on the information. This provides the nodes with a complete view of the non-updated graph.

Note that only the hash values are contained in the  $G_A/G_A^N$  updating packets. Node B will compare the hash values with the existing certificates in  $G_B$ . Only when the result is negative will node B request for certificates and update  $G_B$ . This ensures the operation to be low cost because it uses only hash values and the exchange happens only between neighbour nodes.

**Constructing Updated Certificate Repositories:** As shown in step 3a and 3b of Figure 2.3, this is the operation where the nodes construct their updated graph  $G$ . To be specific, node A update  $G_A$  in two approaches. It can update  $G_A$  based either on  $G_A^N$  for a certificate if node A can communicate with directly the issuer or on  $G^N$  where  $G^N$  stands for the updated graph of neighbors.

**Certificate Revocation:** Each user is able to revoke a certificate it issued if it believes that the certificate becomes invalid. This can be achieved in two ways, namely explicit revocation and implicit revocation. The explicit revocation acts in a way that the issuer sends out explicit revocation statement. This statement will reach other users with the exchange of the

certificate graph. The implicit revocation makes use of the certificate's lifetime. Certificates will expire automatically when the lifetime is reached.

Generally speaking, in security area when two users intend to communicate they need to provide proof of themselves so that the counterpart can trust them. The proof could be a certificate issued by a CA that binds its ID with public key. It could also be an ID that has been registered in a KGC. However, such function cannot be found in the certificate chain-based key management scheme. The reason that A trusts B is not detailed in [16]. But the authors suggested that a side secure channel may exist, which is similar to the mobility-based key management scheme.

Although both certificate chain-based and mobility-based key management schemes are fully self-organized and remove the necessity of any type of trusted third parts, the former one improves certain levels of efficiency from the latter one. The main reason lies in the fact that the mobility-based key management scheme uses peer-to-peer certificate exchange while the certificate chain-based key management scheme allows users to broadcast their certificate graph.

However, due to its lack of certificate authority, the authentication is considerably weak. In [7], Van Der Merwe, Dawoud and McDonald commented that it is difficult to ensure valid transitive trust chain for more than two links. Furthermore, a chain is as strong as its weakest link. Any malicious node along the chain may cause false authentications.

## 2.4 Conclusion

This chapter reviews the existing MANET key management schemes which can be mainly categorized into three types. Firstly, as for pre-established

key management schemes, the ideas are comparatively straightforward while the implementations are complicated and yet the efficiency and scalability remains problematic. By contrast, the schemes with transitive trust models are much complex, but they are efficient and purely self-organized. Finally, the complexity of the remained model lies in the middle of the former two. Theoretically this model needs OTTP at the initiation stage. However, it becomes self-organized afterwards. Moreover, the authenticity of this model is better than the other two. To this end, we implemented the distributed model in our key management scheme. We will discuss it at Chapter 4. But before that, let us look at the existing secure routing protocols in the coming chapter.

## Chapter 3

# Secure Routing Protocols Review

In the literature, AODV, DSR and DSDV are three major routing protocols in MANET. The easiest way to secure routing is to add secure aspects on the existing successful protocols. To this end several secure routing protocols are presented, namely, ARAN, SAODV, Ariadne, SLSP and SEAD.

### 3.1 ARAN

A secure routing protocol for ad hoc networks (ARAN) [25] was proposed by Sanzgiri et al. It is based on an on demand protocol called AODV. In this section we firstly introduce AODV, then we will discuss ARAN.

**AODV** AODV [11] is an on demand routing protocol. As we discussed before, users of on demand routing protocols do not perform routing unless a request is received or generated. Moreover, AODV uses three types of control messages: Route Request (RREQ), Route Reply (RREP) and Route Error

(RERR) to control the whole network.

In order to discover a destination node (DN), firstly the source node (SN) will broadcast a RREQ message. A sequence number is given to each node which has received the RREQ message. On receiving this message, the intermediate node will rebroadcast it until it gets to the DN. When this RREQ message finds its way to the DN, a RREP message is generated, sending back to the SN via exactly the same route as the RREQ came from, and thus a connection is established.

In addition, every route will be assigned with a lifetime. Once a message is transferred via this route, the lifetime is refreshed. When the lifetime expires, the route becomes invalid.

In this case, once a route is broken, the nearest node will generate a RERR packet back to the SN. Then the SN will check if there is any alternative route to the DN storing in its cache. If not, it will invoke route discovery again.

**ARAN** In ARAN, each node must obtain a valid certificate along with its public key from a trust party prior to joining the network. The route request (RREQ) packets and the route reply (RREP) packets are functioning in Table 3.1.

If a source node (SN) wants to establish a route to a destination node (DN), it will broadcast a RREQ packet, as well as the address of the DN, the certificate of itself, a randomly picked number  $N$ , a time stamp  $t$  and a signature of the packet, signed with the public key of SN. The number  $N$  and the time stamp together ensure the freshness of the packet.

Any intermediate node (node IA, IB and IC in our example) other than the DN on receiving this packet will have to remove the signature and certificate of the former intermediate node (if there is any), replace it with its

Route	Data
SN broadcast	$(RREQ, DN, Cert_{SN}, N, t)pk_{sn}$
IA broadcast	$((RREQ, DN, Cert_{SN}, N, t)pk_{sn})pk_{IA}, Cert_{IA}$
IB broadcast	$((RREQ, DN, Cert_{SN}, N, t)pk_{sn})pk_{IB}, Cert_{IB}$
IC broadcast	$((RREQ, DN, Cert_{SN}, N, t)pk_{sn})pk_{IC}, Cert_{IC}$
DN to IC	$(RREP, SN, Cert_{DN}, N, t)pk_{DN}$
IC to IB	$((RREP, SN, Cert_{DN}, N, t)pk_{DN})pk_{IC}, Cert_{IC}$
IB to IA	$((RREP, SN, Cert_{DN}, N, t)pk_{DN})pk_{IB}, Cert_{IB}$
IA to SN	$((RREP, SN, Cert_{DN}, N, t)pk_{DN})pk_{IA}, Cert_{IA}$

Table 3.1: RREQ and RREP in ARAN

own signature, and its certificate is added as well.

Once the RREQ finds its way to the DN, the DN will reply with a RREP packet, down through the same route that RREQ comes from.

The ARAN has comparatively low efficiency, since it requires users from every hop to sign the routing packets. In the cryptography area, signing a packet is considered to be high cost, compared with hash functions. For a certain input data, the hash procedure requires only a hash algorithm while a sign procedure requires users keys and a much complex algorithm.

The authentication is another drawback. This is because the whole network depends on a trusted third part, which makes the trusted third party a single point of failure.

The network is also vulnerable against denial of service (DoS) attacks. Any malicious user with a valid certificate is capable of flooding the network with RREQ packets.



## 3.2 SAODV

The usage of certificates in ARAN decreases the efficiency of the network. To this end, Zapata and Asokan proposed another protocol based on AODV, which is named Secure AODV (SAODV) [19], to secure the routing information for MANET.

In their scheme, it is assumed there is a separate key management scheme distributing keys for the MANET users. It is also assumed that every user will be able to verify others' public keys and certificates. The major modification from ARAN is that in Zapata and Asokan's scheme a routing packet is firstly divided into two parts, namely the non-mutable fields and the mutable fields. Then a digital signature is used over the first part and a hash chain (may also be recognized as per-hop hashing in some articles) is used to process the second part. The main purpose is to process them separately and to increase efficiency, as the processing of a hash function is faster than signing a packet.

**Basic SAODV with Hash Chain** A SAODV packet contains the following attributes: a Max\_Hop\_Count field which is normally set to be the Time\_To\_Live value of an IP packet; a Hash\_Field to set the current value of hash function; a Hash\_Function value to indicate which type of hash function is in use. MD5 and SHA1 are two recommended functions. Moreover, SAODV allows users to define their own hash function.

Every time a RREQ or RREP packet is generated, the Source Node (SN) firstly selects a random number to be the Initial Value of hash function. Then it calculates a Top\_Hash value using the equation:

$$\text{Top\_Hash} = h^{\text{Max\_Hop\_Count}}(\text{Initial\_Value})$$

where  $h$  is a hash function and  $h^i(x)$  refers to the result of hashing  $x$  for

$i$  times. Once any intermediate node or the destination node (DN) gets this packet, it calculates the value of  $h^{Max\_Hop\_Count-Hop\_Count}(Hash\_Field)$ . If the result equals to  $Top\_hash$ , then the route of the packet is secured.

On one hand the Max\_Hop\_Count and the Hash\_Function are signed with the digital signature, along with the routing information, to ensure the integrity. They are signed by the SN, and no intermediate node should be able to modify them. On the other hand, the Hash\_Field is not signed. When a packet is transmitted, the value of Hash\_Field changes from hop to hop because it needs to be modified by every intermediate node. Thus, to sign them consumes lots of resource. To this end, those information is hashed so that no one can forge it whilst the process remains efficient.

**An Advanced Version of SAODV** In order to make routing more efficient, AODV allows intermediate nodes to reply RREQ messages if these nodes have a valid route to the DN. This causes problems when this feature is applied over SAODV: nodes other than DN cannot reply a RREQ even though they maintain valid routes to the DN. SAODV is a peer-to-peer secure protocol, where RREP messages need to be signed by DN, in other words, RREP can only be generated by the DN and other nodes cannot reply on behalf of the DN. To solve this problem, SAODV suggests two methods.

The first one is forbidding intermediate nodes from replying RREQ messages. Every intermediate node forwards the RREQ messages, no matter whether it has a valid route to DN or not. This method solves the problem, but sacrifices the efficiency.

The other method is adding a prefix to the RREP packets. The prefix is signed by the DN and stored in the intermediate nodes. When an intermediate node gets a RREP from a DN, it stores the RREP with the prefix.

If there is another RREQ packet searching for the same DN, this intermediate node can reply with the prefix on behalf of the DN. This prefix should have a life time which is signed by this intermediate node. The SN needs to check both the signature of the lifetime (from intermediate nodes) and the signature of the RREP (from the DN).

One of the advantages of SAODV is that it is a resource economical protocol compared with ARAN. One of the main characteristics of ARAN we discussed before is it requires every intermediate node to sign the routing packet. This is very energy consuming, compared with SAODV. By contrast, SAODV divides the routing messages into two parts, mutable and non-mutable, and secures them with different approaches, namely hash function and sign procedure respectively. The mutable fields are signed by hash chain functions. This mechanism ensures that the size of the routing packets stays constant. The non-mutable fields are signed by digital signatures. Only the signatures of the source and the destination nodes are required, which is also an improvement from ARAN.

### 3.3 Ariadne

The per-hop hashing mechanism in SAODV is a great improvement for secure routing. Nevertheless, the authentication issue remains problematic. In [2], Hu and Perrig proposed a new model to strengthen the authentication during the routing. A secure on demand routing protocol for ad hoc network (Ariadne) they proposed is an efficient secure routing protocol based on DSR using symmetric key cryptography.

**DSR** Dynamic Source Routing (DSR) protocol [13] is a reactive routing protocol which makes use of caches and nodes' full addresses to provide

reliable routes. Similar to AODV, DSR allows users broadcast the route request packet to search the DN. The difference lies on the fact that DSR uses nodes full address to distinguish users, while AODV uses sequence numbers to represent users. On receiving any routing requests, users should rebroadcast the packets, appending with their own addresses, as long as they are not the DN. Furthermore, DSR is a cache enabled protocol, where users store former routes in their cache for further use. This is a special feature that AODV cannot perform, because every routing request packet in DSR contains full addresses of previous nodes, every user will be able to update the knowledge of the network and store the network topology in its cache.

**Ariadne** In Ariadne, it is assumed every two nodes, namely node A and node B, are capable of obtaining two keys,  $K_B^A$  and  $K_A^B$ , each key for a direction of communication. In their paper, Hu and Perrig [2] adopted mechanism of SAODV by separating the mutable and the non-mutable parts. They use per-hop hashing to secure the hop count field, and another three methods to provide packet integrity: TESLA, digital signatures and message authentication codes (MAC). In this thesis, we will primarily discuss TESLA for an example.

**Per-hop hashing** Per-hop hashing is used to prevent any attacker from modifying or removing any hop from the route. It works the same way as we mentioned in SAODV. If any hop is removed or redirected from the route, the hash chain will be broken.

**TELSA** TELSAs is an efficient broadcast authentication scheme used in multi-hop communications. In TELSAs, each node picks a random number  $K_0$  as its initial key. These keys are exchanged by certain symmetric key

management schemes. Then the nodes generate their session key on a one way trapdoor hash function  $h$  and a time value  $t$ .

$$K_N = h(K_{N-1}) = h^N(K_0)$$

$$N = (current\_time - initial\_time)/t$$

Then the user publishes its keys in a reverse way:  $K_N, K_{N-1}, K_{N-2}$ . When another user intends to send a route request to this user, it has to examine the current key corresponding to the current time. In this way, only one MAC is required for a route request packet. When the DN gets this message, it also examines the key in the message with the former key stored in its memory in a reverse way. Because the hash function is a one way trapdoor function, it is impossible to anticipate the future key for any user other than SN.

$$K_{former} = h^{(former\_time - current\_time)/t}(K_{current})$$

While TELSA is enabled, each intermediate user authenticates new information in the route request packet with its current key. The DN will not send any route reply until all the intermediate nodes' keys are verified. Then the DN will generate a route reply packet with a list of intermediate users' keys.

### 3.4 SLSP

Other than the protocols we discussed before, secure link state protocol (SLSP) for mobile ad hoc network [20] is a protocol based on the proactive routing protocol.

In their paper, Papadimitratos and Haas assumed that each user is guaranteed with a valid public/private key pairs. The certificates binding the public keys, the IP addresses and the medium access control addresses are also pre-established for users.

**Neighbour Discovery** Neighbours are discovered by periodically broadcasting hello messages. An entry to the routing table will be set up if a neighbour is found and verified. The routing table is maintained by an integral part of SLSP named neighbour lookup protocol (NLP). It mainly has three responsibilities.

1. Maintaining routing tables with neighbours' IP and MAC addresses;
2. Discovering malicious users with duplicated IP or MAC addresses;
3. Controlling the traffic rate sending to/receiving from neighbour users.

**Neighbour maintenance** After building up the routing table, users will periodically broadcast their link states using link state updates (LSU) packets. The hash chain mechanism is adopted to control the propagation range of the LSU packets, so that only specific neighbour users will receive these packets.

**Flooding control** SLSP is a lightweight flooding prevention protocol. The NLP assigns higher priority to the users who produce less traffic over the heavy producers. This mechanism provides certain level of resistance against DoS attacks.

### 3.5 SEAD

The protocols we discussed so far are all based on either symmetric or asymmetric cryptography. These cryptography approaches are considered to be low efficient, compared with a hash function. To this end, Hu, Johnson and Perrig [8] proposed secure efficient distance vector routing protocol for mobile wireless ad hoc network (SEAD), which only use efficient one-way hash functions and does not use any asymmetric operations.

**DSDV** The SEAD protocol is partially based on destination-sequence distance-vector (DSDV) routing protocol [6]. In the distance-vector routing, each node maintains a routing table with entries to other nodes. The distance to other nodes are measured by a distance vector (usually is the hop count between these two users), which is known as the metric in that table entry. The lower value of distance vector for a route generally means a smaller transmission cost. Those routes have higher priorities to be picked, thus the routing becomes more efficient. DSDV adopts this mechanism and makes some modification by adding a sequence number in each routing table entry. The usage of the sequence number effectively prevents the routing loops. When implemented to MANET, DSDV sends periodic routing update packet.

**Metric and Sequence Number Authentication in SEAD** In SEAD, an efficient one way hash function is used to provide authentication. Each node in SEAD uses a specific single next element from its hash chain in the routing update packet that it sends to itself (metric 0). This mechanism provides a lower bound authentication, since the nodes can only use the next element in the hash chain, which means nodes can increase the metric, but cannot decrease it. The authentication of the entry for a specific sequence

number in the routing table uses same hash chain. Assume  $m - 1$  to be the upper bound of the network diameter, also assume the hash chain is  $h^0, h^1, h^2 \dots h^n$ , where  $n$  is divisible by  $m$ . For a sequence number  $i$ , denote  $k = n/m - i$ , a group of hash values  $h^{km}, h^{km+1}, h^{km+2} \dots h^{km+m-1}$  is used to authenticate the routing update packets for that sequence number.

**Neighbor Authentication in SEAD** The authentication with the hash chain over the metric and the sequence number is based on the assumption that all the nodes are acting positively (with no malicious attackers or selfish users). Although the hash chain effectively prevents the malicious node modifying routing update packets from other nodes, it does not forbid the malicious nodes sending faulty packets about themselves. Thus, to determine a neighbour node remains a problem. This issue is called neighbor authentication in [8]. Hu, Johnson and Perrig presented several approaches to solve this problem.

One of the approaches is TELSA. It requires synchronized clock thus may bring either an authentication delay or a relatively high communication overhead.

Another approach is using message authentication code. The secret key is involved in this approach. This is beyond the purpose of designing SEAD, where Hu, Johnson and Perrig intended to present a highly efficient routing protocol without any asymmetric key cryptography. Nevertheless, the message authentication code scheme does not entirely solve the problem. If the message authentication code is not enabled, the problem is neighbour node authentication; while if it is enabled, the problem becomes the distribution of the public keys and the certificates.



## 3.6 Conclusion

In this chapter we review five existing secure routing protocols. Interestingly, we noticed that most of them are based on re-active routing protocols, while secure routing protocols that are based on existing proactive routing protocols, for instance, OLSR, are rare. To this end, we suggest that proactive secure routing protocols could also be efficient. We present our own secure routing protocol in Chapter 5.

In the next chapter we will present our key management scheme.

## Chapter 4

# Certificateless Cryptography over MANET

In this chapter, we consider a different approach to the existing solutions, namely to incorporate the certificateless cryptography into MANET. As we shall show in this chapter, the adoption of certificateless cryptography to the MANET scenario is not very straightforward. Nonetheless, by combining the secret sharing schemes with the certificateless cryptography, we obtain an efficient and secure MANET scheme. Our contribution is to apply the existing certificateless cryptography into MANET using a threshold secret sharing scheme. We firstly create a generic model based on the above ideas and then we propose our scheme that comprises of a combination of certificateless cryptography and secret sharing scheme. To support our idea, we implement our schemes in OPNET to analyze its efficiency and practicality.

## 4.1 Certificateless Cryptography

In 2003, Al-Riyami and Paterson [27] proposed a new system known as certificateless cryptography. The idea of certificateless cryptography is to gather the strength of both the public key cryptography and ID-based cryptography and to avoid the drawbacks that these two systems have. In this system, there is a trusted authority called the Key Generation Centre (KGC) that will need to generate a partial secret key for the users, given the users' identity. Nonetheless, each user also needs to generate his/her own partial secret key and based on these two pieces of information (partial secret keys), the user can generate the public key that needs to be published. Although this system incorporates a public key, this public key does not need to be certified as this public key has been 'implicitly' certified by the partial secret key issued by the KGC. Hence, to verify the authenticity of the public key, the KGC's public key needs to be involved. We note that there is no key escrow problem in this model as the KGC does not know the user's secret key. The KGC can only know the partial secret key but not the complete secret key as some part of the secret key is generated by the user himself/herself.

## 4.2 Generic Model

We assume that at the beginning of the network there is a Key Generator Center (KGC) which generates partial secret keys for all the users. We also denote  $n$  to be the number of original nodes and  $t$  to be the pattern of security level of the threshold system. Those  $n$  nodes collectively form a Distributed Key Generator Center (DKGC). After the initiation, the KGC will go offline, and the network becomes self-organized. We define those nodes that get partial secret keys from the KGC to be the original nodes,

those nodes that get partial secret keys from DKGC to be the new-joint nodes and those nodes that collectively form the DKGC to be DKGC nodes.

- **Setup:**

This algorithm takes as input a security parameter  $1^k$  and returns the master private key  $msk$  and master public key  $mpk$ . This algorithm is run by the KGC, in order to setup a certificateless ad hoc system.

- **Extract-partial-secret-key:**

This algorithm takes as input the master public key  $mpk$ , the master private key  $msk$  and an identity  $ID=i \in \{0,1\}^*$ . It outputs a partial private key  $d_i$ . This algorithm runs by KGC once at the initiation of the network.

- **Extract-master-secret-key-shares:**

This algorithm takes as input the master private key  $msk$  and an identity  $ID=i \in \{0,1\}^*$ . It outputs a master secret key shares  $msks_i$ . This algorithm runs by KGC once at the initiation of the network.

- **Extract-partial-secret-key-share-and-master-secret-key-share:**

This algorithm takes as input the master public key  $mpk$ , the master private key share  $msks_i$  from a DKGC node and an identity  $new$  of a new-jointly node. It outputs a share of partial user private key  $ds_{new,i}$  and a share of master secret key share  $msks_{new,i}$ ,  $i \in \{0,1\dots n\}$ . This algorithm runs by DKGC nodes.

- **Extract-master-secret-key-shares-DKGC:**

This algorithm takes as input the master public key  $mpk$ , an identity  $ID=new \in \{0,1\}^*$ , and  $t$  shares of master private key share  $msks_{new,i}$ ,  $i \in \{0,1\dots n\}$ . It outputs a master secret key share  $msks_{new}$ . This algorithm runs by the new-joint node.

- **Extract-partial-secret-key-DKGC:**

This algorithm takes as input the master public key  $mpk$ , a user identity  $ID=new$  and  $t$  shares of partial user private key  $d_{new,i}, i \in \{0, 1, \dots\}$ . It outputs a user partial secret key  $d_{new}$ . This algorithm runs by the new-joint node.

- **Set-user-keys:**

This algorithm takes as input the master public key  $mpk$ , a user identity  $ID=i$ , a partial private key  $d_i$  and a secret value  $x_i$ . It outputs a user public/private key pair  $(pk_i/sk_i)$  or an error symbol. This algorithm runs by all the nodes.

- **Encryption:**

This algorithm takes as input the master public key  $msk$ , a user's identity  $ID=i$ , a user's public key  $pk_i$  and a message  $msg$ . It outputs a cipher text  $c$ .

- **Decryption:**

This algorithm takes as input the master public key  $msk$ , a user's private key  $sk_i$  and a cipher text  $c$ . It outputs a message  $msg$ .

### 4.2.1 Fully Distributed System

In the fully distributed system, all the nodes will have a share of  $msk$ . They together maintain the stability of the system. At the initiation stage, the KGC generates a master public/private key pair  $(mpk/msk)$  using **Setup** algorithm. It then generates user partial keys using **Extract-partial-secret-key** algorithm and divides  $msk$  with **Extract-master-secret-key-shares**. The user partial keys  $d_{ID}$  and master secret key shares  $msks_{ID}$  are distributed to all the origin nodes. Once this is done, the KGC goes offline, and all

the original nodes become DKGC nodes.

We use threshold cryptography to provide authentication for new nodes. A new-joint node needs to successfully contact at least  $t$  DKGC nodes. Those DKGC nodes will run **Extract-partial-secret-key-share-and-master-secret-key-share** algorithm for the new-joint node. Once this new-joint node obtains  $t$  shares of  $msks_{new,i}$  and  $t$  shares of  $ds_{new,i}$ , it will be able to derive a master secret key share  $msks_{new}$  and a partial secret key  $d_{new}$  by **Extract-master-secret-key-shares-DKGC** and **Extract-partial-secret-key-DKGC** respectively, and it becomes a DKGC node. The number of DKGC nodes rises with the increase of node numbers.

DKGC nodes use **Set-user-keys** algorithm to calculate their own public/private keys. The public keys will be broadcast all through the network so that nodes can communicate to each other with **Encryption** and **Decryption** algorithms.

### 4.2.2 Partially Distributed System

In a partially distributed system, a certain number of nodes will become DKGC nodes. The  $msk$  is only shared between these nodes. They are responsible for issuing partial secret key for new coming nodes. This system differs from fully distribution system that :

1. For a new-joint node, the DKGC nodes only issue partial secret key shares  $ds_{new,i}$ , without any master secret key shares  $msks_{new,i}$ .
2. Once a DKGC node goes offline, a random non-DKGC node will be picked. Other DKGC nodes will give this node master secret key shares  $msks_{new,i}$ , so that this chosen one will become a new DKGC node. In this model, the number of DKGC nodes does not increase.

In our model we pick all the initiation nodes to be the DKGC nodes. The relationship among the number of DKGC nodes, the total number of nodes and threshold of the system will be further discussed in Section 4.4.

### 4.3 Detailed Scheme

The first certificateless public key encryption scheme was proposed by Al-Riyami and Paterson. We incorporate their work and adopt it to MANET key management with CL-PKE. The scheme is as follows:

- **Setup:**

We assume  $IG$  is a Bilinear Diffie-Hellman parameter generator and  $k$  is the security parameter for the system. This algorithm has four steps.

1. Run the  $IG$  generator on an input  $k$ , it outputs  $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$  where  $\mathbb{G}_1$  and  $\mathbb{G}_2$  are groups of prime order  $q$ .  $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$  is a pairing.
2. Choose an arbitrary generator  $\mathbb{P} \in \mathbb{G}_1$ .
3. Select a master private key  $msk$  uniformly at random from  $Z_q^*$  and set  $\mathbb{P}_0 = msk \times \mathbb{P}$ .
4. Choose four cryptographic hash functions  $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}_1$ ,  $H_2 : \mathbb{G}_2 \rightarrow \{0, 1\}$ ,  $H_3 : \{0, 1\}^m \times \{0, 1\}^m \rightarrow Z_q^*$  and  $H_4 : \{0, 1\}^m \rightarrow \{0, 1\}^m$ , here  $m$  will be the bit-length of plaintexts.

The master public key  $mpk = \langle \mathbb{G}_1, \mathbb{G}_2, e, m, \mathbb{P}, \mathbb{P}_0, H_1, H_2, H_3, H_4 \rangle$ . The master private key is  $msk \in Z_q^*$ . The message space is  $M = \{0, 1\}^m$  and the ciphertext space  $C = \{0, 1\}^{2m} \times \mathbb{G}_1$ .

- **Extract-partial-secret-key:**

This algorithm takes as input an  $ID \in \{0,1\}^*$  and carries out the following steps.

1. Compute  $\mathbb{Q}_{ID} = H_1(ID) \in \mathbb{G}_1$ .
2. Output the partial private key  $d_{ID} = msk \times \mathbb{Q}_{ID} \in \mathbb{G}_1^*$ .

Any user can verify its partial secret key by checking  $e(d_{ID}, \mathbb{P}) = e(\mathbb{Q}_{ID}, \mathbb{P}_0)$ .

- **Extract-master-secret-key-shares:**

We assume a polynomial  $f(x)$  can be defined as

$$f(x) = msk + \sum_{i=1}^t (a_i x^i)$$

Where  $a_1, a_2 \dots a_t$  are uniformly distributed over a finite field  $F$ . This algorithm takes as input an  $ID \in \{0,1\}^*$  and outputs a master secret key share  $msks_i = f(ID_i)$ . From this formula we can compute  $msk$  by

$$f(0) = msk = \sum_{i=1}^{t+1} [\hat{L}(0, ID_i) \times f(ID_i)] \in Z_q^*$$

and we also have

$$f(x) = \sum_{i=1}^{t+1} [\hat{L}(x, ID_i) \times f(ID_i)]$$

where  $\hat{L}(\alpha, \beta)$  is the appropriate Lagrangian coefficients. Assuming  $S = \{ID_1, ID_2, ID_3 \dots ID_{t+1}\}$ , then

$$\hat{L}(\alpha, \beta) = \frac{\prod_{\gamma \in S, \gamma \neq \beta} (\alpha - \gamma)}{\prod_{\gamma \in S, \gamma \neq \beta} (\beta - \gamma)}$$

- **Extract-partial-secret-key-share-and-master-secret-key-share:**

Giving a master secret key share of node  $i$   $msks_i$  and a new-joint node's  $ID = new$ , this algorithm takes the following steps.



1. A partial secret key share is calculated by

$$\mathsf{ds}_{new,i} = \hat{L}(0, ID_i) \times msk s_i \times \mathbb{Q}_{new} = \hat{L}(0, ID_i) \times f(ID_i) \times \mathbb{Q}_{new} \in \mathbb{G}_1$$

2. A master secret key share is calculated by

$$msk s_{new,i} = \hat{L}(ID_{new}, ID_i) \times msk s_i \in Z_q^*$$

- **Extract-partial-secret-key-DKGC:**

This algorithm takes as input  $t$  partial secret key shares  $\mathsf{ds}_{new,i}$ , the partial secret key  $\mathsf{d}_{new}$  can be calculated by

$$\mathsf{d}_{new} = \sum_{i=1}^{t+1} \mathsf{ds}_{new,i} = \sum_{i=1}^{t+1} \hat{L}(0, ID_i) \times f(ID_i) \times \mathbb{Q}_{new} = msk \times \mathbb{Q}_{new} \in \mathbb{G}_1$$

- **Extract-master-secret-key-shares:**

This algorithm takes as input  $t$  master secret shares  $msk s_{new,i}$  and the  $msk_{new}$  can be calculated by

$$msk_{new} = \sum_{i=1}^{t+1} msk s_{new,i} = \sum_{i=1}^{t+1} \hat{L}(ID_{new}, ID_i) \times msk s_i = f(ID_{new}) \in Z_q^*$$

- **Set-user-keys:**

This algorithm takes as select a user's secret value  $x_{ID} \in Z_q^*$ , input the master public key  $mpk$  and user's partial secret key  $\mathsf{d}_{ID}$ . It outputs user's secret key  $\mathsf{sk}_{ID} = x_{ID} \times \mathsf{d}_{ID}$  and user's public key  $pk_{ID} = \langle \mathbb{X}_{ID}, \mathbb{Y}_{ID} \rangle$ , where  $\mathbb{X}_{ID} = x_{ID} \mathbb{P}$  and  $\mathbb{Y}_{ID} = x_{ID} msk \mathbb{P}$ .

- **Encryption:**

For a message  $msg \in M$  and an identity  $ID \in \{0, 1\}^*$  with its public key  $pk_{ID} = \langle \mathbb{X}_{ID}, \mathbb{Y}_{ID} \rangle$ , the encryption algorithm takes as follows:

1. Check the public key by  $e(\mathbb{X}_{ID}, \mathbb{P}_0) = e(\mathbb{Y}_{ID}, \mathbb{P})$ . If the result is negative, abort the encryption and output an error symbol.

2. Compute  $\mathbb{Q}_{ID} = H_1(ID) \in \mathbb{G}_1^*$ .
3. Choose a random number  $\sigma \in \{0, 1\}^m$ .
4. Set  $r = H_3(\sigma, msg)$
5. Compute and output ciphertext:

$$c = \langle r\mathbb{P}, \sigma \oplus H_2(e(\mathbb{Q}_{ID}, \mathbb{Y}_{ID})^r), \\ msg \oplus H_4(\sigma) \rangle$$

• **Decryption:**

Suppose  $c = \langle \mathbb{U}, V, W \rangle \in C$ . To decrypt this cipher text with private key  $sk_{ID}$ :

1. Compute  $V \oplus H_2(e(sk_{ID}, \mathbb{U})) = \sigma'$ .
2. Compute  $W \oplus H_4(\sigma') = msg'$ .
3. Set  $r' = H_3(\sigma', msg')$  and test if  $\mathbb{U} = r'\mathbb{P}$ . If not, output an error symbol and reject the ciphertext.
4. Output  $msg'$  as the decryption of  $c$ .

• **Correctness:**

$$\begin{aligned} \sigma' &= V \oplus H_2(e(sk_{ID}, \mathbb{U})) \\ &= V \oplus H_2(e(x_{ID}d_{ID}, r\mathbb{P})) \\ &= V \oplus H_2(e(x_{ID}msk\mathbb{Q}_{ID}, \mathbb{P})^r) \\ &= V \oplus H_2(e(\mathbb{Q}_{ID}, x_{ID}msk\mathbb{P})^r) \\ &= V \oplus H_2(e(\mathbb{Q}_{ID}, \mathbb{Y}_{ID})^r) \\ &= \sigma \\ msg' &= W \oplus H_4(\sigma') \end{aligned}$$

$$\begin{aligned}
&= \text{msg} \oplus H_4(\sigma) \oplus H_4(\sigma') \\
&= \text{msg} \\
r'\mathbb{P} &= H_3(\sigma', \text{msg}')\mathbb{P} \\
&= H_3(\sigma, \text{msg})\mathbb{P} \\
&= \mathbb{U}
\end{aligned}$$

## 4.4 Issues and Design Principles

We incorporate a distributed system to replace the KGC, so that the network becomes self-organized. This fully distributed system is based on the threshold cryptography with two patterns  $(t, n)$ . The pattern  $t$  represents the threshold of the model, which means any  $t+1$  malicious users can break the system (hence, the system is upperbounded by  $t+1$ , which means that as long as there are at most  $t$  malicious users, then the system is considered to be at the ‘secure’ state). The pattern  $n$  represents the total number of users. We denote  $n'$  to be the maximum number of users, and  $t'$  to be the number of malicious users in the network at the initiation state.  $t'$  should be less than  $t$  to get the network initiated.

Unfortunately, we cannot anticipate if a new-joint node is malicious or not. If the system is based on fully distributed model, then in the worst case, all the new-joint nodes are malicious, which add up to  $n'-n+t'$  malicious DKGC nodes. In order to keep the system running well, this  $n'-n+t'$  should be smaller than  $t$ . The system becomes vulnerable when  $t-t'$  nodes join the network.

If the system is based on the partially distributed model, every DKGC sends its data to a random non DKGC node before it goes offline. When  $t-t'$  original nodes goes offline, and they all replicate themselves to new-joint

node, the system becomes vulnerable.

Fully distributed systems are more efficient, but only allow a small number of new-joint nodes. Partially distributed system can be secure as long as a certain amount of origin nodes stay online, but it requires cooperation between DKGC nodes and new-joint nodes, and it brings along with extra communication overhead searching for DKGC nodes. Different systems should be chosen over different scenarios.

## 4.5 Conclusion

This chapter presented the design of a key distribution scheme over mobile ad hoc network, based on the certificateless cryptography and threshold secret sharing scheme. In this work, we have successfully issued public/secret keys for users without providing certificates. Our scheme also ensures that system can work on self-organized networks after the initiation. The simulations of our scheme will be discussed in Chapter 7. In the next chapter we will present our secure routing protocol.

# Chapter 5

## Secure Routing Protocol

### 5.1 OLSR

#### 5.1.1 Basic Mechanism of OLSR

Optimized Link State Routing (OLSR) [5] protocol is one of the most popular proactive routing protocols in ad-hoc networks. Most of the proactive routing protocols pre-established routes before communication occurs in a way that periodically broadcasting beacons, called "Hello" message, to examine the network topology. However, OLSR differs from other proactive routing protocols as follows:

Firstly, a "Hello" message is used to obtain information from all 2-hop neighbours. Then a distributed selection of multi-point relay (MPR) nodes is performed to achieve optimized link state routing. The MPR strategy is the major difference between OLSR and other proactive routing protocols. The MPR nodes are a sub-group of source node's 1-hop neighbours. They are carefully selected that a source node can connect to all its 2-hop neighbours through those MPR nodes without any 1-hop nodes other than MPR nodes.

Finally, a Topology Control (TC) message is broadcasted by the source node to share the topology between other nodes. It contains topology information of its source node. Each node will be able to find an optimized path to other node after collecting others' TC messages. The transmission of TC message is a flooding process. In order to reduce the control overhead, TC messages will only be transmitted over MPR nodes. After gathering enough information through the TC messages, nodes are able to send packets to other nodes.

### 5.1.2 Remaining Problems of OLSR

One of the major contributions of OLSR is its MPR mechanism which optimizes link state so that nodes can perform routing with least cost. However, the selection of MPR nodes is based on the coverage. Unfortunately, in real scenarios, best coverage does not guarantee best performance. For instance, OLSR have not presented any mechanism to measure the link quality, nor to ensure that a misbehaving node has not been selected.

We believe that the MPR nodes should be selected based not only on coverage but also availability. We think that to select a node that is worthy of trust is more important than other aspects. To this end, we adopted a reputation system into the original OLSR protocol.

## 5.2 Reputation System

In current literature, there exists several reputation systems. In [21], Damiani et al. presented one of the first reputation system, XRep. It is designed for Gnutella, one of the first peer-to-peer file sharing protocols. It uses an encrypted Polling message to collect votes from others, and determines the quality of resources base on those votes. With this mechanism, XRep man-

aged to reduce the number of malicious intruders and low quality resource distributors. However, according to Curtis, Safavi-Naini and Susilo [22], the XRep failed to address the problem of whitewashing (see section 6.5 for more details), which turned out to be one of the major drawback of most reputation systems.

In 2004, Curtis, Safavi-Naini and Susilo solved this problem by presenting a new protocol named  $X^2Rep$  [22]. They claimed that heavy punishment on attackers can efficiently stop the whitewashing attack, hence making the system robust. Although  $X^2Rep$  was designed for Gnutella, thus might not be applicable for MANET, we believe that the same penalty mechanism can be successfully intergraded into the reputation system on the MANET.

With the development of peer to peer file sharing technologies, comes with new reputation system. In 2006, Yu, Susilo and Safavi-Naini [30] introduced a reputation system specially designed for Bit Torrent,  $X^{2BT}$  Trusted Reputation System. To make the system unique for Bit Torrent,  $X^{2BT}Rep$  suggested an introduce approach, where one user can introduce friends and files to the polling server. The opinion of a user towards any other users or files is valued by a number. By collecting several votes, the polling server will rate the files, and users will be able to tell which one to download.

Unfortunately, as the same reason with  $X^2Rep$ ,  $X^{2BT}Rep$  does not fit MANET. It requires a polling center to collect votes, which MANET cannot provide.

## 5.3 Reputed-OLSR

### 5.3.1 Assumptions

To simplify the design of the protocol, we assume that each node is guaranteed with an ID, a public key and a secret key, while the public key is blinded with the ID, meanwhile each node will be able to validate other's ID and public key with a certain method (certificate/certificateless/ID-based). We also assume the wireless channels are bidirectional, which means if node A is in the propagation range of node B then node B is in the propagation range of node A as well.

We denote "[ ]" to be the sign procedure while "{ }" to be the encryption procedure, " $R_B^A$ " to be B's reputation given by A, while  $f()$  to be a function to calculate the reputation.

In our protocol, each node maintains three tables:

- A node table which contains 1-hop and 2-hop neighbours, MPR information, and the entry to 2-hop neighbours;
- A topology table which contains routing information of all other nodes;
- A reputation table which contains reputations for both 1-hop neighbours and routes;

### 5.3.2 Generic Model

In this section we propose our scheme that takes 5 steps to perform routing.

**Step one, Neighbour Discovery** During this step, every source node (SN) sends out hello messages to their 1-hop nodes. Then those 1-hop nodes reply with list of SNs' 2-hop neighbours. This step ensures the SN gets a



correct topology of its neighbours. Every packet in this stage needs to be signed with corresponded user.

**Step two, Polling** In this step, SN sends out polling requests to 2-hop neighbours via 1-hop neighbours. 2-hop neighbours then reply with their opinion on each 1-hop neighbours. Polling requests need to be signed by SN while polling replies need to be signed by 2-hop neighbours and encrypted with SN's public key.

**Step three, Reputation Calculation** This step takes as input opinions collected from 2-hop neighbours and output the reputation of certain 1-hop neighbour. Based on those reputations, SN select certain nodes as MPR nodes.

**Step four, TC sharing** In this step, every SN floods its Topology Control (TC) messages. Upon receiving those messages, every node updates their network knowledge. The TC messages should be signed by SN and every intermediate user.

**Step five, Communication** In the final step, users start to communicate. Hash functions and digital signatures are used to provide integrity.

### 5.3.3 Detailed Protocol

In this part, we carry out our scheme in a simple Mobile Ad-hoc Network which consists of 10 nodes that are showing below. As shown in Figure 5.1, node A is crowded by 4 1-hop neighbours and 5 2-hop neighbours, and it tries to establish route to those neighbours by following steps.

#### **Step one, Neighbour Discovery**

As highlighted in Figure 5.2, the first step involves 2 procedures. Firstly, a source node *A* broadcast Hello messages to all 1-hop neighbours. Then, upon receiving those messages, node *B, C, D* and *E* reply with HelloReply

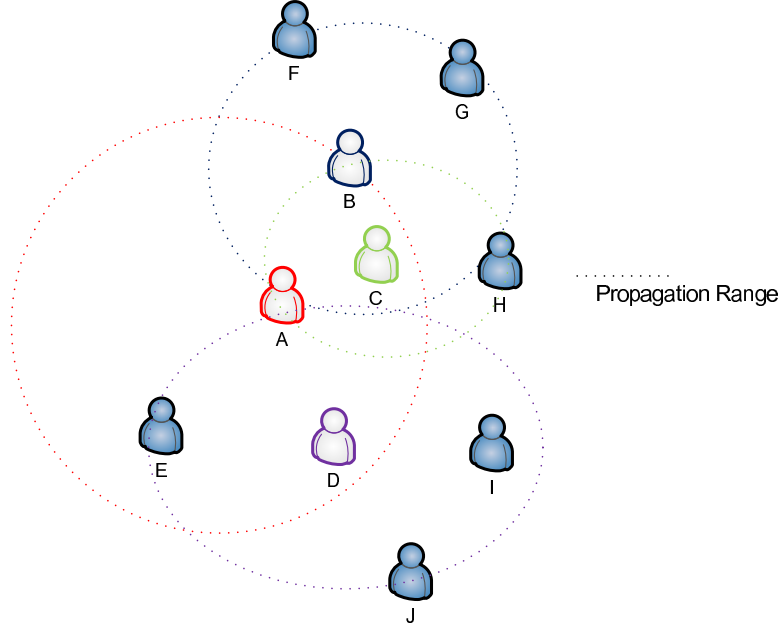


Figure 5.1: Network Setup

messages. The HelloReply should contain information about each individual's 1-hop neighbour nodes so that the source node  $A$  can expand its 2-hop neighbour knowledge.

The detailed transmission are shown as follows:

$$A \longrightarrow B, C, D, E : [Hello]_{ska}$$

$$B \longrightarrow A : [HelloReply : C, F, G, H]_{skb}$$

$$C \longrightarrow A : [HelloReply : B, H]_{skc}$$

$$D \longrightarrow A : [HelloReply : E, H, I]_{skd}$$

$$E \longrightarrow A : [HelloReply : D]_{ske}$$

Based on the HelloReply messages, node  $A$  builds a MPR reputation table, Table 5.1. Node  $F, G, H$  and  $I$  are 2-hop neighbours, so they will have initial reputations given by  $A$ . Meanwhile, node  $E$  has no connection to 2-hop neighbours, thus its reputation is the initial reputations given by  $A$  as

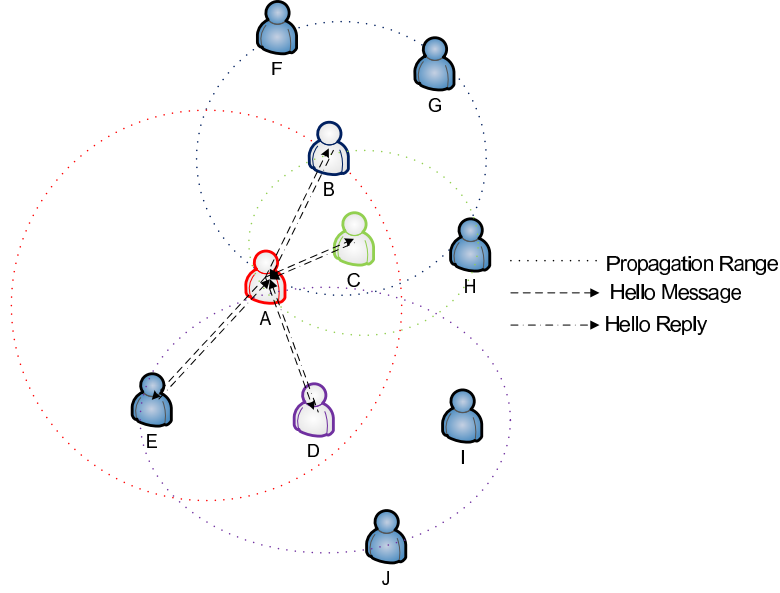


Figure 5.2: Neighbour Discovery

well and will not be selected as MPR node. On the contrary, node  $B$ ,  $C$  and  $D$  has multiple routes to other nodes, so their reputations are not set yet. The source node  $A$  sends out Polling message (Figure 5.3) to collect their reputations.

### Step two, Polling

The Polling messages are sent to 2-hop neighbours, in terms of  $F, G, H$  and  $I$ . Those messages are signed with  $A$ 's secret key so that intermediate nodes, for instance, node  $B$ , cannot forge them. The 2-hop neighbours then reply with Polling reply message. The reply contains their opinion on certain nodes thus should be encrypted with  $A$ 's public keys so that only node  $A$  can read.

$$A \longrightarrow F, G, H : [PollRequest : R_B^F, R_B^G, R_B^H]_{ska}$$

$$A \longrightarrow H : [PollRequest : R_C^H]_{ska}$$

$$A \longrightarrow H, I : [PollRequest : R_D^H, R_D^I]_{ska}$$

Node	Entry	MPR	Reputation
B	A	Not Set	Not Set
C	A	Not Set	Not Set
D	A	Not Set	Not Set
E	A	Not Set	$R_E^A$
F	B	N/A	$R_F^A$
G	B	N/A	$R_G^A$
H	B,C	N/A	$R_H^A$
I	D	N/A	$R_I^A$

Table 5.1: MRP Reputation Table, Stage 1

$$F, G \longrightarrow A : \{[PollReply : R_B^F]_{skf}\}_{pka}, \{[PollReply : R_B^G]_{skg}\}_{pka}$$

$$H \longrightarrow A : \{[PollReply : R_B^H, R_C^H, R_D^H]_{skh}\}_{pka}$$

$$I \longrightarrow A : \{[PollReply : R_D^I]_{ski}\}_{pka}$$

**Step three, Reputation Calculation** Based on the replies, node  $A$  will calculate reputations on 1-hop neighbours. The reputation table is correspondingly updated as follows:

$$R_B^A = f(R_B^A, R_B^F, R_B^G, R_B^H)$$

$$R_C^A = f(R_C^A, R_C^H)$$

$$R_D^A = f(R_D^A, R_D^H, R_D^I)$$

We suggest that the  $f()$  function is defined as follows:

$$R_j^i = f(R_0^i, R_1^i, \dots, R_n^i) = \text{Min}(1, R_j^i + \Delta * \sum_{n=0}^{k=0} R_k^i / n)$$

In our simulations we assume that  $\Delta = 0.01$  so that the system is robust against several attacks. When this reputation system is implemented, users can freely choose their own  $\Delta$  to increase the efficiency, (for instance, in a small network with 10 users, the reputation does not go up as fast as in a

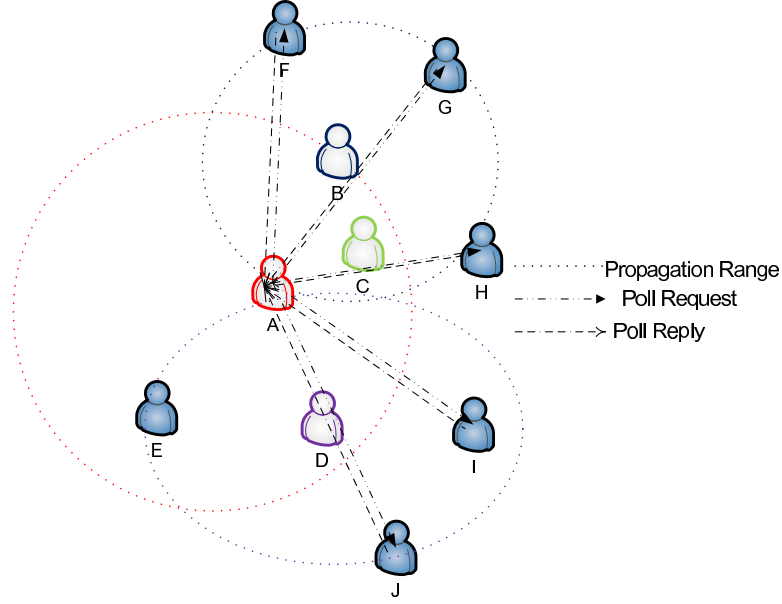


Figure 5.3: Polling

large network because less votes are collected, in this case we can increase the  $\Delta$  to accelerate the incensement.) as long as this function ensures that:

1. the more feedbacks (more valid connection between candidate and its neighbors) the higher reputation;
2. the higher value of the each feedback, the higher the reputation;
3. if the reputations for two routes are the same, then a node with more availability is chosen. This mechanism can avoid the system from a cold start at the initiation stage.

In the MPR reputation table, the reputations for  $B$   $C$  and  $D$  are now set (Table 5.2).

#### Step four, TC sharing

The source node selects MPR nodes. In our case because the initiate reputation for every node is the same (0.5), so node  $B$  and  $D$  are chosen

Node	Entry	MPR	Reputation
B	A	Not Set	$f(R_B^A, R_B^F, R_B^G, R_B^H)$
C	A	Not Set	$f(R_C^A, R_C^H)$
D	A	Not Set	$f(R_D^A, R_D^H, R_D^I)$
E	A	No	$R_E^A$
F	B	N/A	$R_F^A$
G	B	N/A	$R_G^A$
H	B,C,D	N/A	$R_H^A$
I	D	N/A	$R_I^A$

Table 5.2: MRP Reputation Table, Stage 2

because of better coverage and availability. Node  $A$ 's reputation table is then updated as Table 5.3.

The source node  $A$  then broadcasts the Topology Control (TC) message to 1-hop neighbours, namely  $B, C, D$  and  $E$ . As illustrated in Figure 5.4, only  $B$  and  $D$  who are MPR nodes will forward the TC messages. Upon receiving a TC message, all nodes will update their topology table.

#### Step five, Communication

With a knowledge of the network topology, nodes are able to send any message to other nodes in the network. Routes are selected through best reputation nodes. During the transmission, if a node successfully forwards a packet, its reputation will rise, otherwise it will fall sharply. The reputation range is  $(0, 1]$ .

if (SUCCESSFUL\_TRANSMIT)

$$R_j^i = \text{Min}(1, R_j^i + \Delta * \sum_n^{k=0} R_k^i / n)$$

else

$$R_j^i = 0.01$$

Node	Entry	MPR	Reputation
B	A	Yes	$f(R_B^A, R_B^F, R_B^G, R_B^H)$
C	A	No	$f(R_C^A, R_C^H)$
D	A	Yes	$f(R_D^A, R_D^H, R_D^I)$
E	A	No	0.5
F	B	N/A	0.5
G	B	N/A	0.5
H	B,C,D	N/A	0.5
I	D	N/A	0.5

Table 5.3: MRP Reputation Table, Stage 3

In our scheme, the penalty of misbehavior or selfish behavior is to set the reputation to 0.01, while if it behaves positive, the reputation will be the minimum value between 1 and  $R_j^i + \Delta * \sum_{n=0}^{k=0} R_k^i / n$ . This mechanism is used to stop the whitewashing attacks in section 6.5 .

## 5.4 Conclusion

This chapter presented the design a of a secure routing protocol over mobile ad hoc network, based on OLSR and reputation system. In this work, we have successfully issued reputations for users and picked MPR nodes wisely and efficiently. Our scheme also ensures that system is robust against several attacks. The simulation will be carried out in Chapter 7.

In the next chapter, we will discuss the security issue that might occur in our solution.

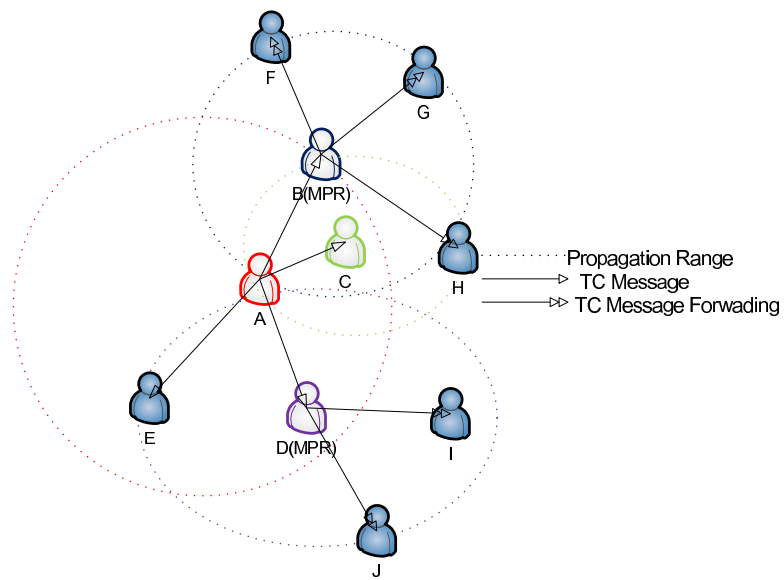


Figure 5.4: TC Message Forwarding



# Chapter 6

## Security Issues

MANET is mostly used in such a scenario, where one user joins a network with no body to trust, thus, in Chapter 4 and 5, we proposed two schemes to ensure secure communication. Meanwhile, there could be several types of malicious users who can perform different types of attacks as highlighted in Table 6.1. The following conclusions can be drawn from this table: our solution is as robust as the XRep protocol. In this chapter, we will discuss these attacks and our solutions.

	XRep	X <sup>2</sup> Rep	X <sup>2BT</sup> Rep	OLSR-Rep
Fake route(file)	Invincible	Invincible	Invincible	Immune
Man in the Middle	Invincible	Invincible	Invincible	Invincible
ID stealth	Invincible	Invincible	Invincible	With CL-PKE
Pseudo spoofing	Invincible	Invincible	Invincible	Invincible
Whitewashing	Invincible	Invincible	Invincible	Invincible
Shilling	Vulnerable	Invincible	Invincible	Vulnerable

Table 6.1: Six Types of Attacks

## 6.1 Fake Route Information

A fake route information attack occurs when a malicious node sends false route information to other nodes. For example, a good node  $A$  and a malicious node  $B$  are neighbours.  $B$  sends faulty information about  $B$  neighbours by saying it has got a neighbour named node  $C$ , in order to get a better reputation.

Our solution solves this problem in a way that when  $A$  sends  $C$  a Poll request, node  $C$ , either not exist or not being  $B$ 's neighbour, will not be able to reply. Furthermore, because the reply requires  $C$ 's signature,  $B$  cannot forge the reply. Thus  $A$  will notice that  $B$  was lying.

## 6.2 Man in the Middle

A man in the middle attack occurs when a malicious user lies in the middle of two honest users. For example, a malicious user  $B$  is in the middle between honest users  $A$  and  $C$ .  $B$  can intercept  $A$ 's packet to  $C$ , and replace it with other information.

Our key scheme first distributes public/secret keys for each user and then enables users to verify others' keys. Meanwhile, our routing scheme requires signature of packets from their original sender. Thus,  $C$  should be able to verify the signature of the packets with  $A$ 's ID or public key, which  $B$  cannot forge. In this case, if  $B$  discards the packet, then  $A$  will consider that route  $A$  to  $C$  via  $B$  is broken, while if  $B$  modifies the packet,  $C$  will easily tell. Either way,  $B$  cannot perform a man in the middle attack.

## 6.3 ID Stealth

An ID stealth attack occurs when a malicious user claims to be someone else who is honest. This attack can be stopped by our key management scheme because we associate IDs with public/private key set and a false ID can be easily detected.

## 6.4 Pseudo Spoofing

A pseudo spoofing attack is an alternative attack of ID stealth attack. It occurs when one malicious user uses multiple IDs thus it is always able to use new IDs when been categorized as unhonored.

This attack is prevented with the help of our key management scheme, because given a specific ID, there will only be one valid set of public/secret keys. Once a certain ID is detected as malicious, or associated with really low reputation, it cannot use other public/secret key set.

However, there is still one drawback of our key scheme that pseudo spoofing attackers can perform. We associate one pair of public/private key with one certain ID so that user cannot obtain more than one key set. Unfortunately we cannot stop malicious user to swap to another ID. In our key management scheme, we assume that any user who gains majority's (at least  $t$  DCA nodes) trust will be issued a public/private key set. Thus, one attacker will be approved with new key set if it is trusted by  $t$  trust.

Fortunately, we use MAC addresses as IDs. To fake a MAC address is costly. To gain trust from DCA nodes is also time consuming. So although our solution is vulnerable to pseudo spoofing attack, we believe that our scheme is robust from this attack in real world scenarios.

## 6.5 Whitewashing

A whitewashing attack is an attack used specifically against reputation systems. It occurs when a malicious user acts positively to earn enough reputation before performing negatively. This attack is avoided by given heavy penalty to those users who act badly. According to our reputation calculation equation and our penalty mechanism,

if (SUCCESSFUL\_TRANSMIT)

$$R_j^i = \text{Min}(1, R_j^i + \Delta * \sum_n^{k=0} R_k^i / n)$$

else

$$R_j^i = 0.01$$

a misbehavior will decrease the node's reputation to 0.01. Assume a bad node is surrounded by 20 very honest nodes with highest reputation of 1, 44 times of continuously successfully transmission is required to raise its reputation to an average value (0.5), if it is surrounded by 10 bad nodes with an average reputation of 0.1, the times of continuously successfully transmission will increase significantly to 152. For more details, the simulation results in chapter7 indicate how many packets a bad node needs to forward to get its reputation increased.

With the heavy penalty given to the misbehaviour nodes, we can ensure that our system is robust against the whitewashing attacks.

## 6.6 Shilling

A shilling attack is also such an attack that is used specifically against reputation systems. It occurs when a multiple number of malicious attackers work together to raise their reputations. The malicious nodes will have higher reputation thus will have the priority to be selected.

Unfortunately, we cannot guarantee that our solution is immune from this attack. However, our reputation calculation equation and our penalty mechanism ensures that every single misbehavior will be punished heavily that it is not worth performing shilling attack.

Ideally, when one malicious behavior has been detected, the attacker's reputation will drop to 0.01. In extreme situations, where the attacker is surrounded by all malicious users who are willing to raise the reputation for the attacker, it will still take times of forwarding packets (this is the same scenario with whitewashing attack when all the neighbours' reputations are set to be 1).

# Chapter 7

## Simulation and Results

### 7.1 CL-PKE Over MANET

#### 7.1.1 Simulation with C

**Setup** In this simulation, we implement our scheme with C code. The programming is based on Pairing Based Cryptography library (PBC) and GNU MP library (GMP), which define a large amount of efficient functions over pairing calculations. Table 7.1 indicates the programming environment.

CPU	Intel T2250 1.73GHz
Ram	1GB
Hard Disk	80GB at 5400rpm
OS	Ubuntu 7.01
GCC version	4.1
PBC lib version	0.4.17
GMP lib version	4.2.2

Table 7.1: Programming Environment for Key Generation

**Result** In this simulation, we assume that the network propagation delay is  $0ms$ , which means once the partial secret key is generated, it will be sent to the correspondent node immediately.

As shown in Table 7.2, if the partial secret key comes from the KGC, it takes  $142.7ms$  for a node to get its key. This time is of the time partial secret key generated by the KGC and the time a node generates its secret key/public key based on this partial secret key. If the partial secret key comes from DKGC nodes, the total generating time increases to  $156.7ms$  for a network with 5 nodes,  $224.3ms$  for a network with 10 nodes and  $313.8ms$  for a network with 20 nodes. This time is comprised of the time for each DKGC node to generate the partial secret key shares ( $10-13ms$ ) and the time the node generates the key based on these shares.

Number of nodes	5	10	20
Keys from KGC	142.756	142.756	142.756
Key shares	13.165	11.315	10.189
Keys from DKGC	156.739	224.295	313.790

Table 7.2: Simulation Results of CL-PKE over MANET with C

Note that this time will not change too much because all DKGC nodes generate partial secret key shares separately. The reason that key generating time is much higher than partial secret key generating time is that the key generating process involves a few pairing calculation over groups, while the partial secret key generating process only involves calculations over the infinite field.

### 7.1.2 Simulation with OPNET

This simulation runs over the OPNET modular. Firstly we will introduce the OPNET modeler.

**OPNET modeler** OPNET modeler is a powerful network simulation software which was developed by OPNET Technologies Inc. OPNET modeler provides a comprehensive development environment which can support both the modeling of communication network and the distributed system. In OPNET modeler, data was collected by running discrete event simulations (DES). It simulates the network's behavior and collects data by producing discrete events. By using OPNET modeler, we can perform model design, simulation, data collection and data analysis [14].

In OPNET world, the whole network is made up of several different nodes, an example of which is the MANET nodes that will be used in our simulation later. Each node is made of different process. During the simulations we collected status of different processes, nodes and networks.

**Simulation Scenarios** The second simulation runs over six scenarios:

1. 10 nodes in total running in partially distribution system , consist of 5 DKGC nodes, 1 type I attacker, 1 type II attacker and 3 normal nodes.
2. 10 nodes in total running in fully distribution system , all of them are DKGC nodes, consist of 1 type I attackers, 1 type II attackers and 6 normal nodes.
3. 10 nodes running in pure AODV system, with 1 type I attacker and 1 type II attacker.



4. 20 nodes in total running in partially distribution system , consist of 10 DKGC nodes, 2 type I attackers, 2 type II attackers and 6 normal nodes.
5. 20 nodes in total running in fully distribution system , all of them are DKGC nodes, consist of 2 type I attackers, 2 type II attackers and 16 normal nodes.
6. 20nodes running in pure AODV system, with 2 type I attackers and 2 type II attackers.

The attackers are defined as follows:

- Type I attacker does not forward any packets. It works simply as a sink.
- Type II attacker does wrong routing. It sends packets to any node other than the correct node. During the simulation, all the type II attackers forwards their packets to type I attackers.

**AODV parameters** The parameters of AODV are shown in Table 7.3.

In the simulation, all the nodes' movement follows the random waypoint model [4] with a pause time of 1 second and a maximum velocity of 10m/s. This mobility model defines that a node will pick some random waypoint in the wireless domain and move towards the waypoint with a velocity randomly picked between 0m/s(exclusive) and 10m/s(inclusive). Once a node gets to its destination, it will pause for 1 second and then move to the next waypoint. The movement repeats until the end of simulation.

The space of the wireless domain is 100m  $\times$  100m, and the propagation range for each node is 35 meters. When the simulation starts, there

Maximum Velocity	10m/s
Mobility Model	RWM
Pause Time	1 second
Dimensions of Space	100m $\times$ 100m
Radio Range	35m
Initiation Time	100 seconds
Background Traffics	1 packet per second
Packet Size	1024bits

Table 7.3: The AODV Parameters

is an initiation time for 100 seconds, during which time, no traffic is generated, except that between nodes and the KGC. After that stage, the KGC goes offline and each normal node (including DKGC nodes) will generate a background traffic, which is 1 packet per second in our simulation. Once a packet received/generated, it takes 0.04 second for a node to process it. This 0.04 second is the OPNET standard average propagation and processing delay. This delay increases to 0.055 second for DKGC nodes, which is because DKGC nodes need to have some extra time (10-13ms) to calculate partial secret key shares and validate public keys. The extra 10-13ms comes from the result of the first simulation.

**Results** As we can see from the Figures 7.1, 7.2, 7.3 and 7.4, our scheme successfully increased the efficiency and strengthened the security. As for a network with 10 nodes, the packet drop rate (Figure 7.1) was slightly higher with our scheme than it is with a pure AODV network during the initiation stage. Fortunately, the packet drop rate of our distributed system dropped to one third of a pure AODV network when the system became stable, with

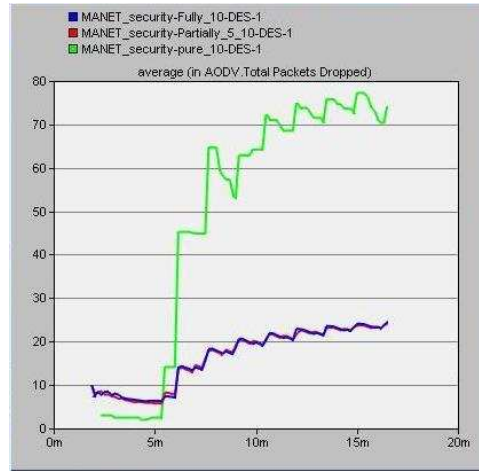


Figure 7.1: Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Packet Dropped

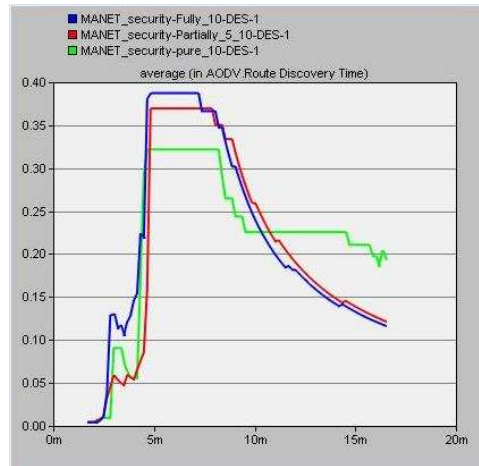


Figure 7.2: Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Route Discovery Time

approximately 20 packets per minute and over 60 packets per minute respectively. Despite that the routing traffic sent (Figure 7.3) and received (Figure 7.4) was comparatively noticeably higher, which, theoretically is be-

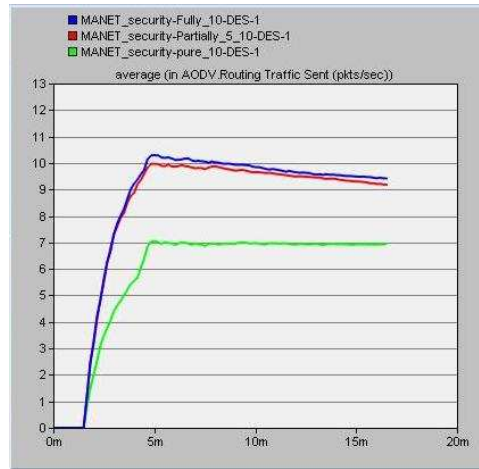


Figure 7.3: Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Packet Send

cause of the key distribution procedure, we managed to maintain the route discovery time (Figure 7.2) to the same level of pure AODV networks.

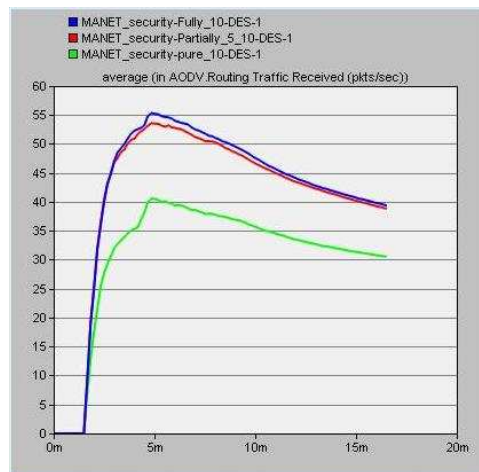


Figure 7.4: Simulation Results of CL-PKE over MANET with OPNET: 10 Users, Packet Received

By contrast, as illustrated in Figure 7.5, 7.6, 7.7 and 7.8, the improvement

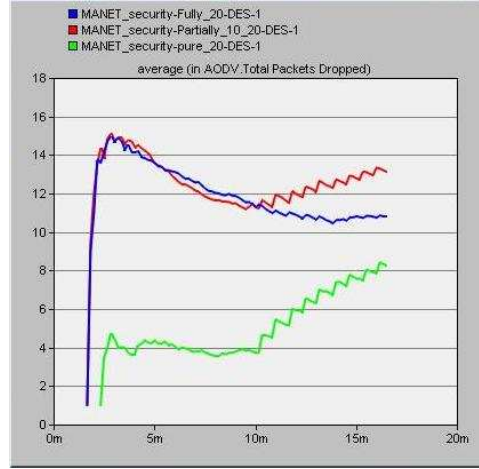


Figure 7.5: Simulation Results of CL-PKE over MANET with OPNET: 20 Users, Packet Dropped

of our scheme in a network of 20 users is more significant. Ideally, when the nodes density increases, the packet drop rate will decrease. Despite of that, our scheme still contributed to the packet drop rate (Figure 7.5), achieving an over 60% decrease from pure networks. Moreover, unlike the 10 user network where more time was spent to establish a route, the average route discovery time (Figure 7.6) was reduced from 0.71s of pure AODV networks to 0.41 of CL-PKE enabled AODV networks. Nevertheless, the communication overhead (Figure 7.7 and 7.8) was higher than pure AODV network. This is probably because our scheme produces a lot more traffic overhead and some of them are dropped because of the Type I attacker.

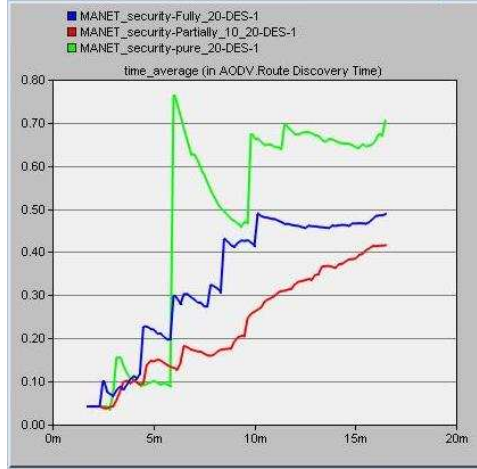


Figure 7.6: Simulation Results of CL-PKE over MANET with OPNET: 20 Users, Route Discovery Time

CPU	Intel T2250 1.73GHz
Ram	1GB
Hard Disk	80GB at 5400rpm
OS	Ubuntu 7.01
GCC version	4.1

Table 7.4: Programming Environment for Simulation against Whitewashing Attack

## 7.2 Rep-OLSR

### 7.2.1 Simulation against Whitewashing Attacks

In this simulation, we implemented our scheme with C codes. The simulations were carried out in three networks, with 5,10 and 20 users. Table 7.4 indicates the programming environment.

In each network, we examined 10 different values of neighbour reputations

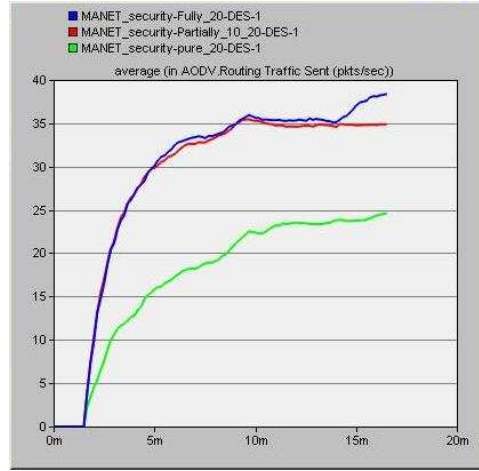


Figure 7.7: Simulation Results of CL-PKE over MANET with OPNET: 20 Users, Packet Send

(from 0.1 to 1 with an incensement of 0.1) and 6 different values of target's reputations (0.01, 0.02, 0.04, 0.08, 0.16 and 0.32). The results are shown in Appendix A.

The results indicated that for a 10 nodes network with an average reputation of 0.6, to recover the penalty of any mis-behaviors or selfish actions users need to forward at least 51 packets before it can send any packets. In the next simulation we observed that this penalty is heavy enough to stop the attackers.

## 7.2.2 Simulation with OPNET

**Simulation Scenarios** All the statistics are collected in four scenarios.

1. 20 honest users using pure OLSR protocol;
2. 20 honest users with 20% misbehavior users using pure OLSR protocol;
3. 20 honest users using Rep-OLSR protocol;

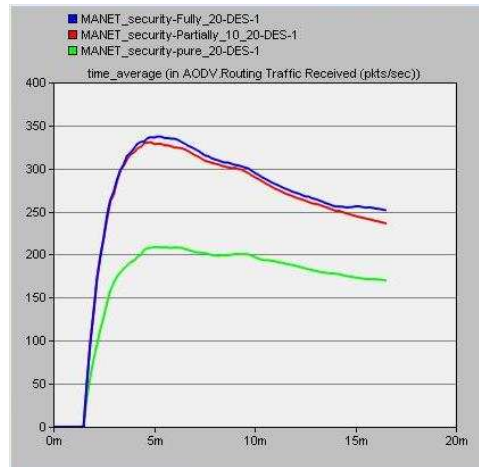


Figure 7.8: Simulation Results of CL-PKE over MANET with OPNET: 20 Users, Packet Received

4. 20 honest users with 20% misbehavior users using rep-OLSR protocol;

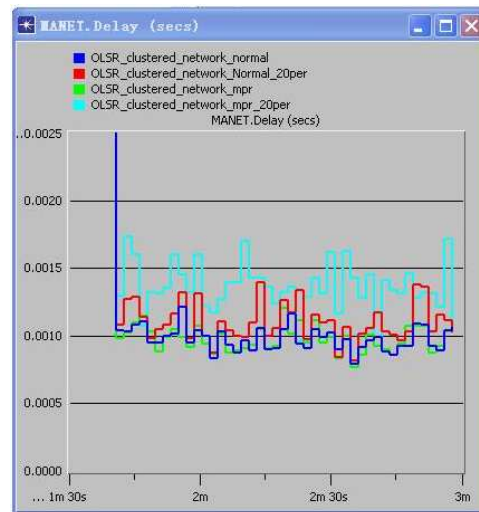


Figure 7.9: Simulation Results of Rep-OLSR over MANET with OPNET: 20 Users, Routing Delay



**OLSR Parameters** In the simulation, all the nodes' movement follows the same random waypoint model [4] as the simulation of CL-PKE, with a pause time of 1 second and a maximum velocity of 10m/s.

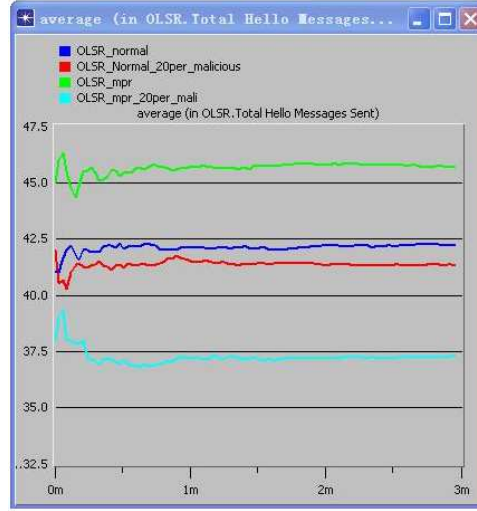


Figure 7.10: Simulation Results of Rep-OLSR over MANET with OPNET: 20 Users, Hello Message

The other aspects of the network remain the same with CL-PKE, for instance, the space of the wireless domain is  $100\text{m} \times 100\text{m}$ , and the propagation range for each node is 35 meters. Moreover, when the simulation starts, there is also an initiation time for 100 seconds, during which time, no traffic is generated, expect the hello messages and polling messages. After this stage, users start to communicate by generating a background traffic, which is 1 packet per second in our simulation.

Unlike CL-PKE, we mainly considered six features in Rep-OLSR:

- *Routing Delay* The time of routing delay;
- *Number of Hello Messages* The number of hello messages and polling messages;

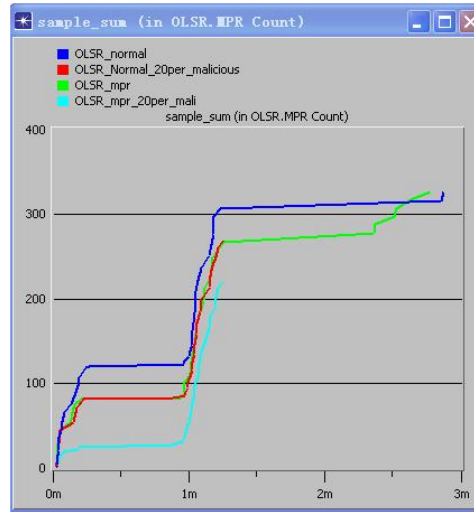


Figure 7.11: Simulation Results of Rep-OLSR over MANET with OPNET:  
20 Users, MPR Count

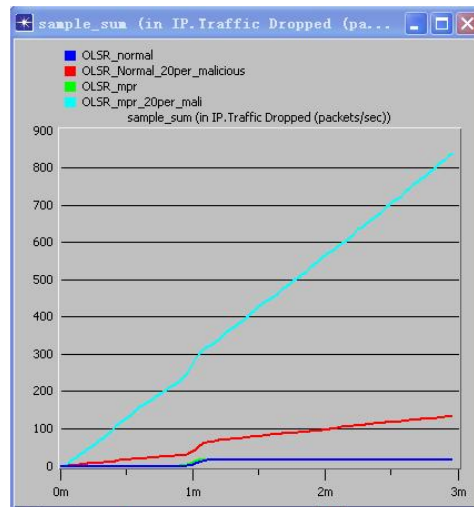


Figure 7.12: Simulation Results of Rep-OLSR over MANET with OPNET:  
20 Users, Traffic Dropped

- *MPR count* The number of Multi-Point Relay nodes;

- *Traffic Dropped* The number of packets that are dropped;
- *Traffic Received* The number of packets that are received;
- *TC Message Forward* The number of Topology Control messages that transmitted and generated.

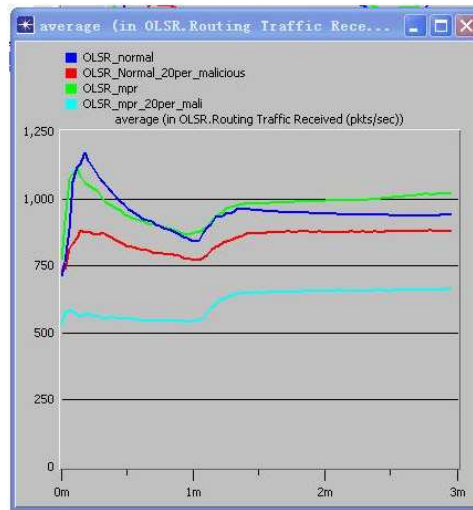


Figure 7.13: Simulation Results of Rep-OLSR over MANET with OPNET: 20 Users, Traffic Received

**Results** As we can see from the figures 7.9, 7.10, 7.11, 7.12, 7.13 and 7.14, our scheme successfully increased the efficiency and strengthened the security. As for the Routing Delay (Figure 7.9), the pure OLSR and the Rep-OLSR shared a same value (0.001s), although the Rep-OLSR with 20% malicious users added a insignificant amount, compared with pure OLSR with 20% malicious users. However, the statistics of Hello Message (Figure 7.10) are different. We managed to reduce 5 packets by using Rep-OLSR when there were intruders. As for the total MPR counts (Figure 7.11), the numbers

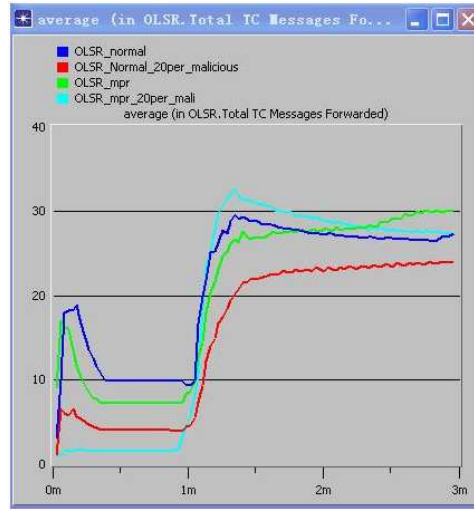


Figure 7.14: Simulation Results of Rep-OLSR over MANET with OPNET:  
20 Users, TC message forward

evened out in four scenarios, which means although we were using a reputation system, this approach will not add any extra load to users. However, routes are wisely selected. The last three figures (Figure 7.12, 7.13 and 7.14) also indicated that, the Rep-OLSR contributed to the stable communication. Furthermore, when there exist certain amount of selfish users, the Rep-OLSR can actually improve the overall performance of the network.

# Chapter 8

## Conclusions

This thesis presented a solution of secure communication over mobile ad-hoc networks. To achieve this goal, we firstly suggested a certificateless cryptography system with distributed model to distribute keys and certificates. The MANET is unique, so modern cryptography technologies might not perform as good when they are brought into MANET. Both public key cryptography and ID-based cryptography have been brought into this area. However, they all have inherent drawbacks. To this end, we adopt the certificateless cryptography which avoids the drawbacks while the system is still self-organized.

Then we proposed a reputation system to perform secure routing. Reputation systems have a long history, nonetheless, the adoption to the MANET is not straightforward. We attempted the reputation system on the OLSR routing protocol, because it is an pro-active routing protocol which uses hello message and MPR mechanism, where an addition of polling message will not add too much load. Collecting votes from any on demand routing protocols is much more complicated, where the user does not necessarily know its neighbour nodes thus, might not be able to send polling messages.

Finally, we carried out our schemes with simulations. The results indi-

cate that our schemes add a comparatively low amount of communication overhead while the intruders and malicious users are successfully prevented.

# Bibliography

- [1] A.A.Pirzada and C.McDonald. Kerberos assisted authentication in mobile ad-hoc networks. 27th Australasian Computer Science Conference, 2004.
- [2] Y.C.Hu A.Perrig and D.B.Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. pages 21–38. Wireless Networks 11, 2005.
- [3] R.L.Rivest A.Shamir and L.Adleman. A method for obtaining digital signatures and public-key cryptosystems. pages 120–126. Communications of the ACM 21, 1978.
- [4] C.Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. ACM SIGMOBILE Mobile Computing and Communications Review, Volume 5 Issue 3, 2001.
- [5] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). RFC3626, 2003.
- [6] C.Perkins and P.Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. pages 234–244. Proc. SIGCOMM’94 Conference on Communications Architectures, Protocols and Applications, 1994.

- [7] J.Van Der Merwe D. Dawoud and S. McDonald. A survey on peer-to-peer key management for mobile ad hoc network. pages Article 1 (April 2007), 45 pages. *ACM Comput. Surv.* 39, 1, 2007.
- [8] Y.Hu D.B.Johnson and Perrig. Sead: Secure efficient distance vector routing in mobile wireless ad hoc networks. pages 3–13. *Proc.4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 02)*, IEEE Press, 2002.
- [9] D.Boneh and M.Franklin. Identity-based encryption from weil pairing. pages 586–615. *SIAM J. Computing* 32(3), 2001.
- [10] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE IT*, 22:644–654, 1976.
- [11] SC.Perkins E.Belding-Royer and S.Das. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, 2003.
- [12] G.Montenegro and C.castellucia. Statistically unique and cryptographically verifiable (sucv) identifiers and addresses. pages 87–99. *Proc.symp. Network and Distributed Systems Security (NDSS 2002)*, Internet Society, 2002.
- [13] D. Johnson Y. Hu and D. Maltz. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. RFC 4728, 2007.
- [14] OPNET Technologies Inc. Modeling overview, help files of opnet modular 12.0.
- [15] S.Capkun J.Hubaux and L.Buttian. The official pgp user’s guide. IIT press, Cambridge, MA, 1995.



- [16] S.Capkun J.Hubaux and L.Buttyan. Self-organized public-key management for mobile ad hoc networks. pages 52–64. IEEE Transaction on Mobile Computing, Vol.2, No.1, 2003.
- [17] S.Capkun J.Hubaux and L.Buttyan. Mobility helps peer-to-peer securitys. pages 43–51. IEEE tans. Mobile Computer, volumn 2, issue 1, 2006.
- [18] L.Zhou and Z.J.Hass. Securing ad hoc networks. pages 13,6,24–30. IEEE Netw, 1999.
- [19] M.G.Zapata and N.Asokan. Securing ad hoc routing protocols. ACM 1-58113-585-8/02/0009, 2002.
- [20] P.Papadimitratos and Z.J.Haas. Secure link state routing for mobile ad hoc networks. pages 27–31. Proc. IEEE workshop on Security and Assurance in Ad Hoc Networks, IEEE Press, 2003.
- [21] E.Damiani S.D.C.di Vimercati S.Paraboschi P.Samarati and F.Violante. A reputation based approach for choosing reliable resources in peer to peer networks. pages 207–216. 9th ACM conference on Computer and Communications Security, 2002.
- [22] N. Curtis R.Safavi-Naini and W. Susilo. X<sup>2</sup>rep: Enhanced trust semantics for the xrep protocol. pages 205–219. Applied Cryptography and Network Security. Second Internation conference ACNS2004. LNCS 3089, 2004.
- [23] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, November 1979.

- [24] Adi Shamir. Identity-based cryptosystems and signature schemes. *Advances in Cryptology - Crypto '84, Lecture Notes in Computer Science 196*, pages 47–53, 1985.
- [25] Kimaya Sanzgiri Bridget Danhill Brain Neil Levine Clay Shields and Elizabeth M.Belding-Royer. A secure routing protocol for ad hoc networks. IEEE International Conference on Network Protocols (ICNP'02) 1092-1648/02, 2002.
- [26] H.Luo P.Zerfos J.Kong S.Lu and L.Zhang. Self-securing ad hoc wireless networks. Proceedings of the Seventh International Symposium on Computers and Communications (ISCC02), 2002.
- [27] S.S.Al-Riyami and K.G.Paterson. Certificateless public key cryptography. page 452C473. C.S. Lai (ed.) *Advances in Cryptology C Asiacrypt 2003, Lecture Notes in Computer Science*, 2003.
- [28] William Stallings. cryptography and network security: principles and practices, fourth edition. pages 210–218, 2006.
- [29] William Stallings. cryptography and network security: principles and practices, fourth edition. pages 334–344, 2006.
- [30] L.Yu W. Susilo and R.Safavi-Naini.  $X^{2BT}$  trusted repupation system:a robust mechanism for p2p networks. pages 354–380. CANS 2006, LNCS 4301, 2006.
- [31] S.Yi and R.Kravers. Practical PKI for ad hoc wireless networks. Tech. rep. UIUCDCS-R-2002-2273, UILU-ENG-2002-1717. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 2001.

# Appendix A

## Results of Whitewashing attacks

Nmbr Of Nghr Nodes	Ave rep of nghbr nodes	Rep of Taget Nodes	Packets Required
5	0.1	0.01	152
5	0.2	0.01	90
5	0.3	0.01	71
5	0.4	0.01	63
5	0.6	0.01	57
5	0.7	0.01	54
5	0.8	0.01	52
5	0.9	0.01	51
5	1	0.01	49
5	0.1	0.02	140
5	0.2	0.02	80
5	0.3	0.02	62
5	0.4	0.02	54
5	0.6	0.02	48
5	0.7	0.02	46
5	0.8	0.02	44
5	0.9	0.02	42
5	1	0.02	41
5	0.1	0.04	125
5	0.2	0.04	70
5	0.3	0.04	53
5	0.4	0.04	45
5	0.6	0.04	39
5	0.7	0.04	37
5	0.8	0.04	35
5	0.9	0.04	33

Nmbr Of Nghr Nodes	Ave rep of nghbr nodes	Rep of Taget Nodes	Packets Required
5	0.1	0.08	106
5	0.2	0.08	58
5	0.3	0.08	42
5	0.4	0.08	35
5	0.6	0.08	29
5	0.7	0.08	28
5	0.8	0.08	26
5	0.9	0.08	25
5	1	0.08	23
5	0.1	0.16	79
5	0.2	0.16	43
5	0.3	0.16	30
5	0.4	0.16	25
5	0.6	0.16	20
5	0.7	0.16	18
5	0.8	0.16	17
5	0.9	0.16	16
5	1	0.16	15
5	0.1	0.32	41
5	0.2	0.32	22
5	0.3	0.32	15
5	0.4	0.32	12
5	0.6	0.32	9
5	0.7	0.32	8
5	0.8	0.32	7
5	0.9	0.32	7

Nmbr Of Nghr Nodes	Ave rep of nghbr nodes	Rep of Taget Nodes	Packets Required
10	0.1	0.01	131
10	0.2	0.01	79
10	0.3	0.01	64
10	0.4	0.01	57
10	0.6	0.01	51
10	0.7	0.01	50
10	0.8	0.01	48
10	0.9	0.01	47
10	1	0.01	45
10	0.1	0.02	121
10	0.2	0.02	71
10	0.3	0.02	55
10	0.4	0.02	49
10	0.6	0.02	43
10	0.7	0.02	42
10	0.8	0.02	40
10	0.9	0.02	39
10	1	0.02	37
10	0.1	0.04	110
10	0.2	0.04	62
10	0.3	0.04	47
10	0.4	0.04	41
10	0.6	0.04	35
10	0.7	0.04	33
10	0.8	0.04	32
10	0.9	0.04	31

Nmbr Of Nghr Nodes	Ave rep of nghbr nodes	Rep of Taget Nodes	Packets Required
10	0.1	0.08	96
10	0.2	0.08	52
10	0.3	0.08	38
10	0.4	0.08	32
10	0.6	0.08	27
10	0.7	0.08	25
10	0.8	0.08	24
10	0.9	0.08	23
10	1	0.08	22
10	0.1	0.16	75
10	0.2	0.16	39
10	0.3	0.16	28
10	0.4	0.16	22
10	0.6	0.16	18
10	0.7	0.16	17
10	0.8	0.16	15
10	0.9	0.16	14
10	1	0.16	14
10	0.1	0.32	39
10	0.2	0.32	21
10	0.3	0.32	14
10	0.4	0.32	11
10	0.6	0.32	8
10	0.7	0.32	7
10	0.8	0.32	7
10	0.9	0.32	6

Nmbr Of Nghr Nodes	Ave rep of nghbr nodes	Rep of Taget Nodes	Packets Required
20	0.1	0.01	121
20	0.2	0.01	74
20	0.3	0.01	60
20	0.4	0.01	54
20	0.6	0.01	49
20	0.7	0.01	47
20	0.8	0.01	46
20	0.9	0.01	45
20	1	0.01	44
20	0.1	0.02	113
20	0.2	0.02	66
20	0.3	0.02	52
20	0.4	0.02	46
20	0.6	0.02	41
20	0.7	0.02	40
20	0.8	0.02	38
20	0.9	0.02	37
20	1	0.02	36
20	0.1	0.04	103
20	0.2	0.04	58
20	0.3	0.04	44
20	0.4	0.04	38
20	0.6	0.04	33
20	0.7	0.04	32
20	0.8	0.04	30
20	0.9	0.04	29



Nmbr Of Nghr Nodes	Ave rep of nghbr nodes	Rep of Taget Nodes	Packets Required
20	0.1	0.08	91
20	0.2	0.08	49
20	0.3	0.08	36
20	0.4	0.08	30
20	0.6	0.08	25
20	0.7	0.08	24
20	0.8	0.08	23
20	0.9	0.08	22
20	1	0.08	21
20	0.1	0.16	72
20	0.2	0.16	37
20	0.3	0.16	26
20	0.4	0.16	21
20	0.6	0.16	17
20	0.7	0.16	16
20	0.8	0.16	15
20	0.9	0.16	14
20	1	0.16	13
20	0.1	0.32	38
20	0.2	0.32	20
20	0.3	0.32	14
20	0.4	0.32	10
20	0.6	0.32	8
20	0.7	0.32	7
20	0.8	0.32	6
20	0.9	0.32	6

## Appendix B

## Glossary

AODV	An On-demand Distance Vector routing protocol
ARAN	A secure Routing protocol for Ad-hoc Networks
Ariadne	a secure on demand routing protocol for ad hoc network
CA	Certificate Authority
CL-PKE	Certificateless Public Key Encryption
CREP	Certification Reply
CREQ	Certification Request
DCA	Distributed Certificate Authority
DKGC	Distributed Key Generation Center
DN	Destination Node
DoS	Denial of Service
DSDV	Destination-Sequence Distance-Vector
DSR	Dynamic Source Routing
ID	Identity
ID-based	Identity-based
KGC	Key Generation Center
LSU	Link State Update
MAC	Message Authentication Code
MANET	Mobile Ad-hoc Network
mpk	master public key
MPR	Multi-Point Relay
msk	master secret key
OLSR	Optimized Link State Routing Protocol
OTTP	Offline Trusted Third Party

PGP	Pretty Good Privacy
PKG	Private Key Generator
Rep-OLSR	Reputation system over OLSR
SLSP	Secure Link State Protocol
SAODV	Secured AODV
SEAD	Secure Efficient distance vector routing for MANET
SLSP	Secure Link State Protocol
SN	Source Node
SUCV	Statistically Unique Cryptographically Verifiable address
TC	Topology Control