Encryption Key Management in Wireless Ad Hoc Networks

A thesis presented in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy in Computer Science

at Massey University, Auckland, New Zealand

Alastair Jon Nisbet

2010

Abstract

Communication is an essential part of everyday life, both as a social interaction and as a means of collaboration to achieve goals. Networking technologies including the Internet have provided the ability to communicate over distances quickly and effectively, yet the constraints of having to be at a computer connected to a network access point restricts the use of such devices. Wireless technology has effectively released the users to roam more freely whilst achieving communication and collaboration, and with worldwide programs designed to increase laptop usage amongst children in developing countries to almost 100%, an explosive growth in wireless networking is expected. However, wireless networks are seen as relatively easy targets for determined attackers. Security of the network is provided by encrypting the data when exchanging messages and encryption key management is therefore vital to ensure privacy of messages and robustness against disruption.

This research describes the development and testing through simulation of a new encryption key management protocol called SKYE (Secure Key deploYment & Exchange) that provides reasonably secure and robust encryption key management for a mobile ad hoc network. Threshold cryptography is used to provide a robust Certificate Authority providing certificate services to the network members using Public Key Infrastructure. The protocol is designed to be used in an environment where communications must be deployed quickly without any prior planning or prior knowledge of the size or numbers of the potential members. Such uses may be many and varied and may include military, education or disaster recovery where victims can use the protocol to quickly form ad hoc networks where other communication infrastructure has failed. Many previous protocols were examined and several key

features of these schemes were incorporated into this protocol along with other unique features. These included the extensive tunability of the protocol allowing such features as increasing the number of servers that must collaborate to provide services and the trust level that must exist along a certificate chain before a request for a certificate will be accepted by a server. The locations of the servers were carefully selected so that as these parameters were altered to increase security, performance remained high. For example, when two servers were required for certificate issuance, a certificate request would succeed 92% of the time. By doubling the servers required and therefore considerably increasing resilience against attack of the certificate authority, this figure dropped only moderately to 78%. The placement of the servers proved to be a critical parameter and extensive experiments were run to identify the best placements for servers with the various parameters chosen.

Simulations show that the protocol performs effectively in a developing and constantly changing network where nodes may join and leave the network frequently and where many of the members may be mobile. The many tunable parameters of the protocol ensure that it is useful in a variety of applications and has unique features making it effective and efficient in a highly dynamic network environment.

Acknowledgements

I would like to thank the many people who have provided encouragement and advice during the time that this research was undertaken. Thanks to my supervisor Dr Mohammed Rashid who provided enthusiastic supervision that ensured the research continued its course towards completion. His knowledge, advice and encouragement during difficult times were a source of inspiration.

Thanks also to my co-supervisor Dr Fakhrul Alam who provided advice and support with the technical aspects of this research and to the School of Engineering and Applied Science at Massey University that provided resources and facilities required to complete the research.

I am very grateful to my previous supervisor Dr Ellen Rose who provided a wealth of knowledge and experience to ensure the research took an unfaltering path during her time as my supervisor.

The completion of this thesis is a testament to the unwavering support of my wife, Ruth who provided many years of support whilst I undertook a lifelong ambition. To her I will be eternally grateful.

This thesis is dedicated to my wife and our two wonderful little children, Thomas and Skye who have suffered the trials and tribulations of Doctoral research at times as much as me. Thank you to my wonderful family.

Table of Contents

List of Figures

List of Tables

List of Publications

Nisbet, A. J. *Wireless Networks in Education – A New Zealand Perspective*. Proceedings of the IIMS Postgraduate Conference 2004, Massey University, Auckland, 2004.

Nisbet, A.J. *An Improved Encryption Key Management System for IEEE 802.16 Mesh Mode Security Using Simulation.* Proceedings of the Fifth New Zealand Computer Science Research Student Conference, Waikato University, Hamilton, 2007.

Nisbet, A.J. Rashid M.A., *A Scalable & Tunable Encryption Key Management System for Mobile Ad Hoc Networks.* Proceedings of the 2009 International Conference on Wireless Networks, Las Vegas, USA, 2009.

Nisbet, A.J, Rashid M.A, Alam, F. *The Quest for Optimum Server Location Selection in Mobile Ad Hoc Networks Utilising Threshold Cryptography.* Proceedings of the 7th International Conference on International Technology: New Generations. Las Vegas, USA, 2010.

Nisbet, A.J, Rashid M.A. *Performance Evaluation of Secure Key Deployment and Exchange Protocol for MANETs.* Accepted for International Journal of Secure Software Engineering (IJSSE), 2010.

Abbreviations and Acronyms

AAA	Authentication Authorisation and Accounting
ACL	Access Control List
AES	Advanced Encryption Standard
AP	Access Point
ARP	Address Resolution Protocol
AS	Access Server
BPSK	Binary Phase Shift Keying
BSS	Basic Service Set
CA	Certificate Authority
CCK	Complimentary Code Keying
CCM	Clear Channel Assessment under MAC
CPU	Central Processing Unit
CRC	Cyclic Redundancy Checksum
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DBPSK	Differential Binary Phase Shift Keying
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
DQPSK	Differential Quadrature Phase Shift Keying
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol
EAPOL	Extensible Authentication Protocol Over LAN
ETSI	European Telecommunications Standards Institute
ERP	Extended Rate PHY
IBSS	Independent Basic Service Set
IC	Integrity Check
ICV	Integrity Check Value
ID	Identity
IEEE	Institute of Electronics and Electrical Engineers
IP	Internet Protocol
IPSec	Internet Protocol Security
IV	Initialisation Vector

FCC	Federal Communications Commission
FH	Frequency Hopping
GHz	Gigahertz
ISM	Industrial Scientific and Medical
KDC	Key Distribution Centre
KGS	Key Generation Server
KM	Key Management
KMS	Key Management Service
L2F	Layer 2 Forwarding
LAN	Local Area Network
MAC	Media Access Control
MANET	Mobile Ad Hoc Network
Mbps	Mega bits per second
MIC	Message Integrity Check
MIMO	Multiple Input Multiple Output
MPDU	MAC Packet Data Unit
MSDU	MAC Service Data Unit
MTM	Man in The Middle
NAS	Network Access Server
OFDM	Orthogonal Frequency Division Multiplexing
PBCC	Packet Binary Convolutional Coding
PCI	Peripheral Component Interconnect
PCMCIA	Personal Computer Memory Card International Association
PDA	Personal Digital Assistant
PHY	Physical Layer
PKI	Public Key Infrastructure
PMK	Pre-shared Master Key
PPTP	Point To Point Tunnelling protocol
PRNG	Pseudo Random Number Generator
PSK	Pre-Shared Key
PTK	Pre-shared Temporal Key
PTMP	Point to Multi Point
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying

RADIUS	Remote Access Dial In User Service
RFMon	Radio Frequency Monitor Mode
RTS/CTS	Request to Send / Clear to Send
STA	Station
TGT	Ticket Granting Ticket
TKIP	Temporal Key Integrity Protocol
TTP	Trusted Third Party
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
WiFi	Wireless Fidelity
WiMax	Worldwide Interoperability for Microwave Access
WPA	WiFi Protected Access
XOR	Exclusive Or

Chapter I

INTRODUCTION

1 .1 Introduction

Communication is a vital part of everyday life for most people. The ability to communicate directly with another person allows us to collaborate with ideas and actions. The ability for computers to communicate quickly and efficiently in a network allows collaboration over vast distances bringing people together to work towards a common goal. Networking was a revolution in the development of the computer age increasing productivity, exchanging ideas and information and allowing people to participate in tasks that otherwise would have been impossible. However, fixing a device to a single location where it needs to be wired to a network socket to join a network is a severe constraint on how and when it can be used.

The introduction of wireless technologies in the 1990s allowed people to take their computers with them and connect to corporate networks or the Internet from cafeterias, parks or locations around offices that were near to wireless access points. This freedom of movement led to uses for computers that had not been possible previously. Areas such as health informatics, education and military applications to name just a few have been revolutionised by wireless communications. However, the constraint of communicating through a fixed access point means that the device must still be within a few hundred metres at best of fixed infrastructure to connect to the network. Eliminating the access point and allowing direct communication between devices allows a host of new uses providing collaboration between mobile devices anywhere and at anytime. This type of rapidly deployed network that can be set up and disbanded as required is called ad hoc networking. The main constraint to the acceptance of Mobile Ad Hoc

Networking (MANET) is security concerns that must be solved before people will accept and trust these types of networks. Security not only provides privacy for message exchange, but protects the network against malicious behaviour that can seriously disrupt the operation of the network itself. This thesis describes the development and performance studies of a new and unique encryption key management (KM) protocol to help solve the problem of security within an ad hoc network.

Wireless devices communicate with other wireless devices by sending and receiving messages across the airwaves using radio frequency. Generally, this radio signal is omni-directional and propagates out for a distance in all directions, much like the ripples on a pond from a stone thrown into the centre. This is significantly different from wired communication, where much of the security of wired devices can be found in the ability to secure the interconnecting wire from intruders, thus making the communications physically difficult to intercept. However, with wireless signals sent out across the airwaves, an intruder only needs to be within range of the signal to intercept the communications with a similar wireless device. Capturing the wireless communications with a latent device is almost undetectable so the victim may not be aware that the communications are being monitored (Pietro and Mancini 2003). It is these very properties of wireless communications that makes wireless devices so attractive and vulnerable to attack.

The structure of a wired network incorporates control over the network so that if wanted, only authorised users can join and participate. Rules can be created so that restrictions on the use of the network, bandwidth used, protocols permitted and entry and exit traffic from the network can be filtered and even blocked if necessary. A

network of devices connected with wires may incorporate a variety of security elements depending on the level of security required. The security can be controlled by separate servers dedicated to the task of monitoring network use and permitting only those devices or users authorised to use the network to join. Much of the security can be handled separately from the devices using the network allowing very computationally powerful machines to be dedicated to the task of handling security which may remain largely transparent to the user.

This relative ease of control over the network is not enjoyed to the same extent by a wireless network. Whilst wireless networks are much quicker and simpler to deploy, especially in areas where wiring devices may be difficult or even against the law such as in historic buildings, the freedom from wires comes with a security cost. It is not just the broadcasting of data over the airwaves that may be targeted by an attacker, but the ease with which an unauthorised device can attempt to join the network that makes securing the network difficult. A distinction is drawn between associating with a network and being authorised on the network. Layering the security so that unauthorised users can not associate with the network, which is the first step in joining the network, can be done using access control methods such as permitting only specified devices to join that have been previously authorised. However, as we shall see, preventing association can be fairly easily worked around so that preventing authorisation is the major weapon in the prevention of attack of the networks. Table 1.1 shows the six aspects of security for a network.

Table 1.1: The six aspects of security in a network.

Confidentiality	The ability to send messages to a recipient and be sure that only the intended recipient can read the message.
Integrity	The assurance that the message received has not been altered and is identical to the message sent.
Non-repudiation	The ability to prove that the persons who sent the message cannot deny that they sent it
Authentication	The assurance that the message sent is genuine and has not been sent by a third party masquerading as the genuine party
Authorisation	The ability to grant access to the network to authorised users and deny access to unauthorised users
Availability	The ability to ensure that the network is robust and stable enough to remain available at all times to authorised users

For any security protocol to be considered complete, all of these points must be satisfied to some degree. Whilst security tends to be a compromise between robustness and ease of use, it is possible for protocols to satisfy all security elements providing complete and robust security.

1.2 Statement of the Problem

With communications broadcast over the air and the inability to prevent an unauthorised node from receiving those communications, security in a wireless environment relies on encrypting messages before they are sent, and providing the ability for only the intended recipient to decrypt the messages. Encryption and decryption of messages requires two artifacts, a suitably robust encryption and decryption algorithm and an encryption key of an appropriate length. The encryption algorithms are generally openly available to any person who wishes to scrutinise them and therefore it is the encryption key that must be kept secret to prevent an attacker from decrypting a message. Much research has been performed in the area of wireless network security utilising infrastructure mode where a base station is used to provide a centre of control for the network. Much less research has been undertaken in the area of MANETs where much greater challenges to security exist because of the lack of any centre of control within the network.

The creation, distribution, revocation and reissuance of encryption keys is known as encryption key management, and it is the management of encryption keys in a Mobile Ad Hoc Network environment that is the focus of this research. In a truly ad hoc network the members have no prior knowledge of each other and all key management tasks are undertaken after the network has formed. This makes providing security particularly challenging as malicious or untrustworthy members can easily join the network, meaning identifying and ejecting misbehaving members is a vital requirement in maintaining security.

Almost all previous protocols utilise offline configuration prior to network deployment or alternatively use a secure side channel to exchange encryption keys prior to communication. Providing a robust key management (KM) protocol for entirely on-the-fly formation of a MANET allows utilisation of these types of network in areas where hesitation has existed because of security concerns. It is much more difficult to perform key management after network formation and with devices that have no prior knowledge of each other. Key management not only allows privacy of messages between users, but also enforces authrorisation to join the network and provides a way to easily revoke the encryption key to eject a misbehaving member of the network. Whilst it is a significant challenge to develop protocols that provide online network formation, the benefits that will result from success make the challenge a very worthwhile undertaking.

The problem we want to solve is the following. What could be a new encryption key management protocol for wireless ad hoc networks that will perform better than previous protocols in areas such as rapid deployment, versatility, availability,

redundancy and tunability? We hypothesise that there exists significant scope to design and develop a new encryption key management protocol that will meet the stated goals.

1.3 Research Methodology

A methodology is a tool to ensure that the research undertaken follows accepted methods and ensures the validity of results. A methodology should organise and structure the tasks undertaken to achieve the goals, include the methods and techniques for accomplishing those tasks, and prescribe the order and method for achieving the desired objectives (Nance and Arthur 1988). The task of progressing through the research from defining the problem to be studied through to the conclusions that can be drawn is divided into stages and each stage is completed before moving onto the next.

The first stage in this research involves defining the problem that is to be studied. This involves formulating and stating the goal of the research so that the results of the research can be compared with the initial goal. The problem is defined previously in Section 1.2.

The next task is to develop the protocol that will meet this goal. This firstly involves a thorough review of the literature relating to encryption and encryption key management, wireless network security protocols and previous protocols that have been developed for wireless ad hoc networks. This provides an understanding of the current state of development and gives ideas that can be utilised in the new protocol.

Next, the most relevant protocols are identified and the features of those protocols at each stage of network development are isolated. Then, the weaknesses and strengths

that have been identified in these protocols are noted. The desirable features can then be listed and incorporated into the new protocol where possible. This builds a base for features that can be modified and enhanced leading to the development of the complete protocol. Once this has been successfully completed, the new protocol can then be tested. It is most appropriate in this case to use simulation research to test the protocol so that a large number of scenarios can be experimented with.

Simulation research involves using simulation to model a system and observe the system in operation. There are distinct types of simulations and models used in simulation. These are:

- Dynamic v Static Models: A dynamic simulation involves changes over time as the simulation progresses. A static simulation involves a snapshot of a single point in time of a system.
- Deterministic, Stochastic and Chaotic Models: In a deterministic model, the system is entirely understood and can be predicted. Stochastic models involve behaviour which cannot be entirely predicted where the input variables lie within a range rather than a single point. Chaotic models are deterministic models where the behaviour cannot be entirely predicted.
- Discrete Simulation: A discrete simulation involves events occurring at discrete points in time. The events occur at a countable point in time. The simulation of this protocol steps at one second intervals where events occur predictably.
- Continuous Simulation: In a continuous simulation, the changes are made in a continuous way from one state to another giving an infinite number of states.

The type of simulation used in this research is a discrete, dynamic simulation with stochastic modelling. As the simulation progresses, the network changes over time as events occur at intervals. Some of these events are based on the use of random variables meaning the exact occurrence can not be predicted.

The stages in the simulation design, testing and running of the experiments follow that proposed by Pegden and Shannon (Pegden, Shannon and Sadowski 1990) and are more thoroughly covered in Chapter six. Once the simulations have been performed, the results are analysed and the performance measured and where possible the results are compared to published results of other protocols. This allows performance comparisons to be made. The results of these comparisons form the conclusions in Chapter eight where the level of success against the stated goal is made.

The final stage involves documenting the results. If desirable for future work, documenting the software and providing details on how to use it can be done to allow future researchers to fully understand the simulation software so that its use and any modifications that may be desirable for later studies can be made.

1.4 Motivation

The one common problem that must be overcome for any of these uses is the ability to reliably and securely send messages between wireless devices and ensure that only the user of the receiving device can read the message. If the problem of maintaining security in a dynamic, mobile network can be improved, the use of this type of network will become significantly more accepted. Security is a major concern to all users of networks, with one study of New Zealand organisations in 2005 finding that 48% of

managers said that security concerns were a major reason for non-deployment or non-consideration of installing a wireless network (Houliston and Sarkar 2005). The same study found that 28% of those organizations that had wireless networks were using them in ad hoc mode. Whilst use of wireless networks is continuing to grow, security concerns are still seen as holding back that growth. To overcome this reluctance, security must be shown to be robust and efficient if wireless networks, and especially ad hoc networks, are to enjoy the explosive growth originally predicted.

Uses for ad hoc networks fall into three broad categories: commercial, military, and private which includes such uses as education and disaster relief. Whilst the uses for the IEEE 802.11 and HyperLAN series of wireless protocols are generally considered for home, office or commercial applications connecting to the Internet, providing coverage to rural areas that are distant from the transmitting base station remains difficult and costly. By having outlying users join the network and utilising neighbours as hopping points for data, the range of the network can be greatly enhanced. Additionally, providing alternative routes for messages to flow between the base station and end user allows redundancy of routes meaning greatly improved reliability. By combining fixed wireless devices with ad hoc networks and a base station, the combination of wireless devices can utilise the base station to act as a gateway for the network allowing some control over the network.

Military uses allow ad hoc networks to be rapidly set up in a battleground situation where communications are extremely vital, and the ability to be able to communicate with peers can give significant tactical advantage. Military situations generally involve high levels of control, and in this type of situation a central point of control for the

network is most desirable. Also, military applications require very tight control over who is permitted to join the network and participate. As this protocol is designed to allow anyone entry to the network, it may be unsuitable for certain military applications. However, entirely ad hoc networks could provide the ability to rapidly create a network and provide communications in situations where security threats may be limited.

In contrast to these two types of networks, education and disaster relief situations may require rapid deployment of the network, but control over the network is much less of an issue. For a disaster situation, the primary objective for the network is communications, whether that is communicating with a central access point giving Internet access or to a central relief station. However, should an ad hoc network be set up where the users can initially only communicate with each other, there are still significant benefits over having no communication. In this type of situation, rapid deployment and range of communications are of primary importance, with security being less of a priority.

One of the major challenges presented by ad hoc networking is the inability to plan in advance. A truly ad hoc network is distinguished by several factors:

1 The network is set up ‘on the fly’ for a specific purpose.

2 It is temporary and will disband when no longer required.

3 Members have no prior knowledge of each other.

4 Many other parameters are not known such as:

- node joining and leaving rate
- geographical spread
- numbers of mobile nodes and speed of the mobile nodes

With so much not known about the potential characteristics of the network after deployment, it is an advantage to be able to tune the network to the requirements. As the network grows, requirements may change and so being able to make minor tuning adjustments to the network parameters may also be a useful feature.

Whatever the use the ad hoc network is put to, several criteria are common to all three applications. That is, the network must be able to be rapidly set up, it must be reliable and it must provide an appropriate level of security so that communications cannot be disrupted and messages can remain secret when required. With many nodes in an ad hoc network running with battery power, a further consideration is that the security protocols running on the nodes be not overly complex so that nodes' batteries will not be exhausted quickly. Therefore the important criteria for a new key management system (KMS) in an ad hoc network is the efficiency of the network communications and the effectiveness of the encryption key management system.

1.5 Contribution of the thesis

This thesis provides an investigation and discussion of previous work from which several of the features of the new protocol have been drawn. By combining some of the best features of previous work with many new features, a unique protocol that provides tunability and versatility in a variety of deployment scenarios has been developed. The contributions made by this thesis are:

- The problem of encryption key management in ad hoc networks has been thoroughly studied and identified.
- An extensive and up-to-date review of the relevant literature relating to MANET key management is provided.

- The key features required by such protocols along with the parameters that should be tunable to the application is identified, defined and specified.
- A unique encryption key management protocol has been developed with the identified features.
- A unique simulation model has been developed in MATLAB for performance studies of the protocol.
- Further direction for future research is provided.

The development of the protocol has provided a full encryption key management scheme that may be useful in a variety of applications. Several possible uses for this type of protocol are:

1. Any group of people who wish to quickly and simply form a network can do so for a variety of possible uses. This may include education where students can deploy into a field equipped with wireless devices such as laptop computers. They can quickly and securely form a MANET and exchange information in real time to collaborate and conduct experiments or similar.

2. Disaster Recovery: People affected by a natural disaster such as an earthquake or flood can utilise their wireless devices to rapidly and simply set up and operate an ad hoc network prior to any official rescue efforts being deployed. Provided their device is within radio range of another device, communication can begin between the two parties. Any other devices within range can utilise the new protocol to request authorisation to join the network and be assured of security sufficient to provide all six aspects required of network security.

3. Military Training: As the protocol utilises online configuration it may not be suitable where a high likelihood of unauthorised requests to join the network is present, or where a determined and skilled attacker may be present. However, in areas where unauthorised nodes are not present, then this protocol could be used with caution in a training exercise. This may be useful because of the attributes of rapid deployment and handling of mobility. However, the authentication of the device rather than the user would mean that in certain military operations it may not be suitable. Military operations may require a protocol that has entirely offline configuration and all network members and devices are preapproved before deployment.

Whatever the application that the protocol is used for, the attribute of entirely online configuration allows rapid network formation and high scalability meaning its uses may be many and varied.

1.6 Thesis Structure

Chapter one of this thesis has provided an introduction to the topic of wireless ad hoc networking. A brief discussion of what wireless networks are and the comparison between wireless and wired network security has been made. The challenges of providing security in a mobile ad hoc wireless environment have been discussed and the motivation for meeting those challenges with this research has been made. A list of the three likely areas of use for this type of protocol is given, and the method in which this new design meets those challenges specific to ad hoc networks is discussed. Finally, the contribution of this work is discussed showing that the new protocol is useful for advancement of research into providing security within a wireless ad hoc environment

and that it lays the groundwork for further development of the protocol or development of a new protocol that may use some of the new features provided by this work.

Chapter two discusses mobile ad hoc network security and looks at why security in this type of network is so difficult to implement effectively. This chapter looks at how the security of wireless networks has developed over the past few years and how using the standard protocols has been effective for networks in infrastructure mode but does not implement well in ad hoc networks. A discussion on the various methods of ad hoc security implementations is briefly made with a view to looking more thoroughly at cryptography and the evolution of Mobile Ad Hoc networking in the following chapter.

Chapter three discusses the evolution of cryptography in computing from the early beginnings over fifty years ago through to the latest algorithms used in today's environment. The use of cryptography in MANETS is looked at and the benefits and drawbacks of the main methods for encrypting and decrypting messages is discussed. This leads to a discussion of the chosen methods for the new protocol with a justification of why the two general types of cryptography, symmetric and asymmetric, have been used in this protocol.

Chapter four is the review of literature relating to mobile ad hoc network security. An in-depth discussion of previous work is made by looking at the features of previous protocol designs. The benefits and drawbacks of the features are discussed and the key features that have proven to be significant advancements are looked at. The reasons for choosing these key features in this new protocol are discussed with a view to supporting the current design with a thorough analysis of prior works.

Chapter five discusses the new design for the protocol looking at each stage of development and relating the choices along the way to previous work. The key features of the design are thoroughly discussed and an in-depth analysis of how the protocol works in each possible situation is made. An examination of the method used to test the protocol design is introduced with a more thorough analysis discussed in the following chapter.

Chapter six looks at the many thousands of simulations performed on the protocol and the variables that are changed for the different simulations to identify performance characteristics of the protocol. These performance trends are identified and a discussion of the implementations relating to security and efficiency levels is made.

Chapter seven discusses the results of the simulations and identifies conditions under which the protocol performs most efficiently. The trade-off between efficiency of this scheme and security level is analysed and any limitations of the protocol is identified and discussed. As the input parameters are altered, the effectiveness and efficiency of the protocol changes and the results of these changes are compared to provide a reference framework that can be used when implementing the protocol.

Chapter eight discusses the conclusions that can be drawn from the results of the simulations. It looks specifically at what inferences from the results can be drawn and how the many tunable parameters can be adjusted to provide the security and efficiency required for the applications the protocol may be used for. Finally, future work that can be undertaken to further improve the protocol is discussed.

Chapter II

MOBILE AD HOC NETWORKS SECURITY AND KEY MANAGEMENT

2.1 Introduction

This chapter introduces wireless networks and discusses both networks that utilise a centre of control, and truly ad hoc networks where all stations are equal and no centre of control exists. It looks at the most common wireless protocols and the security that has been built into those protocols at design time. A discussion of the evolution of the protocols and security standards is made, and how those standards have evolved over time as vulnerabilities have been exposed and new protocols have been developed to eliminate those vulnerabilities. The chapter looks at both standards based and non-standards based security protocols found in wireless networks, and the reasons why those standards by themselves are not suitable for ad hoc networks. Finally, a brief discussion of encryption key management in wireless ad hoc networks is made and why this is important if ad hoc networking is to be adopted as a topology for wireless networks worldwide.

2.2 Mobile Ad Hoc Networks

Wireless networks are being utilised in a variety of environments. Even with the initial wireless protocols providing limited range and bandwidth, they were ideally suited as an alternative to wired networks in an office or home environment. With the newer protocols providing much greater range and bandwidth, many more possible uses are being realised. Firstly, commercial uses for providing Internet availability as an alternative to wired telephone lines is gaining popularity, as many potential Internet users live in areas where it is impossible or very expensive to provide wired broadband

Internet services. Particular interest is being shown in rural areas where the ability to interconnect wireless devices (often referred to as nodes) to provide much greater range to the network is of particular benefit. Additionally, military forces around the world are already deploying wireless networks in battlefield situations where the ability to rapidly set up a mobile network is of great benefit. Further uses for ad hoc networks include disaster relief situations such as in a flood or earthquake where transport and communications infrastructure may be severely affected. Here, the ability to set up an ad hoc network by placing nodes on rooftops and high terrain, especially with the ability to use intermediate nodes as hopping points to increase the distance for communications, means messages can be sent quickly over vast distances The potential to operate a network in this mesh mode is extremely attractive as it not only adds the ability to extend the range of the network, but also adds redundancy of routes from node to node greatly enhancing the reliability of connections. Without the need for a central base station, this type of network can be initialised with little or no forward planning meaning it can develop and grow on the fly, when needed and for a particular purpose. Figure 2.1 shows a mesh network where nodes remain stationary

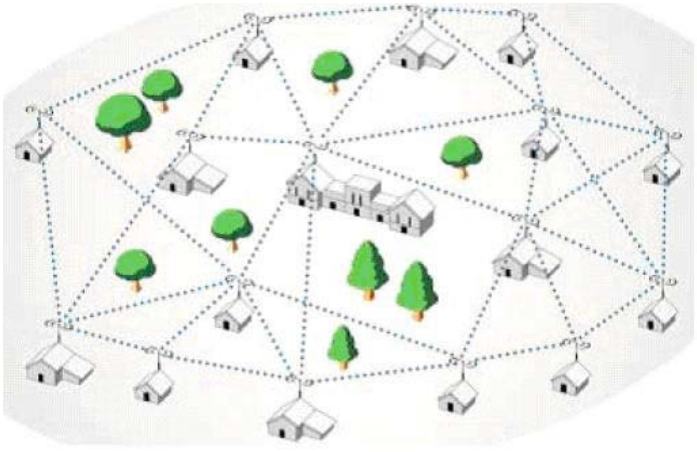


Figure 2.1: A wireless mesh network (Higgins, Egan, Hurley and Lemur 2006).

However, to take this type of topology one step further greatly increases its potential. That step is to add a dynamic ability to the mesh network, therefore creating a MANET. This topology is distinguished by being able to dynamically set up a network of peer nodes quickly, being able to allow members of the network to leave and new members to join, and to provide the ability for members to move through the network whilst maintaining connectivity and security of communications. The security of these types of networks present the greatest challenges, both because of their dynamic nature and because there is no prior planning or preconfiguration.

Ad Hoc networks can be spread over large geographical areas with numbers of members ranging from just a few, to thousands of wireless devices all communicating with each other. The network may grow and shrink as devices join the network and leave again, and secure communications must also be achieved for devices that wish to communicate with each other that may be out of direct range. Networks may spring up separately and as they grow in numbers and area may join together or separate apart forming bigger networks, or smaller separate networks. Add to this the fact that no prior knowledge of geographical size, number of nodes, mobility or prior knowledge of each other exists and the security requirements become very difficult to implement. These devices must rely on intermediate stations to pass on messages reliably and securely ensuring that the intended recipient receive the message without it being read or modified by any devices other than the sender and receiver. For these reasons, the topology of ad hoc networks demand special security requirements, especially in the area of authentication and authorisation that other topologies can more easily implement (Akyildiz and Wang 2005).

Ad Hoc networks where nodes pass messages on can be used either with a central base station or as truly ad hoc networks where no centre of control or base station exists. One option for ad hoc networks is to extend the range of a network that utilises a base station to connect to an outside network such as the Internet. Whilst several of the standards are not suitable for these types of networks as they have very limited bandwidth and range intended for a localised personal area network (PAN), the HiperLAN and HiperMAN suites and both IEEE 802.11 (WiFi) and IEEE 802.16 (WiMax) suites of protocols have great potential in this area. A HiperMAN or WiMax base station serving several subscriber stations by linking them to the Internet via an ISP may use those subscriber stations as base stations for the HiperLAN or WiFi devices, thus utilising the high bandwidth of HiperMAN/WiMax to provide Internet access to several much lower bandwidth HiperLAN/WiFi devices. This may be useful in offices, homes and businesses such as restaurants and cafes or parks or other public areas that provide Internet access to anyone with a wireless device and browser. This type of topology provides the best of both worlds: high speed wireless Internet access whilst mobile, and a central base station that can provide security services such as encryption key management. Figure 2.2 shows an ad hoc network used to extend the range of a Metropolitan Area Network using mobile laptop computers.

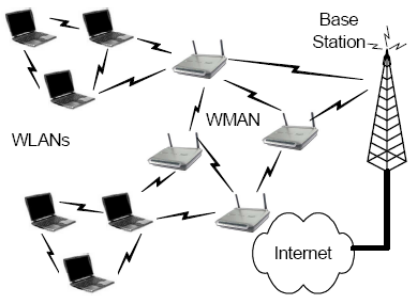


Figure 2.2: Ad Hoc network with base station (Zhou and Fang 2006).

Whilst the precise standard used for the network does not alter the method that can be used for encryption key management, it does give a guide as to the likely parameters for the radio communication that may be used. With the WiFi standards being the most popular standards worldwide, it is most likely that a user wishing to join a mobile ad hoc network will have a device equipped with IEEE 802.11 a, b, or g as these are the most popularly used standards. The limited range of approximately 300 metres outdoors provided by these standards, and with a bandwidth of no more than 54 Mbps for the g standard which is the most common, means that truly ad hoc networks have the ability to pass messages quickly over great distances provided a route from sender to receiver exists, even if only briefly.

This raises the probability that in an ad hoc network, communication between nodes will at times involve intermediate nodes acting as hopping points and passing on the messages. In theory, with enough nodes present, the network could have unlimited range, and every node in the network could contact every other node in the network. Figure 2.3 shows an ad hoc network where nodes are using intermediate nodes to act as hopping points for the messages. The circles represent half the radio ranges of the devices.

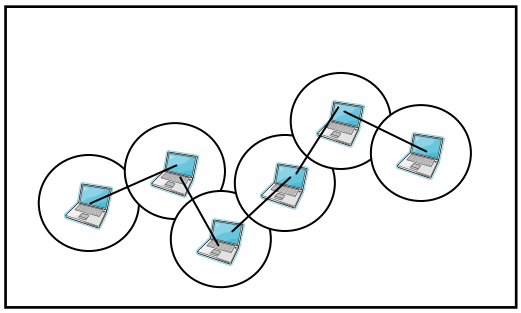


Figure 2.3: Ad Hoc network extending range.

This limited range works for efficiency in the network because nodes communicating in one area will not interfere with communications in another area if they are out of each others' range. Security therefore, relies on nodes acting responsibly and passing messages on without interfering with them. A robust security system needs to monitor the network, both by checking for malicious behaviour such as failing to forward messages or attempting to alter messages, or Byzantine behaviour where multiple nodes collude to disrupt the network. The special security needs created by truly ad hoc networks require customised security to ensure the six attributes required of a full security protocol, confidentiality, integrity, non-repudiation, authentication, authorisation and availability, can be met.

2.3 Wireless Networking Topologies

The equipment used for wireless networking includes wireless pcmcia cards, pci cards, access points, routers or gateways, bridges, and omni-directional and directional antennas. Whilst the maximum distance for reception is designed to be 300 metres with both 802.11b and 802.11g and 100 metres for 802.11a, experiments have shown that reception ranges of several kilometres are possible with high gain antennas and favourable conditions. There are two broad categories of wireless networks: infrastructure networks and ad hoc networks. The following sections examine each of these types of topology.

2.3.1 Infrastructure Mode

It was envisaged that the primary use of wireless network technology would be the ability to connect a wireless access point to a company Local Area Network (LAN), thus providing ease of joining the networks, especially for users who may only wish to

connect as a one-off event, or for mobility where the user's laptop computer can be used at different locations around the office. The access point is known as the Basic Service Set (BSS) with the wirelessly connected computers generally called stations (STA). If the LAN is connected to an access point, then the access point is referred to as an Extended Service Set (ESS). Figure 2.4 shows a wireless network operating in infrastructure mode.

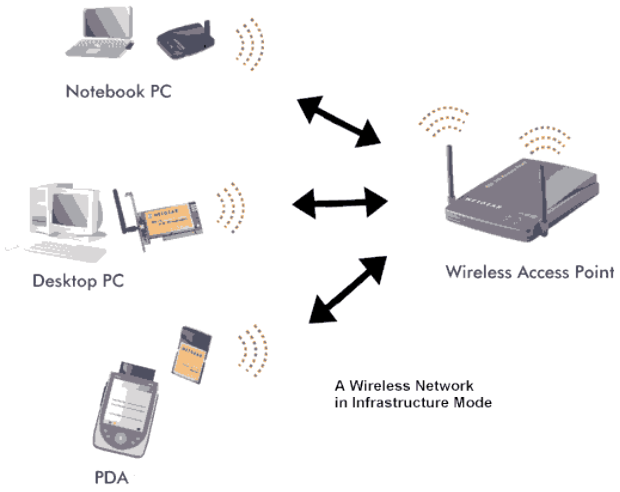


Figure 2.4: Wireless network in infrastructure mode (Netgear 2009).

This configuration allows employees or other authorised people to connect to the corporate LAN from anywhere within or near the offices, and communicate with the wired LAN or Internet. All that is needed to connect is a notebook computer or similar device with either a wireless pcmcia card, or built in wireless technology. Infrastructure mode need not necessarily include a connection from the access point to the LAN. The access point can be used as a standalone centre of control of the network, where security settings for the network can be configured and enforced over the entire network. This gives the network administrator an easier task for implementing security than if no centre of control were used. In this mode, wireless stations must contact other wireless stations through the access point, even if they are in direct range of each other. Security

can easily be enforced through Access Control Lists where the station ID (MAC address) is contained in a list within the access point. If the address is not on the list, the station is not permitted to join the network. Additionally, encryption keys can be entered into the access point and only stations using those same authorised encryption keys will have messages relayed. This centre of control makes implementing, removing and enforcing rules a relatively simple task.

If the access point is connected to the corporate LAN, then the ability to extend the LAN to any stations within range of the access point or points is very beneficial to the users. They can move around the office whilst maintaining connectivity even whilst at another part of the building that it may not be possible to supply wired access to. One substantial advantage of this implementation is that connections from the access point to servers can be made with wires rather than wirelessly. This adds a layer of security to the network making it physically more difficult for an attacker to intercept messages. The access point may be connected to a server whose sole task is to provide security to the network, thus removing much of the security messages from the wireless medium.

Whilst it is much simpler to implement and enforce security rules with infrastructure mode, considerable planning and building of the network prior to use is required. Additionally, users must always be within direct range of an access point to connect to the LAN or to each other, even when the stations may be physically nearby. Infrastructure mode is the most commonly implemented topology; however the IEEE standards also make provision for the connecting of stand-alone computers in an ad-hoc or peer to peer network. The following section describes the mobile ad hoc networking topology.

2.3.2 Ad Hoc Mode

In this mode, the connected computers are known as Independent Basic Service Sets (IBSS). The ability to connect multiple computers together without a central computer is an advantage for collaboration between people wishing to communicate without the constraint of having to be within range of a wireless access point. Figure 2.5 shows an ad hoc network where mobile devices are communicating directly with each other.

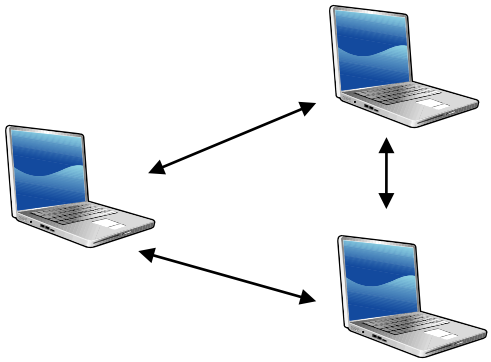


Figure 2.5: Basic ad hoc network.

This has proven especially useful for collaborative learning or sharing of data whilst out in the field away from the educational or corporate environment. However, whilst the vendor supplied software, or the more sophisticated operating systems, make this type of network relatively simple to set up, it does raise some serious security concerns, especially if there is sensitive data being sent from one computer to another without the wish for anyone else connected to the network to have the ability to view the data.

As with most wired network cards, wireless cards can operate in promiscuous mode allowing them to read all data transmitted on the network that they are connected to. However, of more concern is that many of wireless network cards have the ability to be put into Radio Frequency Monitor Mode (RFMon). This allows them to read all transmitted data within range, without being authorised on the network. The security

implications of this are huge. By running packet sniffing software all that is required for an attacker to capture wireless packets is for the attacker to be within range of the communicating wireless equipment. With a directional high gain antenna, this could be hundreds of metres away or more.

Awareness of the ease with which wireless network traffic can be captured has been an ongoing concern. The final World Wide Wardrive in 2004 found 228000 IEEE 802.11 access points with only 38% of those using the built in security protocols WEP or WPA encryption (WWWD4 2004). This figure is up from 30% in 2001, a relatively insignificant increase over four years showing that many network administrators and home users are still not taking the security threats to their wireless networks seriously. This trend appears to have continued, both because of a greater awareness by network administrators of the vulnerability of open wireless networks, and because the vendors are tending towards encryption being the default settings for devices with non-encryption needing a conscious choice not to implement.

This growing trend in wireless networking to eliminate the access points and instead use ad hoc networks with several mobile laptops interconnected is providing users with uses that were previously not considered. This mode is especially useful in meetings or public areas where several people wish to transfer data between computers or PDAs. The main constraint to this mode of operation is security concerns, with users concerned over transmitting sensitive information without a centre of control of the network. In the New Zealand survey conducted in 2004 (Houliston and Sarkar 2005) mentioned in Chapter 1, 28% of respondents who had wireless networks installed in their organisation were using them in ad hoc mode, with 83% using them in infrastructure mode. This suggests that business organisations are beginning to realise the potential of using these

types of networks in a peer to peer mode, something previously not seriously considered as a business use for wireless networks.

If used in a static manner, this topology is known as mesh networking. Mesh networks can allow one station to be connected to the Internet, and the other computers associated with the network to share the Internet connection, or they can be used purely for communication within the network between network members. Ad hoc routing protocols provide seamless handoffs from one device to another to maintain connectivity as devices move around the network. This ability to roam within the network footprint means there is a substantial increase in the usefulness of this type of network. If mobility is incorporated into a mesh networking architecture, then this topology is called a Mobile Ad Hoc network (MANET).

Whilst there are considerable benefits to eliminating the need for access points, with benefits there are usually drawbacks. The main concern with using intermediate devices to pass on communications is whether those devices can be trusted with what may be sensitive or critical data. Can these intermediate devices reliably pass on communications traffic, yet be prevented from reading, altering, incorrectly routing or deleting the traffic? From a management standpoint this can be a very difficult question to answer. Each wireless device in the network must be configured correctly to provide the best possible security, yet the very nature of these networks ensure that mobility and ease of joining or leaving the network is ensured. Implementing security in these ad hoc networks is difficult because of the lack of a centre of control of the network, but if the devices (nodes) can be pre-configured before network formation, then at least securely installing encryption keys or digital certificates can be assured. This reduces the problem to maintaining security and dealing with new devices that wish to join an

already configured ad hoc network. If the ad hoc network devices have no prior knowledge of each other and there is no pre-configuration or even prior knowledge of geographical size, number of nodes or mobility, then the security implementation is extremely challenging. This problem, where a truly ad hoc network requires formation and maintenance, has thus far not been satisfactorily solved and available protocols tend to deal with pre-configuring nodes offline. For a truly ad hoc network that is formed without pre-configuration, encryption key management is the cornerstone of satisfying the security requirements that the network will require. Whilst in any protocol that allows anybody to join the network, it is the device rather than the user of the device that is authenticated, this adds a significant level of confidence to the users. They can, at the very least, be sure that the device they are exchanging messages with is authorised on the network and can be identified if any malicious misbehaviour is detected. As a unique identification of every device is made, any continued misbehaviour will result in permanent exclusion from the network. Whilst a rogue user may attempt to masquerade as another person, often communication will be with people who are known to each other. This may be especially true in the educational or disaster relief applications. If necessary, testing the person desirable of communication with the other user may be carried out by questioning. If serious doubt of the other user's identity is aroused, then they can be permanently barred from any more attempted communication. If satisfied that they are who they purport to be, then confidential messages can quickly be passed between them.

The following section examines wireless networking protocols in more detail and their role in an ad hoc networking environment.

2.4 Wireless Protocols

Whilst many variations on connecting devices wirelessly have been produced, it is the suite of IEEE standards that have been most accepted worldwide. This section looks at the evolution of those suites and others. A technical comparison of the standards is made through the various protocols that have added bandwidth and security designed into the standard. This discussion sets the baseline for how wireless networking is most commonly implemented and how security of the networks is dealt with when using these standards in infrastructure mode with a central base station or access point. This is then concluded with an introduction to the ad hoc networking topology which is most likely to be implemented in an ad hoc network.

When first introduced, early wireless networking protocols identified problems with communication between computers using the radio spectrum. Firstly, they were susceptible to interference from other devices using the radio frequencies, and secondly there were security problems with broadcasting sensitive data over the airwaves. However, the benefits outweighed the drawbacks, and it was envisaged that most drawbacks would be overcome by advances in technology given time. Within a few years, serious development had progressed on wireless network technology to the point that in the early 1990s, standards were in development that would become ratified and available to the general public.

The following is a list of the most common technologies and their attributes. This gives a baseline for the likely technologies that will be used when networking in a MANET that this protocol may be utilised for.

- IEEE 802.11

In 1997 the Institute of Electrical and Electronics Engineers (IEEE) defined a standard for wireless networks called 802.11. This standard set down what was required for a wireless network to be compliant with their standard, including the data transfer rate which was 1 or 2 Mbps. This first standard used the 2.4GHz Industrial, Scientific and Medical (ISM) band that is available without a licence in most countries worldwide. Range was a maximum of 300 metres outdoors.

- IEEE 802.11a

This supplement to the 802.11 standard, produced by the IEEE, specifies the standards that an 802.11a wireless network must conform to for it to be classified officially as 802.11a. The 802.11a standard ratified by the IEEE on 15th September 1999, and this document describes the enhancements to the base standard that a vendor's product must conform. Most parts of the document are mandatory, but some do have an optional content left to the various vendors.

The 802.11a standard uses the 5 GHz band instead of the more freely available 2.4 GHz band and uses Orthogonal Frequency Division Multiplexing Physical (OFDM PHY) instead of Direct Sequence Spread Spectrum (DSSS) used by the 802.11b standard. OFDM is needed as an addition to the base 802.11 standard to allow the transmissions in the 5 GHz range. The OFDM system allows data payload transmissions in the 6, 9, 12, 18, 24, 36, 48 and 54 Mbps ranges. The standard specifies that support for transmitting and receiving at the data rates of 6, 12 and 24 Mbps is mandatory, but the other rates are left up to the vendor. Depending on the data transmission rate being used, various parameters may be specified for that rate. Table 2.1 shows the rate-dependent parameters for 802.11a.

Table 2.1: Rate-dependent parameters.

Data Rate Mbps	Modulation	Coding Rate (R)	Coded bits per subcarrier	Coded bits per OFDM symbol	Data bits per OFDM symbol
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16-QAM	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	16-QAM	2/3	6	288	192
54	16-QAM	3/4	6	288	216

When transmitted the encoded data bits are interleaved by a block interleaver, and then sent over the 5 GHz frequency to the receiver. The valid operating channels are shown in Table 2.2.

Table 2.2: Valid operating channels.

Regulatory Domain	Band (GHz)	Operating Channel Numbers	Channel Centre Frequencies (MHz)
United States	U-NII lower band (5.15-5.25)	36 40 44 48	5180 5200 5220 5240
United States	U-NII middle band (5.25-5.35)	52 56 60 64	5260 5280 5300 5320
United States	U-NII upper band (5.725-5.825)	149 153 157 161	5745 5765 5785 5805

- IEEE 802.11b

The 802.11b standard, like the 802.11a standard is an extension to the 802.11 standard published by the IEEE in 1997. It was also ratified by the IEEE on September 15th 1997. The main reason for the extension is to give greater data rates. The 802.11 standard calls for 1 or 2 Mbps, whilst 802.11b is required to support 1, 2, 5.5 and 11 Mbps. It also differs from the 802.11a standard in many other respects. Firstly, it uses the much more

publicly available 2.4 GHz Industrial, Scientific and Medical (ISM) band, and it's peak data rate is 11 Mbps as opposed to the 802.11a rate of 54 Mbps. Additionally, peak reception distance outdoors is 300 metres against 100 metres for the 802.11a standard. However, because it operates without a licence in most countries around the world, it has become much more popular and therefore is by far the most common standard that was implemented up to the later g standard being released. To operate at the 2.4 GHz range, the technical implementation needs to be quite different than the 802.11a standard. To ensure compatibility between devices, all stations must initially transmit control frames at one of the supported rates. This ensures all stations can read the control frames, even though vendor specific rates can be used between stations from the same manufacturer.

The 802.11b standard calls for a high rate extension to the 802.11 standard. This extension uses the Direct Sequence Spread Spectrum (DSSS) technique to provide the higher rates of 5.5 and 11 Mbps. To provide the higher rates, Complementary Code Keying (CCK) is employed as the modulation technique. This allows the higher rates to be achieved whilst using the same bandwidth as the lower rates.

The frequency range for 802.11b for the FCC (USA), (IC) Canada and (ETSI) Europe covers the spectrum from 2.4 to 2.4835 GHz. For Japan it is 2.471 to 2.497, France is 2.4465 to 2.4835, and Spain is 2.445 to 2.475. Four modulation formats and data rates are specified for the High Rate PHY. The basic rate is 1 Mbps DBPSK modulation, with the enhanced access rate of 2 Mbps using DQPSK. For the extended direct sequence specification, two additional rates are defined. The High Rate access rates are based on the CCK modulation scheme for 5.5 Mbps and 11 Mbps, with the optional PBCC mode provided for potentially enhanced performance.

The hop sequences for each of the specified geographical areas are defined with two sets. The first set uses non-overlapping frequency channels to allow the High Rate systems to minimise interference degradation. The synchronisation of frequency hopping is performed by the MAC sub layer management entity. The second set uses half-overlapping frequency channels with 10 MHz centre frequency spacing to enable interoperability with 1 Mbps and 2 Mbps frequency hopping (FH) systems. Figures 2.6 and 2.7 show the operating channels for North America.

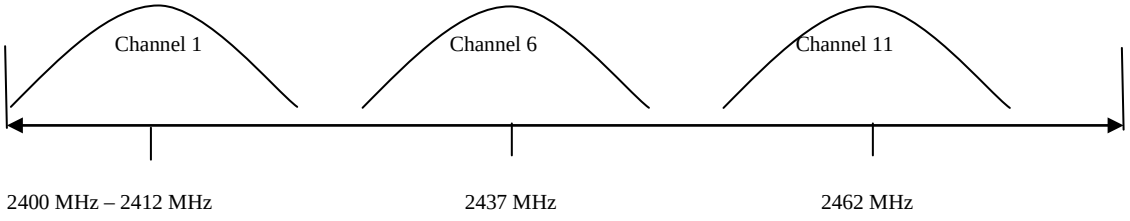


Figure 2.6: North American operating channels: non-overlapping (IEEE 1999).

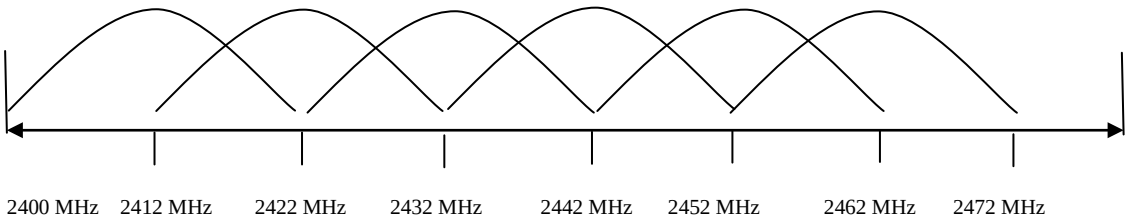


Figure 2.7: North American operating channels: overlapping (IEEE 1999).

The 802.11b standard is still widely used around the world even as the newer ‘g’ standard becomes more common. This is due partly to the use of the freely available 2.4GHz ISM band that it uses, and partly because of its greater range than 802.11a. However, it does suffer from two significant deficiencies when compared with 802.11a. Firstly, the data rate is significantly less, and secondly it has only three non-overlapping channels compared with twelve. What was called for was a new standard that allowed

for the use of 2.4GHz but also allowed for significantly increased data rates. This came in 2003 with the ratification of IEEE 802.11g. However, as it uses the 2.4GHz frequency, the standard still only permits three non-overlapping channels. If channels are chosen for access points within range of each other that are not at least four channels apart, some interference will occur, significantly reducing the data rates of both access points.

- IEEE 802.11g

The IEEE 802.11g standard is a later amendment to the original 802.11 standard, and was ratified on June 12th 2003. The 802.11g standard has become the most popular standard because it offers the best of 802.11a with the best of 802.11b. It operates in the ISM 2.4 GHz frequency spectrum, meaning that 802.11g equipment can be installed in most countries throughout the world without a licence. Additionally, it offers high data throughput of 54 Mbps, and a range of 300 metres outdoors. The standard specifications ensure that it is backwards compatible with 802.11b equipment, meaning that organisations can transition to 802.11g without having to eliminate all of their 802.11b equipment to do so. For the purposes of this review, I shall concentrate on the main differences with the 802.11b standard, as much of the 802.11g standard is similar to that of 802.11b.

The 802.11g standard implements the Extended Rate PHY (ERP) specification, to allow data rates greater that 11 Mbps. To accomplish the higher data rates, the use of Direct Sequence Spread Spectrum Orthogonal Frequency Division Multiplexing (DSSS-OFDM) is used.

The 2.4 GHz ISM band is a shared medium, and coexistence with other devices such as other types of STAs is an important issue for maintaining high performance. The ERP modulations (ERP-OFDM, ERP-PBCC, and DSSS-OFDM) have been designed to coexist with existing 802.11b STAs. This coexistence is achieved by several means, including virtual carrier sense Request to Send or Clear to Send or Clear to Send to Self (RTS/CTS or CTS-to-self), carrier sense and collision avoidance protocols, and MSDU fragmentation.

As can be seen, there are many changes to the basic 802.11 standard that are required in order to achieve the extended transmission rates. The 802.11g standard is designed so that it will operate in a fall-back mode to be compatible with the 802.11b equipment already deployed. To achieve this, the 802.11g equipment will operate at the slower speeds of 802.11b, and the transmission packets retain all necessary fields to ensure compatibility with the earlier standard. One area that has caused some problems with this intended compatibility is with the preamble field. The original 802.11 standard defined only a long preamble. The 802.11b standard gave an option for the implementation of a short preamble in 802.11b devices to increase transmission speed. However, 802.11g devices must be able to transmit and receive with both a long and short preamble field. If an 802.11g device is configured to use a short preamble, and an 802.11b device that is only equipped to use the long preamble is attempting to transmit and receive with it, the two devices will fail to communicate.

This is a fairly minor problem, and as older 802.11b equipment is replaced, 802.11g equipment has become the more accepted standard because of its added benefits.

- IEEE 802.11n

This standard, ratified in 2009 and provides much higher data rates by using multiple input multiple output (MIMO) technology. Whilst similar in operations to prior IEEE 802.11 standards, several enhancements have increased performance. Both the 2.4 GHz and 2.5 GHz frequencies are used and along with OFDM data rates are increased up to 300 Mbps. This makes it more suited to environments where higher data rates can be adequately handled on devices it may be used with such as broadband routers. Range of the transmissions is extended over previous protocols to 600 metres in optimal conditions.

- HiperLAN/1 (Hi Performance Radio LAN)

A standard developed in Europe and defined by the European Telecommunications Standards Institute (ETSI). The standard was developed in 1991 and approved in 1996. It uses the 5 GHz frequency which in many countries worldwide can be used without a radio licence provided that it is of low power and limited range. The maximum range is 50 metres. Modulation using Frequency Shift Keying (FSK) and Gaussian Minimum Shift Keying (GMSK) is used to achieve a data transfer rate of 10 Mbit/s.

- HiperLAN/2

Data throughput is raised to 54 Mbit/s by using more efficient modulation techniques such as Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK). Security is designed into the standard using symmetric encryption with the Data Encryption Standard (DES) and Triple DES.

- HiperMAN

This variation is designed to provide broadband services similar to the IEEE WiMax standard. It operates between 2 and 11 GHz but mainly uses the 3.5 GHz frequency. It is designed for non line-of-sight networks utilising quality of service such as video and voice communications. It has Point to Multi Point (PTMP) and Mesh Network configurations designed into the standard. Security is designed into the protocol in the form of DES and Triple DES. The range of HiperMAN signals is a maximum of 55 km and a data throughput of up to 70 Mbit/s. Whilst supporting mesh mode configurations, it is used with a central base station such as a television transmitter and the mesh nodes used to pass on the signals to extend the range. It is not designed for a mobile ad hoc network environment.

Of the two primary organisations that develop wireless networking standards, it is the IEEE's suite of protocols that have become the most accepted worldwide. As the 2.4 GHz spectrum is the most freely available to use without a licence, IEEE 802.11g has taken over as the most adopted standard with most new wirelessly equipped devices having this standard built into them.

One criticism of the early protocol development in both HiperLAN and IEEE 802.11 was the lack of security that was built into the standard at design time. This left security up to the individual vendors to provide, which many chose not to do. The consequence of this was a mistrust of wireless networking for sensitive data exchange because many networks were deployed without any security. Both ETSI and IEEE rectified this when the second variations of their protocols were released. The IEEE included a protocol called Wired Equivalent privacy (WEP) with its release of IEEE 802.11a and IEEE

802.11b. The following section looks at this security scheme and highlights some of the problems with it that have been rectified in the IEEE 802.11g release.

2.5 Wireless Network Security

The following section discusses the security issues specific to wireless networks and looks at the potential attacks on networks that result from weak security. It then looks at security protocols that have been developed for wireless networks to combat those potential attacks.

2.5.1 Wireless Network Attacks

Any wired network can be vulnerable to an attack. The Internet has allowed anyone with sufficient knowledge from almost anywhere to launch an attack on a network connected to the Internet. This includes intrusion into the network, monitoring of data within the network and denial of service attacks on web servers within the network. The advent of wireless networks has significantly increased the vulnerability of networks that incorporate wireless devices.

A host of free and commercially available tools have been produced, that are intended mainly to allow network administrators to test the security of their wireless networks. However, many of these tools also assist attackers to breach the security of insecure networks. As these tools are freely available to even the most unskilled network attacker, they can be simply and easily used for malicious purposes whenever an improperly secured network is detected. The following table lists some of the more common tools.

Table 2.3: Wireless network attack tools.

Name	Purpose	Effect
AirJack	Traffic injection & reception	Man in the Middle attack or fake packet injection
Airsnort	Crack WEP key	Capture packets and then crack the WEP key
ARPPoison	ARP Cache poisoning	DoS or MTM attack
coWPAtty	WPA dictionary attack	Password discovery
LinkFerret	Packet sniffer	Capture wireless packets
FakeAP	Generates fake access points	Confuse wardrivers or act as honey pot to steal passwords
Kismet	Wireless Access Point Logger	Wardriving
Netstumbler	Wireless Access Point Logger	Wardriving
Parasite	ARP Cache Poisoning	DoS and MTM attack
SMAC	Spoof MAC address	Bypass Access Control List
Void11	Network penetration by DoS	Authentication & Deauthentication flooding
Wellenreiter	Wireless Access Point Logger	Wardriving

The following is a description of several wireless network attacks that can be perpetrated against insecure networks.

2.5.1.1 Access Control List Avoidance

As with a wired network, a wireless network can be configured so that only those network card MAC (Media Access Control) addresses contained in an Access Control List (ACL) are permitted to authenticate to the access point. When the wireless client associates with the access point, a check is done on a list of authorised MAC addresses to see if the client's MAC is contained in the ACL. If not, authentication fails and the client is disassociated from the access point.

To authenticate to the targeted access point, the attacker must first gain knowledge of an authorised MAC address. To do this, the attacker needs to be within range of the access point or a currently authenticated client who is communicating with the target access point. Then, the attacker runs packet sniffing software with the wireless network card on his computer set to Radio Frequency Monitor Mode (RFMon). This mode allows the

wireless card to accept all transmissions within range without having to be authenticated on the network as it would have to be whilst running in promiscuous mode. The transmissions are observed by the attacker, with each transmission showing the sender and receiver's MAC address. The attacker merely needs to observe a transmission to the target access point to learn an authorised MAC address. Here, the attacker has two choices. He can either wait until the authorised client disassociates from the access point, or can send a spoofed disassociate message to the access point forcing the client to be disassociated. Then, using the spoofed MAC address of the authorised client, the attacker can quickly associate and authenticate to the access point. As the spoofed MAC address is contained in the ACL, the attacker will be permitted to join the network and will be given all the network privileges that the genuine client has.

2.5.1.2 Denial of Service (DoS)

There are several ways to carry out a denial of service attack against a wireless network. If the network is operating in infrastructure mode, then the target of the attack will be the wireless access point or a client computer communicating with the access point. The purpose of the DoS attack is to prevent clients from using the network, either by disassociating them from the network or overwhelming the network so that performance degrades significantly.

2.5.1.3 Denial of Service with Frequency Jamming

In May 2004, three Ph.D students at the Queensland University of Technology published an account of a DoS attack made possible by a minor modification to the hardware of a wireless device (Wullems C 2004) The Australian Computer Emergency Response Team (AusCert) immediately issued an advisory (AusCert 2004) describing

the method and potential problems caused by such an attack. The attack works by targeting the Clear Channel Assessment (CCM) protocol used by the 802.11 devices as an integral part of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. It is successful against 802.11 devices using Direct Sequence Spread Spectrum (DSSS) operating at 2.4GHz, and includes all 802.11b devices, and those 802.11g devices operating below 20 Mbps. It works by continually sending a signal over the channel, ensuring that any devices within range will wait until the channel is clear before transmitting. This means that any clients within range, or any clients within range of an affected access point will be denied transmission until the signal ceases. The attack will be effective as long as the transmission continues, and there is currently no known mitigation for the attack. Further, the source of the attack could be very difficult to find, as any 802.11 device including PDAs could be modified to carry out the attack.

2.5.1.4 Address Resolution Protocol (ARP) Cache Poisoning

This attack works only when an attacker is connected on the same local network as the target machines. Therefore it is only effective on networks that are connected using switches, hubs and bridges, but not routers (Fleck and Dimov 2001). Most wireless access points act as transparent MAC layer bridges that allow network traffic, including ARP packets to pass freely between the wired and wireless networks.

The APR protocol maps IP addresses to MAC addresses on local networks. If a client wishes to send a message to an IP address on the LAN, it sends a broadcast packet requesting the MAC address for the computer assigned that IP address. The ARP request asks, “Is your IP address x.x.x.x? The host with that IP address returns a packet with their MAC address. The message is sent to that MAC address, and the sender

stores the IP-to-MAC reference for future messages. The IP-to-MAC mappings are updated whenever an ARP request or reply is received.

The protocol was designed this way in an effort to minimise network traffic. However, a flaw in the design was discovered that allows an attack on the ARP mappings. As ARP is a stateless protocol, an ARP reply sent will be accepted whether or not a request for an IP-to-MAC address ARP message has been sent or not, and this new mapping of addresses will overwrite any older mapping (Whalen 2001). This flaw can be exploited by an attacker to poison the IP-to-MAC mapping cache, even in switches that have port security features that bind MAC addresses to individual ports. This is because no MAC addresses are actually changed, and the attack occurs at the IP layer which the switch does not monitor.

The ARP cache poisoning attack can target both wired and wireless machines on the network in several variations of a Man in the Middle style attack. If computer A is wishing to send a message to computer B, the attacker can poison the ARP cache of the switch connecting those computers so that the attacker receives and passes on all messages sent between the two computers. Provided a switch, hub or bridge is used, this attack is equally effective for a wired and wireless network, and with a wireless access point connected to the switch, it allows the attack to be performed by an attacker wirelessly. This means that the installation of an access point connected to the wired LAN creates vulnerabilities on the wired LAN as well as to other wirelessly connected computers.

This type of attack can be difficult to detect but there are several methods for mitigating the attack that can be used. Firstly, a Virtual Private Network can be used with all machines on the local network. This prevents an attacker from authenticating to the network, and therefore prevents them from sending spoofed ARP packets. Secondly, an arpwatch tool can be used which sends an email notification to the network administrator whenever an IP-to-MAC address is changed. Thirdly, an Intrusion Detection tool may be able to alert a network administrator to excessive numbers of ARP packets on the network. Whilst this is not a preventive solution, it may go some way to ensuring the ARP poisoning is at least detected. Finally, a switch can be configured to allow only manual changes to the ARP cache by the network administrator. ARP messages can then be received by the switch, but will be ignored preventing the cache from being poisoned.

2.5.1.5 Eavesdropping

This is the most basic form of attack, and is a passive attack where the attacker simply observes communications traffic between devices. If the communications are unencrypted, then the attacker merely needs to be within range of the communications, and armed with a wireless packet sniffer can observe and log the communications. If encryption is used for the communications, then the packets can be logged, but decryption will be difficult or impossible depending on the type of protocol being used for encryption.

2.5.1.6 Man in the Middle (MTM)

A MTM attack is performed by an attacker acting as a middle-man between two communicating wireless devices without their knowledge. This can be achieved in

several ways, but commonly it is done by spoofing a MAC address so that the communicators each think that they are directly communicating with each other. If the targets can be fooled, then even a secure connection using Secure Sockets Layer can be spoofed by the attacker setting up two secure connections, with the attacker as the initiator. By doing this, confidential information including passwords can be captured and read by the attacker. Alternatively, the attacker can provide false or misleading information to the subject without the subject being suspicious because they believe they are directly communicating with a trusted device such as an access point of other client. This may include fake web pages being displayed, fake emails being downloaded, or other fake confidential communications controlled by the attacker.

2.5.1.7 Replay Attack

In this type of attack, the attacker captures packets of data with wireless sniffer software and their wireless network card operating in promiscuous or RFMon mode. The attacker resends the packets over the wireless network so that the target computer receives the data as genuine. This can act as a DoS attack if a sustained resending of the data is undertaken. However, the more usual reason for the attack is to maliciously get the computer to redo something that was intended to be done only once. For example, an attack on a bank transmission may be to repeatedly send the instruction to transfer a sum of money to an account.

One of the criticisms of the IEEE WEP security protocol discussed later in the chapter was that there was no replay protection built into the protocol. With WiFi Protected Access (WPA), the later replacement for WEP also discussed later, as with WEP, the Initialisation Vector (IV) is used as a packet sequence number, but WPA encrypts the

IV as well with the data. The new Message Integrity Check (MIC) known as Michael is appended to the packet and ensures that the data, including the IV, have not been altered. This makes forged IVs impossible for an attacker. The Temporal Key Integrity Protocol (TKIP) ensures a new key is used for encryption of each packet of data sent, and therefore a new key is used for the encryption of the IV as well. Both the receiver and transmitter reset the packet sequence space to zero whenever new TKIP keys are set. The transmitter also increments the sequence number by one with each packet sent. If the receiver identifies a packet as out of sequence, that is with the IV the same or smaller than a previously received packet, then that packet is dropped and the receiver increments a replay counter by one in order to keep track of the number of suspicious packets. If more than one suspicious packet is received then counter measures are implemented to prevent the assumed attack from continuing (Walker 2002).

2.5.1.8 Spoofing

This is the act of a node masquerading as another node. The identification of the node, such as MAC or IP address may be spoofed by an attacker. This may allow an unauthorised node to join a network using an authorised ID, or may trick nodes into communicating confidential messages to the attacker as they believe they are communicating with an authorised node.

All of these attacks can be used against an ad hoc network, but with the exception of signal jamming, all the attacks can be mitigated by a robust encryption key management system. To be robust, the KMS must include a non-changeable identity method so that nodes cannot masquerade as other nodes, take on multiple identities, or rejoin the network once ejected and banned for misbehaviour.

2.5.1.9 WiFi Protected Access Password Discovery

With the introduction in 2003 of WiFi Protected Access, all known vulnerabilities in WEP were effectively fixed. Whilst only intended as an interim measure until WiFi Protected Access 2 was introduced in 2004, WPA has proven to be a popular and robust protocol that continues to be used widely. One advantage it has over WPA2 is that it was designed to be used in either of two modes, making it useful for home or small office use where expenditure on a dedicated Remote Authentication Dial-In User Server (RADIUS) is impractical.

In Enterprise Mode there is per-user authentication which is used in conjunction with the 802.1x security framework and an authentication server. The 802.1x protocol uses an Authentication, Authorisation and Accounting (AAA) server such as RADIUS, and TKIP and Michael offer per-packet key mixing, a message integrity check and re-keying for every packet sent. In Consumer Mode a pre-shared key (PSK) is used between the communication wireless devices, and TKIP and Michael are used for per-packet re-keying and message integrity checking.

Consumer mode consists of five essential tasks that must be performed for an effective communication.

1. the client associates with the access point
2. the client is authenticated and the Pair-Wise Master Key (PMK) is distributed
3. the Pair-Wise Transient Key (PTK) is created and installed
4. an integrity check is done on the messages
5. a successful wireless session begins using TKIP based on the PTK

In 2004, a paper was published that exposed a vulnerability in the consumer mode of WPA (Moskowitz 2004). The paper showed how the consumer mode of WPA is vulnerable to an offline dictionary attack because enough information is broadcast during the creation and verification of a session key for an attacker to discover the pre-shared key.

The following shows how the PMK and PTK are created (Takahashi 2004).

PMK = PBKDF2(passphrase, ssid, ssidLength, 4096, 256)

PTK = PRF-512(PMK, “Pairwise key expansion”, Min(AP_Mac, Client_Mac) || Max(AP_Mac, Client_Mac) || Min(ANonce, SNonce) || Max(ANonce, SNonce))

With this information available to an attacker using a packet sniffer and wireless card in RFMon mode, the attacker can capture an association session and run freely available software that will do a brute force attack on the pre-shared key password. The software was developed and posted on the Internet shortly after the possibility of an attack was discovered. Whilst the attack requires many calculations to check one possible password, it is theoretically possible to discover the password if less than twenty characters are used for the key. For passwords of more than twenty characters, the time involved to crack the password is such that it is not reasonably feasible to do so with today’s computers.

The following table lists some of the most common types of attacks and the areas that are targeted by those attacks.

Table 2.4: Network attacks.

	Confidentiality	Integrity	Non-Repudiation	Authentication	Authorisation	Availability
Eavesdropping	☑					
Jamming						☑
Denial of						☑
Replay			☑			☑
Man in the	☑	☑	☑	☑		
Spoofing	☑		☑	☑	☑	
Wi-Fi PA	☑	☑	☑	☑		

2.5.2 Standards Based Security Solutions

The special requirements for security of wireless networks pose challenges to implement because of the broadcast nature of the medium. This has resulted in security standards being developed alongside networking standards so that when the standards are ratified, security protocols are built into them. This section begins by examining security protocols that are standards based, followed by a discussion of non-standards based protocols that have been developed separately from the wireless protocols.

2.5.2.1 Wired Equivalent Privacy (WEP)

When the IEEE was developing the 802.11 ‘a’ and ‘b’ standards, it was decided that a security protocol should be developed and written into the standard. The security protocol is called Wired Equivalent Privacy (WEP) and is so called to describe the fact that it is designed to provide the same level of security for a wireless network as that of a wired network. It uses a well known and accepted symmetric stream cipher called RSA. A symmetric stream cipher means that encryption of the data is done at the transmitting end with the same key as that used to decrypt the data at the receiving end. This has advantages for networks where guests may wish to join the network as a one-off situation, in that the pre-shared key (PSK) can be entered into the guest computer allowing the guest to join the network without any configuration to the software on the

network side. As ease of use for both users and network administrators was identified as an important factor in wireless network adoption, this seemed like an ideal encryption scheme.

One problem identified with the IEEE 802.11 standards, is that it does not specify how key management should be done. Consequently, different vendors implement it in different ways. If WEP is used on a wireless network, the sending and receiving station use the same pre-shared secret key to encrypt data. This key is specified in the standard as 40 bits in length, with an additional 24 bits used as an Initialisation Vector (IV). Longer keys are optional, and many vendors allow the use of 104 bit keys, plus the 24 bit IV making a total key length of 128 bits. One flaw in the way the IV is used is that it is contained in the header of the packet, and only the data portion of the packet is sent encrypted. This means that anybody with a packet sniffer can read the unencrypted header of the packets and see the IV being used for that packet. Additionally, the way the IV is incremented is also vendor specific. Most wireless cards will set the IV to zero when they are initialised, and increment the IV by one for each packet. Some cards however, switch between two different IVs with every packet sent, and some use random IVs.

A wireless network can be configured to use WEP or not as the administrator chooses. If WEP is chosen, then a secret key *k* is chosen, and shared between the Access Point (AP) and the client station (STA). To compute the encrypted WEP frame, the plaintext frame data M is first concatenated with its non-cryptographic checksum *c(M),* to produce *M.c(M)* where '.' denotes concatenation. Then, a per-packet IV is prepended to the secret key to create a packet key, *IV.k.* The RC4 stream cipher is then initialised

using this packet key, and the output bytes of the cipher are XORed with the check summed plaintext to generate the ciphertext.

$$C = (M.c(M)) \text{ XOR } RC4(IV.k)$$

The actual WEP data is the per-packet IV prepended to this ciphertext, C. Figure 2.8 shows the WEP data unit.

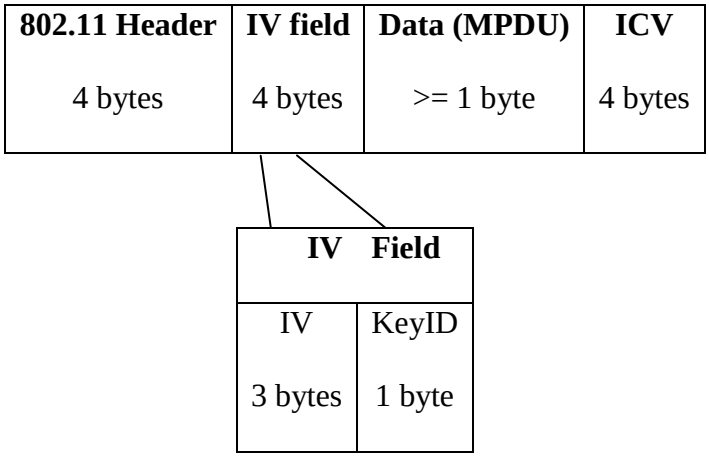


Figure 2.8: WEP data unit.

In 2001, a paper was published describing a theoretical attack on a WEP key (Fluhrer, Mantin and Shamir 2001), which identified 'weak' keys that were produced during the encryption process. These 'weak' keys could lead to the discovery of the WEP key if enough data could be captured that used the same key to encrypt it. The following year a successful attack on WEP encipherment was described (Stubblefield, Ioannidis and Rubin 2002). Figure 2.9 shows how the WEP encipherment works and how the IV forms part of the key.

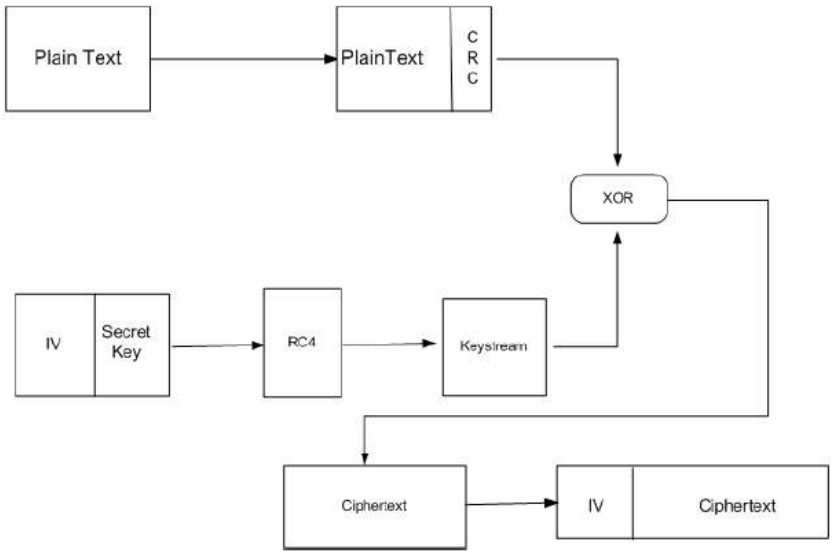


Figure 2.9: WEP encipherment (Vibhuti 2005).

The IV has a length of 24 bits, and is concatenated to the secret key (40 or 104 bits). This results in the seed for the WEP Pseudo Random Number Generator (PRNG) of 64 bits. The WEP PRNG is based on the RC4 algorithm. The output of the WEP PRNG is a key sequence of the same length as the text to be encrypted, given by the length of the plaintext and an Integrity Check Value (ICV) corresponding to a Cyclic Redundancy Check (CRC-32) of the plaintext. The key sequence and the plaintext are then Exclusive-Ored (XOR), and the resulting encrypted text is sent over the air, concatenated with the IV which is sent in the clear.

The IEEE 802.11 standard defines two different authentication schemes, Open System Authentication and Pre-shared Key Authentication. The Open System Authentication method is actually a NULL authentication, in that anyone wishing to join the network can do so. The Pre-Shared Key Authentication is a challenge-response type authentication scheme. Station A sends an authentication request and its station identifier to station B. B replies with an authentication message containing a random

challenge of 128 bits. Station A encrypts the message and sends it in encrypted form back to B. Station B, who has encrypted a copy of the message and is waiting for A's response, checks that the response from A matches the message that B encrypted. As the same key is used for encryption and decryption by the two communicating stations, if the messages match, then B must have the correct key. If so, B is then authenticated and permitted access to the network. Figure 2.10 shows station A requesting to communicate with station B.

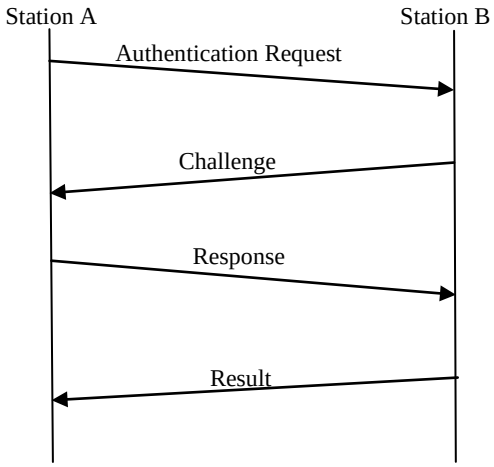


Figure 2.10: Authentication request and response.

The standard defines the use of up to four different pre-shared keys, shared by all the stations within a BSS. To allow for stations to move between different access points easily, many implementations use the same WEP key on all access points. This bad practice means that if the WEP key is compromised, the attacker has access to all stations and access points. The standard does consider the possibility of having per-user WEP keys, or better still per-MAC address WEP keys. However, only a very limited number of vendors have implemented this in their products, partly because of the management difficulties in administering this type of key distribution.

2.5.2.2 Weaknesses in WEP

Many flaws have been discovered in WEP framework. Notably, a paper published in 2001 (Borisov, Goldberg and Wagner 2001) described several serious weaknesses that could lead to key discovery, and these weaknesses are discussed below. WEP relies on a secret key *k* shared between the communicating parties to protect the body of the transmitted frame of data. The process of encryption is described as follows:

- Checksumming: An integrity checksum is computed on the message, *c(M).* The two are concatenated to obtain a plaintext *P = (M, c(M))*. Note that *c(M)* and therefore *P* do not depend on the key *k*.
- Encryption: The plaintext *P* is then encrypted using the RC4 algorithm. First, an Initialisation Vector (IV), *v*, is chosen automatically. The RC4 algorithm generates a keystream, a long series of pseudorandom bytes, as a function of the IV and *k*. This keystream is denoted by RC4*(v,k).* Then the plaintext is exclusive ored (XOR) denoted by $\oplus$ with the keystream to produce the ciphertext.

$$C = P \oplus RC4(v,k)$$

The receiver of the frame decrypts the message in the reverse of this process. The checksum is then checked against the decrypted plaintext by splitting it into the form (*M',c'*) recomputing the checksum *c*(*M'*) and checking that it matches the received checksum 'c'. This ensures that only frames with a valid checksum will be accepted by the receiver.

The WEP protocol was designed with three main security goals in mind.

1. Confidentiality.
2. Access control.
3. Data integrity.

In all three of these cases, the security of the protocol relies on the difficulty of discovering the secret key through a brute-force attack. There are two different versions of WEP. That stated in the IEEE 802.11 standard which specified a 40 bit key, and the optional version left to individual vendors which can have a 104 bit key. The reason for the standard specifying a 40 bit key was to comply with US government regulations which only allowed the export of encryption with a maximum key size of 40 bits. This however changed in January 2000, and longer keys are now permitted. A 40 bit key is not secure against a brute-force attack with the power of today's computers, but a 104 bit key is and will remain so for many years. However, as shall be described, an attacker would not need to mount a brute-force attack to discover the secret key, and therefore the length of the key makes no difference to the time taken to discover the key.

A well-known weakness of stream ciphers is that encrypting two messages with the same IV and key can reveal information about both messages (Borisov, Goldberg and Wagner 2001).

For example:

If $C1 = P1 \oplus RC4(v,k)$
And $C2 = P2 \oplus RC4(v,k)$
Then

$$C1 \oplus C2 = (P1 \oplus RC4(v,k)) (P2 \oplus RC4(v,k))$$
$$= P1 \oplus P2$$

In other words, XORing the two ciphertexts that have been encrypted with the same IV and key together causes the keystream to cancel out and the result is the XOR of the two plaintexts. This can lead to a number of different attacks. Firstly, if the plaintext of one of the messages is already known (known plaintext attack) the plaintext of the other can be easily discovered. Secondly, even if neither plaintext is known, an attacker can simply try different combinations until likely words are discovered. This can also be

done by looking for two English plaintexts that XOR to the given value. Thirdly, if there are *n* ciphertexts that all reuse the same keystream, then there is a problem known as *depth n.* Reading traffic in depth becomes easier as *n* increases.

To prevent these types of attack, WEP uses per-packet IVs to vary the keystream generation process for each packet that is transmitted. However, the IV is sent in each packet unencrypted, so an attacker easily knows if an IV and therefore the key is being reused. Unfortunately, key management is not specified in the 802.11 standard, and different vendors compute the IV for the packet in different ways. Some alternate between two IVs for every packet, some randomly choose an IV for every packet, but most set the IV to zero when the WLAN card is initialised, and increment the IV by one for every packet that is sent. If the card is removed and reinserted, or the computer is rebooted, the IV is set to zero again. Even if this third method is used, a busy WLAN operating at 11 Mbps and sending 1500 byte packets at an average of 5 Mbps, will exhaust all available IVs in about half a day, and the IVs and therefore keys will begin to be reused. Worse, an implementation that randomly selects IVs will on average incur a collision of IVs every 5000 packets, which is only a few minutes worth of transmission.

Another area that WEP promises to secure is that of message authentication. Unfortunately, this implementation also suffers from a bad design and therefore fails to achieve its promise. WEP uses a CRC-32 checksum which is included as part of the encrypted payload of the packet. Cyclic Redundancy Checksums (CRCs) are designed to detect random errors in the message, but they are not and were never designed to be resilient against a malicious attacker. This weakness is exacerbated by the fact that the message payload is encrypted using a stream cipher.

Further problems with the WEP algorithm are summarised below:

- The IV is part of the key. The input for the WEP PRNG is the WEP key concatenated with the IV, which is not a suitable way to use it.
- Small IV. The IV is given by only 24 bits, which implies a relatively small IV space, thus causing a high probability of key stream reusing in networks with high amount of traffic.
- Small WEP key. The 40 bits WEP key is too small, since it corresponds to only five characters. Considering that passwords are usually chosen as ASCII strings, dictionary attacks have high probability of success.
- No actual integrity protection. The ICV used in WEP is a simple CRC-32, which is not keyed and can thus be obtained by anyone.
- No replay protection.
- The used stream cipher and the CRC are linear.

This last point means that an attacker can alter the destination IP address to his own, thus causing the reply to the message to be sent to the attacker, where it can be read. Further, there is no mutual authentication defined in the standard. This could be done, with the challenge-response procedure being repeated in the other direction, but as it was not written into the standard, it is not implemented.

It was discovered that for a baseline attack against a 128 bit WEP key, between 4000 000 and 6000 000 packets were required to discover the key (Stubblefield, Ioannidis and Rubin 2002). This represents just a few hours of traffic on a busy network.

This was something of a disaster for wireless networking as security concerns had plagued the adoption of the IEEE 802.11 standards from the beginning. Security

concerns are seen as one of the main reasons for the slow adoption of wireless networks in organisations, and consistently rank as a major concern amongst organisations who have installed wireless technology.

With the practical implementation of the FMS (Fluhrer, Mantin and Shamir) attack on a wireless network, there was now proof that WEP was not secure, and there was no secondary security standard available that could be used. This led to a flurry of activity by vendors who tried to design their own propriety security implementations. The difficulty with vendors own standards is that they will only work with equipment from the particular vendor using that technique, and they are generally not thoroughly tested before being released. This further caused concern amongst potential users of wireless networks, and so a thoroughly tested and official standard was needed quickly. What followed was the design of a new security protocol called Wi-Fi Protected Access 2 (WPA 2). This protocol was seen as being a very secure protocol, and thorough testing was planned before it would be released. However, The FMS attack was publicised in 2002, and WPA 2 was not planned for official release until 2004. There was clearly a need for an interim security protocol that would be available as a software or firmware upgrade to all Wi-Fi vendors. In conjunction with many of the vendors, the Wi-Fi Alliance developed a scaled down version of the proposed new protocol that was introduced in October 2003. It was dubbed Wi-Fi Protected Access (WPA).

2.5.2.3 WiFi Protected Access (WPA)

Whilst still using the RC4 encryption algorithm, it had several new features that meant it was extremely secure. A pre-shared key between the sender and receiver was still used as a base key, but it used a new key scheduling algorithm called Temporal Key Integrity Protocol (TKIP) which used a hash function to choose a new key for each

packet of data sent. Further, rather than a packet sequence counter used as an Initialisation Vector and as part of the key pre-pended to the packet in clear text, a new 48 bit IV was used as an input to the mixing function to determine the per-packet key. Additionally, the use of a new Message Integrity Check (MIC) appended to the sent packet ensured that the data had not been modified by an attacker after it had been sent. With 128 bit encryption used in conjunction with the RC4 encryption algorithm, and a 48 bit Initialisation Vector and new MIC know as Michael, security concerns were seen as being resolved in the interim until WPA 2 arrived.

Task group I, or TGi was established by the IEEE 802.11 to resolve the security problems with WEP. The problems with WEP are so serious and so numerous, that an entirely new protocol is needed. However, millions of WEP-based devices are in use around the world, and the industry has an obligation to fix the security defects that are installed on these machines if possible. The 802.11 equipment is comprised of both hardware and software, and it is not cost effective to install new hardware chips in these devices. This implies that any upgrade to WEP will of necessity be able to operate on these devices with a software upgrade. Access points are usually equipped with the cheapest microprocessor available, typically an i486, ARM7 or PowerPC running at 40 or even 24MHz. With the WLAN traffic using up to 90% of the available CPU cycles, this leaves very few spare cycles for new functions. Cryptographic functions tend to be very CPU-hungry, and in order to make it viable to use already deployed 802.11 equipment with a new security protocol, the protocol must by necessity remain fairly basic. Whist the new 802.11i protocol is designed to run on new, more powerful hardware, the interim measure is designed to be a software upgrade to existing equipment.

The Temporal Key Integrity Protocol (TKIP) is a suite of algorithms wrapping WEP, to achieve the best security that can be obtained with the design constraints that are in force. TKIP adds four new algorithms to WEP:

1. A cryptographic Message Integrity Check (MIC) called Michael to defeat forgeries.
2. A new IV sequencing discipline, to prevent replay attacks.
3. A per-packet key mixing function, to de-correlate the public IVs from weak keys.
4. A rekeying mechanism, to provide fresh encryption and integrity keys.

2.5.2.4 Michael

The MIC has three parts made up of a secret key *k*, a tagging function and a verification predicate. The tagging function takes the key *k* and the message M as its inputs, and generates a tag T called the message integrity code, as its output. M is protected from forgery by a protocol that has the sender compute and tag T and send it with the message M. A check for forgery is made by the receiver inputting *k*, M and T into a verification predicate. If it computes to what is expected then T evaluates to TRUE or to FALSE otherwise. The MIC is considered secure if it is not feasible for an attacker to select the correct tag for some new M without first knowing the key *k*. The Michael key is 64 bits, represented as two 32 bit little-endian words (*k0,k1)*. On the 802.11b equipment, an attacker is able to create a forgery against Michael in approximately 2 seconds worth of messages which calculates to about two minutes. The level of protection afforded by this process is much too weak to provide security by itself, so TKIP assists Michael with further measures. If a TKIP implementation detects two failed forgeries in one second then the message process considers that an attack is underway against it. The process is then for the station to delete its keys, disassociate,

wait for one minute and then re-associate. This process will interrupt communications briefly but is necessary to ensure a successful attack is thwarted quickly.

2.5.2.5 Defeating Replays: IV Sequence Enforcement

The MIC is not capable of detecting a replayed packet attack. To do so, the WEP IV field is used by TKIP as a packet sequence number. The sender and receiver initialise the packet sequence space to zero whenever the new TKIP keys are set. The sender will increase the sequence number of each packet by one. TKIP views an out-of-sequence packet as one that has the same or a smaller sequence IV than a MPDU that has previously been received as correct. If an out-of-sequence packet is received, the packet is discarded and the receivers replay counter is incremented by one. One limitation of the TKIP replay detection is that it is not designed to work with Quality of Service (QoS) enhancements introduced by the IEEE 802.11 Task Group e in 2005 as 802.11e.

2.5.2.6 Defeating Weak Key Attacks: Key Mixing

As WEP seriously misused the RC4 algorithm, the TKIP per-packet key construction is implemented to correct this flaw. Called the TKIP mixing function, the per-packet key construction, replaces a temporal key for the WEP base key, and calculates the WEP per-packet key. There are 2 phases to the key mixing function. Phase 1 eliminates the same key from use by all links, while phase 2 de-correlates the public IV from the known per-packet key. Phase 1 combines the 802 MAC address of the local wireless device and the temporal key by iteratively XORing each of their bytes. This is used to index into an S-box1, to produce an intermediate key. Mixing the local MAC address into the temporal key way causes different stations and access points to generate different intermediate keys whether or not they start with the same temporal key. This

type of situation is commonly found in ad-hoc networks and so this type of key construction fits well with these types of networks. This key generation forces the stream of generated per-packet encryption keys to be different at every station. The Phase 1 intermediate key is constructed only when the temporal key has been updated. To improve efficiency, most implementations cache this value for better performance. Phase 2 encrypts the packet sequence number with a small cipher. The intermediate key is utilized to produce an encrypted a 128-bit per-packet key. The leaves the first 3 bytes of Phase 2 output to correspond exactly with the WEP IV, and the last 13 bytes to correspond to the WEP base key. This is done as the existing WEP hardware expects to concatenate a base key to an IV to create the per-packet key. The second design goal is achieved by this process by making it very difficult for an attacker to match IVs and per-packet keys. The security analysis of the key mixing function is not as high as that created by Michael but there is consensus by the cryptographic community that the security goals have been achieved.

2.5.2.7 Defeating Key Collision Attacks: Rekeying

TGi rekey architecture is dependent upon the hierarchy of at least three key types. These are the temporal keys, key encryption keys and the master keys. TKIP uses two temporal key types: a 128 bit encryption key and a 64 bit key for data integrity. TKIP utilises two different temporal keys in each direction of a security association. Therefore, each association has two pairs of keys making a total of four temporal keys. A WEP KeyID is used by TKIP to identify the set of keys with a two-bit identifier. When an association is first established the first set of temporal keys is bound to one of these two WEP keyids. When new keys are created, the association alternates between the two keyids, with the new set of temporal keys being associated with the least-recently bound keyid. When the keys are bound with the new pair of temporal keys,

TKIP will still accept packets with the old key id and its temporal keys. However, they will now only transmit under the new keyid and its keys. New temporal keys are required to have not been used before, at least in a previous session with the same or any other peer. This is required even after a reboot, and there must be no algorithmic relationship amongst the set of keys. New temporal keys require careful implementation for this reason. TGi achieves this aim by using special rekey messages. The stations and the access point derive the new set of temporal keys from this special rekey message. This rekey exchange must also be secure to prevent an attacker compromising the keys during the message exchange. Temporal keys are protected by the next level of the hierarchy. This is the encryption keys that protect the temporal keys. Two key encryption keys are utlised. There is one to encrypt the new keying material with the second to protect the rekey messages from forgery. As the station and access point must establish this new set at each association, it is required that these keys are similar to those for temporal keys.

All these different aspects of TKIP work together with WEP to provide greatly enhanced security. When the station transmits an MSDU using TKIP, the process uses the temporal Michael key to compute the MIC of the source and destination MAC addresses and the MSDU data. TKIP adds the MIC to the data field which adds 8 bytes to the data payload. The process then fragments the MSDU into several MPDUs as required. Once this has been done, each packet fragment is given a packet sequence number and the key mixing function is used to create a per-packet encryption key for each packet. This is represented as a WEP IV and a base key. From here on, the steps are identical to WEP which is most likely done in hardware by the station and access point. The Integrity Check Value (ICV) field is appended to the data field ICV of each

of the fragments. The encryption now consumes the IV and base key and encrypts the data field. This includes the MIC and ICV. The IV and the key id of the set of temporal keys are also encrypted and added into the WEP IV field. This completes the encapsulation process and the MPDU in its encrypted form is now ready to send.

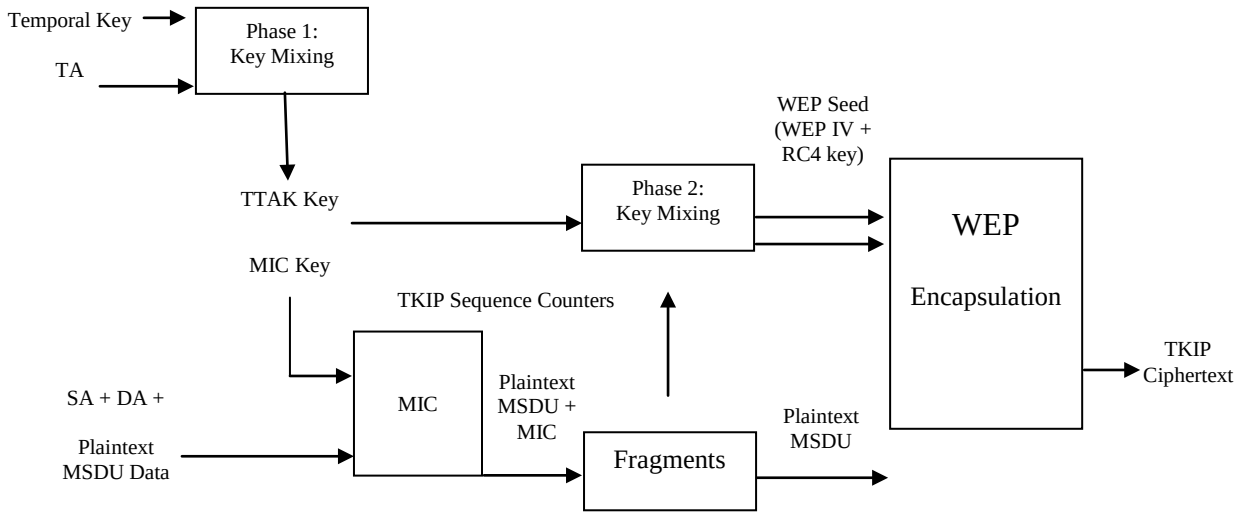


Figure 2.11: TKIP encapsulation (Walker 2002).

2.5.2.8 WiFi Protected Access 2 (WPA 2)

On 24th June 2004, the IEEE ratified WPA 2, officially now known as the IEEE 802.11i standard. With the use of the Advanced Encryption Standard (AES), which was adopted as the official encryption algorithm by the United States Government in October 2000, and a host of other changes, there seemed to finally be a secure protocol for wireless transmissions that would remain secure for many years. However, with a more complex protocol came some drawbacks.

Firstly, AES needs a fairly powerful processor to encrypt or decrypt the packets as they are sent or arrive. As wireless networks may be implemented on older computers, this may not always be possible. Secondly, 802.11i uses another of the IEEE's standards

known as 802.11x. This standard, known as Extensible Authentication Protocol (EAP) requires the use of separate server to provide authentication. The infrastructure needed is a wireless access point wired to an Authentication, Authorisation and Accounting (AAA) server such as a Remote Access Dial In User Service (RADIUS) using one of several protocols such as token cards, Kerberos, one-time passwords, certificates, and public key authentication. The following is the steps taken to provide authentication and authorisation with EAP.

- The client sends an EAP-start message.
- The access point replies with an EAP-request identity message.
- The client sends an EAP-response packet containing the identity to the authentication server.
- The authentication server uses a specific authentication algorithm to verify the client's identity.
- The authentication server will either send an accept or reject message to the access point.

The access point sends an EAP-success or reject packet to the client. If the authentication server accepts the request, then the access point will change the client's port to an authorised state and the client is now authorised on the network. The basic 802.1X protocol enforces effective authentication f whether or not encryption is implemented. If dynamic key exchange is utilised, then the 802.1X authentication server is permitted to return session keys to the access point along with the accept message. The access point uses the session keys to build, sign and encrypt an EAP key message that is then returned to the client immediately after sending the success

message. The client uses the key message to define applicable encryption keys. (Modified from an article by Jim Geier (2002))

The IEEE 802.1X port-based network access control standard is one method of increasing security in IEEE 802 networks. A framework is used for centralised authentication, access control and key exchange. However, it fails to specify any security mechanism that should be implemented to achieve this goal. If the connection between two devices is a point-to-point architecture, then the 802.1X protocol is suitable. This is the case in a wireless LAN environment. If this is the case, the access point takes responsibility for enforcing authentication and access control and will allow or refuse access to the network to other devices. An authentication server, separate to the access point, is used perform the task of authenticating devices. The access point and server will form a security association and often use the Authentication, Authorisation and Accounting (AAA) protocol. This protocol encapsulates messages from the access point and relay them to a Remote Authentication Dial-In User Service RADIUS server. Figure 2.13 shows the IEEE802.1X architecture utilising a AAA server.

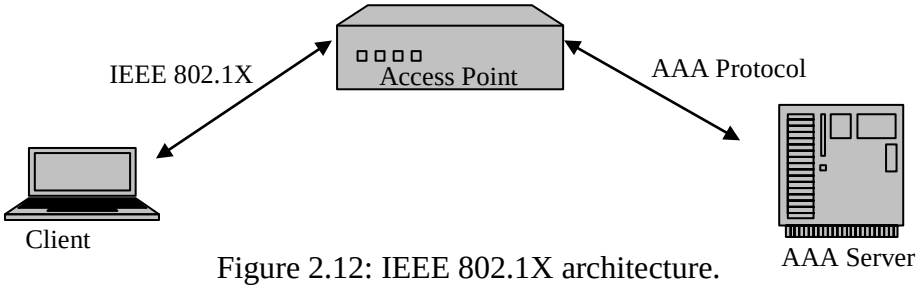


Figure 2.12: IEEE 802.1X architecture.

The IEEE does not mandate any authentication mechanism, but rather it defines an encapsulation technique known as Extensible Authentication Protocol over Local Area Network (EAPOL). This protocol is aimed at enhancing the Point to Point Protocol

(PPP) with additional security. EAPOL defines a way to carry EAP packets in the frames of a LAN, which means that every authentication mechanism defined for EAP is available for LANs deploying IEEE 802.1X for authentication and authorisation. Briefly, when a client station (STA) wishes to associate with an access point (AP) and therefore join the WLAN, the AP asks the STA to authenticate itself. The message from the STA is forward by AAA protocol to the AAA server, and messages flow through the AP between the STA and the AAA server. By using this method, much of the authentication process is removed from the wireless side of the LAN to the wired side, where it is much more secure from attack.

Whilst the 802.1X framework is an elegant solution to authentication and authorisation for organisations with the infrastructure in place, it can be expensive and difficult to implement in less formal environments such as the home or small office. Additionally, it requires the use of an access point connected to a server to provide the authentication. Whilst this can be done for a wireless network operating in infrastructure mode, it cannot be done for wireless networks operating in ad hoc mode. This has meant that 802.11i has been adopted in many larger organisations, but WPA and even WEP are still widely used in smaller environments, and in ad hoc mobile networks. Table 2.5 compares the properties of WEP, WPA and WPA2.

Table 2.5: Standards based encryption methods.

	WEP	**WPA**	**WPA2**
Cipher	RC4	RC4	AES
Key Size	40 bits	128 bits encryption 64 bits authentication	128 bits
Key Life	24 bit IV	48 bit IV	48 bit IV
Packet Key	Concatenated	Mixing Function	Not Needed
Data Integrity	CRC-32	Michael	CCM
Header Integrity	None	Michael	CCM
Replay Attack	None	IV Sequence	IV Sequence
Key Management	None	EAP-based	EAP-based

2.5.3 Non-Standards Based Security Solutions

For non-standards based security solutions, the choices are somewhat greater as they are not tied directly to WLAN security. The first choice to be made for a non-standards based solution is as to where the security protection will occur. Three possibilities are considered:

- At the Access Point.
- Using a Network Access Server (NAS) separating the WLAN from the infrastructure network.
- Using remote enforcement of security such as a Virtual Private Network (VPN) gateway.

The choice will depend on exactly what is needed to be achieved. This may involve confidentiality, integrity, authentication and access control. Perhaps the most typical solution for a corporate environment will be a VPN. Whilst not WLAN specific, it can be an extremely good solution as it may fit in well with the company's security infrastructure. The VPN establishes an encrypted tunnel between the mobile terminal and an endpoint within the company's network. The terminal and VPN gateway share a common secret, or the mobile terminal may have a digital certificate issued by the company's Certification Authority (CA). After an initial authentication and negotiation phase in which a session key is exchanged, the mobile terminal and the company's VPN gateway start to transmit data encrypted under the exchanged session keys. The VPN may be based on Internet Protocol Security (IPSec), Layer 2 Tunnelling Protocol (L2TP), Point to Point Tunnelling Protocol (PPTP), and Layer 2 Forwarding (L2F).

A further choice may be to use Kerberos. This involves four components, the Kerberos Client, the Kerberos Server, the Authentication Server (AS), and the Key Granting

Server (KGS). The AS and KGS are usually concentrated in the Key Distribution Centre (KDC). Every Kerberos Client and every Kerberos Server share a secret key with the KDC. The Kerberos Client makes contact with the AS, asking for a Ticket Granting Ticket (TGT), which can later on be presented to the KGS, asking for a ticket for a specific server whose services the client wishes to access. All replies sent to the Kerberos Client are encrypted with the client's secret key. When the client wishes to make use of the services for which it asked for a ticket, it will contact that server and present the ticket, encrypted with the server's secret key, and containing the name of the client and the session key. The server decrypts the ticket and is then able to mutually authenticate with the client. If successful, the client then has use of the server's services. All communication is integrity and confidentiality protected, and replay attacks are avoided by using timestamps. When deployed on a WLAN, the mobile terminal plays the role of the Kerberos Client, the access point plays the part of the Kerberos Server, and the KDC is placed somewhere on the infrastructure network. Whilst the Kerberos security architecture may offer a very robust authentication mechanism, it is a very complex and expensive solution to implement. For this reason, it is really only suited to large corporate environments, where the protection of data and the heavy usage of the WLAN demand a robust and secure solution.

Further non-standards based solutions may involve an Access Control List based on the clients MAC address or a firewall separating the WLAN from the wired LAN. A solution may also be to link the firewall to the Dynamic Host Configuration Protocol (DHCP) server, so that control can be maintained over the MAC address and IP address of clients wishing to join the network.

2.6 Summary

Whatever vulnerabilities exist in a wireless network can be largely overcome by robust encryption and an effective encryption key management system suitable for the topology and application it is used for. With encryption of messages coupled with other security attributes added to the data the only effective method of ensuring all six requirements for security can be met in the network, the management of encryption keys forms the very heart of the security implementation.

The following chapter examines cryptography and key management within a wireless network environment, from the early, basic security protocols to the more advanced methods available today.

Chapter III

CRYPTOGRAPHY IN MOBILE AD HOC NETWORKS

3.1 Introduction

This chapter introduces the concepts of encryption and describes mobile ad hoc networking topologies. Encryption requires two communicating parties to possess two artifacts; firstly an encryption algorithm and secondly an encryption key. Encryption algorithms are generally publicly known and available for scrutiny, meaning a non-secret agreement can be made as to the algorithm to use. This leaves the encryption key which either must remain secret when exchanged or a method must be used where publicly disclosing the key does not allow an attacker to decrypt messages that are encrypted with the key. A full encryption key management system (KMS) will perform more than mere key exchange. It will allow for creation, distribution, updating if necessary and destruction of the key. This chapter begins by discussing basic encryption techniques through key exchange, symmetric key encryption and asymmetric key encryption.

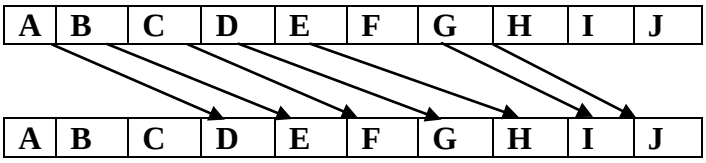
Following encryption techniques is an introduction to Threshold cryptography and identity-based cryptography. These types of cryptography techniques involve splitting the key management process into parts and assigning several contributors to create their own part of the key, certificate or signature. Tying the nodes identity to the certificate allows nodes to be positively and permanently identified and is useful for security in a network such as a MANET. The highly dynamic nature of these types of networks presents special challenges to implementing effective and efficient KMS systems. This chapter concludes with a summary of encryption key management choices and which of these may be best for a highly dynamic ad hoc network.

3.2 Background

The problem of two parties communicating privately when others may read the message is one that extends back in time many centuries. Two thousand years ago Julius Caesar used a very rudimentary method of encryption known as a fixed shift cipher, commonly called the Caesar cipher. The technique is a form of substitution cipher where each letter is replaced by another letter to render the message unreadable. The plain text letters are shifted to the right of the alphabet by a fixed number of letters with the alphabet continuing in a circular fashion so that once ‘z’ is reached, the next letter will be ‘a’. The process is described below.

Encryption:

1: Select the number of letters to shift to the right (3)



2: Write the message in plain text.

H	E	A	D

3: Rewrite the message by shifting all letters to the right using the key (3).

K	H	D	G

4: Send the encrypted message.

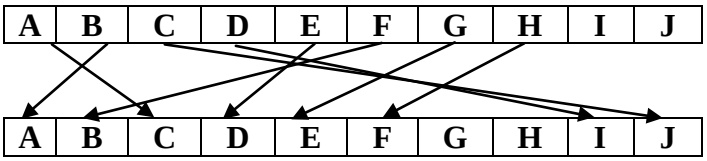
Decryption:

1: Take the encrypted message and shift all letters to the left by the key (3).

H	E	A	D

2: Read the plaintext message

Whilst a very basic technique, if intercepted by an attacker the message may remain secure if the algorithm or key is not known. However, two problems with the technique exist. Firstly, the sender and receiver must know the algorithm and the key, and secondly discovering the technique by intuition and trial and error is fairly trivial. A more complex technique is to use a substitution cipher but rather than having a fixed shift for the letters of the alphabet, a random shift is used.



Whilst more difficult to decipher the message, a more complex problem exists. Not only does the receiver of the message require the algorithm, but each letter of the alphabet requires its corresponding shifted letter to be known. Rather than a single key, there are in effect 26 keys that are required. This technique is known as the ‘one time pad’. Both the sender and receiver have a pad of several pages. Each page shows the alphabet and the substituted letters. The sender and receiver are synchronised so that they know which page of the pad is used at a particular time. Whilst considerably more secure than the original Caesar cipher, it is more complex and requires more information to be securely exchanged. An attack is still possible by means of counting the letters that appear in the message and finding the most common ones. Then, the letters are exchanged to see if words are made. For example, ‘e’ is the most commonly used letter in the written English language, so that if ‘v’ appears the most number of times in the message it is exchanged with ‘e’ to see if that makes a sense. If not, the next most common letter is tried and so on. Some letters may occur twice together giving a clue that they may be vowels or commonly repeated consonants. By trial and error, eventually the message is likely to be decrypted.

It can be seen from these two examples that two features of encryption present themselves. Firstly, the more security a technique provides the more complex the method must be. Therefore security is a trade-off between ease of use and complexity. Secondly, whilst the techniques serve the purpose, prior secure exchange of information (key or pad) must have taken place before the technique can be used. These are common features of encryption and key management protocols that still exist today.

3.3 Encryption

It may be possible to secure a computing device from an attacker by means of physical security, logon and password or encryption of files whilst on the hard drive. However, once the data leaves the device it is open to interception. Whether data travels along wires or through the airwaves, inevitably a determined attacker can often intercept those communications and read the data. With wired communication, the ability to secure the wires as well as the communicating devices away from a potential attacker goes at least part way to providing basic security to the communications. However, with wireless communications the attacker only needs to be within distance of the radio range of the transmitting device to receive the data. Therefore, with data so easily intercepted and read, encrypting the data so that what is captured appears meaningless to all but the intended recipients is the major security technique used for secure communications. Encryption techniques involve encryption key management to provide the encryption keys to the senders and receivers of messages. A technique for 2 parties to securely exchange keys without any prior contact was devised in the 1970s and is still used today in various forms (Diffie and Hellman 1976). An examination of the technique is warranted to set the baseline for key exchange techniques.

In 1976 W.Diffie and M.Hellman wrote a paper describing a technique for two parties to communicate securely without any prior configuration (Diffie and Hellman 1976). The technique, known as Diffie-Hellman key exchange involves two parties, A and B, who wish to communicate securely even though their communication may be monitored by unauthorised parties. To do so, public key infrastructure is employed and the focus of the technique is how to exchange keys openly yet afterwards communicate privately. The demonstration involves two parties who traditionally are called Alice and Bob.

Step 1: Firstly, they openly agree on a very large prime number p, usually at least 1024 bits. They also agree on a generator g.

Step 2: Alice chooses a large random integer that is less than the prime number, $X_A < p$ and keeps it secret. Bob does likewise $X_B < p$, so they both now have their private keys.

Step 3: Alice computes her public key $Y_A \equiv g^{X}{}_A \pmod p$ and sends it openly to Bob. Bob computes his public key $Y_B \equiv g^{X}{}_B \pmod p$ and sends it openly to Alice.

Step 4: Alice computes $Z_A \equiv Y_B{}^{x}{}_A \pmod p$ and Bob computes $Z_B \equiv Y_A{}^{x}{}_B \pmod p$. Here $Z_A < p$, $Z_B < p$. But $Z_A = Z_B$, since $Z_A \equiv Y_B{}^{x}{}_A \equiv (g^{x}{}_B)^{x}{}_A = g^{(x}{}_A{}^{x}{}_B{}^{)} \pmod p$ and similarly $Z_B \equiv (g^{x}{}_A)^{x}{}_B = g^{(x}{}_A{}^{x}{}_B{}^{)} \pmod p$. So this value is then their shared secret key.

The major advantage of such a simple scheme is that it allows a secure and rapid exchange of encryption keys. The main disadvantage is that each key can only be used for communicating between two parties. Should a third party, C, wish to communicate with A and B, then either they must all share the same key in which case C must perform a key exchange with either A or B, or the process must be repeated from A to C and again from B to C with 2 unique keys. If another party wishes to join in the communications and each pair requires unique keys, then the process must be done

again leading to a very time intensive problem of key exchange with nodes joining the network.

Whilst this is a simple method for key exchange for a very small number of nodes, it scales very poorly and is therefore not suitable by itself for larger networks. This scheme falls within the category of individual key exchange and is useful for both symmetric and asymmetric key exchange.

Encryption of messages plays such a vital part in the security of both wired and wireless networks that the choice of security may be integral to the usefulness of the network. Security is generally considered a tradeoff between ease of use and level of security. A very secure system may be very difficult to use and a very simple system to use may be very insecure. Additionally, the use the network will be put to may dictate what, if any, security will be required. Many open wireless networks at cafés or parks provide no security at all as their purpose is to allow anyone access to the network easily and quickly. A military network would expect high management and high security at the expense of ease of use. With modern technologies much of the security of the systems is transparent to the users, so to some extent high security can still be available with relatively simple to use networks. For a wireless ad hoc network, the very ad hoc nature indicates rapid deployment and ease of joining for those who are authorised. Encryption may be required for some messages but not for others and various levels of encryption may be required depending on the message and the path that must be taken for the message. With these choices available, the type of encryption used is an important choice for the protocol designer.

3.3.1 Symmetric Encryption

Symmetric encryption involves both the sender and receiver sharing the same key for encryption and decryption of a message. Generally, one key is used by both parties but it is possible to have the parties share two keys so that if one key is compromised by an attacker, only one side of the conversation can be decrypted. To communicate secretly, the two parties to the conversation must share the same encryption key and encryption algorithm and must ensure that the key is kept secret. One possible process of establishing a symmetric key before beginning message passing is shown in Figure 3.1.

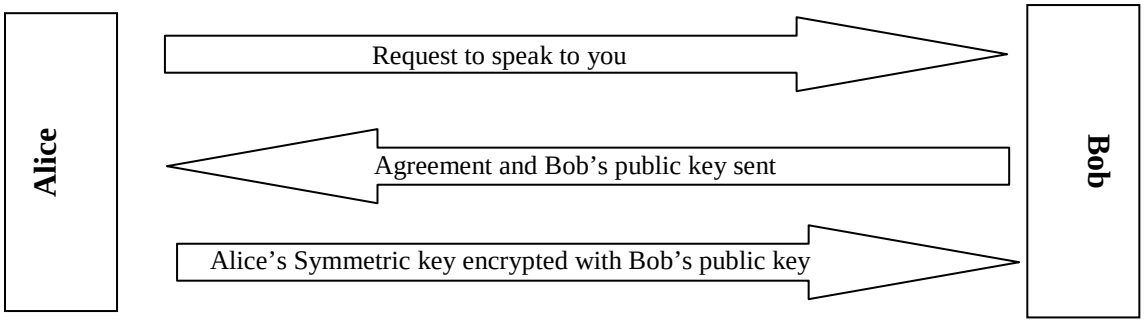


Figure 3.1 Direct symmetric key exchange using Diffie-Hellman key exchange.

This process involves only two parties directly creating and exchanging a symmetric encryption key. If a centre of control for the process is required, then a key distribution centre (KDC) can be employed as a trusted third party (TTP). The KDC takes over responsibility for creating and distributing the keys, and if necessary revoking a key and informing the network members of the revocation. The two parties must first possess a password (K_A for Alice and K_B for Bob). The KDC knows the passwords of all the network nodes and so uses Alice's password along with a nonce to create a ticket (K_{AB}). The KDC also send to Alice K_{AB} encrypted with Bob's password K_B. This is sent to Alice. Using her password, Alice retrieves the secret key and forwards the ticket to Bob.

Bob uses his password to retrieve the secret key. They both now can communicate directly using their shared secret key. The revised process is shown in Figure 3.2.

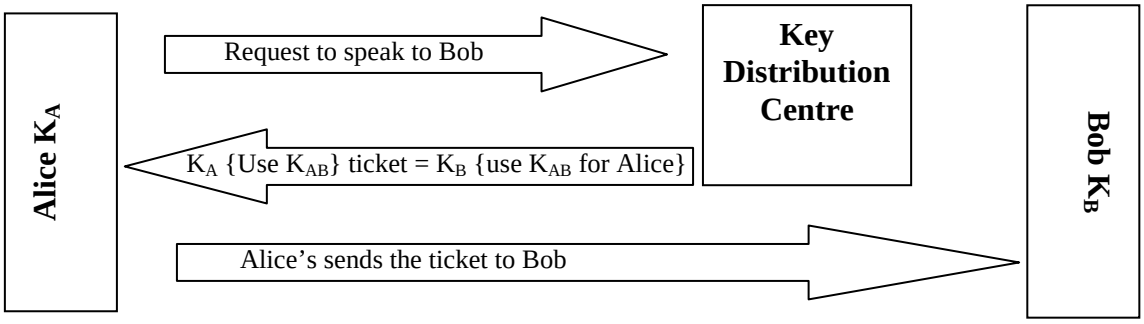


Figure 3.2 Symmetric key exchange using a KDC (Hadjichristofi 2005).

Once the key has been created and sent to the other party, private message sending can begin. The process of encryption and decryption of the messages is as follows:

1. Sender encrypts the message by processing the individual bits through the encryption algorithm with the key.

2. Sender sends the encrypted data to the receiver.

3. Receiver processes the data through the decryption algorithm using the same key.

4. Receiver can read the decrypted message.

By allowing the encryption algorithm to be publicly available, only the encryption key must be kept secret from all but the communicating parties. The encryption algorithms are generally publicly available as this allows intense scrutiny of their method of encryption leading to greater confidence in the security of the algorithm. Some encryption algorithms are kept secret in an effort to enhance security but these are generally not widely accepted for use because their robustness against attack can not easily be tested.

With the problem of secure key exchange being solved, a key management system (KMS) can be designed and the keys safely exchanged. For a KMS to be complete it must perform three basic functions.

1. Key creation: The two parties wishing to communicate must first decide which of them will create the key. For a wired network or infrastructure mode wireless network, this will be undertaken by the access point or a separate server connected to the access point. For an ad hoc network where the nodes are generally considered equal, a rule must be designed into the protocol for this task. Whilst it may be a trivial decision, the rule should state which node should be responsible for key creation. If the network incorporates nodes that are considered superior for some reason such as a server or cluster head node, then it may be that one task of that node is to create the keys when required.

2. Key distribution: As at least two parties will share an encryption key, the key must be distributed securely to those parties. The choices for distribution in a wireless network are:

    a. Offline where the keys are entered manually into the network nodes prior to the network going live. Whilst a very secure method, it does require very high management, especially in a network with a large number of nodes. If one key will be shared by all network members then the task is simplified as only one key need to be entered into each device. If a unique key will be used for each pair, then $k - 1$ keys (where k is the number of network members) must be entered into all other devices for each node.

b. Secure side-channel: The keys are entered into each node before network deployment in a method that is secure such as by infra-red. This is also very high management but less intensive than manually.

c. Online: The keys are exchanged during network formation wirelessly. This is the most efficient and least management intensive but also carries the greatest risk of compromise. However, with secure key exchange it can be performed with a high level of confidence in the key remaining secure. To reduce the time required for key exchange, only those pairs of nodes that will be communicating may exchange keys and if any further exchange is required, this can be done on a demand basis.

3. Key Revocation: A key may be revoked for several reasons. The key may expire, it may be compromised by an attacker or the node may leave the network. Keys that are revoked are not reissued and key revocation may also involve key destruction where a method is in place where the key can be destroyed if necessary. Destruction may be necessary so that a compromised node or one that has left the network can not pass the key to another party who may have been capturing the encrypted messages. If the key were obtained by this unauthorised node, it could then decrypt and read the previously captured messages.

Symmetric encryption is a relatively simple solution for security of messages as it requires only the two communicating parties for it to be implemented. For an ad hoc environment where key management may be done online, one party takes responsibility for key creation and distributes the key securely to the other party. At this point, secure

communication can begin. If the key is no longer required, the parties agree to revoke the key and may destroy it to prevent compromise at a later time. One further benefit that symmetric encryption enjoys is that the processing required for the encryption and decryption of the message is not overly computationally expensive. This means that it can be done relatively quickly and without excessive draining of battery power. The tradeoff with this simple scheme is that the security level it offers is not particularly high for two main reasons. Firstly, there is no authentication provided between the two parties, only a simple key exchange. Secondly, without authentication this technique is open to a Man-in-the-Middle attack which would be undetectable. However, the simplicity of the exchange and the reliance on only two parties to perform the exchange means that symmetric encryption is a good choice for ad hoc networking with battery powered devices. However, if higher security is desirable then a more robust technique involves the addition of a third party to participate in the key management. This technique called Public Key Infrastructure (PKI) is also known as asymmetric encryption and is described below.

3.3.2 Asymmetric Encryption

If two parties wish to communicate directly and have no prior knowledge of each other, then asymmetric encryption allows message passing to begin more quickly because the added steps of symmetric key exchange are not required. When two parties wish to communicate, the initial request may contain the requester's public key and the reply may contain the public key of the requestee. At this point, both parties have their private keys which remain secret and their public keys have been exchanged. The main problem with this type of direct exchange is that no authentication has taken place. Whilst the request has been sent, the requester can not be sure that the reply has come from the

intended party. In a dense network where several nodes may overhear the exchanges, a man in the middle (MTM) attack may be simply performed. As described in Chapter 2, this type of attack involves a third party masquerading as the intended party to both the requester and requestee. The third party reads the messages and then passes them on without the other two parties aware that the messages have been read or altered.

As with symmetric encryption, most types of security attacks can be mitigated by implementing authentication. Authentication involves a node being able to prove who it is. This can be done in a variety of ways. A node may rely on other nodes that trust it to vouch for it such as in a web of trust, or it may possess a digitally signed certificate that it can present proving that its identity is as it claims. Purely relying on trust and therefore honesty of the node is not overly secure. A variety of malicious behaviours are trivial to implement once trust has been built up. A much more robust method is to possess a digitally signed certificate that among other attributes proves the node's identity and public key. The Internet Engineering Task Force developed the X.509 digital certificate standards and the information contained in the current version of that standard certificate is shown in Table 3.1.

Table 3.1: Basic fields of an IETF X.509 v3 Digital Certificate (IEEE 2004).

X.509 v3 Field	Description
Version	Indicates the X.509 version
Serial Number	Unique integer assigned by the CA
Signature	Object Identifier and optional parameters defining the algorithm used to sign the certificate
Issuer	Distinguished name of the issuing CA
Validity	Specifies when the certificate becomes active and when it expires
Subject	Distinguished name identifying the entity whose public key is certified
Subject Public Key Info	Public key and parameters and the identifier of the algorithm with which the key is used
Issuer Unique ID	Optional field to allow reuse of issuer names over time
Subject Unique ID	Optional field to allow reuse of subject names over time
Extensions	The extension data

A node requesting communication with another node will exchange digital certificates and therefore verify identities and public keys. However, certificates may expire or be revoked, either through a node misbehaving or leaving the network, or if a node is known to have been compromised. Therefore, nodes in the network must be able to check the validity of the certificate. This is usually done by having a Certificate Authority (CA) who acts as a trusted third party (TTP). The CA is generally responsible for issuance of the certificate and therefore may keep a record of all valid certificates in a Valid Certificate List (VCL) and any certificates that have been revoked in a Certificate Revocation List (CRL). When a node receives a certificate from another node, it requests the CA to check the validity of the certificate. If the validity is confirmed by the CA, then the node will agree to communication. If not, it ignores the request. This process is shown in Figure 3.3.

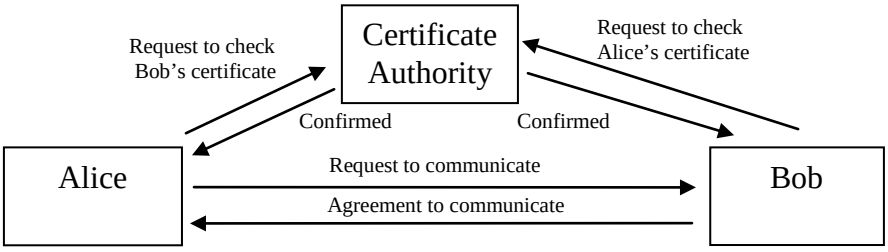


Figure 3.3: A successful request for Alice to communicate with Bob.

A disadvantage of this necessity to check with a third party is that the CA must always be available and contactable. Further, if intermediate nodes are used to pass on messages to the CA, greater opportunity exists for an attacker in the message chain to misbehave with the request. Having a single node acting as a CA therefore places a great burden on the node to remain always available and to process constant requests for certificate services. If new nodes join the network, then the CA may also need to create

and distribute new certificates if the protocol allows this, and if certificates expire, then constant renewals of certificates may be required, especially in a large network with many nodes. One further problem with having a single CA for the network is that it creates a single point of failure. If a successful attack against the CA is performed, then the network may cease functioning or all certificates may be compromised.

This problem can be dealt with in several ways. Firstly, multiple CA's can be deployed that all contain the same information. A node requiring certificate services need only contact any one of the CA's to receive service. The CA's will then periodically merge their CRLs to ensure revoked or expired certificates are not confirmed as valid. Multiple CAs solves the problem of a single point of failure, as failure of a single or even multiple CAs will still allow the network to function. However, a potential attacker now has multiple targets to attack, any one of which may supply all the sensitive key information about the network. One method to deal with this problem is to divide the functionality of the CA across several nodes, each of which performs a part of the key management function but requires several CA's information to be merged to perform the entire task. This collaboration between more than one CA to perform a single task is called threshold cryptography.

3.4 Threshold Cryptography

For a network that has prior planning with members pre-configured, area and numbers of nodes known in advance and geographical placement of the nodes also known, security of the CA nodes can be much more robustly configured than if those attributes are not known. It may be better from a performance viewpoint to have a single CA node, or multiple replications of the CA node where each CA has full ability to provide

key management services. If the network is truly ad hoc with no prior planning, area and node numbers unknown, and no prior knowledge of each other, allowing a single node or multiple nodes to have full CA ability is much riskier. The problem of a single point of failure, or in the case of a replicated CA, multiple targets for an attacker each containing all the sensitive key management information for the network, makes standalone CAs not such an attractive choice. For security reasons, dividing responsibility between multiple nodes that must combine to produce a service is a much better proposition. Threshold cryptography involves a threshold of nodes designated as CAs or key management servers, that each performs a part of the required task for key management. In a network of *n* nodes, *k* nodes is the threshold and *m* is the total number of CA nodes. This threshold k is sometimes chosen to be one number above the number of nodes that it is foreseen could be compromised, if that can be calculated, or it may be the number that provides the best performance without unduly compromising security. If there are more than *k* nodes designated as CAs, then there is redundancy provided which ensures some CAs can go offline or be compromised without affecting the CA's ability to perform key management.

$$k < m < n$$

n = number of nodes in the network, *m* = number of CAs, *k* = threshold required for services

Threshold cryptography is useful for truly ad hoc networks where the dynamic nature of the network makes CA availability at all times difficult to ensure, and CA replication impractical because of vulnerability to compromise.

3.5 Identity Based Cryptography

For truly ad hoc networks without prior configuration, nodes' identities will not be known in advance. If PKI is used where certificates are issued during network

operation, the identity of the node must be fixed so that it cannot change that identity at a later time. Non-changeable identities of nodes ensure that they can not masquerade as another authenticated node, and not rejoin the network under a new identity if they have been ejected, possibly for malicious behaviour. Often, the identity may be the Media Access Control (MAC) address of the network card issued by the manufacturer, or the Internet Protocol (IP) address issued to the node when it joins the network. Whilst these are both unique, they can both easily be changed by a misbehaving node (spoofed). It is much better to tie the identity to some attribute of the node that cannot be changed. One method is to use a hash function on the encryption key issued with the certificate. The public key can be similarly hashed by a prospective communication partner. If it matches the identity then the key and identity are genuine, if not then it is a spoofed identity. If the certificate is pre-installed and cannot be altered, then this method may work well. However, if certificates are issued on a demand basis after network initialisation, then nodes may be issued multiple certificates with multiple requests. Therefore, tying the identity to the public key does not work well for truly ad hoc networks using PKI.

Another method is to tie the identity to some hardware device such as the CPU serial number that is both unique and unchangeable. This method is perhaps the best of the various approaches for an ad hoc network as no prior knowledge of the node is needed and the serial number used as the unique identifier exists for the lifetime of the device, not just the time the node is in the network. Different methods suit different applications, but whatever the method used, identities can be attached to nodes that are non-changeable, at least to the point that if a malicious attempt to alter them is made, it will be obvious to the other nodes in the network.

3.6 Summary

Cryptography involves altering the individual bits in a message in such a way that an intruder who intercepts the message cannot read it. Additionally, for a full key management system, the intruder should not have the ability to alter the message without it being obvious to the intended recipient, who can then ignore the message or request that it be sent again. Encryption requires two artefacts, a robust encryption key and a robust encryption algorithm. Encryption protocols for ad hoc networks generally utilise either symmetric encryption where the same key is used for encrypting and decrypting the message, or asymmetric encryption where different keys are used. Whichever method is chosen will depend on the requirements of the network and the level of security needed for the application. If a non-changeable identity for every member of the network is enforced, then security can be raised to a higher level by monitoring behaviour of network members and permanently ejecting any that may seriously misbehave. Byzantine behaviour, where several network members collude to disrupt the network can be thwarted by the use of threshold cryptography, where the key management services are spread across several nodes acting as servers. These nodes must work together, each providing only part of the encryption key, certificate, or signature. These parts can then be assembled and the requesting node will then have a genuine and authenticated certificate and key. This method provides robustness by allowing for several nodes to be compromised without sufficient information released to the attacker to allow them to disrupt the network. By combining all of these features into a full key management system, an effective, efficient and robust key management protocol can be developed that can be used in a variety of ad hoc applications.

The following chapter reviews the state of the art for mobile ad hoc network key encryption protocols. It discusses the current protocols and their benefits and weaknesses, and looks at what attributes best suit applications that utilise truly ad hoc networks.

Chapter IV

LITERATURE REVIEW

4.1 Introduction

This chapter examines the state of the art in encryption and key management in wireless networks. A discussion of the various topologies is followed by an examination of the parts of various proposed schemes which are most promising for a truly ad hoc network key management system. Whilst the focus of this research is on key management schemes for Mobile Ad Hoc Networks, an examination of several Ad Hoc Sensor Networks is also made. Sensor networks are quite different from MANETS in that they involve sensors that are often static and generally have very limited processing power. However, several sensor schemes have interesting features which can be adapted to MANET schemes and so an examination of those schemes is warranted.

The choices available for implementing a secure and efficient key management system in wireless ad hoc networks are many and varied. Key generation and deployment within ad hoc topologies can be categorised into two general areas: singular keys where keys are used for private conversations between network members, and group key exchange where a single key for the entire cluster or network is dispersed amongst members. Whilst this research deals with key management for mobile ad hoc networks and focuses on single key generation and distribution, useful areas from all types can be employed in Mobile Ad Hoc networks and so are worth discussing.

The dynamic nature of the network makes choosing wireless nodes to act in a superior manner to other nodes difficult. Whilst many schemes select a head node or head nodes for the network, mobility and dynamicity mean that the head node or nodes can move

out of range, leave the network or otherwise become ineffective. Schemes are often categorised in different ways depending on the perspective of the designer of the protocol. From a generalised approach they can be divided into three distinct types. Firstly there are hierarchical schemes, also called centralised, whereby a head node is appointed and effectively acts as the centre for control of the network. Key management messages flow from and to the head node who assumes overall control of the network. Secondly there are contributory schemes where several or all nodes work together to provide key management services for the network. Finally, there is a hybrid of both types where the network is divided into clusters. Here, each cluster effectively acts as a sub-network and head nodes in the clusters control security within their cluster. Inside the clusters, nodes may form into hierarchies or contributory topologies and the clusters themselves then act in a contributory or hierarchical manner.

If the perspective is on the generation and distribution of the encryption keys, then the categorization is best divided into two main areas, contributory schemes and distributive schemes. This categorization refers to the generation of the keys. It describes whether a single node generates the key or multiple nodes work together to generate the key. From that point onwards several nodes may be involved in the distribution of the keys, however the categories relate to the key generation process. The distributive category can be further subdivided into symmetric key schemes and asymmetric schemes employing PKI. Key deployment within these topologies can further be divided into two types: singular keys where keys are used for private conversations between network members, and group key exchange where a single key for the entire cluster or network is dispersed amongst members along multicast or broadcast messaging. Figure 4.1 shows the categories of key management schemes.

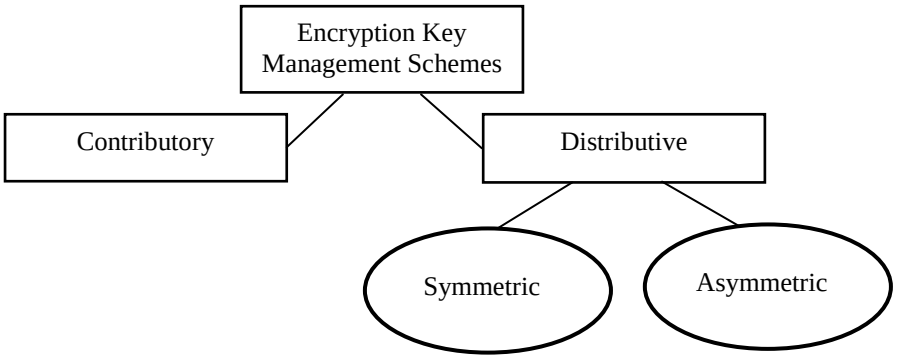


Figure 4.1: Categorisation of KMS schemes (Hegland, Winjum, Mjolsnes, Rong, Kure and Spilling 2006).

Whatever topology or key exchange method is employed, useful areas from all types can be utilised in Mobile Ad Hoc networks and so are worth closer examination. As this research is focused on key generation as well as distribution, the categories used here are contributory and distributive. The following section begins with a generalised examination of various key exchange schemes followed by an examination of contributory schemes and finally distributive schemes.

4.2 Encryption Key Exchange

In Chapter three, Diffie-Hellman key exchange was described where two nodes can securely exchange encryption keys even in a dense wireless network where several nodes may overhear all communication between them. One problem with Diffie-Hellman key exchange is that the communicating parties can not be sure that the node purporting to hold a certain identity is in fact that node with that corresponding identity. For this reason, tying a non-changeable identity to a node is vital to ensure the node can not masquerade as another node. Further to and caused by this problem is the vulnerability of the scheme to a Man-in-the-Middle attack. In Chapter two, this type of

attack was discussed, but fortunately this attack threat can be negated by the use of authentication of the parties involved in the key exchange.

A modification to the Diffie-Hellman key exchange protocol introduced authenticated Diffie-Hellman where both parties must be authorised to participate in the network and therefore exchange encryption keys (Diffie, Oorschot and Wiener 1992). This scheme uses public key cryptography where the two parties both possess public and private key pairs and a digitally signed certificate authenticating their identities and the keys. The digital certificates are generally securely entered into the nodes offline before network formation. Because the nodes are both authorised in the network and can positively identify themselves, the man-in-the-middle attack will be thwarted when a node checks the authenticity of the certificate of an attacking party. With secure key exchange available to schemes employing PKI, many of the protocols have chosen to use PKI at least for initial key exchange, even though many will then employ symmetric key encryption from that point onwards.

4.3 Contributory Schemes

The ING is named after one of its authors and is a group key exchange scheme where the Diffie-Hellman protocol is used to exchange a group key and that key is then passed on from node to node where the nodes are positioned into a logical ring (Ingemarsson, Tang and Wong 1982). The key distribution involves n-1 rounds where each round involves part of the key being forwarded to the next node. When the rounds are completed the group key can be calculated. The main disadvantages with this scheme are that it scales particularly poorly and that misbehaving nodes are difficult to detect and therefore may prevent the key from ever being distributed. Further, if a node leaves

the network the entire process must be performed again to create and distribute a new key. For a highly dynamic network, this could lead to much of the traffic on the network being key generation and distribution messages, leaving little bandwidth for non-management messages.

In contrast to many of the contributory schemes, Burmester and Desmedt propose a scheme where Diffie-Hillman key exchange is not used (Burmester and Desmedt 1994). In their scheme which is designed to generate a group key, only three rounds are needed. The first two rounds require each node to generate and broadcast a public value with the third round used to calculate the conference key from the broadcast value and its own secret. Whilst using fewer rounds than the similar ING scheme, the calculations are considerably more complex requiring more time and more battery power. One other significant drawback of this scheme is that it requires very reliable links between the nodes which is unlikely in an ad hoc topology.

A scheme that was designed specifically for ad hoc networks is that proposed by Asokan and Ginzboorg (Asokan and Ginzboorg 2000). Their scheme called Password Authentication Key Agreement (PAKA) is an extension of another protocol, Hypercube and Octopus (Becker and Wille 1998), which itself is a modification to ING. The H&O protocol logically arranges the nodes in a hypercube and uses Diffie-Hellman key exchange to exchange the symmetric keys. The hypercube structuring significantly reduces the number of rounds needed in ING, but suffers from a lack of authentication of the nodes to each other. PAKA takes this protocol a step further by introducing authentication in the form of a password which must be used during the Diffie-Hellman key exchange. The main drawback of this scheme is that the passwords must be

preconfigured within the nodes offline. This moves away from the truly ad hoc nature of the network that is needed in on-the-fly rapid deployments.

Another extension to the Diffie-Hellman key exchange is called Cliques (Steiner, Tsudik and Waidner 1998). This scheme makes a distinction between the initial key agreement and the later auxiliary key agreement. A group controller, effectively a head node, is used to synchronise the key agreement process. The group controller collects all of the shares of the group key which are unicast by all other nodes to the group controller. The group key is then calculated and distributed. The group controller however becomes a single point of failure for the network and so all nodes have the ability to take over as group controller if necessary. Two drawbacks to the scheme are that a malicious node may be able to take the role of group controller and that this scheme requires reliable links from all nodes to the group controller which would be unlikely in a highly dynamic ad hoc environment.

Contributory schemes appear attractive for ad hoc networking as they fit well into the ad hoc nature of network formation and function where nodes may join and leave frequently. However, all schemes proposed so far have significant drawbacks making them unreliable for such a volatile network as that envisaged. Whether it be requiring preconfiguration, a reliance on reliable links to function, or lack of authentication meaning poor security, no contributory scheme is effective enough in such situations as may be found in the likely applications that a truly ad hoc KMS may be used in. Table 4.1 shows the characteristic of the key exchange and contributory schemes and is modified from that derived by Hegland et al (Hegland, Winjum, Mjolsnes, Rong, Kure and Spilling 2006).

Table 4.1: Characteristics of key exchange and contributory schemes.

	Diffie_Hellman DH	Authenticated DH	ING	Burmester -Desmedt	PAKA	Hypercube	Cliques
Group / Individual	I	G	G	G	G	G	G
Authentication	No	PKI	No	PKI	Password	No	No
Redundancy	No	No	No	No	No	No	No
Scalable	No	No	No	No	No	No	No

4.4 Distributive Asymmetric Schemes

Distributive schemes are quite different to contributory and can employ either PKI or symmetric key encryption. If PKI is employed then a Certificate Authority (CA) must be available at all times to undertake key management responsibilities. These include certificate issuance, valid certificate confirmation to any requesting node and certificate revocation. If a distributive CA is used, then a number of nodes must collaborate to form the CA and perform the tasks. A request for a certificate will involve the CA nodes constructing a certificate and signing the certificate as valid. A choice of whether to collaborate to create the certificate or the signature authenticating the certificate is available. A method for generating a full signature from parts of the key was devised in 2000 and is an elegant solution as it allows the certificate to be generated and distributed but the key to remain secure with the certificate holder once it has been assembled (Shoup 2000). The PKI distributed schemes are examined next.

A partially distributed Threshold Certificate Authority Scheme has been proposed whereby a significant advantage is tolerance to intrusion of the Certification Authority (Zhou and Haas 1999). The private CA is distributed over several nodes designated as server nodes. The threshold for the number of server nodes required is *k* (threshold) out of *n* (total nodes) nodes. When a node requires a certificate, each server node generates

a partial signature using its private key. A server designated as a combiner node then collects all of the partial signatures and combines them to produce a valid signed certificate. This certificate is then securely passed to the requesting node. Periodic updates of the certificates are used to counter any possible attacks that may have compromised a server. The robustness for the scheme comes from distributing the responsibility for certificate generation to several server nodes, meaning that a number of server nodes must be compromised within a limited time frame before enough useful information can be gleaned. The major drawbacks to this scheme are that the initial configuration, including which nodes shall act as servers, must be done prior to network boot strapping offline, and that the combiner node is a single point of failure for any certificate requests. One further problem with the scheme is that of periodic updating of the certificates. Whilst adding security to the protocol, it requires synchronisation of the nodes to ensure their certificates have been updated and old certificates are entered on the CRL. This adds considerable message overhead to the network using up valuable bandwidth and draining battery power.

An extension of this protocol was proposed by Yi and Kravets (Yi and Kravets 2003). This scheme called Mobile Certificate Authority (MOCA) deals with the CA problem by distributing the CA functionality. CA nodes are selected by displaying the best physical security and computational ability. For example, a powerful laptop computer would be better suited to CA responsibility than a computationally and battery power limited PDA. Furthermore, the responsibility of combining the partial certificates into a full certificate is moved from combiner nodes to the requesting node. This adds robustness as the requesting nodes no longer rely on the combiner nodes being available. In Zhou and Haas's scheme flooding is used to request certificate services. In

MOCA, a new protocol is proposed that sends request messages in a unicast format directly to the server nodes. However, if server nodes are not immediately available then the protocol reverts to flooding. This adds efficiency to the communications freeing up bandwidth for other communications, but requires nodes to maintain an extra routing table to that of the underlying routing protocol. This adds complexity and itself requires extra messages for management of the routing tables.

A slight modification to the MOCA protocol produced a scheme called Secure & Efficient Key Management (SEKM) (Wu, Wu, Fernandez and Magliveras 2005). In this modification the servers in MOCA instead form a multicast group to add efficiency to updating of secret shares and certificates. The node broadcasts a request for certificate services to a server group, with the first server to receive the request generating a partial signature and then forwarding the request to k + a servers. Only k partial signatures are required with the additional ones used as redundancy in case one is corrupted or lost. Whilst adding efficiency to the MOCA protocol, it is essentially a minor modification to gain a slight improvement in efficiency. One drawback to the proposed scheme is that it does not describe how the first receiving server should identify that it is the first and advise other servers of the fact.

Whilst MOCA and SEKM use partially distributed CA schemes, Kong et al have suggested a fully distributed threshold CA scheme called Ubiquitous Security Support (Ubiq) (Kong, Zerfos, Luo, Lu and Zhang 2001). Here, all nodes in the network get a share of the private CA key. A coalition of k one-hop neighbours combines to provide CA functionality. It does not require an underlying routing protocol to assist with this but does require at least *k* one-hop neighbours. For this reason, mobility of the nodes

may actually assist with functionality of the CA. The nodes in the network earn trust from other nodes when they prove that they hold a valid certificate, and holding a certificate allows the node to hold a share of the CA private key. The major drawback of this scheme is that it requires offline configuration of certificates for the initial nodes that instigate the network. Further, the threshold value *k* may be difficult to choose. A low value of *k* means that an intruder needs to compromise fewer nodes to obtain useful key information. A higher value of *k* makes intrusion more difficult but requires a higher number of nodes to be easily reachable for key management services. Limiting communications for certificate services to one-hop neighbours is bandwidth efficient and therefore good for scalability, but for a smaller network where nodes may only have one neighbour, this can mean CA services are unavailable.

A further protocol that utilises a fully distributed CA is that proposed by Zhu et al (Zhu, Bao, Deng, Kankanhalli and Wang 2005). Called Autonomous Key Management, this scheme is similar to Ubiq when there are only a few nodes in the network. However, as the network grows in numbers, a hierarchy of key shares is utilised. Instead of receiving a share, the new nodes receive a share of the share of the CA private key. The root CA private / public key pair is bootstrapped by a group of neighbours through distributed verifiable secret sharing (Rosario, Jarecki, Krawczyk and Rabin 2007). Each of the *n* neighbours chooses a secret value and distributes shares of this to the other neighbours using a (*k*, *n*) secret sharing scheme. Whilst the distribution of the shares is contributory, the derivation of the shares is not. To counter too few nodes requiring compromising to discover the key, the shares are split and distributed to more nodes as the numbers grow. This adds a higher level of security to the scheme, but a major drawback is that authentication of nodes is done offline prior to network initialisation.

In an effort to devise a scheme that is fully distributed and therefore in the spirit of true ad hoc networking, Capkun et al developed Self-Organised Key Management (Capkun, Buttyan and Hubaux 2003). It is essentially a Pretty Good Privacy (PGP) scheme adapted to ad hoc networks (Zimmerman 1994). All nodes are equal and generate their own private / public key pairs, distributing their certificates to nodes that they trust. The certificates are stored within each node rather than relying on a central repository. The scheme assumes that trust is transitive, meaning that if 'A' trusts 'B' and 'B' trusts 'C', then 'A' can also trust 'C'. The nodes merge their certificate repositories and attempt to find a verifiable chain of certificates. Certificates may be revoked either explicitly by the issuer, or implicitly after an expiration time. Renewing a certificate requires contact with the issuer which may not always be possible in a mobile network. Periodic exchanges of certificates with neighbouring nodes occur, but this synchronization of exchanges and renewal of certificates is not defined in the protocol. Additionally, certificate expiry requires that a constant renewal of certificates is required, and along with a constant certificate exchange with neighbours means that the scheme does not scale well. Further, Byzantine nodes (misbehaviour by colluding nodes) could compromise a single node to gain information about several certificates. If a node can legitimately join the network, then it could maliciously issue certificates to other malicious nodes allowing them to legitimately join the network and freely be given certificate information of other nodes. This lack of scalability and reliance on transitive trust makes this scheme suffer from fairly poor security.

Yi and Kravets combined their MOCA protocol with certificate chaining to develop a new protocol called Composite Key Management (Yi and Kravets 2004). CKM attempts to take the best points of both protocols and eliminate the poorer points by

providing higher security than PGP alone and increased availability of the CA in the MOCA scheme. Nodes certified by the CA can themselves issue certificates. Nodes must first request a certificate from a CA node, but if a CA node is unavailable then a certificate holding node can take over the certificate issuing duty. One major modification to the two schemes that it is derived from is that the certificates themselves contain a confidence value reflecting the confidence that the issuer has in the node. Confidence is measured from no confidence being a 0 to total confidence being a 1. Each node in the issuing chain multiplies the existing confidence by its confidence level and replaces the value in the certificate with that calculated confidence. In this way, the final confidence value in the certificate reflects the average confidence of all the nodes along the chain. A threshold value for confidence and therefore whether a certificate is issued can be preset for the network, with a higher value requiring more trust for a node to join. This scheme provides better security and greater availability of the CA than the schemes from which it was derived, but the protocol does not describe how revocation of a certificate should be done. Additionally, the problem of bootstrapping the network with no initial trust is dealt with by configuring the initial nodes offline. This is unsuitable for a truly ad hoc network where all key management tasks must be performed after network formation.

In the Mobility-Based Key Management Scheme, security is dealt with by mimicking human behaviour (Capkun, Hubaux and Butty 2003). If two nodes wish to communicate securely then they must physically move close to each other eliminating the need for nodes in between to act as message hopping points. The protocol allows for offline configuration if communicating needs are pre-known and therefore symmetric encryption can be used, or fully self-organising utilising either symmetric encryption or

PKI. If self-organisation is used then the scheme permits one level of transitivity of trust, so that if ‘A’ wishes to communicate with ‘C’ and ‘B’ trusts ‘A’ and ‘C’, ‘B’ can vouch for both ‘A’ and ‘C’ allowing them to trust each other. Before beginning communication, the nodes exchange userID, key and node address, and sign and exchange a certificate stating that an association has been established between the two. One advantage to the scheme is that only the communicating pairs hold each other’s keys and therefore compromise of a node only compromises that node and the one that it is communicating with. However, Byzantine behaviour can be difficult to identify and the process for revocation of a certificate is not defined. Once again, transitivity of trust means a compromised node, if not readily identified may vouch for other malicious nodes allowing them to communicate with other network members. However, the main disadvantage of this scheme lies in the need for communicating pairs to be physically collocated to communicate, which is difficult to achieve in a MANET and it is questionable whether security is enhanced measurably by such a restriction.

A modification to PKI schemes is to use Identity-based schemes as a replacement for certificates. Identities that bind a node to its identity have the advantage of being relatively small in size, typically only several kilobits. For bandwidth limited schemes, identity based schemes use information already contained in messages as identifying information. Such a scheme is Identity-based Public Key (IBC-K) first proposed in 1984 (Shamir 1984). In this scheme, a private key generator (PKG) is used during the setup phase to generate the public system parameters and along with these and the ID of the node, the signature of the communicating node can be verified. The public system parameters include the public key of the PKG and information about the message space. The PKG also generates the private signature keys binding them to the ID of the node.

In the extraction phase the private keys are issued and are uniquely given using the IDs and the PKG master key.

Extensions to identity-based schemes have been proposed including the first to offer a practical scheme (Boneh and Franklin 2001) and later an extension to this scheme to offer message authentication at no additional overhead (Lynn 2002). This scheme used the ciphertext itself to act as a message authentication code. The drawback of such schemes is that the PKG is a single point of failure for the entire network, and that all nodes must be able to contact the PKG for initial issuance of certificates.

A scheme was proposed that combines identity-based cryptography with threshold cryptography specifically for use in ad hoc networks (Khalili, Katz and Arbaugh 2003). The nodes that initialise the network form a threshold PKG spreading the PKG in a (*k*, *n*) threshold manner. This provides much greater security requiring several nodes to be compromised to gain useful information about the PKG master key, and provides key management services as long as *k* server nodes can be reached. A request for a private key is sent to *k* or more server nodes along with the requester's identity. The node is sent a share of the private key from each of *k* nodes. With *k* correct shares received, the node can compute its private key. The major flaw with this scheme for a truly ad hoc network is in the network initialisation phase. Mutual authentication between nodes entering the network is required prior to beginning communications. This implies physical contact between the nodes or a secure side channel that cannot be eavesdropped. Further, this initial phase is vulnerable to man in the middle attacks and so the only secure method is physical, offline preconfiguration which is not in the nature of a truly ad hoc network.

The major drawback in most PKI schemes devised so far for ad hoc networking is that at least a threshold of nodes require offline configuration to be truly secure. Once running, many schemes allow the network to maintain security with some schemes proving to be at least moderately robust. However, the complexity of maintaining accurate and up-to-date CRLs and ensuring all nodes are aware of revoked certificates in a timely manner is difficult to overcome. In a geographically dispersed MANET such as in those envisaged where communications may be at or near maximum distances for radio transmission, knowing whether a node has left the network or has just been temporarily disconnected can be difficult. The action to take in these circumstances can make the difference between an efficiently running network and one overwhelmed with management messages such as certificate creations, requests, revocations and updates. These schemes all show some positive areas for this type of application but none is effective enough as it stands to be truly useful in the likely types of applications. Table 4.2 shows the characteristics of distributive asymmetric schemes. It is modified from that derived by Hegland et al (Hegland, Winjum, Mjolsnes, Rong, Kure and Spilling 2006).

Table 4.2: Characteristics of distributive asymmetric schemes.

	Zhou / Haas	MOCA	SEKM	UBIQ	AKM	PGP-A	MOB	IBC-K
Trust	CA	CA	CA	CA	CA	CA	CA	PKG
Authentication	No	PKI	No	PKI	Password	No	No	No
Redundancy	No	No	No	No	No	No	No	No
Scalable	Poor	Fair	Fair	Fair	No	No	No	No

An examination of distributive symmetric key encryption schemes follows.

4.5 Distributive Symmetric Schemes

Pre-shared Group Key (PSGK) is a scheme whereby a group key is distributed to nodes prior to network messages beginning. Being a group key and utilising symmetric key

encryption, a single key is shared by the entire network and the same key is used for encryption and decryption of messages. Any new node wishing to join must be sent the network key before it can communicate. Initial installation of the key can be done manually or a secure key exchange protocol can be used. For keys to be distributed non-manually, a key distribution centre must be employed to create and distribute the keys to all nodes. If the key distribution centre, effectively a head node, is unavailable, then the network cannot function. Additionally, any compromise of a single node discloses the group key meaning that key can no longer be used. Any new node wishing to join the network must request the network key. If the same key is used continually, then nodes that may have been monitoring traffic before joining can use the key to decrypt those prior messages. Utilising a single key for the entire network is simple and efficient, but messages within the network are not secure with every node having the ability to read every message it is within range of listening to.

A simple modification to this scheme is to manually configure nodes with key pairs so that every node maintains a list of keys to communicate with every other node. However simple as this seems, the problem of key distribution is very resource intensive. A node joining a network comprising of, for example, 100 nodes must generate 100 unique keys and distribute each key securely to every other node. This can be done either manually or by employing a secure key exchange protocol such as Diffie-Hellman. Whilst elegant in its simplicity, this basic method of key management is only suitable for small networks where key management services can be done securely prior to network startup.

A protocol designed specifically for use in ad hoc networks during emergency response is called SKiMPy (Pužar, Andersson, Plagemann and Roudier 2005). In this scheme, the nodes in the network generate a random key and advertise it to their one-hop neighbours through HELLO messages. The best key is then chosen as the local group key. Here, best may mean the key with the lowest ID number, the latest timestamp or similar. Predistributed certificates within the nodes allow a secure channel to be used to distribute the chosen key to all nodes, and allow for ready identification of authorised nodes. Periodically the key is updated to counter any attack on the group key that may be underway. One advantage of the scheme is that key distribution is independent of routing protocol as the key is passed directly to one-hop neighbours continuously until all nodes possess the key. A disadvantage of the scheme is that it is used to spread a group key which allows all nodes to decrypt all transmissions that they receive. For emergency response work or other applications where it is desirable for all information to be shared, this may be acceptable. But if for example it is the victims of a disaster using the network, they may wish to communicate privately with each other. Additionally Byzantine behaviour by nodes is not addressed and could lead to false keys being sent to neighbours advertised as the new group key. However, the protocol is reasonably efficient and does fit well with truly ad hoc deployments.

Self healing Session Key Distribution is a scheme whereby the emphasis is on robustness where unreliable links exists and revocation of the session key can be performed (Staddon, Miner, Franklin, Balfanz, Malkin and Dean 2002). The protocol requires that all nodes contain a pre-shared secret and that a group manager is in charge of key management services. The master key is distributed in a message to nodes that then process the message and extract the key. To exclude a node from the network, new

messages are sent to all nodes but the excluded node lacks a required message to extract the key. The self-healing ability is handled by allowing legitimate nodes who may have missed one or more key updates to still compute the latest session key. All key update messages contain a share of the previous key update messages and later key shares, meaning nodes can use these shares in sequence to compute later keys until arriving at the latest, valid key. The main advantage of this scheme is that it is robust against communicating nodes that may drop in and out of range of the network and therefore miss one or more key update messages. It is also bandwidth efficient as the key update messages are periodically broadcast to the entire network at one time. However, reliance on a group manager for the key creation, distribution and revocation gives it a single point of failure and requires that all nodes can receive messages sent from a single node.

In Logical Key Hierarchy (LKH), the group key is distributed by a group manager with the key encrypted with a node's individual key (Wong, Gouda and Lam 2000). A logical hierarchy is employed to make key revocation a simple process. The hierarchy is logically ordered so that nodes represent the leaves of a tree with the group manager as the root. A modification to this type of scheme was proposed whereby the group manager function is distributed over several group managers each responsible for their cluster of nodes (Rhee, Park and Tsudik 2005). When nodes move from one cell to another, they must contact the new cell manager to receive that cell key. This means that nodes must be aware of which cell they are located in, and effectively makes this scheme a centralised topology. A refinement to the original protocol resulted in LKH++ (Pietro, Mancini and Jajodia 2002). In this scheme, designed for wireless networks, nodes are able to compute the new key from the old key using a one-way function. This reduces the need for rekey messages freeing up bandwidth for other communications. A

major advantage to this scheme is that it is more bandwidth efficient than many others and using individual keys to encrypt messages allows Byzantine behaviour to lead to expulsion. However, having to rekey each time a node joins or leaves the network is bandwidth inefficient especially in a network with unreliable links.

Zhu and others devised a probabilistic key pre-distribution scheme with secret sharing to setup pair-wise keys in mobile ad hoc networks (Zhu, Xu, Setia and Jajodia 2003). A node intending to communicate with another node sends a secret symmetric key to the other node encrypted with different pre-distributed keys. The assumption made by the authors of the scheme is that less messages required to securely communicate is preferable over higher computational requirements for the scheme. In an ad hoc network where devices have limited power and in an application where recharging of batteries may not be possible, computational complexity may be a very undesirable drain on battery resources. The scheme suffers from one major drawback which is the need for key distribution before the network can begin operating.

Several protocols have been developed with wireless sensor networks in mind. Sensor nodes tend to be computationally constrained as the devices can be physically small, and are generally static rather than mobile. However, several protocols show areas of their schemes that are also useful for mobile networks. SPINS is a scheme where individual key pairs between nodes and a central base station are pre-installed (Perrig, Szewczyk, Wen, Culler and Tygar 2001). Whilst this topology is not MANET-based, a scheme within the protocol called μTESLA is used for authenticated broadcast (Perrig, Canetti, Briscoe and Song 2000). This scheme uses a pre-distributed commitment which is effectively the last key in a one-way key chain. New keys are hashes of the old key,

allowing nodes receiving the key to verify that they have originated from the original key. The sender includes a Message Authentication Code (MAC) with the message which is stored by the receiver until the later key is disclosed. The nodes require a level of synchronisation to ensure that the correct key is used at the correct time to verify the MAC. By incorporating μTESLA in the protocol, authentication of messages is possible to ensure that they originated from the base station, adding a level of security to the protocol. Whilst using a base station in the network is not suitable for truly ad hoc situations, the authentication of messages is desirable. If all nodes in a MANET are preloaded with the commitments of the key chains of all other nodes, then each node can operate as a base station for its own messages, providing authentication for all messages sent. One major benefit of authentication for every message sent is that intrusion resistance is very secure and Byzantine behaviour can be quickly dealt with by excluding the rogue nodes.

Whilst SPINS with μTESLA is useful for secure unicast, GKMPAN is designed to be secure for multicast (Zhu, Setia, Xu and Jajodia 2004). Here, the SPINS protocol utilising pre-shared group keys is extended to provide an efficient method for revocation and rekeying. GKMPAN assumes that each node has been pre-installed with a subset of symmetric keys in a large key pool, as well as the group key and a commitment. The commitment is used to authenticate any revocation messages sent from the key server. The ID of the node determines which keys in the key set the node possesses. To revoke the keys of a node, a revocation message is broadcast to all nodes with the ID of the node to be expelled. The nodes can tell from the ID which keys are held by the node and all of those keys are erased. Additionally, the revocation message identifies a key not known by the expelled node that should be used to derive the new

group key. The old key and new key data are used to calculate the new group key for those nodes that possess the update key data. For those nodes that do not possess the update key data, a message encrypted with one of the keys in their key set is sent by the key server. A hierarchical structure is used to disseminate the key update messages to those requiring it. The old group key and new group key coexist until it is confirmed to the key server that all nodes now have the new group key. This adds overhead with confirmation messages having to flow back to the key server until the old group key is no longer used. It also allows a time lapse where the node to be expelled can still receive and decrypt messages with the old key. It must be assumed that all nodes can reply to the key server in a timely manner with their confirmation messages, which in a static sensor network may be a justified assumption. However, in a dispersed network with unreliable links this may mean the expelled node is not actually expelled for a long period of time. Should another Byzantine node wish to disrupt the revocation process, it simply need not reply that it has received the update key and the old key will continue to be used. Whilst bandwidth efficient in the sense that many nodes may be able to calculate the new group key themselves, requiring confirmation from every node that they have received the key will then use valuable bandwidth for administration which is always undesirable.

Another scheme that is designed for wireless sensor networks is called Secure Pebblenets (Basagni, Herrin, Bruschi and Rosti 2001). Here a pre-shared group key is installed in the nodes prior to network formation. The symmetric traffic encryption key is used to encrypt and decrypt HELLO messages between nodes. The main difference from many other wireless sensor protocols is that the network is arranged into clusters with the cluster heads one-hop neighbours of the cluster nodes. The cluster heads form a

backbone for the network and compete to become the key manager. Once chosen the traffic encryption key is distributed to the cluster heads who then distribute it to their nodes. Periodically, the clusters are reorganised and at the same time a new traffic encryption key is distributed. The purpose of the reorganisation is twofold: firstly to ensure the clusterheads' batteries are not exhausted too quickly as they have a higher computational load, and secondly to account for the mobility of the nodes with some nodes moving to within one hop of the clusterheads and some moving further away. The drawbacks of this protocol include a single point of failure by utilising a key manager, no replay protection for messages and the ability for any node with the initial group key to derive later keys even though they may have been expelled. However, utilising clusters does have advantages of bandwidth efficiency by delegating responsibility for communications within clusters to clusterheads rather than a single network head.

A very simple approach to the problem of initial key exchange between nodes is proposed by Anderson et al (Anderson, Haowen and Perrig 2004). Infection is designed for an ad hoc sensor network and here a node will transmit on a very low power setting, sending a message containing its symmetric key. If no reply is heard, the power is increased incrementally until a node replies with its own key. The scheme relies on the probability that a potential attacker will be out of radio range at the time of key exchange. By whispering the key it is hoped that an adversary is too distant to hear the exchange. Keys are exchanged with neighbours in this manner until eventually all nodes in the network share at least a key with one neighbour. Security is minimal with only good luck relied upon for keys to remain secure against unauthorised nodes. Assuming that an adversary has not heard the key is unrealistic, as it is possible that they have and therefore communications are not secure when it is thought that they are. Additionally,

authentication of nodes is not dealt with so that the neighbour whom you are exchanging keys with may be the very adversary you are trying to avoid.

An extension to the protocol to deal with mobility of the nodes is offered by Hwang et al (Hwang, Han and Nam 2006). In their modification, a node that has moved out of range of its initial neighbor may be able to exchange keys with a new neighbour. This is done by using the previous key as authorisation to join. Here, the new neighbour queries the old neighbour by asking if the key and ID of the node are known to them. If they answer positively, then authority is granted to the node to exchange keys with its new neighbour and effectively rejoin the network. If the key or ID are not known, then the node is suspect and a key exchange is denied. Whilst adding the ability to allow a node to move throughout the network, security is not enhanced and Byzantine behaviour may lead to legitimate nodes being excluded or unauthorised nodes eavesdropping on the key exchange.

A further protocol designed for static wireless sensor networks is LEAP+ (Zhu, Setia and Jajodia 2006). This is a modification to the authors original protocol, Localised Encryption and Authnetication Protocol (Zhu, Setia and Jajodia 2003). This scheme is designed for static nodes pre-configured with individual keys and a separate group key. The individual key is used to derive pairwise keys for communicating securely with a one hop neighbour. During one hop neighbour discovery, a master key is derived by the node from its node ID and the pre-installed individual key. The master key is used as authentication to sign HELLO messages to it neighbour. Any node possessing the initial key can calculate the master key of its neighbour and authenticate the message. Once a HELLO message is received and authenticated, the pairwise keys used for

communication can be calculated. Once the pairwise keys have been agreed upon, the network group key is deleted from the node. This gives protection to the network should a node be compromised. However, the major drawback of this protection mechanism is that with the deletion of the key goes the erasure of information allowing communication with any other node. Therefore, once network setup is complete, no further nodes are permitted to join. Communication between several nodes at once in a local broadcast is achieved by nodes distributing a cluster key to all nodes that it has a pair-wise key in common with. Whilst secure once network setup is complete, the inability for mobility and new nodes that are ignorant of the initial group key to join make this protocol unsuitable for a dynamic network. Table 4.3 shows the characteristics of the distributive symmetric schemes. The table is modified from that derived by Hegland et al (Hegland, Winjum, Mjolsnes, Rong, Kure and Spilling 2006).

Table 4.3: Characteristics of distributive symmetric schemes.

	PSGK	SKiMPY	S-HEAL	LKH	Pre-Dis	SPINS	Pebblenets	Infection	Leap+
Group/Individual	G	G	I	G	G	I	G	G	G
Trust	Pre-shared	Head	Head	Head	Head	Head	Offline	No	No
Authentication	Offline	Key	No	Key	Key	Offline	Offline	No	Offline
Redundancy	No	No	No	No	No	No	No	No	No
Scalable	Good	Fair	Fair	Fair	Fair	Poor	Fair	Good	Poor

4.6 Conclusion

There are various approaches to the problem of providing effective and efficient key management in mobile ad hoc networks. Each proposed scheme has benefits and drawbacks. This trade-off between benefits and drawbacks means that schemes can be suitable for one type of application but unsuitable for another type. The following chapter describes the process of designing the new protocol and how these benefits and drawbacks have been utilised to provide a unique scheme that is suitable for the types of applications described in Chapter 1.

Chapter V

DESIGN OF THE KEY MANAGEMENT SCHEME

5.1 Introduction

Many previously developed protocols are a compromise between security and efficient key management. A very secure scheme is often complex and difficult to use. A very simple and easy to use scheme often provides very low security. The challenge therefore is how to have both attributes to a sufficient degree so that the protocol operates efficiently and has effective security. Most protocols implement key management services before network deployment or specify that key exchange must be done using a secure side channel such as infrared. Offline configuration ensures key creation and distribution can be done securely prior to network deployment. Infrared equipped laptops and PDAs generally have a very limited range of only a few metres and require direct line of sight in a narrow beam between two devices, making it suitable only if prior planning for the network is done. Both these approaches make attacking the key exchange process almost impossible. Whilst these methods provide very high security, at least until the network begins key management functions after deployment, it is not practical for many applications. Very few proposed protocols provide key management where no offline configuration or secure side channel is utilised. This is because providing key management entirely after network formation where a network grows from nothing to a highly dynamic network with members joining and leaving, networks merging and splitting, and misbehaving members identified and ejected, is extremely difficult.

As the purpose of this research is the design and testing of a protocol for a truly ad hoc network that begins with a single node growing to a large network with a high number

of nodes, it assumes members have no prior knowledge of each other. This means all key management services must be performed online after network formation with wireless radio communication. Additionally, it is designed to deal with high mobility of some or all nodes and assumes a percentage of malicious nodes that misbehave in some way and must be identified and ejected from the network.

With one possible application for this protocol being disaster recovery where disaster victims may be able to establish a network for communications where other communication infrastructure has failed, the main focus is to very quickly allow anyone to form or join the network and monitor behaviour from that point onwards. Another application may be in an educational environment where students are in a large area collecting data for analysis. Here, every student is considered equal with no node superior to another. These types of applications allow every member of the network to share responsibility for helping to form and maintain communication links, all the while maintaining a fairly high degree of security to ensure the network is resilient against internal and external attacks. Even in these scenarios, security is vital for several reasons. Firstly, some messages may be highly confidential between two members and must remain private from all but the intended recipient. Secondly, if any node can join the network but never be excluded, a misbehaving node could seriously disrupt the running of the network by excessive message sending, failing to pass on messages or sending false messages.

It is desirable then to have a protocol in place that can efficiently provide all three key management functions: key creation, key exchange and key revocation. This unique approach presented challenges that necessitated a very structured approach during

design, beginning with the features this type of network requires. This was formulated into ten requirements of the design. Previous schemes were examined and where possible, the best parts of those schemes were identified and incorporated into the design. Any drawbacks of these schemes were used as warnings to avoid implementing anything that would serve to undermine security or efficiency. After examining many previous schemes, seven protocols were found to have parts of their design that were useful, and these parts or general features were incorporated along with new ideas to form a new and unique encryption key management protocol that is dubbed SKYE, which stands for Secure Key deploYment and Exchange.

5.2 Key Features of the Design

By looking at the requirements for the protocol, the desirable design features can be identified.

Requirement 1*: Any node should be able to join the network. No prior knowledge or offline configuration should be necessary.*

Response: The combination of not having any offline configuration but using digital certificates to bind keys to identities leads to self certification of nodes or servers issuing certificates to nodes. Using a certificate authority requires a choice of how many servers should be required before a certificate can be issued. In the MOCA protocol, a minimum of 20% of nodes are designated as servers, and of those 15-20 are required to be contacted for KMS services. However, this assumes that the network begins with 100 nodes or so as any less would mean that not enough servers are available to provide certificate services. With the network beginning with a single node and growing

dynamically, the new protocol utilises a minimum server's required threshold but overrides that rule until enough servers are present to provide the services.

Requirement 2*: Key management messages should be the least number possible to provide the service. The number of messages utilises more bandwidth than the size of the messages, so larger messages but less of them is desirable. Additionally, using computationally complex algorithms for messages requiring low or no security wastes time and drains valuable battery power and therefore a choice of encryption methods should be available.*

Response: From a security standpoint, some messages do not require high security, or at times even any security. Encryption and decryption calculations require considerable CPU usage. For messages that require little or no privacy, it is therefore desirable to send them unencrypted or with lower encryption saving valuable battery power. This leads to the desirable functionality of ranking messages with a corresponding security level and using the appropriate encryption for the level. There should be three levels of security for messages similar to that used for military communications. That is, unclassified where no encryption is used, classified using symmetric encryption and secret utilising asymmetric encryption. The appropriate level allows for a choice of encryption and therefore computational power required to encrypt and decrypt the message. A default level of encryption could be network specific, but as a baseline all messages should be sent as classified unless otherwise desired.

Requirement 3: *Keys should only be exchanged with nodes that the sender wishes to communicate with.*

Response: Exchanging keys with only those nodes that it is necessary for communication saves considerable key management overhead. Some schemes perform key swapping between all nodes even though they may never use those keys to communicate. The sender first selects the security level, open, symmetric or asymmetric encryption. Then, the appropriate steps are performed until certificate validation and key exchange have taken place.

Requirement 4: ***All authorised nodes should be able to communicate with all other authorised nodes in the network.***

Response: Provided the communicating nodes are part of the same network, the routing protocol employed should ensure that contact can be made between the two nodes. For key management, the choice of encryption utilised for the message will determine what, if any, keys need to be exchanged. Only nodes holding valid certificates are authorised on the network and can exchange messages, but nodes that have not yet been issued certificates can pass on certificate issuance requests.

Requirement 5*: Certificates binding keys to node's identities should be used for high security.*

Response: The use of digital certificates binding keys to the node's identity raises two points. Firstly, the identity of the node must positively identify the node and must not be able to be changed or spoofed. Protocols often utilise the MAC address or IP address of the node as their identity. The IP address is unsuitable for this type of network as a node may leave the network and upon rejoining will be issued a new IP address. However, the MAC address is suitable as it is unique to the device. A more robust method could be implemented where the identifier for the device is constructed when the software is

installed. If the identifier is based on unique hardware features such as CPU number and hard drive serial number and this is hashed and used in a digital certificate contained in the device, then the certificate could be sent along with the request to the servers for a PKI certificate. Altering the internal certificate would be almost impossible, and creating a new certificate would involve reinstalling a fresh copy of the software and changing hardware devices so that the same hash was not created. Whist possible to do this, it would be a reasonably skilled and time consuming exercise making this type of deceptive activity highly unlikely, especially on more than one occasion. Secondly, digital certificates must be created, issued and stored and a record kept by servers in a Valid Certificate List (VCL). Additionally, a Certificate Revocation List (CRL) must be maintained up-to-date and readily available to all nodes at all times. This presents a considerable challenge. To avoid offline configuration to install the digital certificates, self certification may be employed where the nodes create their own certificates permanently bound to their identity. This complies with the truly ad hoc nature of the design where all nodes may join the network initially. To ensure robustness against attack, redundancy which provides fault tolerance and high availability, a distributed CA using threshold cryptography is employed. The CA will comprise of *k* nodes out of *n* nodes in the network. The total servers in the network is *m*, which at times will be more than *k*. If so, a subset of the *m* nodes (*k)* must combine to provide CA services.

Requirement 6: ***If a node misbehaves, it should be identified and if necessary permanently ejected from the network.***

Response: To eject a node from the network, misbehaviour must be identified and noted. Neighbouring nodes are responsible for monitoring each other's behaviour. Each node joins the network with total trust. The trust is measured from full trust of 1 to very

low trust of 0.1. This certificate chain trust calculation is used unaltered from that used in Composite Key Managment (Yi and Kravets 2004). Each instance of malicious behaviour identified reduces the trust in that node from the accuser by 0.1. The trust level is used when a node requests a certificate. The trust level along the certificate chain is calculated along with the attenuation factor and this final calculation is used by a server to decide whether a certificate should be issued or not. This method assumes that the likelihood of a node being compromised is equal for every node in the network with probability *p*. The length of the chain to the server is *d*. Therefore, the probability that the chain has not been compromised is $(1-p)^{(d-1)}$. The calculated level is compared with the network attenuation factor threshold. If it is at or above the threshold a certificate is issued. If below the threshold the request is refused and the node must make another request. Nodes that note misbehaviour advise the CA of their accusations against a node. A threshold of accusations within a time limit must be received by the CA before a node is ejected. To eject a node, the node's identity is added to the CRL and a broadcast message with the nodes identity and the revocation status is sent to all nodes.

Requirement 7: ***A single node should not have the power to eject another node.***

Response: A threshold of accusations is required to eject a node ensuring that no single node can maliciously eject any other node.

Requirement 8: ***An excluded node should still receive vital information.***

Response: Any messages deemed unclassified such as messages about rescue efforts or warnings of impending danger in a disaster situation are sent unencrypted and can be

read by all nodes, including those ejected or those who are not currently part of the network.

Requirement 9: ***The network should be highly scalable.***

Response: For a network to be highly scalable, nodes must be able to communicate at the same time without interfering with others communication. With a maximum radio range of approximately 300 metres it is envisaged that with a widely dispersed network only limited numbers of nodes will be within range of each other. The network uses the limited radio range of the devices along with the key exchange on a demand basis to assist with scalability.

Requirement 10: ***The network should handle mobility of nodes seamlessly.***

Response: Mobility of the nodes should present no problems with the key management as key exchange is on a demand basis and network wide protocols are employed where geographic relocation of the nodes will make no difference to the key management functions employed. Additionally, the CA will be dispersed and the same number of threshold CA servers will need to be contacted for certificate services wherever the node may be. Therefore, mobility of the nodes will effectively be transparent to the members of the network including the servers.

5.3 Design Steps for the Proposed Protocol

The first stage of design is to examine previous protocols and to select those features that are desirable for the current design. Modifications to identified features from previous protocols may be necessary. Eleven previous relevant schemes were identified and entered into a table showing the stages from Initial conditions through to ejecting

nodes from the network for misbehaviour. This list of eleven was then reduced to seven schemes that were the most relevant. A list of benefits and drawbacks to the schemes is made below each scheme. Each scheme deals with stages in different ways. Some schemes only look at certain stages and either the scheme is not designed to deal with some stages, or it ignores the stage and is designed as part of a scheme only. Table 5.1 shows the seven most relevant previous schemes and their characteristics.

Table 5.1: Comparison of Protocol Features.

	MOCA 2003	SOPKM 2003	URSA 2004	Composite Key Management 2004	SEKM 2005	FSOPKM 2005	Improvement of Threshold Signature 2005
Initial Conditions	Certificates installed offline Server nodes chosen	All nodes equal Certificates installed offline	Offline configuration for initial nodes	Any protocol can use this addition eg MOCA	Certificates installed offline Server nodes chosen	Nodes generate their own key pairs	Same as SEKM
Key Generation	Updates performed at server nodes	Updates performed with neighbours	Initial nodes generate tickets		Updates performed at server nodes	Nodes generate their own key pairs	Verification of messages included
Key Distribution	Servers each send their share to the requester. Requester combines shares into certificate	Neighbours compare certificates to repositories at each communication. Update if necessary Repositories merged	Tickets propagate through the network	Certificate chaining used where each node in the chain multiplies the trust level by their trust value	Requester contacts one server who then contacts threshold of other servers	Certificates only exchanged with one-hop neighbours	Message origin traced to prove authentication
Key Revocation	CRL maintained Threshold of accusations must be received within set time	CRL maintained at each node and exchanged with neighbours regularly	Behaviour constantly monitored. Consensus of nodes agree to revoke ticket		CRL maintained Threshold of accusations must be received within set time	Node revokes its own key and generates replacement	Same as SEKM
Benefits	Combiner node not required More server nodes than threshold required increasing availability of servers	Key updates kept local minimizes communication and maximizes availability of services	New nodes can join the network Nodes trusted until prove otherwise	Consensus of trust Threshold for trust can be varied to increase security	Taking turns at being active servers preserves battery life	Certificates generated by nodes Minimal key distribution messages	Increased security False message injection unlikely with message origin authentication
Drawbacks	Offline configuration Server nodes designated before deployment New nodes cannot join unless preconfigured	Offline configuration New nodes cannot join unless preconfigured	Node ejected is permanently barred from the network	Addition to an existing protocol to provide trust evaluation only.	Offline configuration New nodes cannot join unless preconfigured	Node monitors its own behaviour.	

By comparing these seven protocols, the different approaches to solving the challenges at each stage could be easily compared. It was found that some protocols had desirable features that improved security, but the cost of implementing the feature was too high for a highly dynamic and dense network. For example, micro Tesla (μTesla) is a method

of verifying the authenticity of a message by asking the sender if they sent a message that has just been received. Whilst this adds security, the cost of sending messages to verify messages is too high, greatly increasing network traffic for little gain. Therefore, whilst a benefit in security, it has a significant drawback in efficiency and so was not implemented. A table was then created showing the stages of the protocol and the benefits and drawbacks to all seven schemes. This led to a selection of protocol features that were desirable enough to be included in the protocol, and drawbacks that should be avoided. Table 5.2 shows these features.

Table 5.2: Benefits and Drawbacks of Features.

	Benefits	Drawbacks	SKYE Protocol
Initial Conditions	All nodes equal fits in with a truly ad hoc model. Network wide CRL prevents attacker moving to join network after being ejected. Variable parameter for server nodes makes security level tunable.	Base key later used to generate subordinate key means base key cannot be read. If base key is then destroyed even a captured node will not surrender the base key. Offline configuration difficult. Maintaining synchronisation of repository difficult. Transitive trust means checking certificate chain for each new communicating node. Binding identity to a key is difficult if self-created.	Offline configuration is not effective. Self certificate creation and signing scales well and allows unknown nodes to join. Ranking messages may add efficiency and reduce encryption computation. Some messages relevant to all nodes and not sensitive should be sent in the clear. Identity must be bound to the node permanently. IP or MAC can be spoofed.
Step 1: Server Group Join Request	Merging repositories on initial contact propagates certificates efficiently.	Nodes designated as servers puts more computational load on device. Servers create superiority which is not truly ad hoc.	Using a server group makes node to server communication easier. Server groups can handle server to server requests transparently to the rest of the network. Choose server nodes - how? Choose server group by threshold and location – how?
Step 2: Certificate Signing Request or key distribution.	If every node knows every other node's public key, key swapping is no longer necessary. Partial signatures held by server nodes makes intrusion more difficult. Requester only needing to contact one node increases chances of successful request for updated certificate. Certificates expiring increases security. Using routing protocol messages for key swapping is efficient but relies on the routing protocol to function.	Self-generated certificates require the node to self-monitor which is not reliable. Tying a KMS to one specific routing protocol may be too constrained. Using a calculation of trust along a chain gives an average trust rating which adds tolerance against malicious nodes. Key swapping with every node does not scale. Combiner node creates point of failure. Certificate expiry requires recertification	Get key swapping done with only those nodes that you need to communicate on a demand basis. No expiry of certificates – explicit revocation? Using routing protocol for key exchange may be sufficient for proposed protocol. Certificate chaining with a calculation of trust along the chain.
Step 3: Server Certificate Update Request	Servers requiring update may foil a compromised server. Taking turns as servers shares computational drain. Local servers only serving local nodes reduces message requirements.	Servers requiring update adds messages	No update necessary for servers. Taking turns as servers increases efficiency but at the cost of security. Choice.
Step 4: New Server Joins	New servers able to join adds ability for servers to join and leave		Ad Hoc Server creation adds flexibility and scalability.
Step 5: Revoke Node's Certificate or key	Misbehaving nodes can be ejected.	Timer expiring on a certificate requires synchronisation which is difficult. Totally banning a node may be too harsh.	A banned node should still hear relevant messages. Banned nodes must not be able to change identity. Threshold of nodes needed to eject node.

With the desirable features identified, those features were incorporated into SKYE, but at times with necessary modifications. As SKYE is a full protocol, it must have all stages from initialisation to revoking certificates and ejecting nodes. New features to the design were added to increase security and efficiency, and to give choices for implementation to enable the protocol to suit many different applications. These choices included such tunable parameters as the number of servers required for key management services and the trust level along a certificate chain that must be above a certain level for the certificate to be issued.

The design of this protocol is intended to be effective in a large, geographically dispersed ad hoc network environment where network formation can be instigated by the first two nodes in the area that wish to communicate. The incorporation of three levels of encryption ranging from no encryption to public / private key encryption allows a choice of privacy meaning ejected nodes can still read important broadcast messages and highly sensitive messages can be sent in an extremely secure form. Additionally, the use of PKI and digital certificates means nodes can be assured that any node communicating with them is authorised as part of the network. The issuance of a certificate by the CA resulted in the key parts and signature being assembled by the requesting node as proposed by Shoup (Shoup 2000).

Remaining with the truly ad hoc network nature of this design, no offline, side channel or prior configuration of the nodes with certificates or keys is incorporated. This is a departure from the more common method of installing keys prior to network formation or using a secure side channel such as infra red to disperse the keys before the network begins operation. Whilst this adds considerable challenges to producing a secure yet

efficient protocol design, the ability for any node to create or join the network at any time is one of the advantages. The following events that occur as the network develops are discussed.

Event 1: Initial Conditions

No preconfiguration is done with the nodes and so potential network members have no knowledge of the network topology or of any other node that may form the network. Nodes have a non-changeable identity bound to hardware as discussed earlier in the chapter. Nodes must be equipped with the protocol software and this could be available for download from a web site or similar.

Event 2: Network Formation

When an event occurs that will lead to the desire to form an ad hoc network, potential members are not aware if any nodes have already formed a network. Any node that wishes to communicate with another node will initially have no encryption keys or public key certificate. A node first listens for any network traffic to determine if a network has been formed. After time *t* (where *t* can be an arbitrary measure of time in seconds), if no traffic is heard, the node sends a broadcast message to request a certificate. The node listens for a reply. If no reply is heard, the node assumes it is the first node or that it is out of range of other nodes. The node waits for time *t* and repeats the process. If a reply is received, the reply includes information about the number of servers required and how many servers are currently in the network. If a node receives a message from another node, then negotiation takes place to decide on parameters. Default parameters are designed into the protocol as a result of simulation results, but

these parameters are tunable in order to cope with the application the network is utilised for.

Event 3: Server Selection

The protocol recommends a default of 3 servers required to collaborate before a certificate is issued. Servers are chosen using the server rule for the network. A default rule exists where servers are chosen as those nodes having the most number of one-hop neighbours (The Most rule). There are optional rules available if an option will increase efficiency. These options are:

- Random selection (Random)
- Least number of neighbours (Least)
- Most neighbours updated (Most Updated)
- Least neighbours updated (Least Updated)

As all nodes are considered equal, server selection is purely based on optimal location of the nodes. The number of servers in a network is governed by the server percentage rule. The default is for 20% of nodes in the network to be servers. This percentage can be changed if a change will increase efficiency or if greater security is required. Fewer servers lead to fewer targets for a potential attack. Increasing the server percentage means certificate services are more efficient but at the cost of decreasing security.

One problem with strictly enforcing the percentage of servers required for certificate issuance is that the network must contain a reasonable number of members before enough servers are present to issue certificates. For example, if three servers are required and 20% of nodes are servers, then fifteen members must be present before three of the members will be servers. The point of the network creation is to rapidly

afford communication and therefore a compromise has been designed. The protocol rules allow for overriding the server percentage rule until enough members exist where the rule can be reinstated. For example, if three servers are required, then the first three members of the network will all be made servers. At this point, certificates can be issued and communication can begin. For this example, no more servers would be added until twenty members are present where one more server is added making four servers and therefore 20% of nodes are servers. Whilst reducing security, the tradeoff with permitting communication to begin quickly is deemed worthwhile. It is possible that in certain areas, small networks may form that never reach fifteen members, and so over-riding the server percentage rule in this case allows communication where it is possible no communication would ever begin if the rule were strictly enforced.

Event 4: Certificate Request

A node wishing to join the network will first check that there are sufficient servers to issue the certificate. If there are sufficient servers, then the node will send the request to the closest server if this is known. If not, a broadcast message is sent asking servers for a route to them. Any servers receiving the request will reply and the route is then returned to the requester. The node replies to the closest server, where closest means the least number of hops away. If the node itself is a server, then it is considered the closest server and acts in that role. The closest server, dubbed the first server, then takes responsibility for contacting the other servers. When requesting a certificate from the server, the node's non-changeable identity is supplied and incorporated into the certificate. This prevents ejected nodes from disguising themselves and attempting to rejoin the network. For example if one server is required, then the full certificate with signature is created by the first server, which may be the node itself, and returned to the

requester. If three servers are required then the first server contacts the second and third servers. These servers reply with a third of the signature each and the first server adds the certificate and another third of a signature and returns these to the requester. The requester stores the certificate complete with public and private keys, and combines the three partial signatures into a complete signature which is used to sign the certificate as authentic. The requesting node now has a full certificate with signature and is authorised to join the network.

Event 5: Certificate Revocation

Revoking a certificate becomes necessary for several reasons:

Node leaves:

A node may leave the network permanently through choice. If this becomes known by the servers, they will agree to revoke the certificate. They will remove the node's certificate from the VCL and enter it on the CRL. Should the node wish to rejoin the network, it must request a new certificate.

Multiple Certificates:

A node may hold a certificate but have joined another network and been issued with a second certificate. If so, the node must surrender its redundant certificate by advising the servers that it has another, older certificate. The node's certificate will be removed from the VCL and entered into the CRL.

Malicious Behaviour:

If there are a number of malicious actions by a node, the node will be permanently ejected from the network. An action that is considered malicious and is noticed by another node will lead that node to make an accusation against the offending node. An accusation is made to a server with the offending node's identity. The default is that if five accusations are received by the servers from different nodes within the default time of sixty seconds, then the accused node is permanently ejected from the network. An accusation list is maintained by all servers that periodically broadcast the list to each other to ensure it remains updated. If the threshold of accusations is received within the threshold time, the node's certificate is revoked by removing the certificate ID from the VCL and entering it on the CRL and entering the node's ID in the Ejected Node List (ENL).

Event 6: Multiple Networks

Due to the dynamic nature of ad hoc networks, at times smaller networks will join together when at least one member of the network becomes in range of a node from another network. Figure 5.1 shows a total of 47 nodes, some of which have formed into networks and some of which are out of range of another node and are therefore alone (indicated by the large circle representing the 300 metre radio range). A similar event will occur when a node bridging two parts of a network moves or leaves the network. In this case, a larger network will become two smaller networks. This merging and splitting of networks will require adjustment of the servers to ensure the correct numbers of servers are in the networks at all times.

The scenario in Figure 5.1 requires four servers to obtain a certificate (red labels) and so nodes in networks with less than four servers have not yet been issued certificates (green labels).

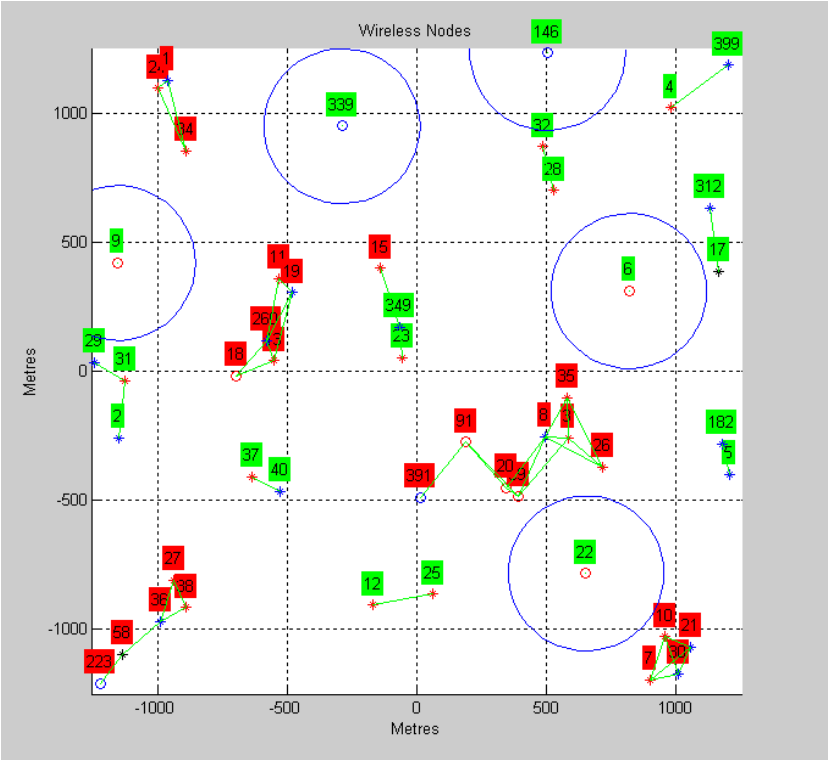


Figure 5.1: Wireless networks – 4 servers required.

Node 15 in the centre has moved from the network to its left where it received a certificate and has now joined a network with only two other members. Node 15 retains its certificate but can only communicate without encryption as the other two network members do not have certificates or encryption keys. A different scenario has occurred in the top left of the network where a fourth node in a network has left the network (shut down) leaving three members. As there were four members and therefore four servers, they have all received certificates. Even though there are now insufficient servers to issue a certificate, the members retain their issued certificates and can continue to communicate. Only one server need be contacted to check a certificate. This compromise allows communications to continue as the certificates were issued securely

when the required number of servers was present. The compromise is that four servers are also required to eject a node. As there are only three servers present, no node can be ejected until at least another one or two servers are present (assuming a server node will not collaborate to eject itself).

Network Merge:

If two separate networks merge together, a calculation must be done to ensure the correct percentage of servers are in the network. First, servers merge their certificate repositories so that nodes with valid certificates can still use those certificates in the modified network. Secondly, a calculation is done to check that the correct percentage of servers is present. If there are too many servers, then the excess number of servers will give up server status. Servers are chosen by being in the worst position. Here, worst is the opposite of best for the server location rule. If the default rule of servers chosen because they have the most number of neighbours is in place, then the server with the least number of neighbours will surrender its server status. If more reduction of servers is required, then the next worst positioned server surrenders server status and so on until the correct percentage is reached.

Network Split:

If a network is split into two smaller networks, then a calculation of the number of nodes and servers is made to check that correct percentage of servers is present. A similar process occurs as when the networks merge to either add more servers or reduce the number of servers until the correct number are present.

Event 7: Message Exchange Requests

Nodes that wish to communicate with another node have three choices for encryption of the message. They may choose no encryption in which case the message is sent unencrypted and therefore insecure. This may be useful for broadcast messages especially in a disaster relief situation where ejected nodes should still receive support messages such as advising of life saving instructions. The second choice is to use asymmetric encryption which is considered very secure but at the cost of considerable CPU cycles and therefore more rapid battery drain. The final choice is symmetric encryption which is less secure than asymmetric encryption but may be sufficiently secure for most messages. The CPU cycles required for encryption and decryption with a symmetric key make this the most likely choice for all but highly sensitive messages or broadcast messages sent unencrypted.

Asymmetric message passing

If a node wishes to communicate with another node using asymmetric encryption, the process is as follows.

1. Sender contacts a server to check the validity of the receiver's certificate.
2. Server first checks sender's certificate.
    a. If not valid, ignore message.
    b. If valid check receiver's certificate.
    c. If the certificate is not valid, server advises sender who does not send a message to the receiver unless they choose to send the message unencrypted.
    d. If the receiver's certificate is valid, advise sender and carry on.

3. Sender sends a message to the receiver requesting communication. This message also contains the sender's public key.
4. Receiver contacts a server to check the validity of the sender's certificate.
5. Server first checks receiver's certificate.
    a. If not valid, ignore message.
    b. If valid check sender's certificate.
    c. If the certificate is not valid, server advises receiver who refuses communication.
    d. If the sender's certificate is valid, advise receiver and carry on.
6. Receiver replies to the sender with a message encrypted with the sender's public key. The message also contains the receiver's public key.

Communication between the sender and receiver can now continue using asymmetric encryption.

Symmetric message passing

For less secure message passing, the sender may choose symmetric encryption for the messages. However, the initial key exchange must be done using PKI and therefore both sender and receiver must have valid certificates. The process is the same as for asymmetric encryption except that once public keys have been exchanged the first message from the sender will contain a symmetric key encrypted with the receiver's public key obtained during the validation of the receiver's certificate with the server. Messages between sender and receiver will now be encrypted using the symmetric key.

Nodes must have a valid certificate to be authorised to send and receive encrypted messages. If no encryption is used for a message, then it can be sent to a node without a

certificate. However, if a node without a valid certificate wishes to send a message, it will be ignored as only authorised nodes are permitted to send messages. Nodes that have not received certificates are expected to pass on messages when required. This is necessary as if nodes waiting for a certificate did not do so then no messages could reach servers unless they were one-hop neighbours. This would mean certificates would rarely be issued as networks form initially without any members having certificates.

5.4 Summary

Performing encryption key management entirely online is very challenging. In a truly ad hoc network, nodes have no prior knowledge of each other and all messages are broadcast omni-directionally and so are easily intercepted. If the network permits anyone to join, then a certain level of trust must exist between members, at least initially. By monitoring behaviour and likely requiring multiple servers to collaborate to eject misbehaving members, responsibility is shared ensuring democratic decisions. Sharing responsibility for certificate issuance and revocation ensures no single node has the power to permit or eject another node, thwarting a malicious node acting alone. Security is inevitably a compromise between ease of use and robustness, and this protocol makes some compromises in security to ensure it can be used in a variety of scenarios that do not require very high security. It is ideally suited for a highly dynamic situation where establishing and maintaining a network quickly and relatively simply is the priority.

The following chapter discusses development of the model and simulation of the protocol. It shows how the input parameters were adjusted to demonstrate the effects on the outputs and identify trends that developed in the performance of the protocol.

Chapter VI

PERFORMANCE SIMULATION OF THE SCHEME

6.1 Introduction

Simulation of a wireless network requires the combination of several artifacts to be utilised together. The first requirement is a computer capable of running the simulation software. The second requirement is a software simulation package, and the third is a method for collecting and collating the results. The software used for these simulations is a purpose built ad hoc simulator written in Matlab. A discussion of the reasons for choosing this software is made later in Section 6.2. Results were collected and converted to graphs using inbuilt Matlab functions combined with custom written programming code. Some criticisms have been made of simulation research where there is a lack of information given about how the simulations were performed and results collected and interpreted (Kurkowski, Camp and Colagrosso 2005). For this reason information is provided describing the simulation setup and testbed specifications. The methodology followed for the research is that of simulation research described by Pegden et al (Pegden, Shannon and Sadowski 1990). The twelve stages in the methodology will be discussed with how each stage was managed. Thousands of simulations were run and the results collated into meaningful information that describes the performance of the protocol. The first section will describe the process of performing the simulations to ensure all relevant information is given. Next, details of the various simulation runs and the parameters will be discussed, followed by a description of the process of collecting and displaying results.

6.2 Justification of Software Choice

There are several wireless network simulators available, many of them at no cost or with an educational licence for students. A requirement for the simulations was to have a network simulator that would display the simulations in real time as they were run, would be very fast as many simulations would be needed to be run consecutively, and needed to be relatively simple to modify as many different experiment parameters would be changed. The most common simulation software packages used are shown in Figure 6.1.

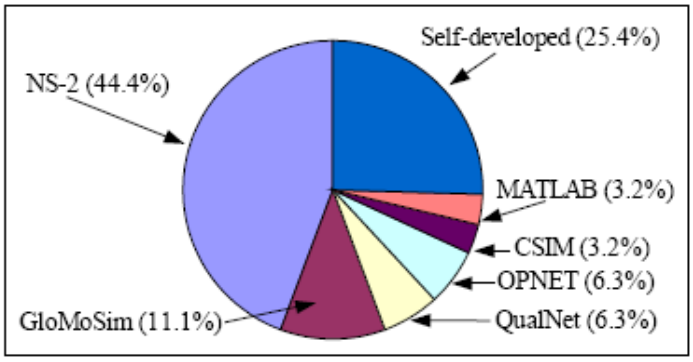


Figure 6.1: MANET simulators used in research (Kurkowski, Camp and Colagrosso 2005).

This research involves testing of the protocol as different variable parameters are altered and observing the change in performance. The focus is on the trends that develop as changes are made, meaning such attributes of wireless signals as attenuation and bit error rate are not necessary to include. They will effectively affect each simulation in a similar way and therefore not alter the outcomes of comparisons that can be made between different settings.

6.2.1 Simulator Software Choices

The most promising of these initially was OPNET (http://www.opnet.com). This is a very powerful simulator encompassing many different parameters of performance including such performance characteristics as bit error rate and signal attenuation. Whilst initially this appeared to be a good candidate, considerable time was spent learning the software. A good graphical interface and the ability to capture output easily were significant plusses. However, the poor manual meant that learning the software thoroughly to modify the base rules, especially for security and key management protocols, proved extremely time consuming and in the end was not possible with the time available. Figure 6.2 shows the graphical display of a network in Opnet.

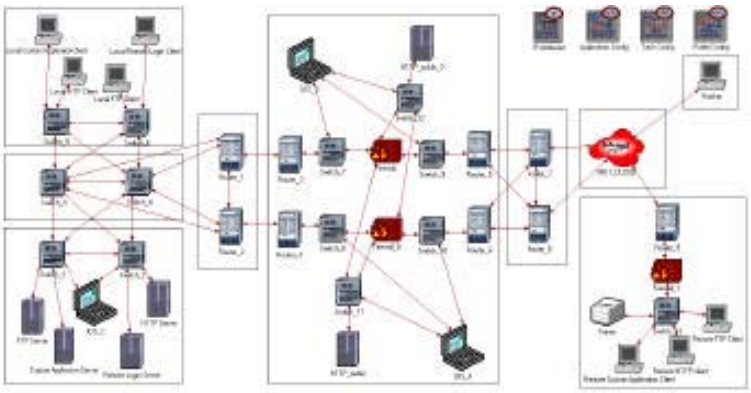


Figure 6.2: Example OPNET Simulation (Zaballos, Corral, Serra and Abella 2003)

NS-2 is a free wireless simulator designed for Linux (http://www.isi.edu/nsnam/ns/). It uses the TCL/C++ language and is well established and often used for simulations. Approximately 44.4% of MANET studies use NS-2 making it the most popular choice (Kurkowski, Camp and Colagrosso 2005). However, it does also have a steep learning curve and as it is constantly updated, available manuals are often out of date (Cavin, Sasson and Schiper 2002). One further drawback of NS-2 is that it does not scale well, especially when more than 100 nodes are present (Naoumov and Gross 2003). As

simulations would encompass several hundred nodes, this simulator was deemed to be unsuitable. Figure 6.3 shows an example NS-2 display.

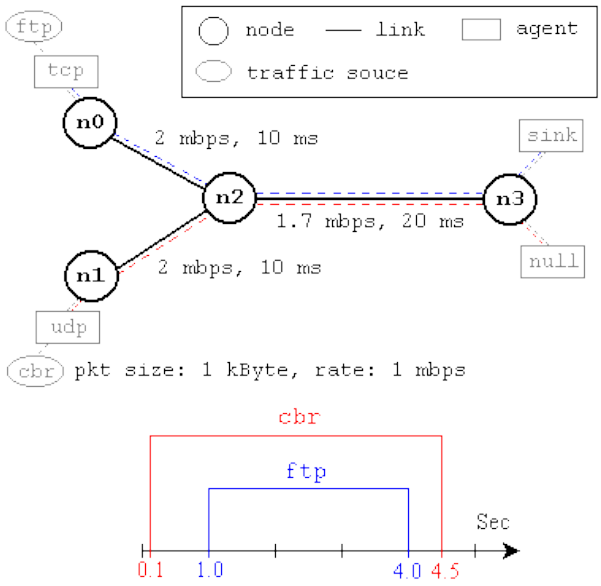


Figure 6.3: Example NS-2 Simulation (Chung and Claypool 2010)

GloMoSim is a simulator developed by the UCLA Computing Laboratory (http://pcl.cs.ucla.edu/projects/glomosim/). The simulator includes several routing protocols and the Random Waypoint Mobility model for node movement. The programming language is Parsec, an uncommon language that was not known. Modification of the protocol rules is difficult and a steep learning curve means considerable time is required to master the software to a sufficient level. Additionally, little information about large network simulations could be found incorporating some risk into using the software, possibly only to find that it could not perform as required. Figure 6.4 shows an example of GloMoSim output.

```
Node:0,Layer:802.11, pkts from network: 0
Node:0,Layer:802.11, BCAST pkts sent to chanl: 69
Node:0,Layer:NetworkIp, Number of Packets Routed


For Another Node: 0
Node:0, Layer:NetworkIp,Number of Packets Delivered
                                    To this Node: 4549
Node:1, Layer: 802.11, pkts from network: 0
Node:1, Layer: 802.11, UCAST pkts sent  to chanl: 2499
Node:    1, Layer:    802.11, BCAST pkts sent to chanl: 67
```

Figure 6.4: Example GloMoSim output (Kunpisut 2010)

Both CSIM (http://www.csim.com/) and Qualnet (http://www.scalable-networks.com/) were looked at as possible choices. However, both are commercial packages that had a significant cost attached. Additionally, they are not commonly used by researchers and so some risk was attached if they were to be chosen. It was decided that these factors made them poor choices in this instance and so both were ruled out.

Finally, Matlab was looked at. Approximately 3.2% of MANET simulation studies use Matlab (Kurkowski, Camp and Colagrosso 2005). However, Matlab uses the Pascal programming language allowing standalone or linked functions to be written. As Pascal was a language already known, this was a significant benefit in reducing the time to learn the software. As the entire simulator was custom written, the simulator falls into both the categories in Figure 6.1 of Matlab and Self-developed and therefore totals almost 30% of those simulators used by researchers. Matlab is primarily a statistical software package which is designed to work with large, multi-dimensional matrices. Matlab was available and it was quickly found that its ease of use was a major advantage. Its ability to deal with large matrices was of significant benefit when considerable numbers of calculations were required very quickly. Finally, the ability to plot nodes with a single ‘Plot’ command made viewing the simulation as it progressed fairly simple. This proved extremely beneficial during testing as the ability to watch the simulations as they progressed formulated new ideas for improving efficiency that were later incorporated into the protocol. With the considerable benefits offered with development of a custom designed simulator, Matlab was chosen as the simulation software and the task of constructing the simulator began.

6.3 Simulation Environment

The simulation setup follows guidelines that ensure every step is thoroughly performed to ensure nothing will be overlooked. Each stage of the methodology is performed before moving on to the next stage, and if necessary stages can be revisited if modifications that are necessary are discovered in a later stage. The following twelve stages were performed during the simulation setup and experimental runs.

Stage one of the methodology involves defining the problem. This stage was performed very early in the research and is discussed in section 1.3 of Chapter 1 in the Research Methodology. The problem is developing a protocol that is superior under certain conditions to previous protocols. The goal of this research is to show, using simulations, that the protocol is efficient in terms of resources available, and effective in terms of the various security attributes that a full protocol is required to provide.

Stage two of the methodology required the planning of resources. This was a straightforward process of ensuring the simulator would run on a desktop computer in a simulated laboratory environment. Matlab is software that is both powerful and efficient in running on a desktop computer. No problems with running a custom design in Matlab were expected and this proved to be the case. Table 6.1 shows the simulation environment.

Table 6.1 Simulation environment.

Artifact	Description
Computer CPU	Intel Pentium 4 2.4 GHz
Operating System	Windows XP Professional v 2002
Simulator Software	Matlab v7.6.0.324
Simulation Type	Dynamic
Pseudo Random Number Generator	Twister
PRNG Initialisation	Varied for each simulation run

Stage three of the methodology involves defining the system to be studied in general terms. This was a long and complex process of designing the protocol and has been thoroughly described in previous chapters.

Stage four involves a conceptual model of the system, including the parameters for the experiments. This first involves defining the fixed parameters for the simulation. Table 6.2 shows the fixed parameters for all simulation runs.

Table 6.2: Simulation fixed parameters.

Simulation area (metres square)	2500
Simulation time (seconds)	600
Nodes present at start	1
Node growth per minute	30
Node leave per minute	10
Node mobility model	RWPM
Node pause time (seconds)	0-10
Nodes malicious (percent)	6%
Malicious message threshold	3
Accusation ejection threshold	5
Accusation ejection timeout (seconds)	60
Simulation runs averaged	10
Communication distance (metres)	300
Private message exchange rate / sec	0.1

Stage five occurs next and it is here that decisions are made as to what variable parameters should be used. This was a complex task as the protocol is designed to provide a very tunable environment meaning many combinations of parameters exist. This process involved two main areas of research. Firstly, previous protocols that were relevant to the current design and secondly the type of application that the protocol is likely to be used for. By examining previous protocols where the authors have provided simulation results, the input parameters could be identified and utilised where appropriate. This gave results that could then be compared to the results of previous protocols to compare performance of the new protocol. Looking at the type of application that this new protocol may be used for, fixed and variable parameters were

identified that are appropriate. These two approaches were not necessarily mutually exclusive. Once simulations had been performed that could directly compare to other protocols in input parameters, comparisons in performance could then be made. The input parameters could then be altered to those more appropriate to the very large and dynamic network topology where comparisons are less easily made as most other protocols have not had similar input parameters or their performance has not been described.

Stage six involves input data preparation where all the inputs for the experiment are decided upon. From stage six onwards proved to be an iterative process, as preliminary experiments were required to help define what variables should be used for the main experimental runs. This meant that as described in stage five, it was necessary to run preliminary simulations to assist with the decision on the range of variables.

Stage seven involved the programming of the code in Matlab to build the simulation software.

This was followed by stage eight where verification and validation of the parameters and outputs was performed. This was part of the iterative process of revisiting stage six through to stage nine where the final simulation runs were prepared.

Stage nine involved initial experiments where results were examined and minor modifications made to the range of variables. From the trial results, server placement modifications were made so that a greater choice existed from the simple random placement. This came from the preliminary results where the possibility of greater

efficiency was identified by choosing different server placement rules. For example, during preliminary runs it was found that up to five servers required for a certificate was the maximum reasonable number as any more than five servers reduced the success rates to an unreasonably low level. This was especially true as the trust level required was also raised resulting in more failures. Very quickly the certificate refusal rate approached 100%. Therefore, the maximum number of servers required was set at five. Any more would be ineffective and so deemed not to be a realistic choice. Other variables such as maximum speed were also experimented with to find a realistic range. With the preliminary simulation results used as a guide, modification to the input variables was made to reduce the experimental runs to a manageable number. Once the ranges of the variables were set, then the simulations could be planned with precision.

Stage ten was next and involved running the final simulations. This took several months due to the very large number of experiments required. Each simulation with the same parameters was run ten times with ten different seeds for the pseudo random number generator (PRNG). This ensured that whenever a random number was generated, each simulation would not generate an identical sequence. Random numbers were used for such things as placement of nodes on the grid, which nodes were mobile or servers, how fast a mobile node would move within the ten kilometer per hour window amongst others. The results of ten simulation runs with the same parameters but with different PRNG seed values was then averaged and the average used as the final result for that simulation. In this way for example, a series of one hundred and ten different simulations would require eleven hundred simulation runs. The list of variable parameters is shown in Table 6.3.

Table 6.3: Simulation variable parameters.

Percentage servers	20-100
Servers required	1-5
Trust threshold	0.1-0.9
Node speed	0-100kmh

During the simulation, nodes in the network would request communication with other nodes in the same network. The process of contacting a server, validating the certificates of both parties, contacting the other node to request communication and receive a reply was performed at each request. The nodes could then choose to use symmetric or asymmetric encryption as desired. The parameter of 0.1 requests for communication per second was set so that nodes that were part of a network and held a valid certificate would request communication with another node once every ten seconds. The purpose of this message exchange was to give an opportunity for malicious nodes to misbehave as the simulation ran so that misbehaviour did not just occur when a certificate issuance request passed through them. Whilst this process added considerably to the realism of the growing networks, metrics were not recorded in the results as this did not add to the performance measurement of the protocol.

Stage eleven involved collating the results and comparing them with what was expected prior to the simulation runs. Results were positive and showed the protocol performed as expected.

Finally, stage twelve involved documenting the results and displaying the identified trends in meaningful ways by construction the tables and graphs of the results.

6.4 Simulation Parameters

Once the first ten stages of the methodology were performed, experimental runs could be made with the protocol. It was necessary to set a baseline of parameters that could be used to compare results with any adjustments made to see if the adjustments resulted in improved performance. By examining previous protocols that use threshold cryptography and any results obtained by simulations of those protocols, appropriate metrics for the results could be chosen. However, choosing appropriate baseline parameters for initial simulations was a complex task which involved some trial and error. As with any new protocol, no previous protocols directly led to assisting with setting the parameters but rather gave generalised guidelines. However, they were used to compare results where possible.

The following sections describe the parameters used for the simulations and the method for obtaining the results.

6.4.1 Blind vs Informed Requests

When a node first appears on the grid, it has no knowledge of whether a network exists or not. Therefore it will broadcast a message announcing its presence and wait for a reply. If no reply is heard it will periodically rebroadcast until a reply is heard. A choice exists once the node becomes part of a network. The node can immediately request a certificate from a server or it can inquire as to whether there are sufficient servers present to meet the minimum servers required rule in place. A request for a certificate when there are insufficient servers is bound to fail, and re-requests will also fail until the minimum servers threshold is met. Experiments were performed to see whether the extra message overhead that is required to ascertain the current server numbers and

threshold required for the network is warranted. If not, then a blind request should be made. If so, then an informed request where the current server rule and server numbers in the network is ascertained before requesting a certificate.

6.4.2 Percentage of Servers

The role of a server node is to perform key management tasks for other nodes in the network. In a truly ad hoc environment, no node is considered superior to any other node. This means that server responsibilities are in addition to the node being a regular member of the network. Therefore, nodes were chosen for no other reason than their location. A node would not consider it desirable to be designated as a server but rather a necessary responsibility that any network member may be asked to perform and therefore must accept. Server responsibilities use resources in the node that it may otherwise use for its own benefit, including drain on battery power and performing key management (KM) services using time that could be used by the node for sending and receiving personal messages. The number of nodes chosen as servers should therefore be kept to a minimum. There should be enough servers to efficiently perform the KM services and provide sufficient security and redundancy for the network, but not more than is necessary and so expose too many targets to attackers who could furnish sensitive information. The baseline for servers was therefore set at 20%. This was less than the MOCA protocol (Yi and Kravets 2003) which is the closest to SKYE in certificate issuance procedures and was found from experimental runs to be sufficient. The MOCA protocol used a baseline of 30% due to strictly enforcing the ‘percentage of servers’ rule which SKYE does not do.

6.4.3 Servers Required Rule

The number of servers required to obtain a certificate and eject a node is a tradeoff between efficiency and security. Requiring more servers adds security by ensuring malicious nodes that become servers cannot maliciously refuse certificate issuance or eject nodes. Byzantine behaviour is malicious behaviour performed by multiple nodes working together to disrupt the network. By increasing the number of servers required for KM services, Byzantine behaviour can be made much more difficult, requiring as many colluding malicious servers as the server rule specifies. Experiments were run with servers required from one to five. One server required meant that any node designated as a server could issue itself a certificate but also meant that a single server could eject a node. Whilst simple and quick to obtain a certificate, security is necessarily low. As the number of servers required is increased, so is the robustness of the certificate issuance and ejection process. As stated, preliminary runs indicated that five servers was a realistic maximum and so experiments with a minimum of one and maximum of five servers were performed.

6.4.4 Trust Threshold

The calculation of trust along the certificate chain from a requesting node to a server was used without modification from the Composite Key Management protocol (Yi and Kravets 2004). In this protocol, trust begins at 1.0 and is reduced by 0.1 for any malicious act noted by a node. Therefore, all nodes began with a trust in each other of 1.0 and this reduced by 0.1 for each malicious act, to a minimum of 0. The threshold for the network which must be met for a server to issue a certificate was varied from 0.1 to 0.9 representing very low security (many misbehaving nodes in the chain) to a very high and secure 0.9. As longer chains had higher chances of having malicious nodes in them,

a threshold of 1.0 was deemed unrealistic, especially as malicious nodes may make false accusations in a deliberate attempt to compromise the chain. Therefore, 0.9 had at least some leeway in malicious behaviour.

6.4.5 Node Growth Rate

It was difficult to determine what a likely node growth rate would be. The experiments were run to determine the performance of the protocol as the network grew from a single node to many, and so reasonable growth should be expected. The likely applications gave some guidelines to the likely growth, but no empirical evidence of growth for these types of networks is available. By examining previous protocols, some guidelines for this figure could be made but overall it was a judgment based on the likely application rather than any empirical data. Using the disaster relief scenario, in a natural disaster such as an earthquake or flood that occurs in a highly populated area, with no prior planning but with residents having laptops equipped with the SKYE protocol, then it is likely they would fairly quickly wish to get communications established. Therefore, a fairly high growth rate could be expected, especially over a large area. The growth rate was therefore set at 30 nodes per minute. It is probable that people would not attempt communication at regular intervals, and so rather than a steady one node joining every 2 seconds, a Poisson distribution is used. This is an established technique for simulating telecommunications joining rates such as phone calls through a telephone exchange. It gives minor bursts and lulls in the growth and so more closely resembles the reality of people joining.

6.4.6 Node Leave Rate

The rate that network members will leave the network by ceasing communications and shutting down their devices would likely be low, especially in the early stages of communication establishment. Turning to the likely applications that this protocol would be utilised for, estimations of leave rates were mostly intuitive. The rates could vary depending on the application and environmental conditions with nodes choosing to leave or communications temporarily failing because of radio interference, radio range exceeded or obstructions blocking signals. The choice of leaving rate was largely intuitive but considered realistic for a network establishing itself in ad hoc conditions. The rate was set at 10 nodes per minute and a Poisson distribution used to more closely represent the reality of leavers with minor bursts and lulls. Any node having left the network had the same chance of being randomly selected to rejoin the network as any other node having left or not yet having joined the network. This mimicked the reality of members temporarily leaving the network due to poor radio communication coverage or simply choosing to take a break from participation.

6.4.7 Node Mobility Model

A mobility model is a rule for how mobile nodes will move within the simulation area. There are several choices available for such movement and these can be grouped into general areas of random models, temporal dependency, spatial dependency and geographic restriction. To decide on which model best mimics the likely movement of nodes, several approaches were taken. First, a review of general literature on mobility models was made. Secondly, a review of previous protocols that employed simulation was made, and finally the likely applications of the protocol. The overwhelming choice for ad hoc simulations and mobility of people and vehicles within areas is the Random

Waypoint Mobility Model (RWPM). Therefore, this model was chosen to simulate mobility of the nodes. An example of a RWPM is shown in Figure 6.1.

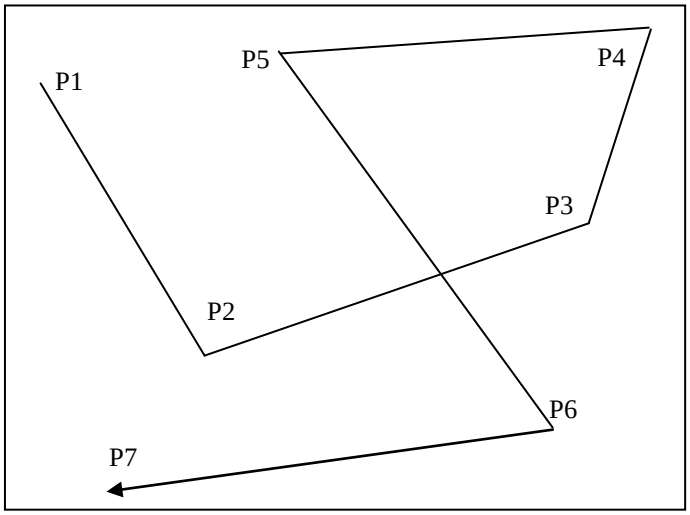


Figure 6.5: Random waypoint mobility model.

In this model, nodes begin at a location on a two-dimensional grid. A random waypoint is calculated and the node moves at a uniform speed towards that waypoint. Upon arrival, the node waits for a certain time (pause time) and then selects another random waypoint and moves towards that waypoint at a uniform speed and so on.

6.4.8 Node Pause Time

Those nodes that were selected to be mobile followed the RWPM model and moved from their origin to various waypoints. The pause times would likely be nil or fairly brief if the nodes were people moving with mobile devices through a disaster area or similar situation. The times would also not be uniform, but rather a random time spent at each waypoint. The pause time was therefore chosen to be between zero and ten seconds and this varied randomly at each waypoint.

6.4.9 Malicious Node Percentage

It is realistic to expect that at some stage, some nodes will act maliciously. An appropriate number of malicious nodes is difficult to quantify for this type of network. Variations would be expected with the application. For example, a military application might expect a high number of attempts at infiltration if it was used in a battlefield situation close to the enemy. If the network was used in a military training situation where only friendly nodes would join, then it is likely no malicious nodes would exist. Once again, the guideline for this figure comes from an examination of the literature and from the possible applications. Very little statistical evidence of malicious numbers of nodes exists and any figures found were generalised. During a natural disaster in 2005 that spread over a large geographical area and may be the sort of situation that this protocol may be employed in, there were reports that rescue helicopters were shot at from victims on the ground. This unusual behaviour shows that at any time attacks could be made on a network even when used to help the very people that may attack it. The number of potential attacks would most likely be low, and turning to the literature for a precise figure led to crime statistics in New Zealand. In 2004, almost six hundred dishonesty offences were recorded per ten thousand head of population (Police 2004). This equates to almost 6% and as malicious attacks in a network are effectively acting dishonestly, for want of more precise data this figure was used as the percentage of nodes within the simulation area that are malicious. As the main goal of the study is to identify trends that occur as variables are manipulated, the precise figures for maliciousness are not important. Provided they remain the same for all simulations the effects will be similar for all results and therefore the trends identified will be similarly affected by maliciousness.

6.4.10 Malicious Message Threshold

The malicious message threshold is the number of times a malicious node will act maliciously. It is necessary to have some tolerance built into the malicious acts as nodes may only misbehave on occasion. For the purposes of the simulation a malicious act is failing to pass on a message to another node when requested to do so. Malicious nodes were set to fail to pass on every third message, whether it was a personal communication or a certificate request. When a message was not passed on, it is assumed the next node in the chain is aware that the previous node has acted maliciously and so an accusation is made to the closest server with the misbehaving node's identity. The server records the accusation and the time that it is made.

6.4.11 Accusation Ejection Threshold and Timeout

This threshold relates to how many accusations against a node must be received within a set time before the node is ejected. This threshold is set at five accusations and the timeout is set to sixty seconds. This means that servers must collectively receive five accusations of misbehaviour against a single node within a sixty second period for the decision to be made to permanently eject the node. The timeout period serves as a buffer for malicious accusations and mistakes where accusations may be made where messages have failed to be passed on accidentally. This also means that malicious nodes must misbehave on a regular basis before permanent ejection from the network results. This is considered a realistic model as some nodes may be considered malicious through accidents such as software problems, moving temporarily out of range or suchlike. It is desirable to punish genuine maliciousness only and so some minor apparent misbehaviour is tolerated.

6.4.12 Communication distance

The radio range of a node is fixed at three hundred metres. This figure corresponds with the quoted range for outdoor reception of IEEE 802.11 b and g signals. As it is more likely that a MANET would be used with this technology than any other, this figure is appropriate.

6.5 Simulations

Once the first nine stages of the methodology have been successfully completed, the tenth stage is to conduct the experiment. In this stage, simulations were performed using the fixed and variable parameters decided in earlier stages. As the focus is to observe trends as variables are adjusted, considerable numbers of simulation runs were performed. Initial trial runs showed that in most cases, the server rule performed best with servers located initially with the most number of neighbours. Nodes that were designated as servers remained so until they either left the network or were excess to the required percentage and surrendered server status. Therefore, for most experimental runs the ‘Most Neighbours’ server rule is used unless otherwise stated. The combination of input variables used can be entered into a calculation to show the number of simulations performed. As the number of simulations required exceeded five hundred thousand, parameters were often incremented in larger steps. For example, server percentage was changed from 10, 20, 50 and 100 rather than in 10% increments and not all input parameters were adjusted against all other input parameters. Additionally, simulations were run to compare inputs where conclusions could be drawn quickly resulting in a single variable used for many of the experiments. An example of this is the server location rule ‘Most Neighbours’ which performed best in most circumstances and so was used for most experiments and similarly with the blind and informed request

rule where informed requests were generally better. This method was sufficient to identify trends and considerably reduced the number of simulations that had to be performed. Table 6.4 shows the simulation runs with the range of input variables which gives the number of variations for each input change.

Table 6.4 Simulation runs.

Input Variable	Range	Variations
Servers Required	1-5	5
Percentage of Servers	10-100	10
Trust threshold	0.1-0.9	9
Mobility - Speed	0-100kmh	11
Mobility - Percentage	10-100	10
Runs per Simulation	10	10

6.5.1 Simulation Metrics

A key management protocol requires two attributes to be truly worthwhile. Firstly, it must be efficient and secondly it must be robust. Efficiency can be measured in several ways. In a MANET using a contention-based communication protocol, it is advantageous to perform any tasks requiring communication between nodes as quickly and with the least number of messages possible. Any nodes within range of the messages but not involved in the message-passing process must wait until the channel is clear before they can send or receive messages. With the certificate request process, a measure of efficiency is the number of nodes involved in the message chain from requester to the first server and then to any other servers required. Therefore, a good measure of efficiency for the certificate request is the number of hops that a certificate request will involve. Both the maximum number of hops during the simulation and the average number of hops per certificate request were collected.

Increasing security with the certificate issuance process involves increasing the number of servers required and increasing the trust threshold for the certificate chain. Any increase in servers required will inevitably involve increasing the number of hops

required for a certificate. A longer certificate chain will inevitably result in a greater likelihood of a malicious node in the chain. The result may be either a decrease in the trust calculation or the request blocked by the malicious node requiring another certificate request by the requesting node. Therefore, the percentage of requests that result in a successful certificate issuance is a good measure of the effectiveness of the protocol. Results were collected for the successful percentage of certificate requests and those certificates that failed to be issued due to the certificate chain threshold dropping below the network threshold resulting in a refusal by the server. This gives an indication of the damage to the process that malicious nodes can cause simply by reduced trust in them by their neighbours or through malicious behaviour. The baseline run is set with the variable parameters in Table 6.5.

Table 6.5: Initial experiment variable parameters.

Servers Required	Servers %	Trust Threshold	Speed kmh	Mobility %
1	20	0.1	0-100	20

The results in Figure 6.6 show the result of the baseline experiment.

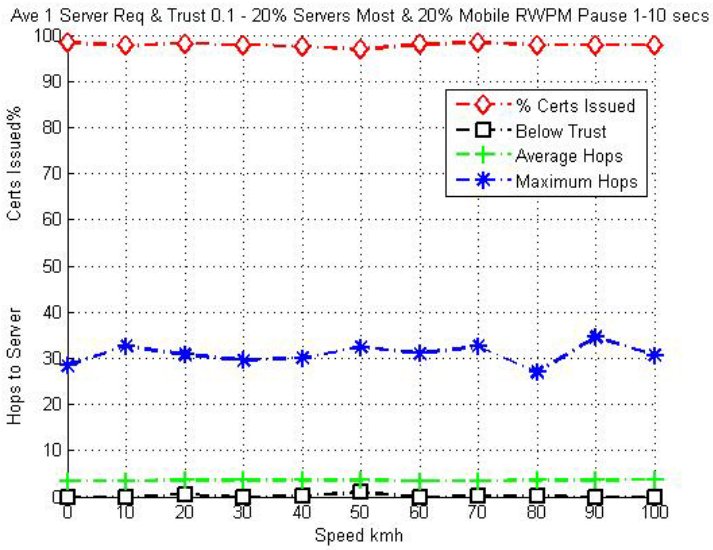


Figure 6.6: Baseline simulation results.

From this initial simulation run, further experiments are run by modifying the input parameters and observing the variance to the output parameters that result. The purpose is to become familiar with the effects that changes in the variables make. This leads to identifying trends in the results that occur due to the changes in the inputs. This will give a guide as to what parameters may be suitable in particular applications or particular network types.

6.5.1.1 Measures

One significant measure of effectiveness for the protocol is how likely a node is to receive a certificate at each request. Ideally, 100% of requests should result in a successful certificate issuance but this is not realistic due to the dynamic nature of ad hoc networks and the volatility of wireless communications. Therefore the desired goal is to achieve as close to a 100% success rate as possible. A measure of efficiency for a distributed CA is the number of hops that a message must make before a certificate is returned to a requesting node. The fewer the number of hops required, the more likely a certificate will be issued and the fewer the number of nodes will drain resources providing hopping points for other nodes' requests. By modifying the input parameters, the effectiveness and efficiency of the protocol can be tuned to provide the best performance possible for the required application.

6.5.1.2 Input Parameters

Following is a list of the input parameters:

1. Servers Required: A minimum of 1 server and a maximum of 5 servers required to obtain a certificate is used.

2. Trust Threshold: The trust threshold for the network is the trust calculated along the certificate chain to the servers. The first calculation is from the requester to the first server. Next, each chain from the first server to any other servers required by the server rule is calculated. All certificate chain calculations must arrive at or below the network threshold for a certificate to be issued. The minimum value of 0.1 is incremented in steps of 0.1 up to 0.9.

3. Mobility: Two input parameters make up the mobility model. Firstly, the speed that the mobile nodes move at. Speed is varied from stationary up to 100 kilometers per hour in 10 kilometer per hour increments. Once assigned a speed, the node will remain mobile at that speed unless pausing at a waypoint or maintaining a correct percentage of mobile nodes necessitates changing a mobile node to a stationary node. Secondly, the percentage of nodes that is mobile within the network. The baseline of 20% was altered through stationary, 50% and 100% mobile.

4. Percentage of Servers: The baseline of 20% of nodes designated as servers is altered to a minimum of 10% up to a maximum of 100%. There are drawbacks to being designated a server. Firstly, there is an increased likelihood of attack as attacks on servers will glean a lot more useful information for an attacker than attacking a non-server. Secondly, drain on battery power of the server as it must perform many more CPU cycles to perform key management creation, distribution and revocation tasks. Finally, a server must perform many communications with other nodes to provide services that it would not have communicated with otherwise. This leaves less time for the server to perform its

own communications. Therefore, the networks should have the minimum percentage of servers possible but still remain effective as a certificate authority.

5. Server Rules: Initially, nodes are placed randomly on the simulation grid and nodes will be designated servers as required to overrule the server percentage rule. When enough nodes are present, the server rule will be enforced and the correct percentage of servers will be maintained. The placement of those servers will have a bearing on the efficiency of contacting a server and therefore the likelihood of successfully receiving a certificate. Shorter certificate chains are desirable and the correct placement of servers will assist in reducing the length of the chains. There are three basic rules for server placement and two further rules that are optional. The first rule is to place the nodes randomly. Secondly, non-server nodes with the most neighbours can be chosen or non-server nodes with the least number of neighbours. The two optional rules are similar to the 'Most' and 'Least' rule but checks are made regularly and if a server node no longer complies with the rule, the server surrenders its role to a non-server node that is in the correct location. These last two rules are optional as they are only suitable where significant trust exists between all nodes and therefore sensitive information can be exchanged frequently without fear of attack.

6.6 Conclusion

As there are a large number of possible combinations of input parameters for the simulations, preliminary runs were made to identify any likely parameters to form a baseline and suitable modifications to the parameters could then be identified. For example, it was found that a maximum of five servers is all that is required due to the

significant fall in performance of the protocol for that number of servers. Therefore, increasing the number above five is not warranted. Similar results were obtained for mobility where maximum speeds were set at 100kmh. It was also found from the preliminary runs that the percentage of servers required for a certificate would significantly increase the number of hops the certificate requests would make. A balance in effectiveness and efficiency was found to be 20% of nodes designated as servers and so this figure was used as a baseline. The server rule for most simulations was set as the non-server node with the most number of neighbours should become a server if another server node is required (‘Most” rule).

The following chapter discusses the results of the simulations and draws conclusions that lead to guidelines on what input parameters should be chosen for particular circumstances that may be found when deploying such a network.

Chapter VII

COMPARISONS AND DISCUSSION OF RESULTS

7.1 Introduction

This chapter discusses the results obtained from the simulations and looks at the trends that develop as the variable parameters are adjusted. By manipulating the variable parameters, the effects can be identified and the differences in effectiveness and efficiency of the protocol due to the changes can be seen. For all baseline simulations, the rule used for choosing new servers when required is to choose the non-server node with the most number of neighbours to become a server.

Periodically screen shots of simulations will be used for descriptive purposes. Table 7.1 shows the legends that are used for the simulation grids.

Table 7.1: Network graphics legend.

Server	*
Non-server	O
Malicious Node	□
Ejected Node	◇

Figure 7.1 is an example of a typical network with three servers required to obtain a certificate in a static network. The figure shows the progression of the network formations at 200 seconds, 400 seconds and the conclusion of the simulation at 600 seconds. The small circles in the third figure show malicious nodes that have been or are about to be ejected and the large circles in the first figure represent the three hundred metre radio range of nodes that are outside the radio range of other nodes.

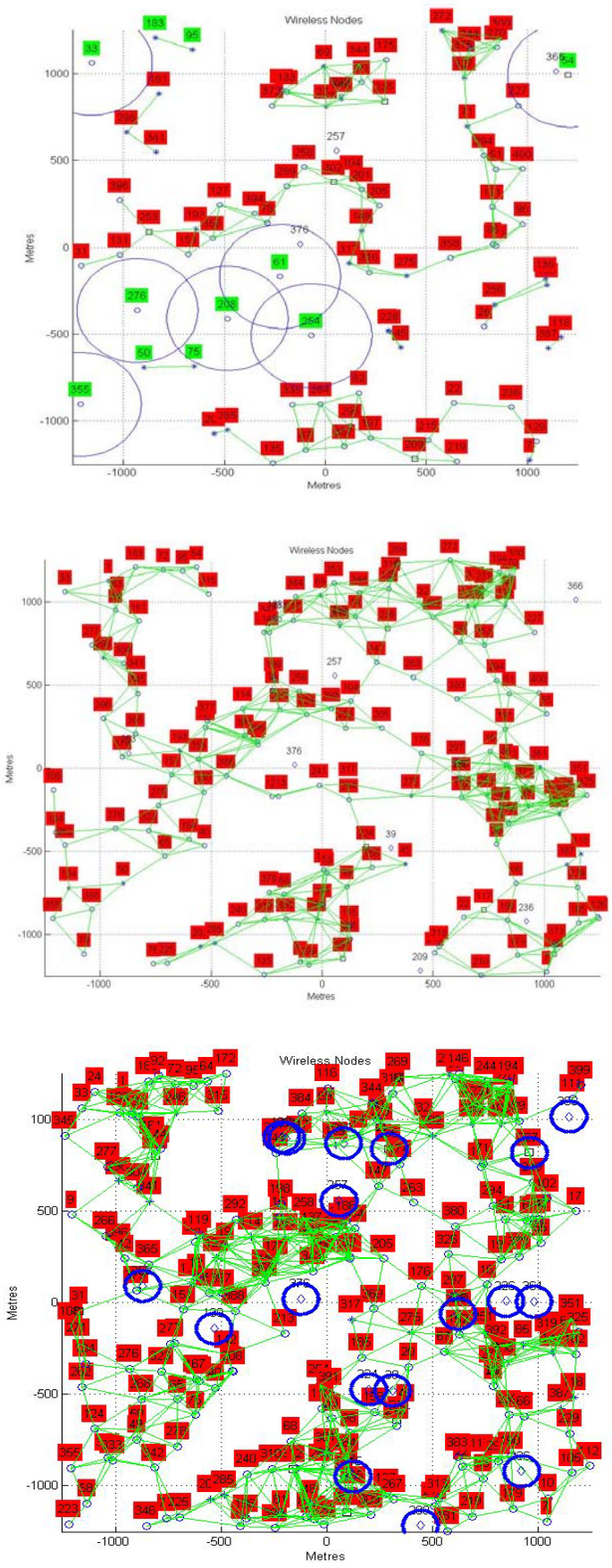


Figure 7.1: Simulation at 200, 400 and 600 seconds for 3 servers required.

These figures show how many smaller networks may come to exist and how the arrival and departure of nodes as well as the mobility of nodes causes networks to join and split apart into larger or smaller networks. Often, simulations will end with several networks or a single large network having been comprised of quite different looking networks whilst the simulation progressed. At 200 seconds there are 86 nodes of which 7 are alone and 79 are in 9 different networks. At 400 seconds this has grown to 158 nodes in a single large network. At the completion of the simulation there are 220 nodes in a single network with 44 nodes designated as servers and 16 nodes having been ejected due to multiple malicious behaviour.

The graph in Figure 7.2 shows the results of the simulation and shows the certificate request success rate, the maximum number of hops required to gain a certificate successfully, the average number of hops for a certificate issuance and the percentage of certificate request failures due to the certificate chain calculation falling below the network threshold. As can be seen, most certificate failures are due to the certificate chain trust value arriving at the server below the threshold required to obtain a certificate. Other failures are either due to malicious nodes failing to pass the request on or routes that have changed between route discovery and message request and therefore the route to the destination node has changed.

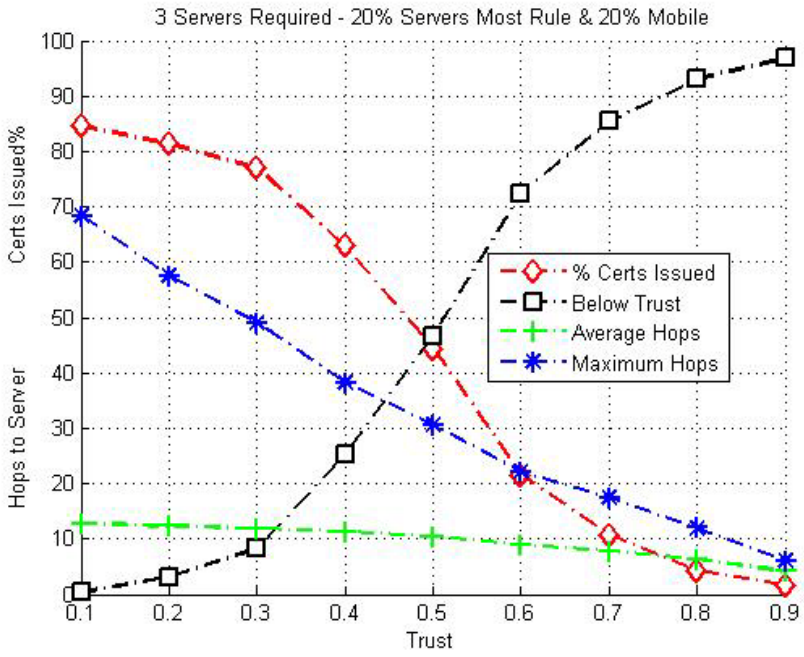


Figure 7.2: Results for 3 servers required with 20% mobility at 10-20kmh vs Trust.

The following sections compare the certificate success ratio to the certificate chain threshold. Certificate success is the number of requests for a certificate compared to the number of requests that result in a certificate issued. The results unless stated otherwise will all come from informed certificate requests. That is, nodes will only request a certificate if they have been informed that there are sufficient servers available to issue a certificate. The trust threshold is the measure of trust along the chain to the server that the certificate request must meet before it is accepted. The trust threshold is set from a very low 0.1 through to a very high 0.9.

The following discussion examines the results of the simulations and begins by comparing informed requests for certificates with blind requests. An informed request is one where a node will only request a certificate if there are sufficient servers in the network to obtain a certificate. A blind request is a request made without any knowledge of the number of servers present. Next are the results showing the effects of

changing the number of servers required for certificate services from one through to five. The results then progress through the changing of the various input variables to identify the effects of the changes. The resulting outputs are compared with the baseline inputs to identify what effect the changes have had. The variations begin with changing the percentage of servers in each network from 10% through to 100%. Following this, the effects of mobility are examined. Firstly, the difference that mobility has on the results measured with static networks through to nodes moving at up to 100 kilometers per hour. Mobility also involves the percentage of nodes that are mobile within the network and so mobility from zero to 100% mobile nodes is looked at. This is followed by comparisons of the rules used to choose new servers based on their location. Next, comparisons are made with other protocols where results obtained can be directly compared with these other protocols. Finally, conclusions are drawn from the results that changing the variables has had on the effectiveness and efficiency of the protocol which leads to guidelines that can be used for settings that can be adjusted for various topologies and applications that the protocol may be used for.

As nodes begin to populate the network grid, they will appear at random locations. As more nodes appear the chances increase that two nodes will be within communication distance. Figure 7.3 shows a network grid of 2500 metres square with 8 nodes of which 3 nodes are within communication range and have therefore formed a network. A single server is required to perform certificate services and so all 3 members now have certificates issued by the server. Those members outside communication range of another node have a large circle surrounding them representing the communication distance of 300 metres.

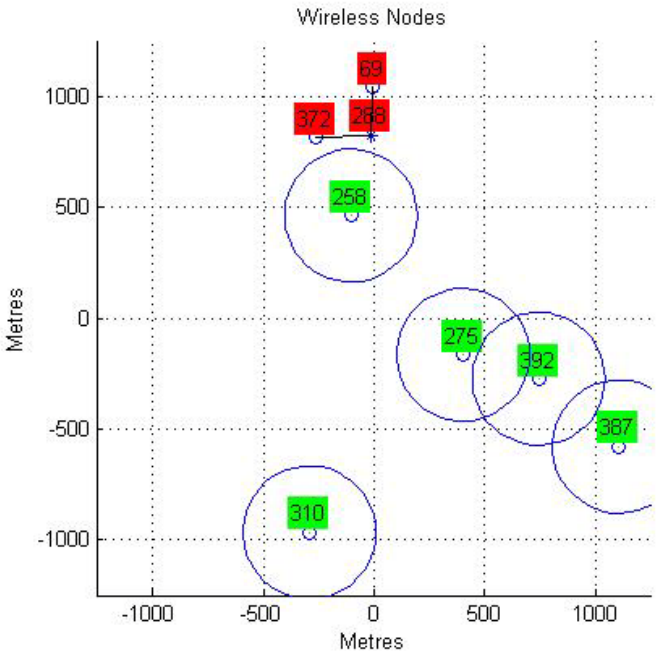


Figure 7.3: Network grid with 8 nodes.

If one or two servers are required to obtain a certificate, then the node is assured that there is almost certainly sufficient numbers of servers present. This is because the server percentage rule is overruled until there are sufficient nodes in the network to provide the minimum number of servers required as a percentage of the total number of nodes. If a node is within communication distance of another node, then a network has formed. If only two members exist and one or two servers are required, then one or both of those members will immediately become servers. Therefore, a node only needs to know whether it is required to assume server status or not. However, if more than two servers are required then a node will not immediately be informed of the number of members in the network or the number of servers. As requesting and receiving network information is message intensive and therefore disruptive for the network, it is desirable to exchange as little information as possible. Therefore, information about the structure and makeup of the network should only be sought if it outweighs any inefficiency caused to the key management processes.

7.2 Blind versus Informed Request

Experiments were run to see whether a new node in the network should request network makeup information to enable a node to make an informed request. If having this information before making a request makes little or no difference to the success rate, then a blind request is preferred. However, if a request for a certificate fails, the node will continue to request a certificate at the network update period of every one second. This process will continue until a certificate request results in a successful certificate issuance. As polling the network for information about the number of servers in the network utilises valuable network resources, this information should only be sought if it will save more information being exchanged for certificate requests than will be exchanged to discover the number of servers. Figure 7.4 shows the results for an informed request versus a blind request. The network is static with 20% of nodes designated as servers and one to five servers required.

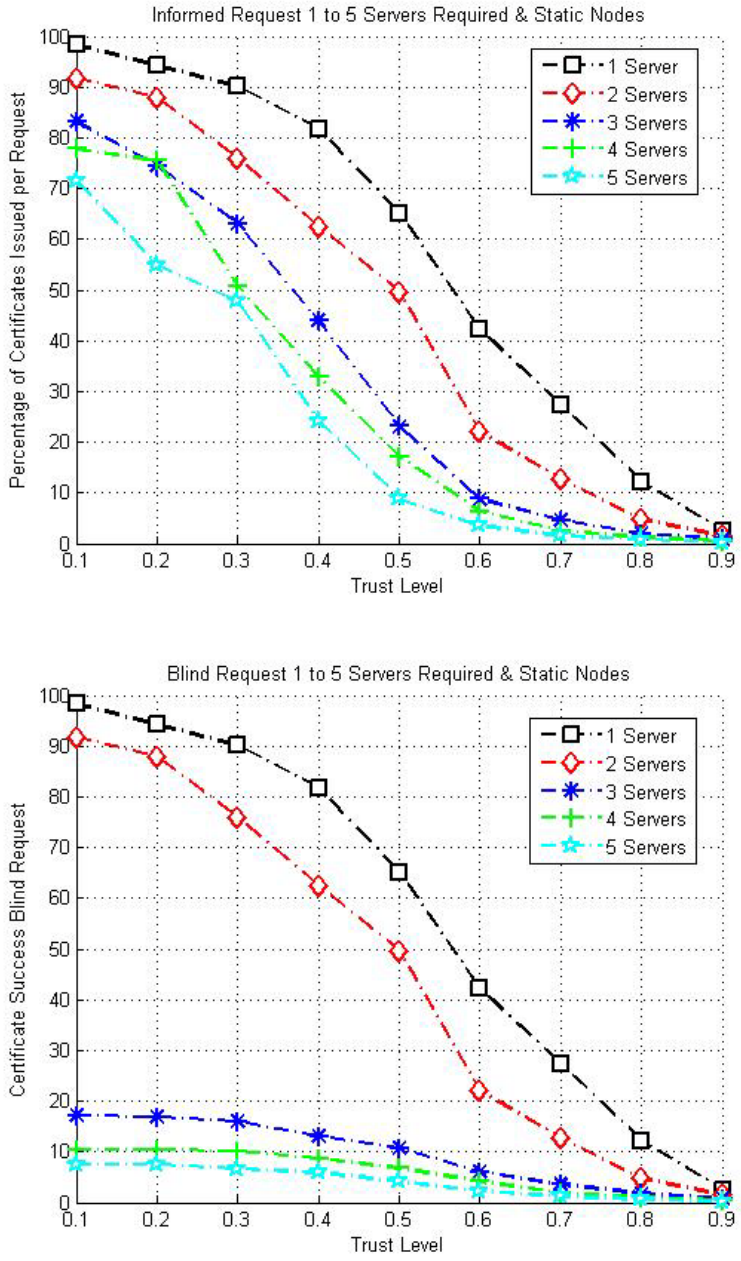


Figure 7.4: Certificate success – Informed Request v Blind Request.

Figure 7.4 shows that if one or two servers are required there is no difference to the success rate of the certificate request. However, if more than two servers are required a significant number of certificate requests will result in failures due to the number of servers present being too few to issue a certificate. These constant requests for certificates with no possibility of success are a waste of network resources. Therefore, a

node joining the network should signify its presence by broadcasting a message and waiting for a reply from any servers in the network. The new member can then maintain a record of the number of servers in the network and when enough servers are present can then request a certificate. These broadcast messages are useful for discovering neighbours, discovering the network makeup of servers and announcing a node's presence so that if necessary that node can become a server.

7.3 Servers Required

A node joining the network will require a certificate signed by the certificate authority before it is authorised to send and receive messages. The number of servers that are required to issue and sign a certificate is a choice the network has and will depend on the application that it is utilised for. This tunable parameter adds security by requiring more servers to collaborate to issue a certificate, but at the cost of requiring more communication and therefore more network resources. Figure 7.5 shows the results for one server required and up to the maximum of five servers required with all nodes stationary. As the trust threshold for the network is raised, the certificate success rate drops. Both requiring more servers to be contacted and requiring a higher trust threshold for the certificate request chain increases security but at the cost of reducing success rates for certificate requests. This figure is used for comparison with other results later in the chapter.

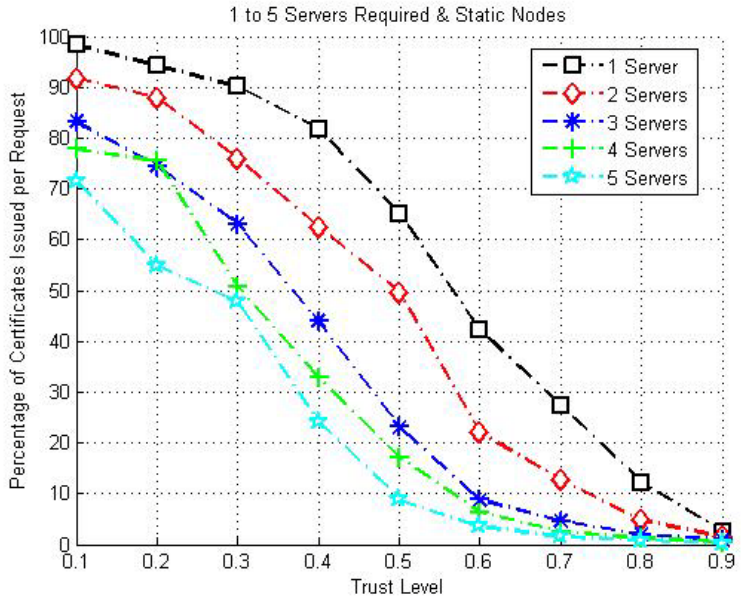


Figure 7.5: Certificate success for 1 – 5 servers required vs Trust and static.

One measure of efficiency for the certificate request process is the number of hops required from requester to servers and back to the requester with the certificate. The more hops involved means the more nodes receiving and passing on certificate requests. This also means that those nodes cannot conduct their own communications, both while passing on a message and whilst other nodes are passing messages within radio range. Therefore, the fewest number of hops required during the certificate issuance process is the desired goal. Figure 7.6 shows the average number of hops required to obtain a certificate in a stationary network with 20% of nodes designated as servers.

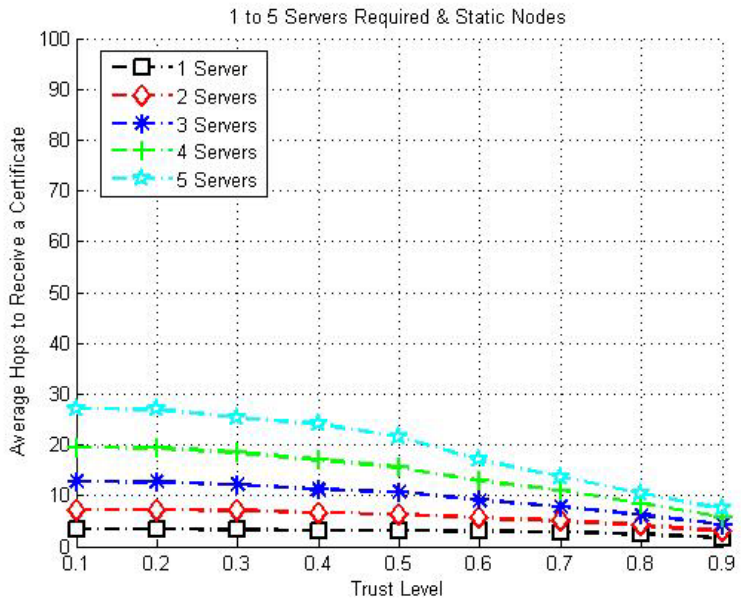


Figure 7.6: Average hops for 1 – 5 servers required vs Trust.

With 20% of nodes designated as servers, requiring a single server to be contacted for a certificate will mean that 20% of certificates will be self-issued. Additionally, a single node has total control of the certificates, both for issuance and for revocation. If accusations of misconduct are made against a node, the server receiving the accusation could choose to deal with it however they wish to and not necessarily by complying with the protocol rules. They could choose to immediately eject the node or ignore all accusations against particular nodes that they may be in collusion with. Additionally, single servers acting as a CA gives a potential attacker a single node to target the attack against. If successful, all knowledge required to attack nodes that have been issued certificates can be gleaned from the compromise of a single node. With servers sharing certificate information, an attacker has as many choices of servers to attack as there are servers in the network. Any server successfully compromised will lead to the attacker gleaning all information about certificates and keys. Therefore, requiring a single server for certificate services is extremely insecure.

7.3.1 Conclusion for Servers Required

As the number of servers required is increased, the robustness against attacks increases. However, increasing the number of servers required to get a certificate results in more failures in certificate issuance. The requesting node will then make another certificate request at the network update period which is set at one second. The major difficulty with a static network is that using a single routing protocol often results in a similar route to the server at each route discovery. If the route has failed previously due to trust issues or a malicious node failing to pass the message on, then it is likely that the same result will occur and the request will fail again. Only a change in the network structure such as a node arriving or leaving or mobility changing neighbours will result in a new route. Additionally, as more servers are required, the number of hops required will necessarily increase. This has the added penalty of making it more likely a malicious node or a node with reduced trust in its previous node in the chain will be encountered. This also makes it more likely that the trust value in the chain will fall below the network trust value and the certificate request will fail. Therefore, increasing the number of servers required for certificate services increases robustness against attack but at the cost of efficiency and reduced certificate success rates.

7.4 Mobility

If the network contains nodes that are mobile, the mobility assists in certificate issuance by quickly changing routes to servers and so leading to a higher chance of success for certificate re-requests. Experiments were run to observe the effect mobility of nodes within the network has on certificate issuance, both in the speed that the mobile nodes travel at and the percentage of nodes within the networks that are mobile.

7.4.1 Speed

Initial runs remained with the baseline of 20% of nodes mobile and then the speed of the mobile nodes was adjusted for the experimental runs. Every node has an equal chance of being selected as mobile as the selection was made randomly. Initially, approximately 20% of servers are mobile as well as approximately 20% of non-servers. Simulations were run in incremental steps of 10 kilometers per hour to a maximum of 100 kilometers per hour. The examples in this section are a sample comparing the static network with 50% mobile and 100% mobile nodes. Further results showing speeds in between these with 20% mobile nodes are contained in appendix 1. In Figure 7.7, 20% of nodes are mobile at speeds between 40 and 50 kilometers per hour.

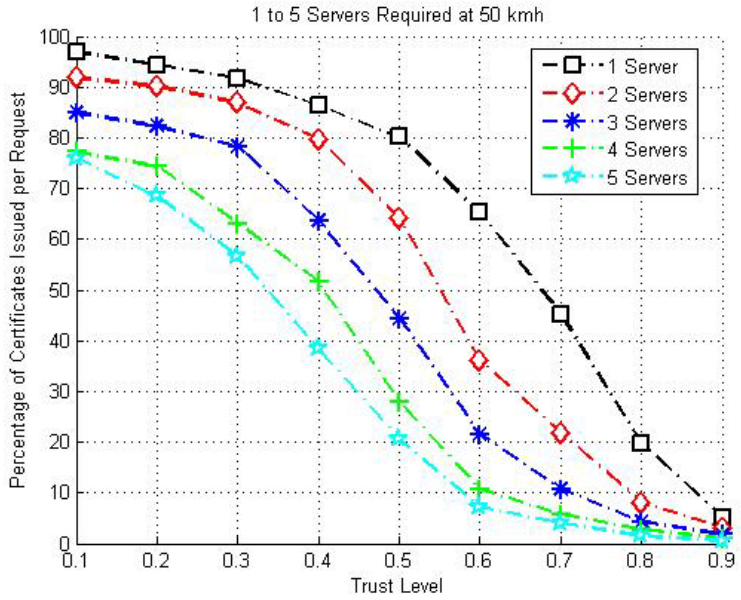


Figure 7.7: Certificate success: 1–5 servers vs Trust and 20% mobile at 40-50 kmh.

By comparing this with the baseline results in Figure 7.5, mobility has increased the certificate issuance success rates across all results. Next, simulations were run to see if faster speeds would increase certificate issuance efficiency further. In Figure 7.8 the results of having 20% of nodes mobile and those mobile nodes moving at between 90

and 100 kilometers per hour is shown. A baseline of 20% of nodes designated as servers is used.

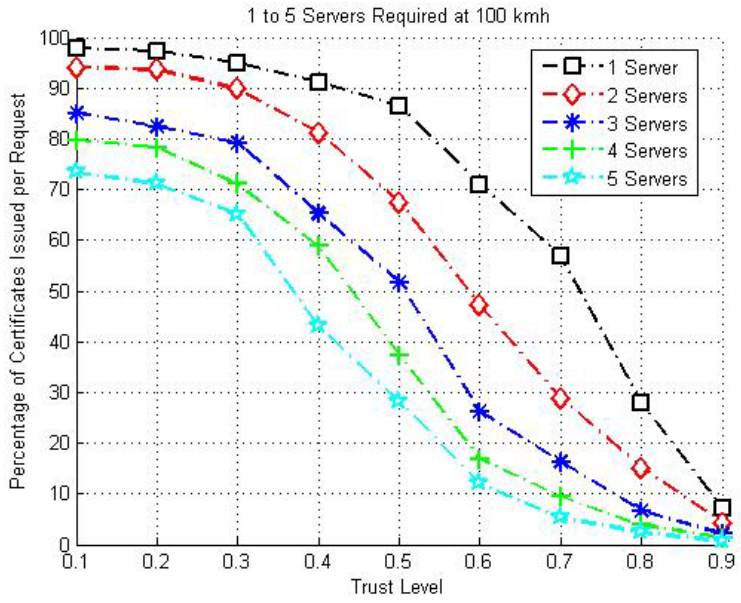


Figure 7.8: Certificate success: 1–5 servers vs Trust and mobile up to 100 kmh.

Comparing the results for the static network and the networks with nodes mobile at 40 to 50 kilometres per hour and 90 to 100 kilometers per hour shows that mobility increases the likelihood of a successful certificate issuance. This is especially true with trust thresholds between 0.2 and 0.8. The faster the mobile nodes are moving, the higher the chance that certificate re-requests will succeed. Table 7.2 compares the three examples and shows the percentage of increase for mobility. The increase is the increase in percentage from static network success rate to the networks with nodes moving at up to 100 kilometers per hour. In Section 7.5 later in this chapter, the various server location rules are discussed. In each of the cases for these simulations, the ‘Most’ rule is used for server locations.

Table 7.2: Static versus Mobile Network Certificate Issuance Efficiency.

Trust Threshold	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Server Static	98	94	90	82	65	42	27	12	3
1 Server 50kmh	97	94	92	86	80	65	45	20	5
1 Server 100kmh	98	97	95	91	86	71	57	28	7
Increase %	**0**	**3**	**6**	**11**	**32**	**69**	**111**	**133**	**133**
2 Servers Static	92	88	76	63	50	22	13	5	2
2 Servers 50kmh	92	90	87	80	64	36	22	8	3
2 Servers 100kmh	94	94	90	81	68	47	29	15	4
Increase %	**2**	**7**	**18**	**29**	**36**	**114**	**123**	**200**	**100**
3 Servers Static	83	75	63	44	23	9	5	2	1
3 Servers 50kmh	85	82	78	64	44	22	11	4	2
3 Servers 100kmh	85	82	79	65	52	26	16	7	2
Increase %	**2**	**9**	**25**	**47**	**126**	**189**	**220**	**250**	**100**
4 Servers Static	78	76	51	33	17	6	3	1	1
4 Servers 50kmh	77	74	63	52	28	11	6	3	1
4 Servers 100kmh	80	78	71	59	37	17	10	4	1
Increase %	**3**	**3**	**39**	**79**	**118**	**183**	**233**	**300**	**0**
5 Servers Static	72	55	48	24	9	4	2	1	0
5 Servers 50kmh	76	69	57	38	21	7	4	2	1
5 Servers 100kmh	73	71	65	43	28	12	5	2	1
Increase %	**1**	**29**	**35**	**79**	**211**	**200**	**150**	**100**	**-**

7.4.2 Conclusion for Speed

The results show that mobility in the network, even when only 20% of nodes are mobile, increases the chances of a successful certificate request. As the trust threshold for the network is raised, mobility has a greater effect on the success rate. Therefore, if a network has high mobility the success rate for certificate requests will be higher than for a network with low or no mobility.

7.4.3 Percentage Mobile

As mobility within the network, both by servers and non-servers, increases the efficiency of the certificate request process by rapidly changing routes to and from servers, experiments were run to see what difference changing the percentage of mobile nodes makes. As the nodes move faster, the more rapidly the routes will change and the

higher the likelihood that a failed request will be followed by a request taking a different route and therefore being more likely to succeed than if it took the same route. It was expected that by having more mobile nodes in the network, a similar result to increasing the speed would occur. That is, higher mobility, whether by faster speed or a higher percentage mobile would lead to rapidly changing routes from node to node leading to higher success rates for certificate issuance. The baseline figure for mobility is 20%, but the application the protocol is used for will dictate the actual mobility. For example, if all nodes are on foot, then speeds above walking speed are unlikely, and speeds above running speed impossible. If the application involves mobility with vehicles, then higher speeds are expected. Additionally, the percentage of nodes mobile will vary with the application and the circumstances. With 20% mobility used as a baseline, experiments were run to see the effect mobility with 50% of nodes mobile would have. Figure 7.9 shows the effect of having 50% of nodes mobile at between 40 and 50 kilometers per hour.

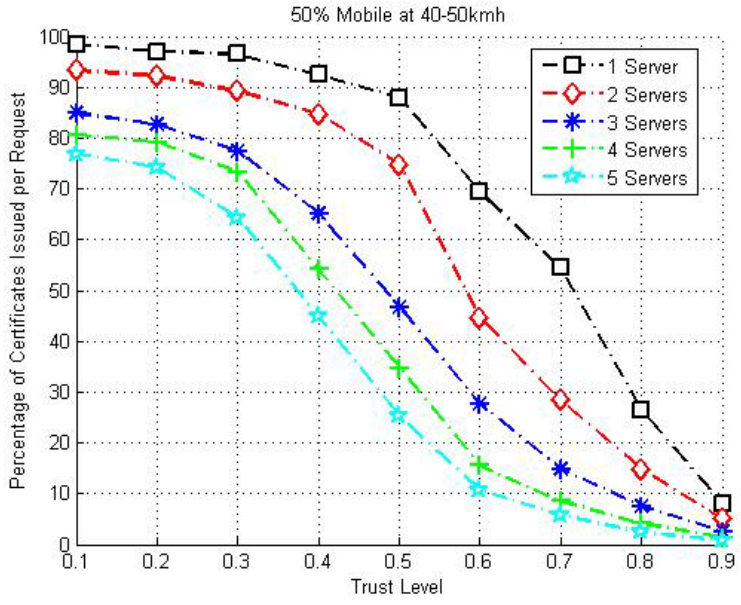


Figure 7.9: Certificate success: 1–5 servers vs Trust and 50% mobile at 40-50kmh.

Comparing this to Figure 7.5, the increased mobility has increased the certificate issuance success rate. Once again the main increase is for trust thresholds over 0.1 and less than 0.9.

7.4.4 Conclusion for Percentage Mobile

Increasing the percentage of nodes mobile within the network has a similar effect to increasing the speed of the nodes. If a network has high numbers of nodes mobile and those mobile nodes are traveling at high speeds, then the certificate issuance success will be considerably higher than for a static network. This increase in effectiveness could, if the users wish, be partially offset by increasing security, either by adding more servers required for KM services or by increasing the network trust threshold.

7.4.5 Percentage of Servers

Nodes are chosen to become servers based on their location relative to other nodes. The network will require a certain percentage of nodes to perform server roles which mainly relates to certificate issuance, authenticating certificates for message exchange and revoking certificates of malicious nodes that have been ejected. As a server's role requires more message exchange and CPU processing than a non-server node, it is not desirable to be designated a server but rather a necessary task that some nodes must perform. Additionally, servers will contain and exchange sensitive information with other servers that could be used by an attacker to infiltrate or disrupt the network. The number of servers required in the network should therefore be kept to the minimum necessary to perform the key management tasks efficiently and maintain robust security. The baseline for the percentage of servers is 20% of nodes. The effect of decreasing this percentage to 10% and then increasing it to a maximum of 100% is examined. Figure

7.5 gave a baseline for certificate success for 20% of nodes in the network performing a server role. Figure 7.10 shows the effect of decreasing this to 10%.

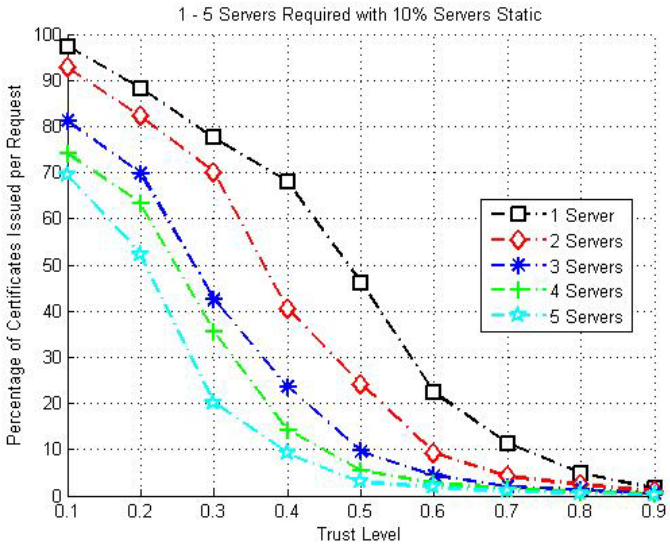


Figure 7.10: Certificate success: 1–5 servers vs Trust and static with 10% Servers.

Whilst only a 10% decrease, the effect is to halve the number of servers in the network from 20%, effectively making a 50% decrease from the baseline. The difference in the success ratio is fairly minor with very low and very high trust thresholds, but the middle trust thresholds of 0.3 to 0.7 have a significant decrease in success. This reduction is due to the failure of the certificate chain to maintain the trust above the threshold upon reaching the servers. Therefore, the increase in the hops required for the certificate request is increased by reducing the percentage of servers in the network. Figure 7.11 shows the effect of having 50% of nodes in the network designated as servers.

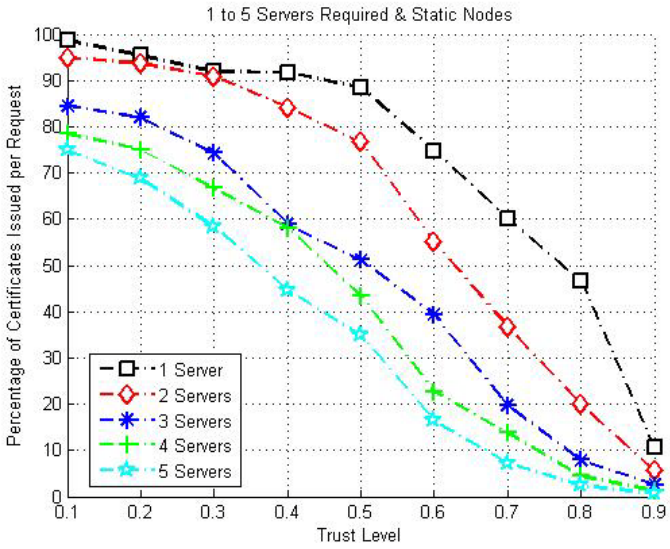


Figure 7.11: Certificate success: 1–5 servers vs Trust and static with 50% servers.

Comparing these results with Figure 7.5, it can be seen that whilst there is an increase in the certificate request success rate, it is not overly significant. At low trust threshold levels the rate is only marginally better, but at higher trust threshold levels above 0.5 the difference is more marked. At very high trust levels, the benefits are more significant, especially when more than two servers are required. The penalty of having so many servers is that Byzantine behaviour is much more likely to be successful from malicious nodes. Therefore, more servers increase the likelihood of mischievous collaboration. If the network application is such that the server nodes can be trusted more than non-server nodes, then a higher percentage of servers has benefits in higher certificate issuance rates. This can then be balanced if desired by increasing the trust threshold for the network. Figure 7.12 shows a network with all nodes designated as servers.

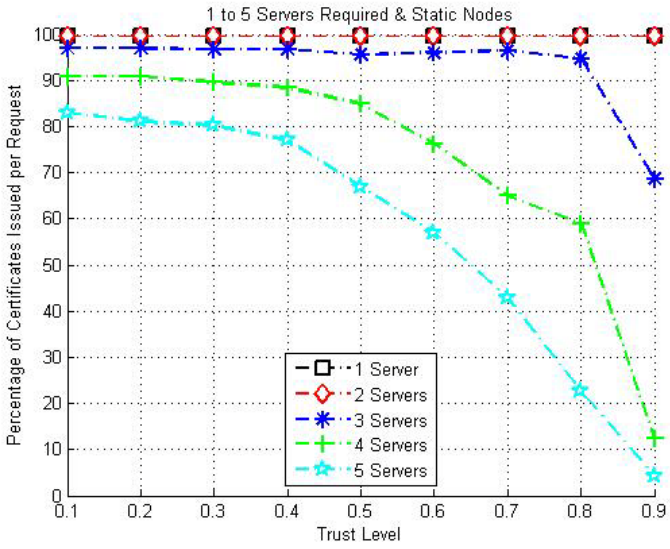


Figure 7.12: Certificate success: 1–5 servers vs Trust and static with 100% servers.

From this figure it can be seen that if one server is required then a certificate request will always succeed. This is because the node is requesting a certificate from itself and therefore a self issued certificate will always be issued. If two servers are required, then a certificate will almost always be issued. A node will only request a certificate if it can communicate with at least one other node and that neighbour will always be a server. Therefore, the neighbour, unless maliciously refusing to cooperate, will issue half the signature to pair with the node's own certificate and signature. If three servers are required, then the request is successful unless a node fails to pass on the message. As the certificate chain will be very short at only four hops required, the trust threshold will almost always be met until the threshold required reaches 0.9. When four or five servers are required, the certificate chains remain short enough that trust thresholds will most likely be met up to a reasonable level of approximately 0.8 in the case of four servers and 0.6 in the case of five servers.

With all nodes designated as servers, the hops required to obtain a certificate will be the minimum possible. Additionally, whichever server rule is in place will make no difference as all nodes are selected to be servers. The average hops required for a successful certificate issuance for 100% of servers is shown in Figure 7.13.

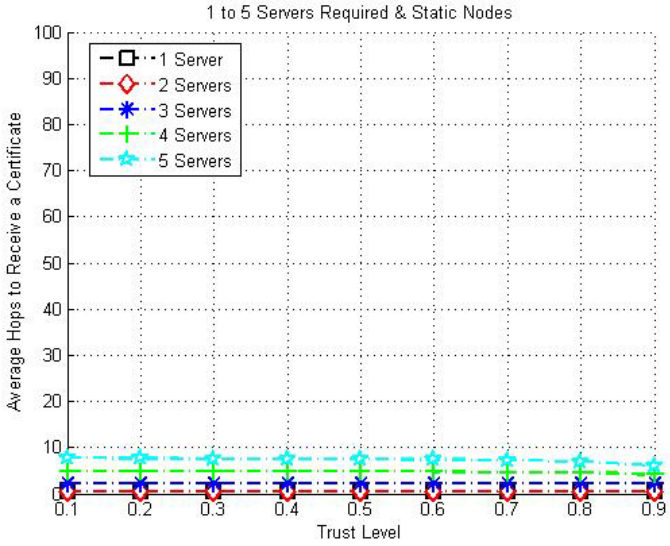


Figure 7.13: Average hops 1–5 servers vs Trust and static with 100% servers.

The reduction in the number of hops required has a significant increase in the success ratio for certificate requests. Table 7.3 shows the hops required with 10, 20, 50 and 100% of nodes designated as servers with 3 servers required as an example.

Table 7.3: Average hops & success percentage for a certificate
10% to 100% servers – 3 servers required.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
10% Servers	17.7	17.1	16.1	14.0	11.8	9.2	7.5	5.7	4.0
Success %	81	70	43	24	10	4	2	1	1
20% Servers	12.8	12.7	12.2	11.2	10.7	9.1	7.8	6.2	4.3
Success %	83	75	63	44	23	9	5	2	1
50% Servers	8.3	8.2	8.0	7.7	7.5	7.3	6.8	6.0	4.5
Success %	84	82	74	59	51	39	20	8	3
100% Servers	2.1	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3
Success %	97	97	97	97	96	96	96	95	69

It can be seen that once a trust threshold of 0.6 is required for a certificate, there is very little difference between 10% and 20% servers for the number of hops required for a successful certificate issuance. Even with 50% of nodes designated as servers, a trust level of 0.7 and above shows almost no difference. However, what is significant is the percentage of failures for fewer servers. This is because as the average number of hops increases, so does the failure rate due to the threshold not being met. Almost every certificate failure will result in a renewed request by the node for a certificate. With 10% servers and a low trust threshold of 0.3, on average every node will request a certificate more than twice (43% success).

7.4.6 Conclusion for Percentage of Servers

As the percentage of servers within the networks is increased, the certificate success rate also increases. With one or two servers required, the certificate success rate is at or close to 100%. This is because nodes will only request a certificate when they are within range of a neighbour and the requester and neighbour will both be servers. Therefore, the hops required will necessarily be only two hops and a certificate will only fail if the neighbour is malicious and does not reply. When three or more servers are required, then the hops will remain few but misbehaving nodes have more chances to fail to pass on requests as the number of hops required is increased. Increasing the number of servers required increases the number of hops required and therefore more certificate request failures occur as more servers are needed. Increasing the servers required and the trust threshold for certificate issuance within the network will increase security. However if only one server is required to perform key management services then security is extremely low. For example, a malicious node can join the network, issue itself a certificate and then eject other nodes maliciously. Therefore, one server

required permits such low security that it is not feasible unless all nodes can be fully trusted. Security is increased primarily by requiring at least four or five servers before reasonably robust security can be assured. Therefore, a balance must be found between system resources available and disclosing sensitive information to more nodes than is desirable. More extensive results for server percentage simulations are available in appendix 2.

7.5 Server Rules

Five different rules were experimented with to identify the best location selection for a new server. The five choices of server rule are:

1. Most Neighbours. This rule states that when a new server is required it is chosen by selecting the non-server node with the most number of one-hop neighbours. If there is more than one server with the same number of neighbours, then the first one identified that complies with the rule is chosen. This tends to select nodes near the centre of networks, often with several servers near each other forming server groups. The nodes on the outside of the network therefore may have several hops to the first server, but if more than one server is required then the hops to the other servers from the first one is generally few. If the network is highly dynamic with high mobility, then after time, long-serving servers will find themselves more randomly placed as the network members move around the network grid. If fewer servers are required to maintain the correct server percentage, then the server with the least number of neighbours surrenders server status.

2. Least Neighbours. This rule states that new servers are chosen by finding non-server nodes with the least number of neighbours. This is the opposite of rule one and

generally places servers on the outer edges of the networks. If the network is highly dynamic then long-serving servers will tend towards looking more randomly placed after time as the network members shift. If fewer servers are required then the node with the most number of neighbours surrenders server status.

3. Random. Servers are chosen entirely randomly from the present non-server nodes. If fewer servers are required than there is present, a randomly chosen server will surrender server status.

4. Most Neighbours Updated. This rule is similar to the 'Most Neighbours' rule but instead of semi-permanently remaining as a server once chosen, the network examines the server's neighbour count at the network update period which is set to one second. If a non-server node has more neighbours than a server node, then the server surrenders server status and the non-server becomes a server. This ensures that at all times servers are those nodes with the most number of neighbours. This forces the rule to be strictly enforced and therefore servers are generally in the centre of networks and grouped together and will remain so for the lifetime of the network. However, this requires a high number of servers to surrender their information to non-server nodes and this constant swapping of roles discloses a considerable amount of secret key management information to network members. Server status is surrendered when necessary by selecting the server with the least number of neighbours.

5. Least Neighbours Updated. This is the opposite of the 'Most Neighbours Updated' rule and generally selects nodes to be servers that are placed on the edges of the

network, often with a single neighbour. This tends to spread the servers apart from each other requiring longer hops between servers.

7.5.1 Server Rule Results: Not Updated

Initially, simulations were run to compare the first three server rules only. As there are significant security drawbacks resulting from constantly enforcing the rules with one second updates, the ‘Most Updated’ and ‘Least Updated’ are considered special option cases that would be used rarely. Therefore, these two rules were left for final simulation runs. The following figures compare the ‘Most’, ‘Least’ and ‘Random’ rules for certificate issuance success and average number of hops required for a certificate to be successfully issued. In all cases 20% of nodes are designated as servers. Figure 7.14 shows that in the case of one and two servers required, randomly choosing the server location has a slight performance gain in certificate issuance over the ‘Most’ and ‘Least’ rule.

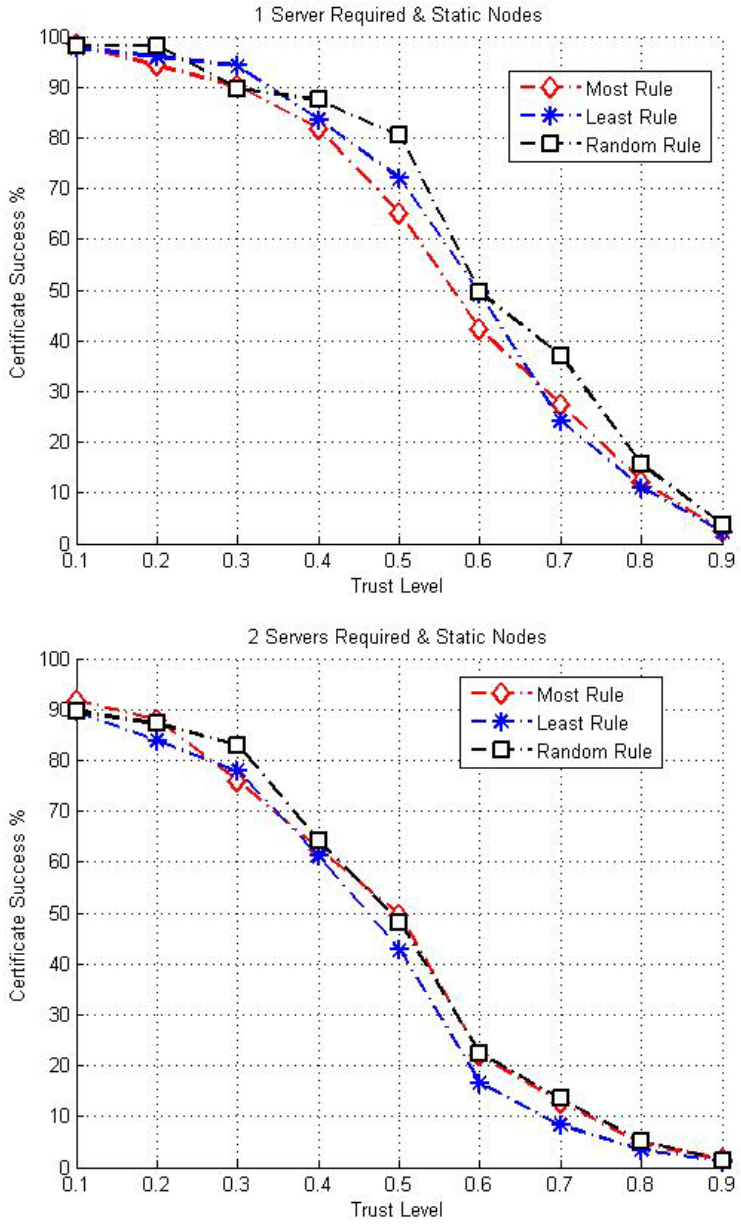


Figure 7.14: Static network with 1 and 2 servers required.

The difference is fairly small as certificate request hops will be few in number. The 'Most' rule performs the worst for a single server required and the 'Least' rule performs the worst when two servers are required. This may be because of the increased hops necessary between servers as more inter-server communication is required. At a trust level of 0.1 and 0.9 all three rules perform at about the same effectiveness. Figure 7.15 compares the cases of three, four and five servers required.

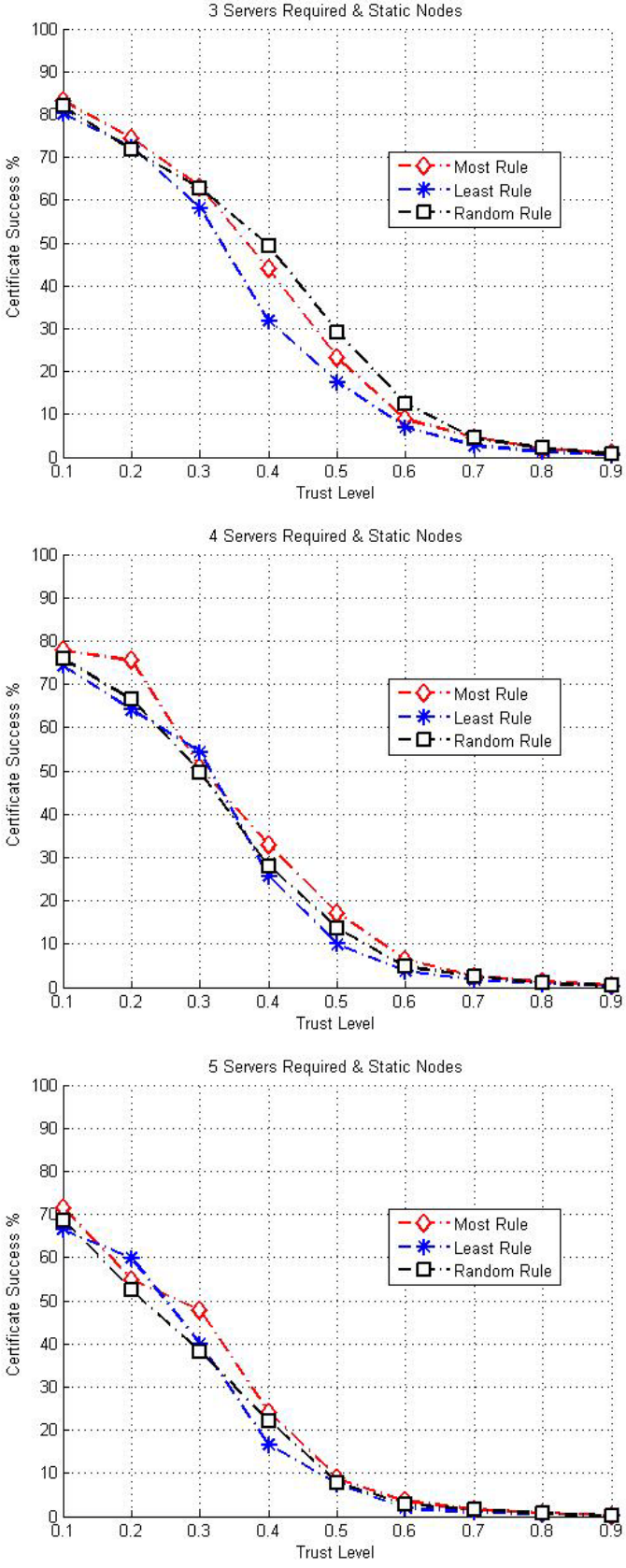


Figure 7.15: Static network with 3, 4 and 5 servers required.

This figure shows that as the number of servers required is increased above two, the random and most rule are slightly superior over the least rule. Once four servers are required, the ‘Most’ rule emerges as the best choice. Whilst the difference is small, even a slight increase in efficiency is of benefit to a busy network. The probable reason for the better performance of the ‘Most’ rule above 1 server, is that servers are assigned where the node has the most number of neighbours. This tends to group servers together near the centre of the network where inter-server communication is usually very close to each other, often as neighbours. This reduction in hops between servers more than compensates for the greater hops to and from the first server from nodes on the outer fringes of the network. Overall, this tends to reduce the average number of hops for a certificate and the greater efficiency created by this rule shows increasing benefits in efficiency as more servers are required. Figure 7.16 shows similar results but with 20% of nodes in the network moving at between 40 and 50 kilometers per hour.

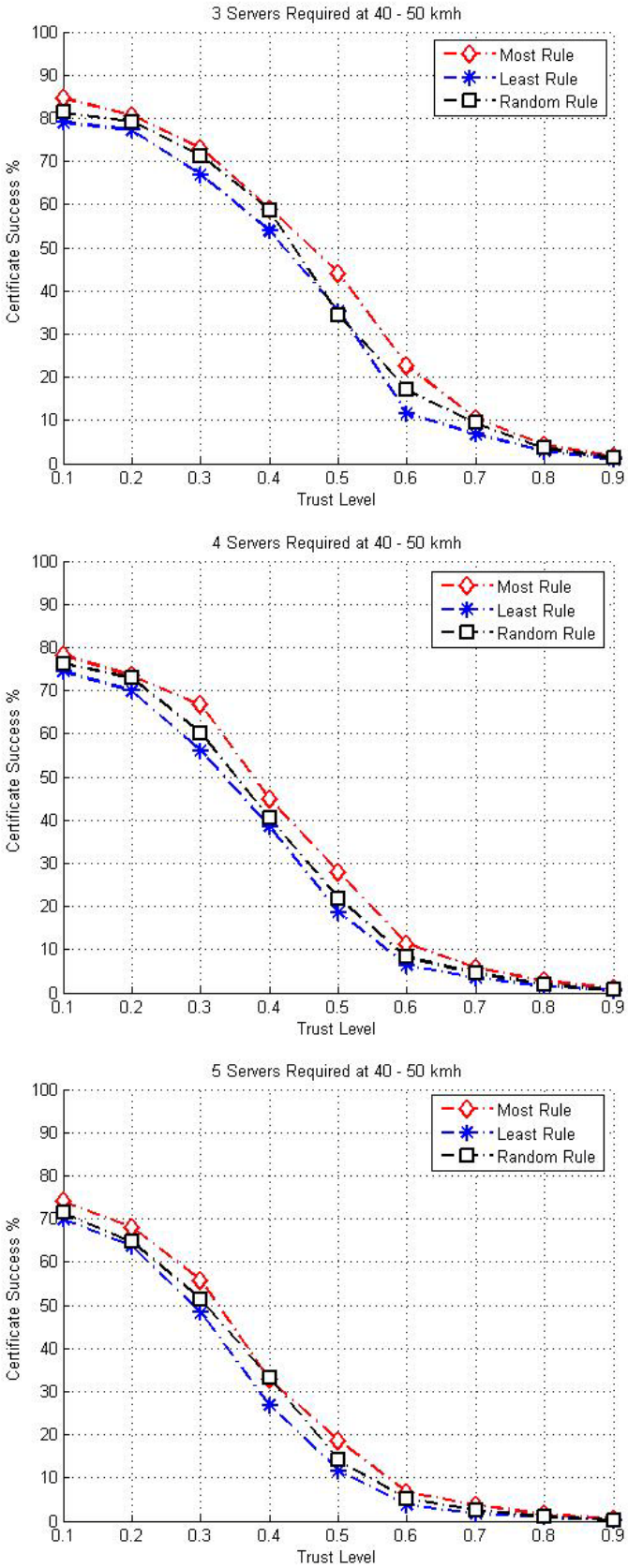


Figure 7.16: Certificate success: 3, 4 and 5 servers with 20% mobile at 40-50kmh.

The results show that for a network with mobile nodes, ‘Most’ is the best rule to use for effectiveness of certificate issuance with the ‘Least’ rule performing worst. Mobility has shown to increase the certificate issuance success rate for all server locations due to rapidly changing routes from source node to destination node. Whilst certificate success ratio measures effectiveness, efficiency is measured by the use of network resources and time taken to perform key management functions. The measure of time and efficiency is therefore the number of hops required to obtain a certificate with the goal being to require the fewest number of hops possible. Figure 7.17 shows the hop counts required for three, four and five servers required in a stationary network.

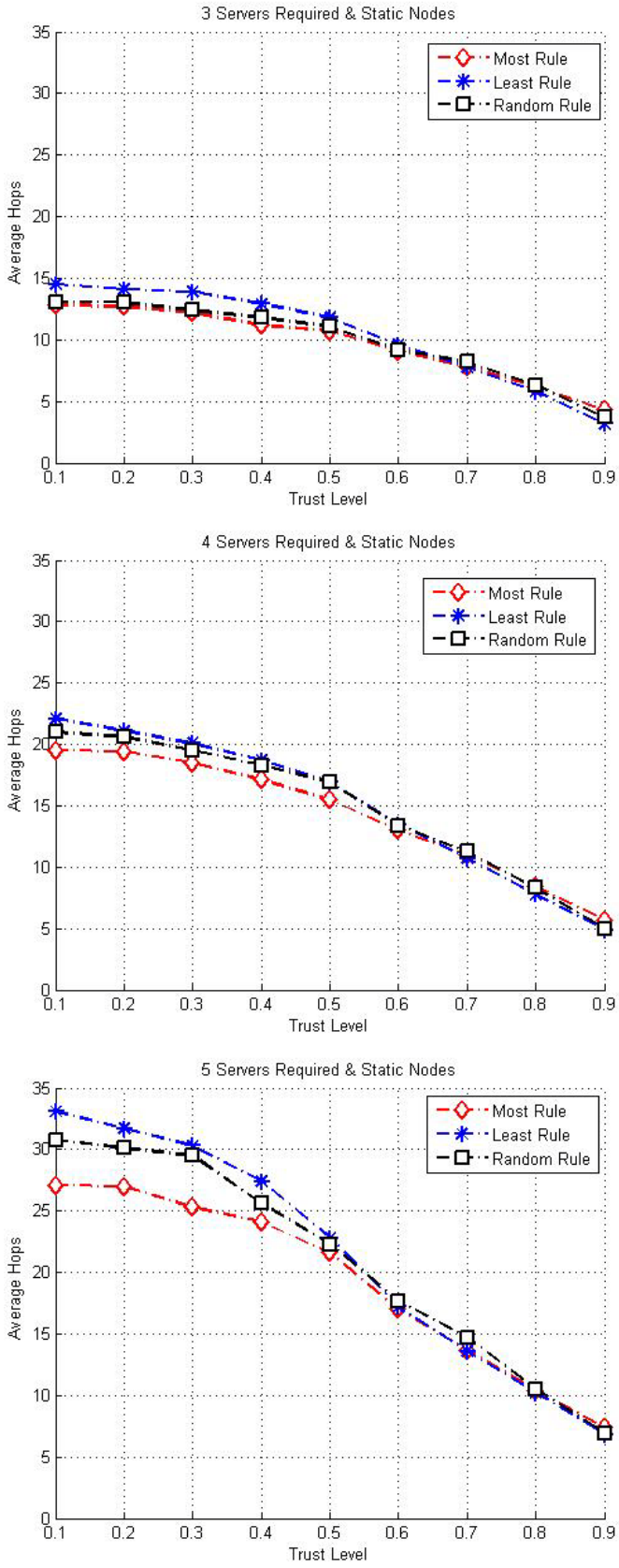


Figure 7.17: Hops required for 3, 4 and 5 servers – static network.

These results show that in a static network for more than two servers, the most efficient rule is the ‘Most’ rule. As more servers are required to obtain a certificate, the gap between the efficiency of the three server rules widens due to the increased inter-server communication. Figure 7.18 shows the average number of hops required to obtain a certificate with 3 and 4 servers required with 20% of nodes mobile at 40 to 50 kilometres per hour, with Figure 7.19 showing the same network settings but with 5 servers required.

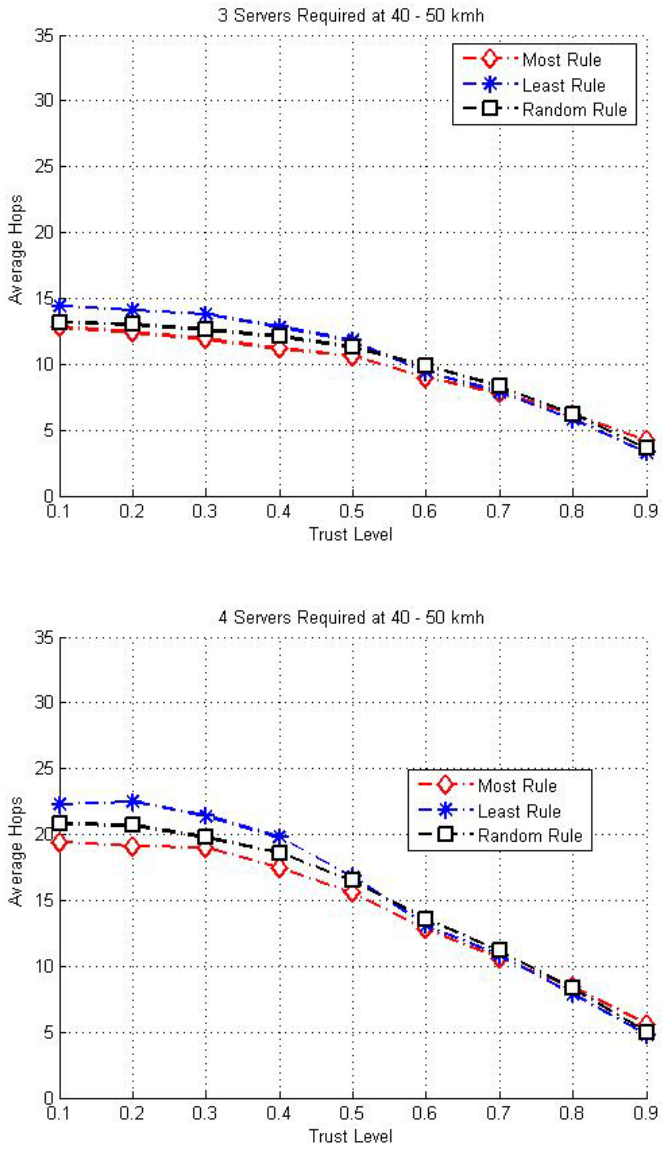


Figure 7.18: Hops required for 3 and 4 servers – 20% mobile at 40-50kmh.

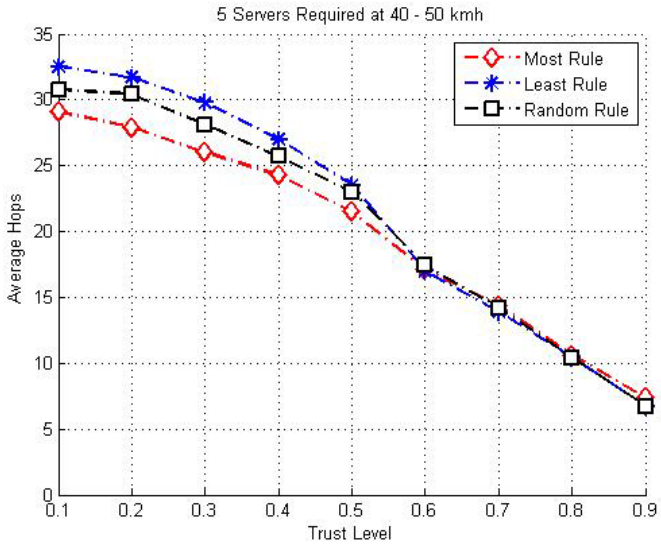


Figure 7.19: Hops required for 5 servers – 20% mobile at 40-50kmh.

Whilst added mobility within the network may increase the success rate for certificate issuance, when a request is successful approximately the same number of hops is required. However, these results show that as the number of servers required increases the gap between the average hop counts also increases. Therefore, as more servers are required for KM services, the location of the servers becomes more critical. In all cases the 'Most' server rule is more efficient requiring fewer hops to obtain a certificate than the 'Least' or 'Random' server placement. The conclusion that can be drawn from the results is that for one server required the 'Random' server location rule should be used. If two servers are required then the 'Most' rule will provide the fewest hops and even though the random rule will provide slightly higher certificate issuance success rates, the increased efficiency makes the 'Most' rule the best choice. If more than two servers are required, then 'Most' rule is clearly the best choice. More results for server location rules are available in appendix 3.

7.5.2 Server Rule Results: Updated

By updating the server selection frequently the server rules can be strictly enforced. The purpose of selecting servers based on their location is to reduce the number of hops required to obtain a certificate. The fewer hops required, the less communication is needed and the higher chance that a certificate will be successfully issued. As the refresh rate for the simulation was set to one second, the server placement update period was also set to one second. That is, every one second the server placements within the network or networks that had developed were checked to see if they complied with the rule in force. If not, a server that violated the rule surrendered its server status to a non-server node that complied with the rule. For example, if the rule in place was to select nodes to become servers that had the most neighbours, a server with fewer neighbours than a non-server would relinquish server status to the better placed node. Figure 7:20 shows the comparison of a typical small network contrasting the placement for the 'Most Updated' versus 'Least Updated' rule with one server required.

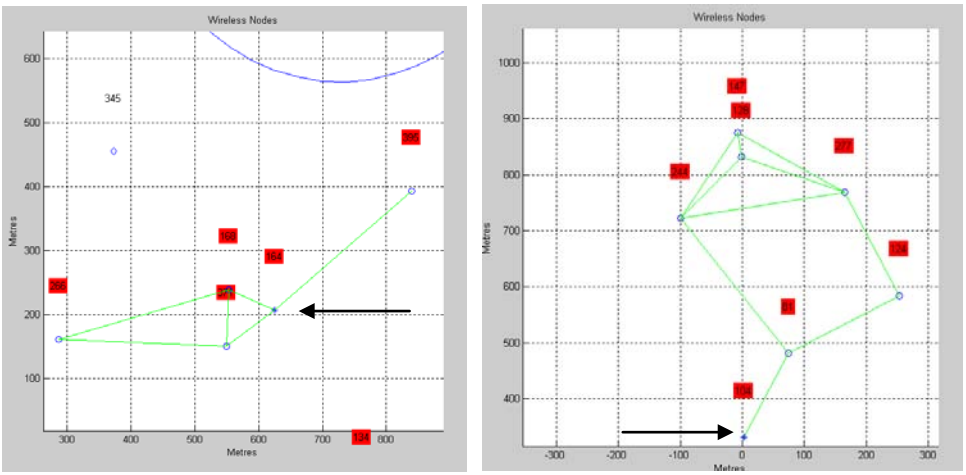


Figure 7.20: Typical network server placement Most Updated vs Least Updated rule.

This shows the placement of the server nodes (* with arrow) when the two rules are strictly enforced. With the 'Most Updated' rule, servers will tend to be placed near the centre of the network where they have the most number of neighbours. In contrast, the

'Least Updated' rule will place servers where they have the least neighbours which tend to be on the outer fringes of the network, often with a single neighbour. Following is a comparison of results for these two rules comparing the certificate success and certificate hop counts to the 'Most Neighbours' rule.

7.5.3 Most Updated Server Rule

This rule places servers where they have the most neighbours, whether those neighbours are servers or not. The observed effect is generally to collect servers together in groups making hopping from server to server require few hops over short distances, often to neighbours. Figure 7.21 shows a certificate being issued to a node where five servers are required to receive a certificate. There are 22 hops required to receive the certificate.

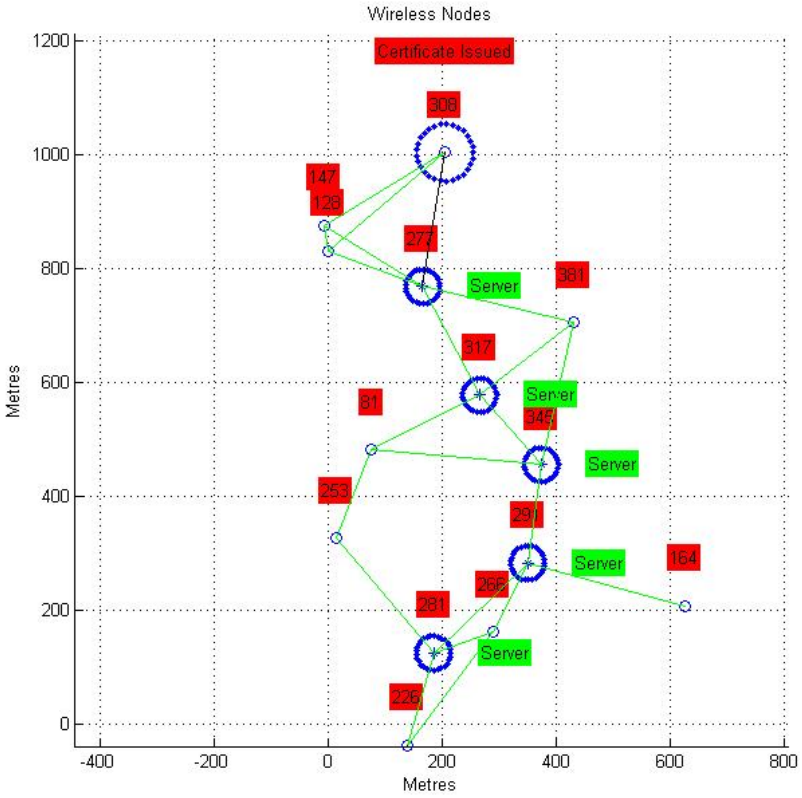


Figure 7.21: Example of Most Updated rule with 5 servers required – 22 hops.

The effect of implementing this rule is most noticeable when the servers required rule is high. If a single server is required, then it is best to have servers located close to the requesting node and so a random placement of servers does this most effectively on average. As the number of servers required increases, it is desirable to have short paths from the first server contacted to the other servers that are required. The longer hop to and from the requesting node to the first server is compensated for by having short hops between servers while the certificate is constructed. The following figure compares the 'Most' server rule with the 'Most Updated' server rule and shows one and two servers required for a static network.

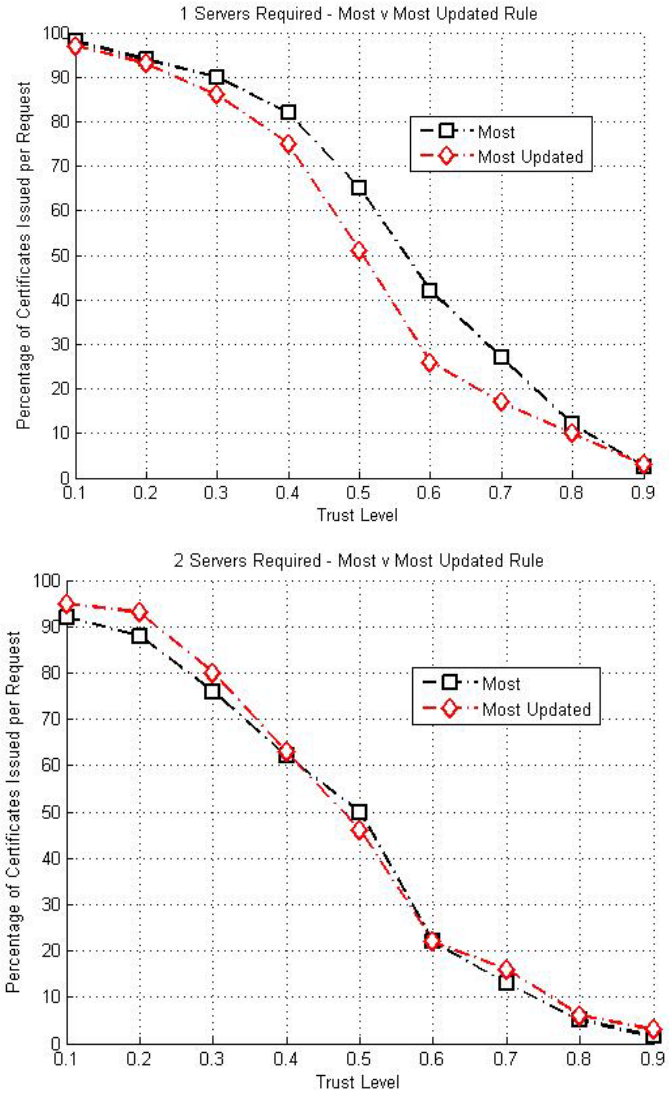
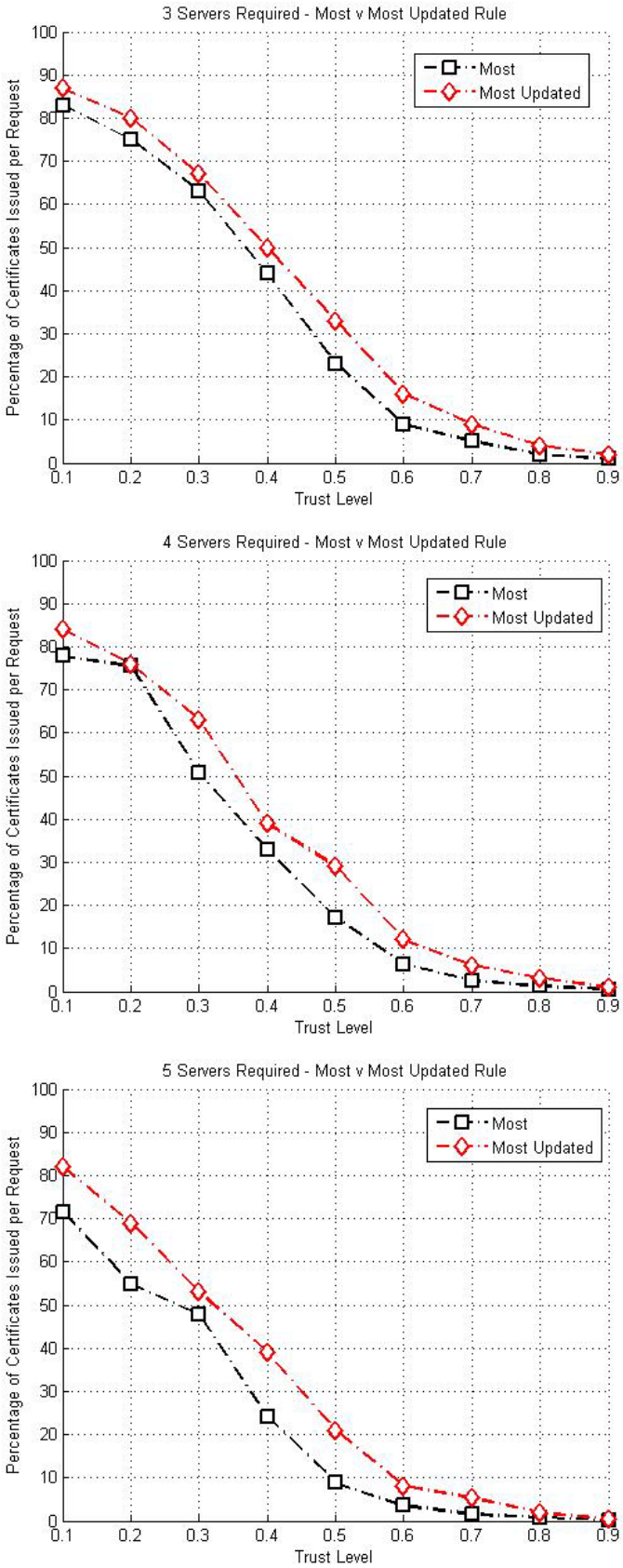


Figure 7.22: Success rate: Most vs Most Updated with 1 and 2 servers required.

As expected, strictly enforcing the server location rule does not improve the success rate for one server. In fact, it actually performs less effectively. This is because nodes are required to contact servers that have moved further away from nodes on the outer edges of the network. The hops required are high because once the first server is contacted it must send the complete certificate hopping back to the requester. There is no benefit to having servers grouped together as server groups. A server can self-issue certificates so a server having other servers as neighbours is of no benefit to them or any other node requesting a certificate. When two servers are required, a balance is struck between nodes having to hop to the centre of the network to contact the first server and the short distance for the first server to contact the second server. This configuration balances out to almost approximate the ‘Most’ rule without updating and so little benefit in success rate is seen. Once more than two servers are required, benefits from grouping the servers together begin to materialise. Figure 7.23 compares the ‘Most’ server rule with the ‘Most Updated’ server rule and shows three, four and five servers required with a static network.



7.23: Most rule vs Most Updated rule with 3, 4 and 5 servers required.

The increase in effectiveness becomes more marked as more servers are required. By strictly enforcing the 'Most' rule for three to five servers, benefit in greater success for certificate requests is seen but with considerably more management messages to maintain the correct server locations. The increase in efficiency and therefore success rate is due to the reduced number of hops to receive a certificate. Interestingly, for two servers required the random placement of servers has slight increase in successful requests overall, but an increased number of hops required for all trust levels. Therefore, even though random placement for two servers gives a slight gain in certificate requests, the efficiency gain in fewer hops for random placement outweighs this. However, due to the increased certificate issuance success rate for two servers the 'Most' rule is used for comparison purposes. Table 7.4 shows the average hops to successfully receive a certificate comparing the random server placement for one server, and the 'Most Neighbours' placement for two to five servers with the 'Most Updated' rule. The network is static with 20% of nodes designated as servers. A positive decrease represents fewer hops and therefore better performance.

Table 7.4: Average hops Random / Most v Most Updated rule.

Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Server Random	3.2	3.2	3.1	3.1	3.0	2.9	2.7	2.4	1.7
1 Server Most Updated	3.8	3.8	3.8	3.5	5.5	3.0	2.8	2.3	1.7
Decrease %	**-19**	**-19**	**-23**	**-13**	**-83**	**-3**	**-4**	**4**	**0**
2 Servers Most	7.3	7.3	7.1	6.6	6.3	5.6	5.0	4.2	3.0
2 Servers Most Updated	6.7	6.5	6.2	6.2	5.8	5.1	4.6	3.9	2.9
Decrease %	**8**	**11**	**13**	**6**	**8**	**9**	**8**	**7**	**3**
3 Servers Most	12.8	12.7	12.2	11.2	10.7	9.1	7.8	6.2	4.3
3 Servers Most Updated	11.6	10.9	10.6	10.1	9.2	8.1	6.8	6.0	6.0
Decrease %	**9**	**14**	**13**	**10**	**14**	**11**	**13**	**3**	**-39**
4 Servers Most	19.5	19.4	18.5	17.1	15.5	13.0	11.0	8.4	5.7
4 Servers Most Updated	15.9	15.6	14.8	14.6	13.4	11.6	9.9	8.1	6.0
Decrease %	**19**	**20**	**20**	**15**	**14**	**11**	**10**	**4**	**-5**
5 Servers Most	27.1	27.0	25.3	24.1	21.6	17.0	13.7	10.4	7.4
5 Servers Most Updated	22.0	20.8	21.1	19.9	18.1	15.2	13.2	10.8	7.8
Decrease %	**19**	**23**	**17**	**17**	**16**	**11**	**4**	**-4**	**-5**

To strictly enforce a server rule, the network must constantly check the server positions and if necessary a server must surrender server status to a non-server node in a better location. This requires a constant polling by the servers of all nodes in the network asking how many neighbours they have. If a non-server node has more neighbours than a server, the server exchanges roles with the non-server including all certificate information that it possesses. This process must be performed on a regular basis and for the purpose of the simulations was done every one second. Only when at least one server exists in the network will it be possible to exchange roles with a non-server. It may take a few seconds of simulation time before two nodes are placed within radio range and a network is formed. Therefore the number of iterations of polling members for their count of neighbours is slightly less than six hundred. This amounts to a huge amount of management traffic flowing to the servers for the purpose of enforcing the rule. The average number of server / non-server role exchanges for the static network is shown in Table 7.5.

Table 7.5: Average server role exchanges Most Updated rule for 1-5 servers.

1 Server	87
2 Servers	113
3 Servers	129
4 Servers	150
5 Servers	145

This table shows the effectiveness of performing the updates every second. Every server role exchange that takes place is a worthwhile result for the server location enquiries.

7.5.4 Conclusion for Most Updated Server Rule

Table 7.5 shows the number of server role exchanges with almost six hundred server location enquiries from servers to all nodes in the network (over the 600 seconds). If

one server is required, on average 87 server role exchanges occur for no reward in effectiveness or efficiency. In fact, for one server this intense message passing results in a penalty for both certificate success and number of certificate message hops. For two servers required, on average 113 server role exchanges take place resulting in approximately the same success rate but with approximately 10% fewer hops required for low to medium trust thresholds. If three to five servers are required, the effort is rewarded with slightly higher success rates and fewer server hops. This increase in efficiency and effectiveness may be useful for applications where performance is absolutely vital. However, this constant exchange of roles resulting in sensitive encryption key and certificate information is highly risky. Not only do non-server nodes receive all information held by the server when exchanging roles, but the information may pass through several nodes along the communication chain to reach the new server. This makes strictly enforcing the server location rules only suitable for networks where more than one server is required to obtain a certificate and where all members can be trusted and certificate issuance performance is the primary concern.

Whilst applications may exist where this is the case, generally security will be such a concern that updating the servers' location is not worthwhile, both in risk of attack and in the intense message passing that is required. Therefore, for one server required for certificate services a random placement of servers should be implemented, and for more than one server required the 'Most Neighbours' rule should be implemented unless special circumstances exist where the 'Most Updated' rule is acceptable.

7.5.5 Least Updated Server Rule

By forcing servers to be nodes with the fewest number of neighbours, this rule tends to select servers towards the outer edges of the networks. This may be beneficial to nodes located near the edges, but if inter-server communication is required, it will often require the network to be traversed multiple times to reach other servers, especially as more servers are required for certificate issuance and revocation. Figure 7.24 shows a typical configuration for a small network where five servers are required to obtain a certificate. The node with the large circle around it is requesting a certificate and the first server is contacting the other four servers to obtain their certificate shares. As can be seen, communication stretches the entire length of the network resulting in thirty hops required to obtain the certificate.

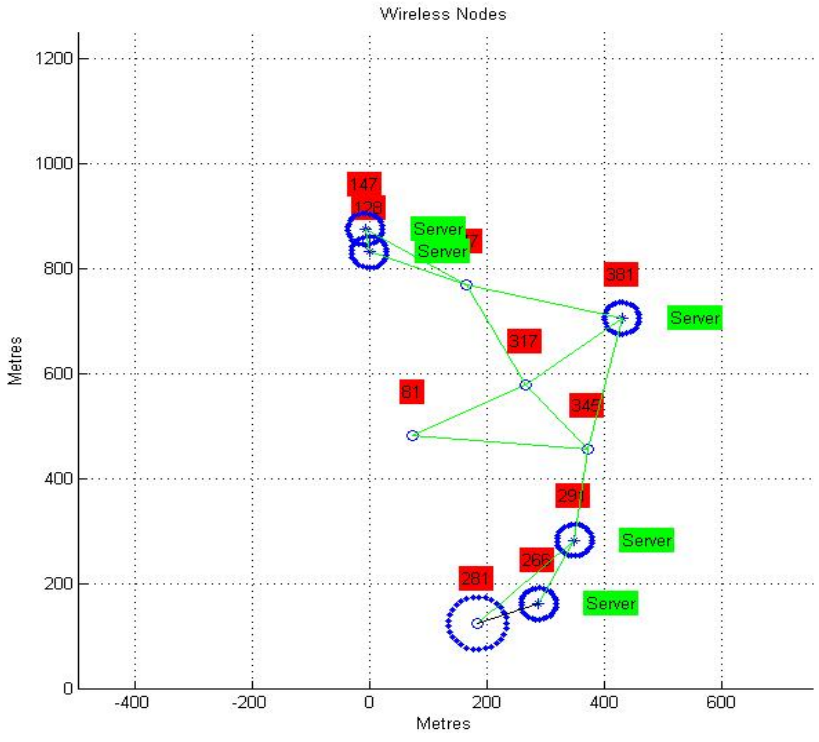


Figure 7.24: Least rule with 5 servers required – 30 hops.

In a small network the effect of server placement is not as noticeable as with a densely populated and geographically spread network. The network in Figure 7.24 shows how the servers will tend towards the outside and on average will result in greater hops to obtain certificates. If the network were considerably larger then servers could spread out more so that a server may not have another server as a neighbour. This additional spread would worsen the problem of inter-server communication resulting in more hops for a certificate and consequently more failures. As the main penalty for dispersed servers is when inter-server communication is required, it is interesting to see the effect when few servers are required compared to higher numbers of servers. As the random placement of servers has been shown to be best when one server is required, the following figure compares the Random server rule for one server and the 'Most' server rule for two servers versus the 'Least Updated' server rule. The following figure shows a static network, firstly with one server required followed by two servers required.

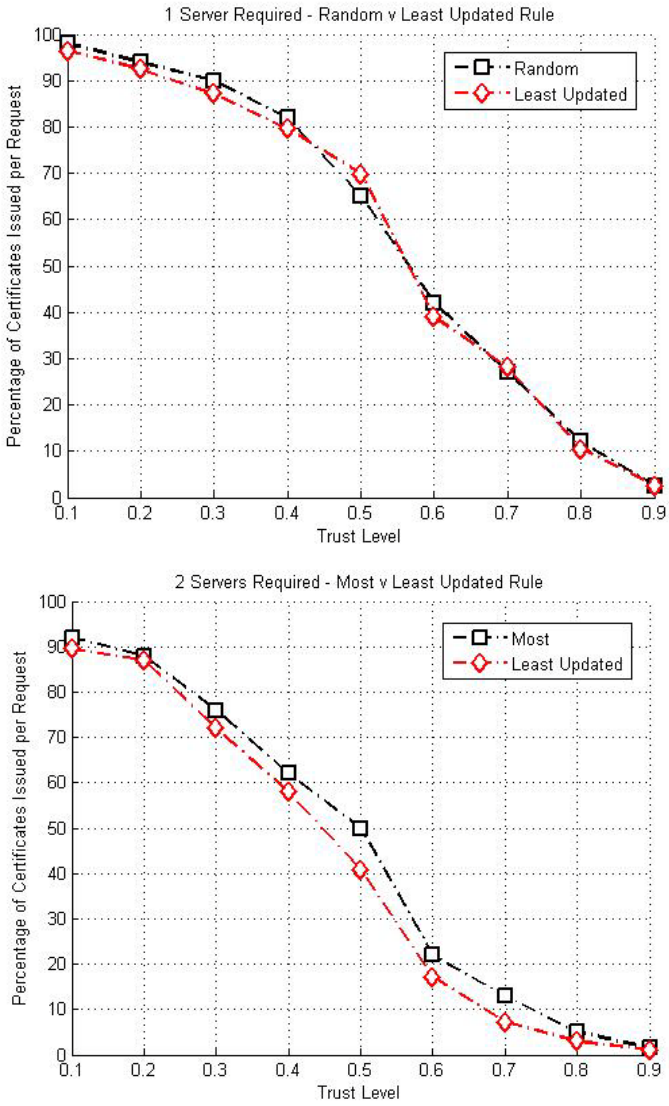


Figure 7.25: Success Rate: Random / Most vs Least Updated 1 and 2 servers required.

As expected, random placement of servers for one server required is best but by a very small percentage. For a 0.5 trust threshold it actually performs slightly worse than for the 'Least Updated' server rule. Overall, there is such a small difference between the two rules that the random placement remains best due to the lower hop count. For two servers required the 'Least Updated' rule results in less certificate success for all trust thresholds. Therefore, for one or two servers required the 'Least Updated' rule should not be used. Figure 7.26 shows network results for three, four and five servers required.

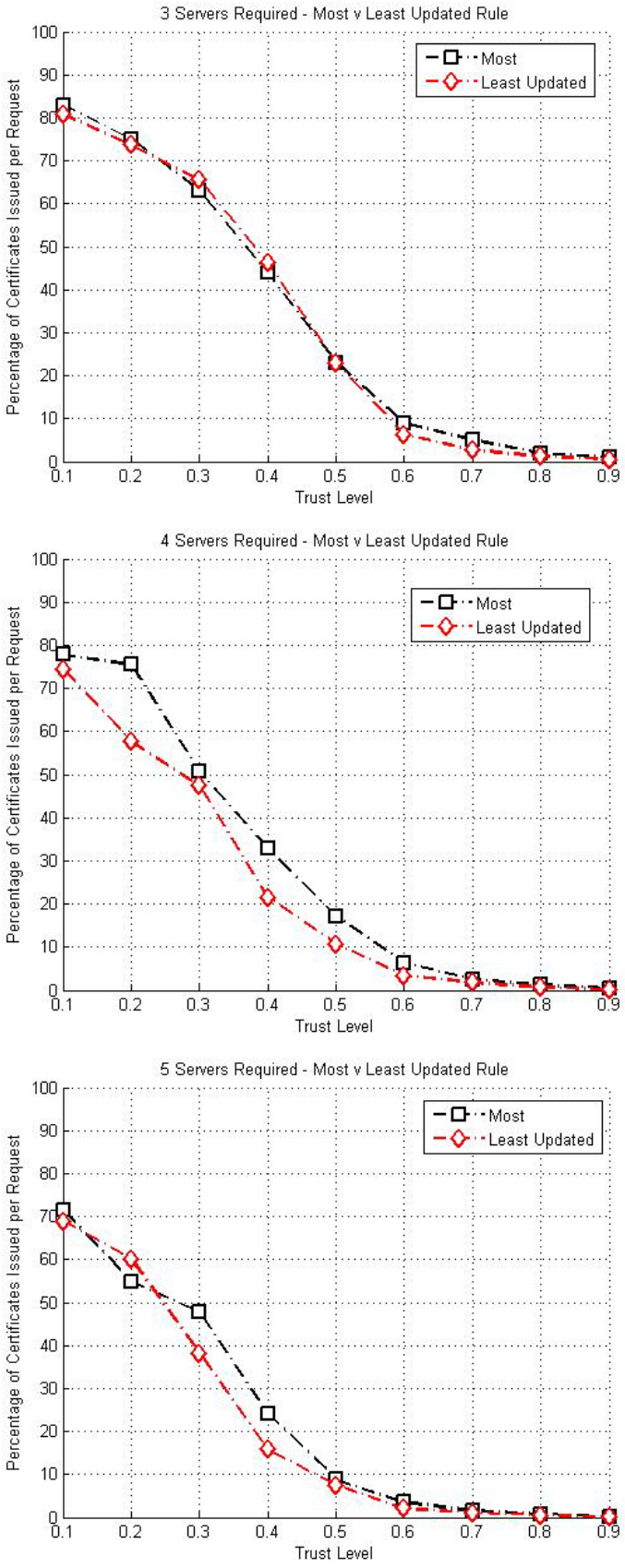


Figure 7.26: Success rate: Most rule vs Least Updated rule: 3, 4 and 5 servers required.

Figure 7.26 shows that when three servers are required the difference between the two rules is negligible except in the lower trust thresholds. However, when four or five servers are required the difference is more marked. Here, the 'Most' rule performs better especially in the mid trust threshold levels. Table 7.6 shows that the 'Least Updated' rule requires more hops to gain a certificate except when there is a trust threshold of at least 0.7 or 0.8. In this case, performance as regards number of hops required is slightly better with the 'Least Updated' rule. In all cases where the 'Least Updated' rule is used, certificate success rates are lower than when either the 'Random' server placement rule is used for one server or the 'Most' neighbours server placement rule is used for two to five servers. Also, the hop counts are greater using the 'Least Updated' rule except for the cases of trust values of 0.7, 0.8 and 0.9 where hop counts for successful certificate issuances are slightly better.

Table 7.6: Average hops Random / Most v Least Updated rule.

Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Server Random	3.2	3.2	3.1	3.1	3.0	2.9	2.7	2.4	1.7
1 Server Least Updated	3.7	3.5	3.5	3.4	3.3	2.9	2.6	2.2	2.0
Decrease %	**-15**	**-9**	**-13**	**-10**	**-10**	**0**	**4**	**8**	**-26**
2 Servers Most	7.3	7.3	7.1	6.6	6.3	5.6	5.0	4.2	3.0
2 Servers Least Updated	8.3	8.1	8.0	7.6	6.9	5.9	5.1	3.8	2.0
Decrease %	**-14**	**-11**	**-13**	**-15**	**-9**	**-5**	**-2**	**10**	**33**
3 Servers Most	12.8	12.7	12.2	11.2	10.7	9.1	7.8	6.2	4.3
3 Servers Least Updated	14.0	13.7	13.6	12.8	11.5	9.5	7.3	5.6	3.1
Decrease %	**-9**	**-8**	**-11**	**-14**	**-7**	**-4**	**6**	**10**	**28**
4 Servers Most	19.5	19.4	18.5	17.1	15.5	13.0	11.0	8.4	5.7
4 Servers Least Updated	23.1	22.5	21.4	19.7	16.6	13.1	10.5	7.4	4.6
Decrease %	**-22**	**-16**	**-16**	**-15**	**-7**	**-1**	**5**	**12**	**19**
5 Servers Most	27.1	27.0	25.3	24.1	21.6	17.0	13.7	10.4	7.4
5 Servers Least Updated	32.3	30.8	30.1	26.7	23.5	17.3	13.6	10.3	6.4
Decrease %	**-19**	**-14**	**-19**	**-11**	**-9**	**-2**	**1**	**1**	**14**

Whilst the 'Least Updated' rule does perform better in some of the cases, the penalty for implementing the 'Least Updated' rule by polling nodes constantly requires a huge amount of management traffic that outweighs any slight benefit the rule provides.

The average number of server / non-server role exchanges for the static network is shown in Table 7.7.

Table 7.7: Average server role exchanges Least Updated rule for 1-5 servers.

1 Server	30
2 Servers	26
3 Servers	23
4 Servers	17
5 Servers	15

If the results of the server role exchanges are compared with the server role exchanges for 'Most Updated' rule in Table 7.5, there are considerably fewer exchanges for the 'Least Updated' rule. Additionally, for the 'Most Updated' rule, adding servers required for certificate services increases the number of server role exchanges that take place. For the 'Least Updated' rule, the opposite effect occurs where adding servers required leads to a reduction in server role exchanges. Figure 7.27 compares the number of server role exchanges for the 'Most Updated' rule and the 'Least Updated' rule.

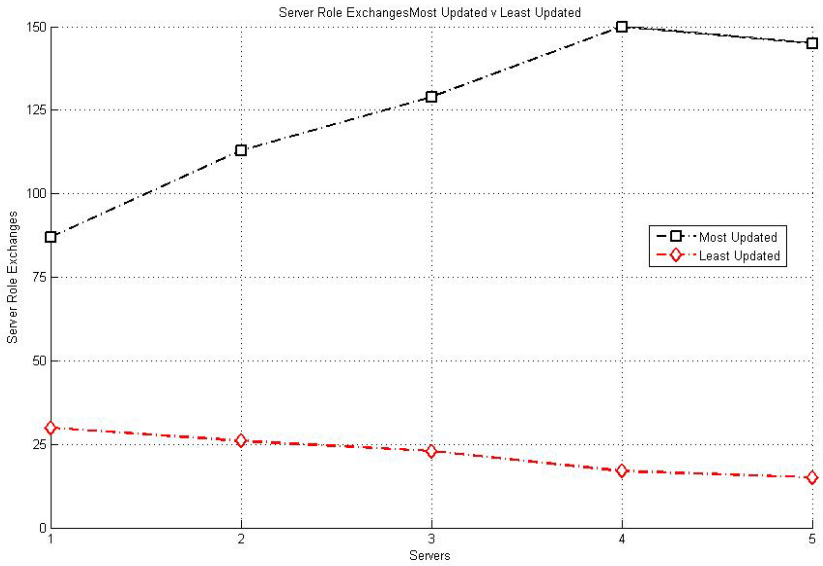


Figure 7.27: Server role exchanges for Most Updated rule vs Least Updated rule.

This figure shows the divergence the two rules have for the number of server role exchanges. Approximately the same number of messages polling the network members for their neighbour counts will result in considerably less actions to exchange server roles with non-servers when using the ‘Least Updated’ rule. This is because the network neighbour counts for nodes nearer the centre of the network are more likely to increase with a growing network as opposed to the neighbour counts of the nodes on the outer fringes which are less likely to change. Placing nodes initially on the outer fringes therefore means they will likely remain as servers for a longer period than for the ‘Most’ Server rule and require much fewer exchanges of roles. One benefit to this is that with fewer server roles exchanged, less sensitive information is also exchanged resulting in greater robustness against attack from malicious nodes.

7.5.6 Conclusion for Least Updated Server Rule

Updating the server rule constantly results in considerably more communication within the network. This constant management message passing is a serious drawback to the viability of this method of strictly enforcing the rules. Constant polling of the members utilises valuable resources of the network leaving availability of those resources for other communication severely limited. Additionally, the exposure of sensitive certificate information results in serious security concerns. All that is required for a malicious node to mount a successful attack is to deliberately place itself in the best location complying with the server location rule. It is likely that the attacker will become a server and receive one server’s entire certificate and key information. If the attacker moves in and out of the best location, it may well exchange roles several times. This means that even if several servers are required to obtain a certificate the attacker may be disclosed as many as are required servers’ information. By collecting several servers’ information

the malicious node may obtain all parts to the issued certificates disclosing all encryption key information to a single, malicious node. This serious security flaw makes updating the rules only suitable for networks where total trust in all members is assured and performance of the certificate management process outweighs any other concerns.

7.6 Comparisons with other Protocols

As SKYE is a new protocol with unique characteristics, direct comparisons of performance with existing protocols is difficult. Often, partial protocols are designed where the full workings do not exist and therefore performance cannot be quantified. Additionally, protocols are often published where sufficient information about their design or performance is not made available and so the performance cannot be accurately judged. Several protocols are improvements on previous protocols in efficiency and security and for this reason only the later developments and most relevant protocols are compared. The differences in the protocol rules have been covered in earlier chapters and so the comparisons made here are more performance based.

The MOCA protocol is a later development of the Zhou and Haas protocol (Zhou and Haas 1999). It uses threshold cryptography with a partially distributed mobile CA. Nodes requesting a certificate must contact a server node by broadcasting a certificate request. Server nodes are configured offline and chosen for their physical characteristics: either for superior processing power or physical robustness against attack. This is a significant drawback of the protocol as it requires prior planning making it unsuitable for truly ad hoc networks. MOCA requires that all servers be

mobile whilst SKYE permits mobility from any node but does not require it. In MOCA, when a node requests a certificate from a server it generally broadcasts the request to all nodes within the network. SKYE makes a distinction between a blind request and an informed request improving efficiency when more than two servers are required for certificate services by ensuring nodes are informed of the rules by periodic broadcast messages that may service multiple new nodes that may wish to join. The requester can therefore contact a server directly knowing where the closest server is located. In a network where nodes may be joining at a high rate, the broadcast network status messages eliminate the need for broadcast requests reducing management message overhead.

When providing certificate issuance facilities, MOCA nodes reply directly to the certificate requesting node placing the burden of ensuring enough authentic replies are received on the requester. SKYE places this burden on the first server to receive a certificate request as the server is better equipped to deal with false replies from servers or routes that fail to meet the required trust threshold. Additionally, servers can cache routes to other servers so that as requests are received by the additional servers, the return routes to the other servers can be tried first without route request messages. This added efficiency makes SKYE more efficient than the MOCA protocol and more able to handle misbehaving server nodes as honest servers can deal with these problems without involving the certificate requester. With extensive experimentation of server placement, SKYE also performs more effectively by ensuring shorter hop counts than MOCA when at least three servers are required for certificate services.

The Self Organised Public Key Management protocol (SOPKM) also utilises offline configuration of nodes where all nodes are considered equal. This is in the truly ad hoc spirit similar to SKYE. All nodes create their own public / private key pair which is efficient but assumes all nodes are entirely trustworthy as authentication of the nodes is not enforced. SKYE ensures nodes cannot fake their identities by requiring enforcement of a non-changeable identity making SKYE less trusting in the nodes and more robust against misbehaviour. Additionally, SOPKM enforces certificate expiry but requires a node to contact the certificate issuing node to renew their certificate. If that node has left the network, the certificate cannot be updated and the node is effectively ejected. SKYE does not enforce certificate expiry but relies on nodes to monitor each other for misbehaviour. This significantly reduces the management traffic required for constant certificate updates and eliminates the risk of failure of the update process making SKYE more efficient and robust.

Finally, the URSA protocol (Luo, Kong, Zerfos, Lu and Zhang 2004) utilises threshold cryptography with monitoring of neighbours for misbehaviour. Offline configuration of a selection of nodes is performed prior to network formation. Nodes may join the network by requesting a ticket and it is the neighbours that monitor behaviour and therefore choose to issue the ticket or not. Solely relying on neighbours rather than a group of servers for certificate services makes certificate issuance more efficient as the number of hops for a certificate will be low.

However, Byzantine behaviour is relatively easy to implement by placing colluding nodes in groups near other nodes and therefore simply voting to permanently eject innocent nodes. The increase in efficiency is more than countered by the increase in

vulnerability to attack making this scheme considerably less secure than SKYE. Additionally, expiry of the ticket requires constant localised message passing to update tickets at regular intervals. This extra management overhead created by expiration of tickets reduces the availability of the network for other, non-management message exchange.

The previous relevant schemes developed for MANETs all require offline configuration of some or all of the nodes before the network can form. This makes them unsuitable for truly ad hoc networks where deployment of the network may be performed quickly as a necessity and with no prior knowledge of nodes to each other. SKYE permits this truly ad hoc deployment and performs efficiently and with sufficient robustness and security to make it superior to previous designs in such circumstances that a truly ad hoc network is likely to encounter.

7.7 Experimental Implementation Scenario

Simulations are a necessary method for this type of research to provide a test bed where a large number of experiments can be performed in a relatively short time and with considerable more flexibility than is possible with a practical experimental setup. Simulations are designed to answer “What if?” questions by changing the variable parameters and simulation scenarios of a protocol and observing the effect on performance. Whilst measurement of performance parameters based on an actual implementation of the protocol would be desirable, it is simply not possible to do so because of the time and resource constraints that prevail and the highly dynamic nature of the network meaning many thousands of experiments would be required to glean enough useful data. A practical test bed would require as many experiments in the field

as simulation runs in this research to be useful in making direct comparisons with the results. That is, each network setting would require ten different experiments, all averaged to obtain a result. The steps that would be required to do such experiments are:

1. Protocol software must be written that would run on several different platforms: Laptop computer, PDA, Smart Phone.
2. Approximately 400 wireless devices would need to be used with the software installed.
3. Approximately 400 people would be required, 1 per device.
4. A 2500 metre square area without obstructions would need to be found.
5. Approximately 50 000 different experiments would need to be run, each requiring a different setup and collection of results. The time between experiments to perform these tasks would be approximately 1 hour.

The logistics of implementing the protocol in a practical test bed setup to compare results with the simulations make such effort an unrealistic task, leaving simulation the only possible method to perform such a large number of experiments. A small experimental setup could be implemented with a few devices to test the basic performance of the protocol. If this was desirable, then a minimum of six devices could be used making five servers and one non-server. If a basic test of the protocol was sufficient, then the devices could be small, static sensor networks in a laboratory environment which could run the protocol and create and distribute digital certificates. Whilst the results would not be sufficient for a thorough verification of performance, this small experiment could be used to verify the logic of the protocol and ensure all requests for certificates resulted in the issuance of a certificate to the requester.

7.8 Conclusion

Performance of the protocol can be measured by a number of factors. A measure of effectiveness for the protocol is the percentage of certificate requests that result in a successful certificate issuance. Efficiency is a measure of how efficiently the process of key management tasks can be performed which is measured by the amount of communication that must be performed to complete the task. With certificate requests, the number of hops required to obtain a certificate should be the minimum required. By manipulating the input variables and observing the output, guidelines as to what input parameters are appropriate can be gauged.

Management messages should be kept to the minimum number required to perform the management tasks. The more management messages that are exchanged leads to a better ability to optimise the key management processes but at the cost of utilising valuable network resources such as time and CPU processing power. Therefore, a balance must be struck between management messages and network resources available.

Results show that a node joining the network should broadcast its presence and any reply should indicate the network settings for servers required and trust threshold. If more than two servers are required, the new nodes should maintain a server count from these broadcast messages until sufficient servers exist in the network to meet the threshold. Placement of the servers has a significant effect on performance. If one server is required, then randomly selecting the servers is the most effective method for minimising certificate request communication. Above one server, server locations should be chosen by selecting those nodes with the most number of neighbours. Only in exceptional circumstances where total trust in members exists and effectiveness of

certificate requests is of utmost importance should the ‘Most Updated’ server placement rule be used. The very high overhead of management messages required to enforce this rule makes its implementation very unlikely and it is difficult to imagine a case where its implementation is warranted. In no circumstances does the ‘Least Updated’ rule provide any significant advantage and so this rule should not be used.

Security within the network relies on threshold cryptography where multiple servers combine to perform a CA role and provide key management services. If one server is required to obtain a certificate, then security is extremely low as any node that joins the network may randomly be selected to become a server. This single node will exchange certificate information with other servers and will then have considerable knowledge of certificates issued in the network. As there are no criteria other than location for selecting a node to become a server, it is just as likely a malicious node rather than a trustworthy node will take on server status. This severe lack of security makes requiring a single server for certificate services a poor choice for all applications other than those that can fully trust all members of the network and require very low resilience against attack.

As the number of servers required is increased, so is the robustness against attack. By having more servers in the network than are required to perform certificate services, redundancy is available making the CA more robust against failure and more available. Increasing the servers required increases the number of messages that must be exchanged to perform the tasks but generally the benefits of increased security outweigh the message passing penalty imposed. Security within the certificate issuance process can further be enhanced by raising the trust threshold for certificate messages. As the

trust threshold is raised, the certificate request failure rate drops due to the certificate request chain calculation dropping below the threshold. This reduction in trust calculation along the chain is the result of malicious nodes that become less and less trusted by their neighbours as they misbehave. Whilst malicious nodes can disrupt the network by their misbehaviour, the protocol is designed to identify and if necessary eject the misbehaving nodes. This robustness against maliciousness, both individually and with Byzantine behaviour is a further attribute of the threshold cryptographic approach.

This protocol provides many tunable parameters that can be adjusted as necessary depending on the application and possibly the trust that may exist between members prior to network formation. The protocol is designed for the truly ad hoc network formation where no prior knowledge of members to each other exists and no prior planning for the network has taken place. The tunable parameters allow secure and robust certificate and key management services making the protocol practical for a wide variety of applications.

As a guide for implementation of the protocol, lookup tables for the input parameters with a moderately mobile network and the resulting success rates are available in appendix 1. For general implementation of the protocol, Table 7.8 provides an example default setting that could be used but with the option to change any of the settings where such changes would make the protocol more suitable for the particular application it is used for.

Table 7.8: Recommended default settings for SKYE.

Servers Required	Servers Percentage	Server Rule	Trust Threshold
3	20	Most	0.6

The following chapter discusses the conclusions that can be drawn from the research including the contributions that this thesis provides to the area of MANET security. Following this is a discussion of the future work that may be performed to further enhance the SKYE protocol.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Introduction

This chapter discusses the general conclusions that can be drawn from the results of the simulations and the contributions to the field of research into MANET security that this thesis provides. Firstly an overview of the protocol is looked at followed by a discussion of future work that may be done to further enhance the protocol.

8.2 Conclusions

The goal of this work was to develop a new encryption key management protocol with enhanced features for mobile ad hoc networks. The purpose of the protocol design is to provide a robust and efficient scheme that has tunable parameters allowing a balance between efficiency of key management services and effectiveness of security. Truly ad hoc networks are distinguished from other types of wireless networks more by what is not known than what is known. Ad hoc networks are often formed entirely on-the-fly for a brief, specific task and then disbanded once they have served their purpose. This on-the-fly nature makes providing security extremely difficult. Often protocols are designed for a particular application or for a network of predetermined specifications such as size, number of nodes and mobility patterns. If these parameters are not known before initialisation, then the ability to tune the parameters to fit the network is highly desirable. Almost all previous protocols assume offline configuration of the security settings and prior knowledge of network size and number of nodes. Additionally, preconfiguration of server nodes and their placement in the network is often performed, but in a truly ad hoc network this is not realistic. SKYE differs from these previous designs in that it assumes no priory knowledge of the network parameters and uses no

prior offline configuration. It therefore utilises extensive tunability to tailor the parameters to the needs of the network. This makes the protocol suitable for a number of different applications where the speed of deployment and scalability are the primary concerns. Such applications are numerous and may include education, disaster relief where victims of natural disasters can establish a network themselves, or military exercises where little likelihood of malicious attack exists. A wide variety of other possible uses exist where any group of people equipped with suitable wireless devices may wish to quickly and spontaneously create a MANET.

Threshold cryptography is employed which spreads the certificate authority over a number of nodes designated as servers. These server nodes must collaborate to provide the key management services and the number of servers that must collaborate to form the certificate authority is a parameter that is tunable for the network. Threshold cryptography makes malicious attack more difficult as an attacker must compromise a number of servers to gain sufficient information to disclose private keys that have been issued to nodes. As more servers are required to collaborate to perform the CA tasks, increased robustness to attack results. However, the penalty for the increase in robustness is more communication between servers to perform the key management tasks. The tradeoff between the robustness and the inter-server communication required can be tuned to provide an acceptable level of efficiency whilst maintaining an acceptable security level. Additionally, if more servers exist than are required to perform the CA role, redundancy of servers exists making the CA more available and more robust against unreliable communications or malicious attack.

The primary goal of effectiveness aims to have as many authentic certificate requests return a certificate to the requester as possible. However, the likelihood exists that as the requests and key management messages are passed from node to node along a chain to their destination, a malicious node may read the request, alter the request or inject their own false request. This likelihood is combated by setting a threshold for the certificate chain that must be met or exceeded for the request to be processed. Each node in the chain multiplies the trust calculated along the chain by their trust value in the preceding node. Any prior misbehaviour by the previous node in the chain will reduce the next node's trust value for it. The longer the chain to the servers, the higher the likelihood that a malicious node will be encountered and therefore the trust level will reduce below the network threshold. A low trust threshold will increase the likelihood of a successful certificate issuance and that likelihood decreases as the length of the certificate chain increases. Setting a higher trust threshold means security along the chain is higher as mistrust is less tolerated. However, if the certificate request fails, a new request must be made by the requesting node. The percentage of requests that are successful is a suitable measure of the effectiveness of the protocol. As the success rate drops, more multiple requests must be made utilising valuable network resources that could be better utilised for inter-node communication.

The second goal was to provide an efficient network. That is, the least number of messages possible to perform key management tasks should be exchanged. If intermediary nodes between a sender and receiver are used to pass on messages, then those nodes and any other nodes within radio range must wait to transmit messages whilst the key management messages are sent and passed on. Therefore, the shorter the chain for the messages means the more efficient is the protocol. This message hopping

involving intermediary nodes requires that ideally servers be placed relative to nodes at locations that facilitate short message chains. The simulations show that random placement is best for a single server CA, and for any more than one server required for key management services the nodes with the most number of neighbours should perform the server roles. Where possible, the results have been compared to other protocols and SKYE was found to perform better in certain circumstances than other protocols. However, the extensive tunability allowing SKYE to be tailored to the application or the users' requirements is a major departure from most other protocols and along with several new features makes SKYE a truly unique and practical protocol. The extensive results provide valuable information about the efficiency of threshold cryptography within a MANET environment and can be used to further enhance the SKYE protocol or to assist development of other protocols that utilise threshold cryptography.

Development of an encryption key management protocol involves considerable review of previous work to set the baseline for enhancement of security or efficiency or a combination of both of those attributes. Simulations have shown that SKYE achieves its goal of providing robust security with sufficient efficiency to enable it to be useful in a variety of applications. Further enhancements of the protocol are expected to provide greater efficiency in key management tasks so that greater security can be applied whilst maintaining acceptable efficiency. As efficiency is increased a balance can be maintained by increasing the security in several ways such as higher numbers of servers required to obtain a certificate or increasing the certificate chain threshold for the network. Simulations show that development of the protocol has been successful and a useful protocol applicable to a variety of real-life situations is the result.

The stated goal of this research has been to develop a new encryption key management protocol that will perform better than previous protocols in areas such as rapid deployment, versatility, availability, redundancy and tunability. Results show that this goal has been achieved, and that under certain conditions the unique approaches to providing encryption key management entirely online has provided a new KMS suitable for the types of applications discussed in Chapter 1.

8.3 Future Work

The simulation results for this protocol have identified trends that can be used to adjust the input parameters to fit the application and users' needs. This tunability relies on the results to predict what settings for the network will provide the best balance between effectiveness of the security and efficiency. Further work that would enhance this protocol may be:

1. A greater study of real-life ad hoc networking to better predict node joining and leaving rates within various likely applications.

2. Better predictability of likely malicious node numbers. Whilst likely that attacks or deliberate misbehaviour would be uncommon, different scenarios may have quite different malicious node rates. Very little data on misbehaviour in ad hoc networks exist and empirical data in this area would provide realistic attack models that could more thoroughly test the protocol.

3. Further research into server location choices so that ideal server locations could be chosen whilst maintaining efficiency. For example, the "Most Updated" rule

was best for efficiency when requesting a certificate but the management messages required to maintain the rule made its use unreasonable. A method for implementing the rule and maintaining efficiency should be sought.

4. Implementing the protocol in practice would be an interesting and worthwhile task. The type of scenario used could be a small wireless network such as a dynamic sensor network. These devices can be obtained at little cost and would be sufficient for a test bed. This would give feedback about performance and provide further research directions for continued development and enhancements.

5. Development of an efficient routing protocol that takes into account trust along the certificate chain would be advantageous. This would ensure the route with the highest trust calculation would be used for certificate requests. A routing protocol called “Least Cost” was experimented with but was found unworkable because of the number of iterations required to find the best route. For example, the dynamic source routing protocol found a path from sender to receiver of ten hops in twenty seven iterations. The same destination with the Least Cost algorithm took thousands of iterations. An effective yet efficient routing algorithm that takes into account the cost of each hop would be a valuable enhancement.

REFERENCES

Anderson, R., Haowen, C. and Perrig, A. (2004). "Key infection: smart trust for smart dust". *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP),* Berlin, Germany, October 5th-8th 2004, pp206-215.

Asokan, N. and Ginzboorg, P. (2000). "Key Agreement in Ad Hoc Networks". *Computer Communications*, Volume 23(17), November 2000, pp1627-1637.

AusCert. (2004). "Denial of Service Vulnerability in IEEE 802.11 Wireless Devices". Retrieved 19 July 2005 from http://www.auscert.org.au/render.html?it=4091&template=1

Basagni, S., Herrin, K., Bruschi, D. and Rosti, E. (2001). "Secure pebblenets". *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing,* Long Beach, CA, USA, October 4th-5th 2001, pp156-163.

Becker, K. and Wille, U. (1998). "Communication complexity of group key distribution". *Proceedings of the 5th ACM Conference on Computer and Communications Security,* San Francisco, California, USA, November 2nd-5th 1998, pp1-6.

Boneh, D. and Franklin, M. (2001). "Identity-Based Encryption from the Weil Pairing". *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology,* Santa Barbara, CA, USA, August 19th-23rd 2001, pp213-229.

Borisov, N., Goldberg, I. and Wagner, D. (2001). "Intercepting mobile communications: the insecurity of 802.11". *Proceedings of the 7th annual international conference on Mobile computing and networking (MOBICOM),* Rome, Italy, July 16th-21st 2001, pp180-189.

Burmester, M. and Desmedt, Y. (1994). "A secure and efficient conference key distribution scheme". *Proceedings of Eurocrypt 1994,* Italy, May 9th-12th 1994, pp275-286.

Capkun, S., Buttyan, L. and Hubaux, J.P. (2003). "Self-Organized Public-Key Management for Mobile Ad Hoc Networks". *IEEE Transactions on Mobile Computing,* Volume 2(1), January-March 2003, pp52-64.

Capkun, S., Hubaux, J.P. and Buttyan, L. (2003). "Mobility helps security in ad hoc networks". *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing,* Annapolis, Maryland USA, June 1st-3rd 2003, pp46-56.

Cavin, D., Sasson, Y. and Schiper, A. (2002). "On the accuracy of MANET simulators". *Proceedings of the Second ACM International Workshop on Principles of Mobile Computing,* Toulouse, France, October 30th-31st 2002, pp38-43.

Chung, J. and Claypool, M. (2010). "NS by Example". Retrieved on December 12th 2009 from http://nile.wpi.edu/NS/

Diffie, W. and Hellman, M. (1976). "New Directions in Cryptography". *IEEE Transactions on Information Theory*, Volume 22(6), November 1976, pp644-654.

Diffie, W., Oorschot, P. C. v. and Wiener, M. J. (1992). "Authentication and Authenticated Key Exchanges". *Designs, Codes and Cryptography*, Volume 2(2), June 1992, pp107-125.

Fleck, B. and Dimov, J. (2001). "Wireless Access Points and ARP Poisoning". Retrieved January 8th 2008 from http://www.ljudmila.org/matej/arppoison.pdf

Fluhrer, S. R., Mantin, I. and Shamir, A. (2001). "Weaknesses in the Key Scheduling Algorithm of RC4". In: Vaudenay, S., Youssef, A.M. (eds.) Selected Areas in Cryptography—SAC 2001. *Lecture Notes in Computer Science*, Volume. 2259, Springer, Heidelberg, pp1–24.

Hadjichristofi, G. (2005). "A Framework for Providing Redundancy and Robustness in Key Management for IPsec Security Associations in a Mobile Ad-Hoc Environment". Computer Science Dept, Virginia Polytechnic Institute and State University. Blacksburg, Virginia. PhD Thesis.

Hegland, A. M., Winjum, E., Mjolsnes, S. F., Rong, C., Kure, O. and Spilling, P. A. (2006). "A survey of key management in ad hoc networks". *IEEE Communications Surveys*, Volume 8(3), 3rd Quarter 2006, pp48-66.

Higgins, K., Egan, R., Hurley, S. and Lemur, M. (2006). "Ad Hoc Networks". Retrieved 11th December 2009 from http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group11/index.html.

Houliston, B. and Sarkar, N. I. (2005). "Wi-Fi Deployment: A Survey of Large New Zealand Organisations". *International Journal of Business Data Communications and Networking*, Volume 1(3), July-September 2005, pp37-58.

Hwang, Y.S., Han, S.W. and Nam, T.Y. (2006). "The expansion of key infection model for dynamic sensor network". *Proceedings of The 8th International Conference on Advanced Communication Technology, 2006,* Korea, February 20th-22nd, pp1-5.

IEEE. (1999). "IEEE Std 802.11b-1999". Retrieved 15th June 2004 from http://www.ieee.com.

IEEE. (2004). "IEEE Std 802.16-2004/Cor 1-2005". Retrieved 10th December 2006 from http://www.ieee.com.

Ingemarsson, I., Tang, D. and Wong, C. (1982). "A conference key distribution system". *IEEE Transactions on Information Theory*, Volume 28(5), September 1982, pp714-720.

Khalili, A., Katz, J. and Arbaugh, W. A. (2003). "Toward secure key distribution in truly ad-hoc networks". *Proceedings of the 2003 Symposium on Applications and the Internet Workshops*, Orlando, Florida, January 27th-31st 2003, pp342-346.

Kong, J., Zerfos, P., Luo, H., Lu, S. and Zhang, L. (2001). "Providing robust and ubiquitous security support for mobile ad-hoc networks". *Proceedings of the 9th International Conference on Network Protocols*, Riverside, CA, USA, November 11th-14th 2001, pp251-260.

Kunpisut, A. (2010). "Network Simulation Using GloMoSim". Retrieved 30th September 2010 from http://www.cpe.ku.ac.th/~anan/courses/204529/document/Sim/gs.ppt#40.

Kurkowski, S., Camp, T. and Colagrosso, M. (2005). "MANET simulation studies: the incredibles". *Mobile Computer and Communications Review*, Volume 9(4), January 2005, pp50-61.

Luo, H., Kong, J., Zerfos, P., Lu, S. and Zhang, L. (2004). "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks". *IEEE / ACM Transactions on Networking*, Volume 12(6), December 2004, pp1049-1063.

Lynn, B. (2002). "Authenticated Identity-Based Encryption". International Association for Cryptological Research, Retrieved 5th June 2009 from http://eprint.iacr.org/2002/072.pdf

Moskowitz, R. (2004). "Weakness in Passphrase Choice in WPA". Retrieved 4th September 2007 from http://wifinetnews.com/archives/002452.html

Nance, R. and Arthur, J. (1988). "The Methodology roles in the Realization of a Model Development Environment". *Proceedings of the 20th Conference on Winter Simulation*, San Diego, California, December 12th-14th 1988, pp220-225.

Naoumov, V. and Gross, T. (2003). "Simulation of large ad hoc networks". *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems (MSWIM 2003)*, San Diego, CA, USA, September 19th 2003, pp50-57.

Netgear. (2009). "Selecting between Infrastructure and Ad Hoc Wireless Modes". Retrieved 19 December 2009 from http://kb.netgear.com/app/answers/detail/a_id/954.

Pegden, C. D., Shannon, R. E. and Sadowski, R. (1990). Introduction to Simulation Using SIMAN. New York, NY, McGraw Hill Inc.

Perrig, A., Canetti, R., Briscoe, T., J and Song, D. (2000). "TESLA: Multicast source authentication transform". Retrieved 16 October 2007 from www.securemulticast.org/msec-bof-5-Perrig-tesla-ietf-bofPDF.

Perrig, A., Szewczyk, R., Wen, V., Culler, D. and Tygar, J. D. (2001). "SPINS: security protocols for sensor networks". *Proceedings of the 7th annual international conference on Mobile computing and networking, Rome, Italy*, July 16th-21st 2001, pp189-199.

Pietro, R. D. and Mancini, L. V. (2003). "Security and privacy issues of handheld and wearable wireless devices". *Communications of the ACM*, Volume 46(9) pp74-79.

Pietro, R. D., Mancini, L. V. and Jajodia, S. (2002). "Efficient and secure keys management for wireless mobile communications". *Proceedings of the second ACM international workshop on Principles of mobile computing*, Toulouse, France, October 30th-31st 2002, pp66-73.

Police, N. (2004). "New Zealand Crime Statistics". Retrieved 15th January 2010 from http://www.justice.govt.nz/publications/global-publications/r/research-on-the-effectiveness-of-police-practice-in-reducing-residential-burglary-november-2005-report-7-case-study-of-the-sydenham-police-area/10-crime-statistics.

Pužar, M., Andersson, J., Plagemann, T. and Roudier, Y. (2005). "SKiMPy: A Simple Key Management Protocol for MANETs in Emergency and Rescue Operations". *Proceedings of the 2nd European Workshop on Security and Privacy in Ad Hoc and Sensor Networks*, Visegrad, Hungary, July 13th-14th 2005, pp14-26.

Rhee, K., Park, Y. and Tsudik, G. (2005). "A group key management architecture for mobile ad hoc wireless networks". *Journal of Information Science and Engineering*, Volume 21(2), March 2005, pp415-428.

Rosario, G., Jarecki, S., Krawczyk, H. and Rabin, T. (2007). "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems". *Journal of Cryptology*, Volume 20(1), January 2007, pp51-83.

Shamir, A. (1984). "Identity-Based Cryptosystems and Signature Schemes". *Proceedings of Crypto '84 on Advances in Cryptology*, Santa Barbara, CA, *USA*, August 19th-22nd 1984, pp47-53.

Shoup, V. (2000). "Practical Threshold Signatures". *Proceedings of the 19th international conference on theory and application of cryptographic techniques*, Bruges, Belgium, May 14th-18th 2000, pp207-220.

Staddon, J., Miner, S., Franklin, M., Balfanz, D. A., Malkin, M. A. and Dean, D. A. (2002). "Self-healing key distribution with revocation". *Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, May 12th-15th 2002, pp241-257.

Steiner, M., Tsudik, G. and Waidner, M. (1998). "CLIQUES: a new approach to group key agreement". *Proceedings of the 18th International Conference on Distributed Computing Systems*, Amsterdam, Holland, May 26th-29th 1998, pp380-387.

Stubblefield, A., Ioannidis, J. and Rubin, A. (2002). "Using the Fluhrer, Mantin, and Shamir attack to break WEP". *Proceedings of Network and Distributed Systems Security Symposium (2002)*, San Diego, CA, USA, February 6th-8th 2002, pp17-22.

Takahashi, T. (2004). "WPA Passive Dictionary Attack Overview". Retrieved May 16th 2008 from http://www.tinypeap.com/docs/WPA_Passive_Dictionary_Attack_Overview.pdf

Vibhuti, S. (2005). "IEEE 802.11 WEP (Wired Equivalent Privacy) Concepts and Vulnerability". Retrieved 8th January 2008 from www.cs.sjsu.edu/~stamp/CS265/projects/Spr05/papers/WEP.pdf

Walker, J. (2002). "802.11 Security Series Part II: The Temporal Key Integrity Protocol (TKIP)". Retrieved 19 July 2005 from http://jcbserver.uwaterloo.ca/cs436/handouts/miscellaneous/Intel_Wireless_2.pdf.

Whalen, S. (2001, April 2001). "An Introduction to ARP Spoofing". Retrieved 6th February 2006 from http://packetstormsecurity.com/papers/protocols/intro_to_arp_spoofing.pdf

Wong, C. K., Gouda, M. and Lam, S. S. (2000). "Secure group communications using key graphs". *IEEE / ACM Transactions on Networking*, Volume 8(1), February 2000, pp16-30.

Wu, B., Wu, J., Fernandez, E. B. and Magliveras, S. (2005). "Secure and Efficient Key Management in Mobile Ad Hoc Networks". *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, Amsterdam, Holland, May 14th-15th 2005, pp288-293

Wullems, C, K. (2004). "A trivial denial of service attack on IEEE 802.11 direct sequence spread spectrum wireless LANs". *Proceedings of the Wireless Telecommunications Symposium*, Pomona, CA, USA, May 14th-15th 2004, pp129-136.

WWWD4. (2004). "World Wide War Drive 2004". Retrieved 16th January 2006 from http://www.worldwidewardrive.org/.

Yi, S. and Kravets, R. (2003). "MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks". *Second Annual PKI Research Workshop Program*, Maryland, USA, April 2003, pp65-79.

Yi, S. and Kravets, R. (2004). "Composite key management for ad hoc networks". *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, Ma, USA, August 22nd-28th, 52-61.

Zaballos, A., Corral, G., Serra, I. and Abella, J. (2003). "Testing Network Security Using OPNET ". *Proceedings of Opnetwork 2003*, Washington, DC, USA, August 2003,

Zhou, L. and Haas, Z. (1999). "Securing Ad Hoc Networks". *IEEE Network, Special Issue on Network Security*, Volume 13(6), November / December 1999, pp24-30.

Zhou, Y. and Fang, Y. (2006). "Security of IEEE 802.16 in Mesh Mode". *Proceedings of the IEEE Military Communications Conference*, Washington, DC, USA, October 23rd-25th 2006, pp1-6.

Zhu, B., Bao, F., Deng, R. H., Kankanhalli, M. S. and Wang, G. (2005). "Efficient and robust key management for large mobile ad hoc networks". *Computer Networks: The*

International Journal of Computer and Telecommunications Networking, Volume 48(4), 15th July 2005, pp657-682.
Zhu, S., Setia, S. and Jajodia, S. (2003). "LEAP: Efficient security mechanisms for large-scale distributed sensor networks". *Proceedings of the 10th ACM conference on computer and communications security*, Washington D.C, USA, October 27th-30th 2003, pp62-72.

Zhu, S., Setia, S. and Jajodia, S. (2006). "LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks". *ACM Transactions on Sensor Networks*, Volume 2(4), November 2006, pp500-528.

Zhu, S., Setia, S., Xu, S. and Jajodia, S. A. (2004). "GKMPAN: an efficient group rekeying scheme for secure multicast in ad-hoc networks". *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems*, Boston, MA, USA, August 22nd-26th, pp42-51.

Zhu, S., Xu, S., Setia, S. and Jajodia, S. (2003). "Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach". *Proceedings of the 11th IEEE International Conference on Network Protocols*, Atlanta, GA, USA, November 4th-7th, 326-335.

Zimmerman, P. (1994). PGP User's Guide. Cambridge MA, MIT.

Appendix 1

A1.1 Lookup tables for 20% mobility with varying speed.

Mobility patterns within an ad hoc network will be dependant on the application and circumstances the network is utilised for. As a guide, the tables show the percentage of certificate requests that result in successful certificate issuance. In all cases for one server required, the random selection of servers is used. In all other cases the node with the most number of neighbours should be selected as a server.

Table A1.1: Static Network.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	98	98	90	88	81	50	37	16	4
2 Most	92	88	76	62	50	22	13	4.8	2
3 Most	83	75	63	44	23	9	5	2	1
4 Most	78	75	51	33	17	6	3	1	1
5 Most	72	55	48	24	9	4	2	1	0

Table A1.2: Mobile 20% at 1-10 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	97	96	93	85	82	51	40	15	4
2 Most	92	88	76	63	50	22	13	5	2
3 Most	83	75	63	44	23	9	5	2	1
4 Most	78	76	51	33	17	6	3	1	1
5 Most	72	55	48	24	9	4	2	1	0

Table A1.3: Mobile 20% at 11-20 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	97	96	93	85	82	51	40	15	4
2 Most	91	86	82	62	50	25	13	5	2
3 Most	78	75	61	52	32	12	7	3	1
4 Most	78	74	56	41	21	7	4	2	1
5 Most	72	64	48	21	12	4	2	1	0

Table A1.4: Mobile 20% at 21-30 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	97	96	96	90	83	66	45	18	5
2 Most	93	91	82	74	67	30	16	7	2
3 Most	83	78	74	53	37	16	8	3	1
4 Most	76	72	57	42	23	9	4	2	1
5 Most	69	62	51	28	13	5	2	1	0

Table A1.5: Mobile 20% at 31-40 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	97	96	96	92	85	67	44	21	6
2 Most	93	91	87	79	65	35	18	8	3
3 Most	83	79	77	63	40	17	8	4	1
4 Most	76	71	57	46	24	10	5	2	1
5 Most	74	69	53	34	16	6	3	1	0

Table A1.6: Mobile 20% at 41-50 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	98	96	95	88	80	70	46	23	6
2 Most	93	93	89	83	65	37	18	8	3
3 Most	85	81	73	59	44	23	11	4	2
4 Most	78	74	67	45	28	11	6	3	1
5 Most	74	68	56	33	19	7	4	2	1

Table A1.7: Mobile 20% at 51-60 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	99	98	97	90	88	69	50	24	7
2 Most	92	90	87	80	64	36	22	8	3
3 Most	85	82	78	64	44	22	11	4	2
4 Most	77	74	63	52	28	11	6	3	1
5 Most	76	69	57	38	21	7	4	2	1

Table A1.8: Mobile 20% at 61-70 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	98	98	97	90	84	72	47	21	5
2 Most	93	91	88	76	66	40	20	10	3
3 Most	84	81	78	56	45	21	12	5	2
4 Most	81	77	64	55	31	12	6	3	1
5 Most	76	71	55	35	21	8	4	2	1

Table A1.9: Mobile 20% at 71-80 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	98	97	95	92	83	64	47	23	6
2 Most	94	93	91	79	67	40	23	11	4
3 Most	85	82	77	67	49	22	12	5	2
4 Most	80	75	66	49	32	12	7	3	1
5 Most	77	69	62	38	23	8	4	2	1

Table A1.10: Mobile 20% at 81-90 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	98	98	96	94	88	72	49	25	7
2 Most	93	92	89	84	66	45	23	11	4
3 Most	85	83	78	68	53	23	14	6	2
4 Most	81	78	73	56	32	13	8	4	1
5 Most	78	73	60	38	20	9	5	2	1

Table A1.11: Mobile 20% at 91-100 kmh.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Random	98	97	95	91	86	71	57	28	7
2 Most	93	92	87	82	67	48	26	12	5
3 Most	85	83	81	71	48	26	13	6	2
4 Most	81	78	69	51	38	17	9	4	1
5 Most	75	73	61	42	24	10	5	2	1

Appendix 2

A2.1 Lookup Tables for Server Percentage

The tables show the percentage of certificate requests that result in successful certificate issuance when varying percentages of servers are used. In all cases there are 20% of nodes mobile at 1-10kmh and the server selection rule is to select the nodes with the most number of neighbours to act as servers.

Table A2.1: Static and 10% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	97	88	78	68	46	22	11	5	2
2 Most	93	82	70	41	24	9	4	2	1
3 Most	81	70	43	24	10	4	2	1	1
4 Most	74	63	36	14	6	3	2	7	0.3
5 Most	69	52	20	9	3	2	1	0.5	0.2

Table A2.2: Static and 20% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	98	98	90	88	81	50	37	16	4
2 Most	92	88	76	63	50	22	13	5	2
3 Most	83	75	63	44	23	9	5	2	1
4 Most	78	75	51	33	17	6	3	1	1
5 Most	72	55	48	24	9	4	2	1	0

Table A2.3: Static and 30% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	98	96	91	89	75	60	44	18	4
2 Most	94	92	84	76	56	36	17	8	3
3 Most	81	78	71	52	36	20	9	4	1
4 Most	81	72	59	49	28	11	5	2	1
5 Most	72	67	54	35	16	7	3	1	0

Table A2.4: Static and 40% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	99	98	97	94	86	63	51	23	7
2 Most	93	91	83	72	57	49	27	12	4
3 Most	83	80	75	61	54	30	17	6	2
4 Most	80	69	62	52	35	17	9	3	1
5 Most	73	68	50	42	28	11	5	2	0

Table A2.5: Static and 50% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	99	95	92	92	88	75	60	47	11
2 Most	95	94	91	84	77	55	37	20	6
3 Most	84	82	74	59	51	39	20	8	3
4 Most	79	75	67	58	43	23	14	4	1
5 Most	75	69	58	45	35	16	7	2	1

Table A2.6: Static and 60% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	98	97	96	93	86	77	60	42	14
2 Most	95	91	85	76	64	57	40	26	7
3 Most	87	85	82	72	60	44	28	12	4
4 Most	83	81	69	60	56	34	15	6	2
5 Most	73	69	63	54	39	19	10	3	1

Table A2.7: Static and 70% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	99	98	98	95	91	85	79	53	19
2 Most	95	95	93	87	83	70	54	35	10
3 Most	86	84	82	73	63	51	37	17	5
4 Most	82	80	76	64	49	36	24	8	2
5 Most	76	72	65	61	43	24	13	5	1

Table A2.8: Static and 80% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	99	98	97	93	94	87	77	64	38
2 Most	95	92	89	89	83	73	68	44	16
3 Most	86	84	82	69	65	55	43	24	7
4 Most	84	81	76	63	58	46	31	13	3
5 Most	78	74	67	60	45	30	17	6	2

Table A2.9: Static and 90% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	99	98	97	97	97	88	84	81	43
2 Most	97	96	94	89	88	78	77	59	26
3 Most	90	88	87	79	72	65	56	33	11
4 Most	85	82	78	73	63	45	39	18	5
5 Most	79	76	71	59	48	34	24	10	3

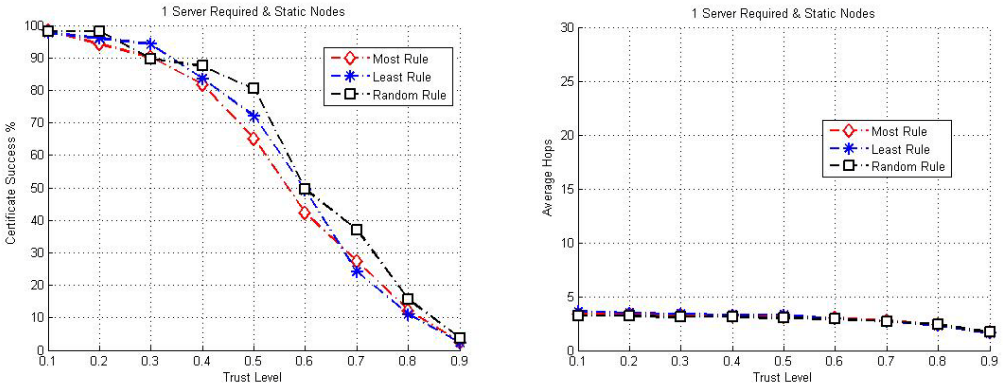
Table A2.10: Static and 100% Servers.

Servers / Trust	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
1 Most	100	100	100	100	100	100	100	100	100
2 Most	100	100	100	100	100	100	100	100	100
3 Most	97	97	97	97	96	96	96	95	69
4 Most	91	91	90	88	85	76	65	59	12
5 Most	83	81	80	77	67	57	43	23	4

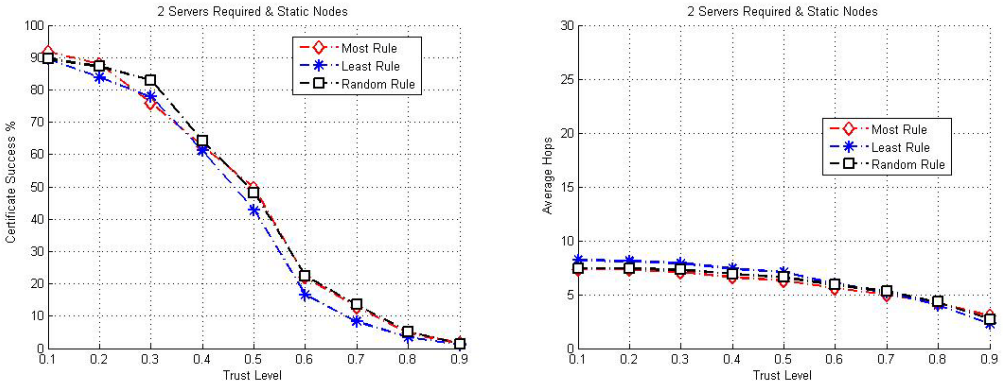
Appendix 3

3.1 Server Location Comparisons

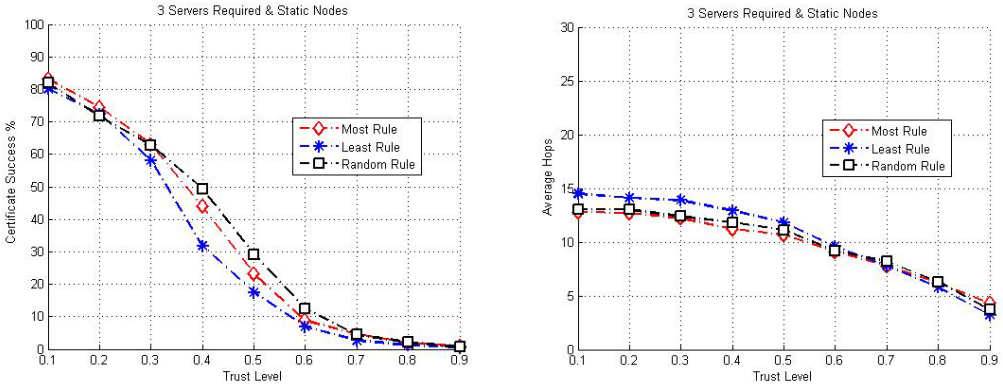
The graphs show the differences in efficiency between the three different server locations where the positions are not updated. In each case the network is stationary.



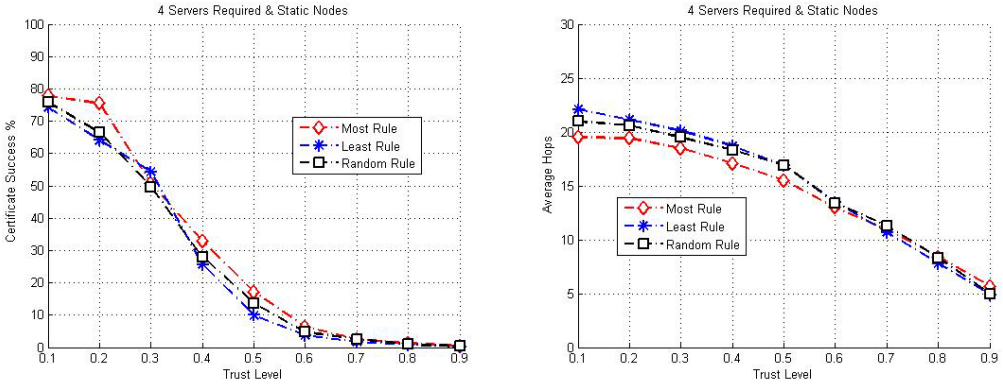
1 Server Required: Success / Hops.



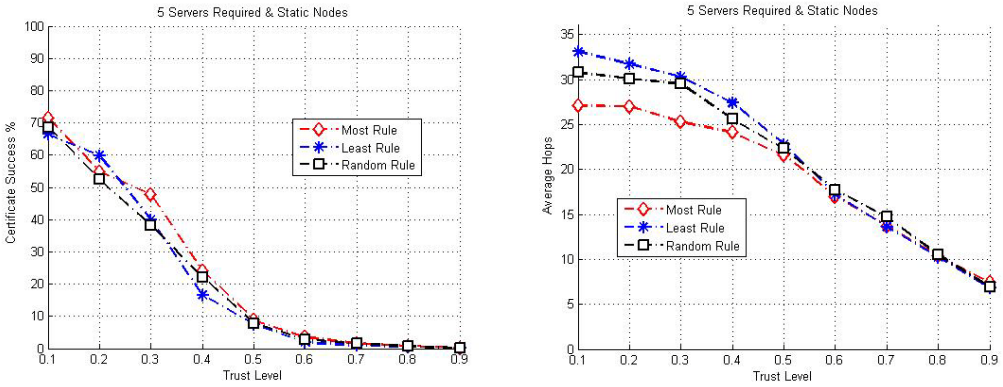
2 Servers Required: Success / Hops.



3 Servers Required: Success / Hops.



4 Servers Required: Success / Hops.



5 Servers Required: Success / Hops.