2007

Two-tier Intrusion Detection System for Mobile Ad Hoc Networks

RAZAK, SHUKOR ABD

http://hdl.handle.net/10026.1/2162

http://dx.doi.org/10.24382/3347
University of Plymouth

Two-tier Intrusion Detection System for Mobile Ad Hoc Networks

by

SHUKOR ABD RAZAK

A thesis submitted to the University of Plymouth in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

School of Computing, Communications & Electronics

Faculty of Technology

February 2007

Abstract

A Two-tier Intrusion Detection System for Mobile Ad Hoc Networks

Shukor Abd Razak

Nowadays, a commonly used wireless network (i.e. Wi-Fi) operates with the aid of a fixed infrastructure (i.e. an access point) to facilitate communication between nodes when they roam from one location to another. The need for such a fixed supporting infrastructure limits the adaptability of the wireless network, especially in situations where the deployment of such an infrastructure is impractical. In addition, Wi-Fi limits nodes' communication as it only provides facility for mobile nodes to send and receive information, but not reroute the information across the network. Recent advancements in computer network introduced a new wireless network, known as a Mobile Ad Hoc Network (MANET), to overcome these limitations.

MANET has a set of unique characteristics that make it different from other kind of wireless networks. Often referred as a peer to peer network, such a network does not have any fixed topology, thus nodes are free to roam anywhere, and could join or leave the network anytime they desire. Its ability to be setup without the need of any infrastructure is very useful, especially in geographically constrained environments such as in a military battlefield or a disaster relief operation. In addition, through its multi hop routing facility, each node could function as a router, thus communication between nodes could be made available without the need of a supporting fixed router or an access point. However, these handy facilities come with big challenges, especially in dealing with the security issues. This research aims to address MANET security issues by proposing a novel intrusion detection system that could be used to complement existing prevention mechanisms that have been proposed to secure such a network.

A comprehensive analysis of attacks and the existing security measures proved that there is a need for an Intrusion Detection System (IDS) to protect MANETs against security threats. The analysis also suggested that the existing IDS proposed for MANET are not immune against a colluding blackmail attack due to the nature of such a network that comprises autonomous and anonymous nodes. The IDS architecture as proposed in this study utilises trust relationships between nodes to overcome this nodes' anonymity issue. Through a friendship mechanism, the problems of false accusations and false alarms caused by blackmail attackers in global detection and response mechanisms could be eliminated.

The applicability of the friendship concept as well as other proposed mechanisms to solve MANET IDS related issues have been validated through a set of simulation experiments. Several MANET settings, which differ from each other based on the network's density level, the number of initial trusted friends owned by each node, and the duration of the simulation times, have been used to study the effects of such factors towards the overall performance of the proposed IDS framework. The results obtained from the experiments proved that the proposed concepts are capable to at least minimise if not fully eliminate the problem currently faced in MANET IDS.

Contents

List of Figures

List of Tables

List of Formulae

Acknowledgement

This study was made possible due to funding from the Universiti Teknologi Malaysia (UTM), and I would like to thank the organisation for the support.

I would like to express my deep and sincere gratitude to my Director of Studies, Professor Dr. Steven Furnell, Research Director of the Network Research Group, University of Plymouth. His wide knowledge and logical way of thinking have been of great value for me. His understanding, support, and personal guidance have made the completion of this thesis possible.

I also wish to express my warm and sincere thanks to my supervisors, Dr. Nathan Clarke, and Dr. Phillip Brooke for their constructive comments and excellent advices that have been very useful to this study. Without their kind support and guidance, it would be impossible for me to finish this research.

During this work, I have received a lot of support from many colleagues for whom I have great regard, and I would like to extend my warmest thanks to those who have helped me with my work in the Network Research Group at the University of Plymouth. Thanks must also go to all my friends in Plymouth as well as in Malaysia for their endless support.

Lastly, and most importantly, I wish to thank my family for their support. In particular, I would like to thank my mother. Without her love, I would never been able to finish this work.

Glossary

AODV	Ad-hoc on-demand Distance Vector
CA	Central Authority
CCS	Credit Clearance System
CD	Compact Discl
CDMA	Code Division Multiple Access
CDPD	Cellular Digital Packet Data
DARPA	Defense Advanced Research Projects Agency
DoS	Denial of Service
DSDV	Destination-Sequenced Distance Vector
DSR	Dynamic Source Routing
FSM	Finite State Machine
GloMoSim	Global Mobile Information Systems Simulation Library
GPRS	General Packet Radio Services
GPS	Global Positioning System
GSM	Global System for Mobile communication
IDS	Intrusion Detection System
IrDA	Infrared Data Association
IREP	Incremental Reduced Error Pruning
MAC	Media Access Control
MANET	Mobile Ad Hoc Network
MIB	Management Information Base
NIST	National Institute of Standards and Technology

NS	Network Simulator
OPNET	Optimized Network Evaluation Tool
PDA	Personal Digital Assistant
QoS	Quality of Service
RERR	Route Error
RFID	Radio Frequency Identification
RREP	Route Reply
RREQ	Route Request
SNMP	Simple Network Management Protocol
SVM	Support Vector Machine
TCP	Transmission Control Protocol
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WWAN	Wireless Wide Area Network

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award. This study was financed by the Universiti Teknologi Malaysia (UTM) with support from the Network Research Group at the University of Plymouth. Relevant seminars and conferences were regularly attended at which work was often presented. Several papers were published in the course of this research project. The work presented in this thesis is solely that of the author.

Signed ....

Date 4/5/2007

Chapter 1

Introduction and Overview

1.1 Introduction

Recent progress and advances in the communication technologies have introduced a new computer network technology, the Mobile Ad Hoc Network (MANET). Such a network comprises nodes that are anonymous, autonomous, and freely distributed across the network without any fixed wire and topology, and thus offers feasibility to create an instant network in various ad hoc environments such as in a meeting room, military battlefield, and disaster relief operation. This emerging technology was first introduced in US military research and has become increasingly popular in commercial applications. In addition, with the potentials that it could offer, MANET also might be used as the main tool for communication, supporting the wired network in the near future. This advancement has without doubt introduced new security concerns and for that reason, new security mechanisms have to be investigated.

This thesis analyses current security measures proposed for MANET environments. Results from the analysis show that a number of research projects have been undertaken to provide security mechanisms for MANET but most of them were focusing on prevention measures, leaving the detection measures nearly unexplored. Prevention mechanisms can be very useful as a first defensive wall to protect MANET from adversaries. However, no matter how strong the prevention measures are embedded in a network, security breaches can always exist that an adversary can exploit (King, 2002).

This situation led the direction of this study, which is to give more attention on the detection measures, and to be specific, an Intrusion Detection System (IDS). Employing an

IDS as a second line of defence could be very useful whenever prevention mechanisms failed to protect the network. Various IDS have been proposed to defend wired network technology (CERIAS, 2006). However, current IDS employed in wired network cannot be simply migrated into MANET environment. The reason is mainly due to the unique characteristics of a MANET, which not only add difficulties to detect similar security threats inherited from a wired network but also require new techniques to detect novel security threats accentuated by the MANET itself. Details on how such attacks could threaten MANET operations also have been extensively reviewed in this study.

When this study was started in late 2003, only a few works have been done to secure MANET operations using an IDS. The first effort came from Zhang et al. (2003), which suggested that an IDS for MANET environments need to be specially designed so that it could suit well with such network's characteristics. After that, a few other techniques have been proposed by other researchers to improve their initial work. However, the numbers are still small and none of them has shown how their techniques could resist attacks from colluding blackmail attackers.

A blackmail attack occurs when an attacker sends false accusations about other nodes' integrity in the networks. Such attacks could exist in many forms and usually targeting against systems or networks operations that demand nodes collaborative participation. In a case of MANET IDS, a blackmail attack is capable to bring down the whole global detection and response mechanisms. The main objective of such attack is to make the victim nodes look bad in the eyes of other nodes. As a result, other nodes might refuse to cooperate with the victim nodes, and thus deny their participation in networks operations. A blackmail attack might not be seen as a big issue in a wired network as there is a Central

Authority (CA) to monitor such misbehaviour. However, the scenario is different in a MANET environment. The absence of a CA to control all network operations in a MANET environment makes the blackmail attack more difficult to detect. The situation will become worse if the attackers decided to collude with each other and launch more severe attack, namely a colluding blackmail attack.

This thesis proposes a novel IDS framework aiming to improve the global detection and response mechanisms of MANET IDS in the presence of colluding blackmail attackers. The proposed framework exploits the concept of friendship between nodes as a self-defensive measure against such attack. The friendship concept proposed in this study is capable of eliminating the problem of node's anonymity, which is the main reason why a blackmail attack could exist in MANET environments (Liu et al., 2004).

1.2 Aims and Objectives of the Research

The aim of this research is to propose a novel IDS framework that is suitable for MANET environments and capable to protect such a network against security threats. In order to achieve this aim, several issues need to be thoroughly investigated and analysed, as follows:

1. to investigate the nature of MANET operation and the characteristics that create more challenges to fulfil its security requirements;

2. to identify security threats in MANET environments that demand new security measures to be deployed;

3. to analyse and highlight the deficiencies of existing security measures that have been proposed for MANET environments, which should support the idea that there is a need for an IDS;

4. to propose and design a novel IDS framework based on the analysis of capabilities and limitations of existing IDS previously designed for MANET environments;

5. to evaluate the performance of the proposed IDS framework;

1.3 Thesis Structure

The outcomes of research which addressed the aforementioned objectives are presented in this thesis as follows.

Chapter 2 begins with an overview of MANET, which describes how such a network is different from the other types of wireless networks. This is followed by a summary of MANET's characteristics and the environments where it is usually deployed so that a better understanding on how such networks operate. Having addressed the background of MANET, the chapter then outlines some of the research challenges that must be faced and summarises security requirements that must be fulfilled to ensure MANET usability and feasibility as a viable future networking technology.

Chapter 3 presents the outcomes of an investigation carried out to identify important security threats in MANET environments. This investigation is essential because in order to design a reliable security measure for MANET, a detailed understanding on how attacks are launched against such a network is required. The chapter begins with a discussion about attack variations, which suggests that attacks against MANET exist in many forms depending upon in which environment the attacks are launched, what communication layer the attacks are targeting, and which type of ad hoc network mechanisms are being targeted. This is followed by a discussion about different characteristics of attackers that exist in MANET environments. In this study, special attention has been given to attacks that could be launched against MANET routing protocols. This is because routing is the most vital mechanism in MANET, and to understand attacks that are threatening its operation is very important. For that reason, this chapter also outlines some examples of attacks related to MANET routing protocols.

Having investigated how security threats could be launched against MANET, Chapter 4 presents some of existing security solutions that have been proposed to ease the impacts of those attacks. The chapter first introduces different types of prevention mechanisms which include authentication, secure routing, and cooperation enforcement mechanisms. This is then followed by a discussion to establish a reason why these prevention mechanisms are insufficient and must be accompanied by suitable detection mechanisms to protect MANET from both internal and external attackers. Several solutions have been proposed in response for the need of IDS that suit well in MANET environments. Most of the efforts have generally attempted to address one of the following vital issues in the intrusion detection and response mechanisms: how to collect the audit data; what is the appropriate method to detect an intrusion; how to minimise false alarms; and how to respond to the intrusion. This chapter summarises some of the efforts proposed by researchers in addressing these issues.

Chapter 5 presents a conceptual design of the proposed intrusion detection framework, which has been designed based on a thorough investigation of previous works. The chapter begins with a list of the objectives and criteria that the proposed IDS are intended to address. This is then followed by a discussion that justifies the ideas behind the proposed design. This chapter also presents a brief overview about each component of the proposed detection framework.

The novel element in this study is on the implementation of the friendship concept to assist the global detection and response mechanisms of the proposed IDS framework. Chapter 6 describes this concept in detail along with experiments aimed to justify its applicability in MANET environments. The chapter begins with a discussion that justifies the needs of a trusted community in MANET environments. This is then followed by a description of the trust framework proposed in this study. Following that, the chapter outlines some of the key features of the proposed trust framework that make it different from other trust frameworks suggested earlier for MANET. Finally, the chapter presents results from the experiments carried out to evaluate the capability of the proposed friendship concept in creating a trusted community in MANET environments.

Based on the conceptual framework presented in Chapter 5, Chapter 7 describes simulation designs of the proposed IDS framework aimed to evaluate its performance. The chapter begins with a detailed description on the implementation of misuse and anomaly detection engines. This is then followed by a discussion on the global detection and response mechanisms, which utilise the friendship concept. This chapter also describes the need for trust and signature management mechanisms, which are also part of the main modules of the proposed IDS framework.

Following the design of local and global detection mechanisms as presented in Chapter 7, Chapter 8 presents the results from simulation experiments evaluating the performance of the proposed IDS framework. The chapter begins by presenting the performance of the proposed IDS framework in detecting intrusions via local (using its own detection capability) and global (with the help of friend mechanism) means. This is followed by a comparison of local and global detection performances to show that the friendship concept introduced in the global detection mechanism is capable of increasing the IDS detection rate. The chapter then continues by presenting the performance of the proposed IDS framework in terms of its capability to globally response to intrusions. The chapter also presents the capability of the proposed IDS framework in resisting against a colluding blackmail attack.

Finally, Chapter 9 summarises the research conducted and presented in this thesis. The chapter begins with a discussion on the achievements and limitations of the proposed framework based on theoretical and experimental analysis. This is followed by a discussion to outline the potential of the proposed trust framework in other MANET research areas. The chapter then concludes with suggestion on some possible extensions to the research for future work.

The thesis also includes a number of appendices containing additional information to support the discussion presented in the main chapters. Source code from the experiments described in Chapters 6, 7 and 8, as well as the raw results are included on an accompanying CD. Finally, a number of publications arising from the project are also included in this thesis.

Chapter 2

MANET Overview and Security Issues

2.1 Introduction

Networks are now one of the most popular manifestations of computing technology, and the most significant example is the Internet. Online banking, email applications, online trading, instant messaging, and news broadcasting are only a few examples to illustrate the massive use of the Internet. Recent progress and advances in the communication technologies have introduced another type of computer network, the Mobile Ad Hoc Networks (MANET). MANET represents a combination of peer-to-peer techniques, wireless communications and mobile computing, and has become an important field of research in recent years. This new technology has been widely used to support communications in an environment that might not allow the deployment of infrastructure networks, such as in military battlefields and disaster recovery sites. In addition, this technology might be used to replace infrastructure networks where employing the wireless networks is more practical (Loo, 2004). Although MANET utilises a wireless medium for communications, as in other wireless networks, it has its own unique characteristics that make it different from the others.

This chapter begins with an introduction and comparison of MANET and other wireless networking technologies. MANET characteristics and the environment where such networks usually deployed will also presented in this chapter. This chapter also outlines research challenges that must be faced and security requirements that must be fulfilled to ensure MANET usability and feasibility as a future technology.

2.2 MANET and the Wireless Network Technologies

Wireless networks assist in the communication activities between two nodes to provide more flexible and easier connectivity. According to the NIST (Karygiannis & Owens, 2002), wireless networks can be categorised into three main categories: Wireless Wide Area Network (WWAN); Wireless Local Area Network (WLAN); and Wireless Personal Area Network (WPAN).

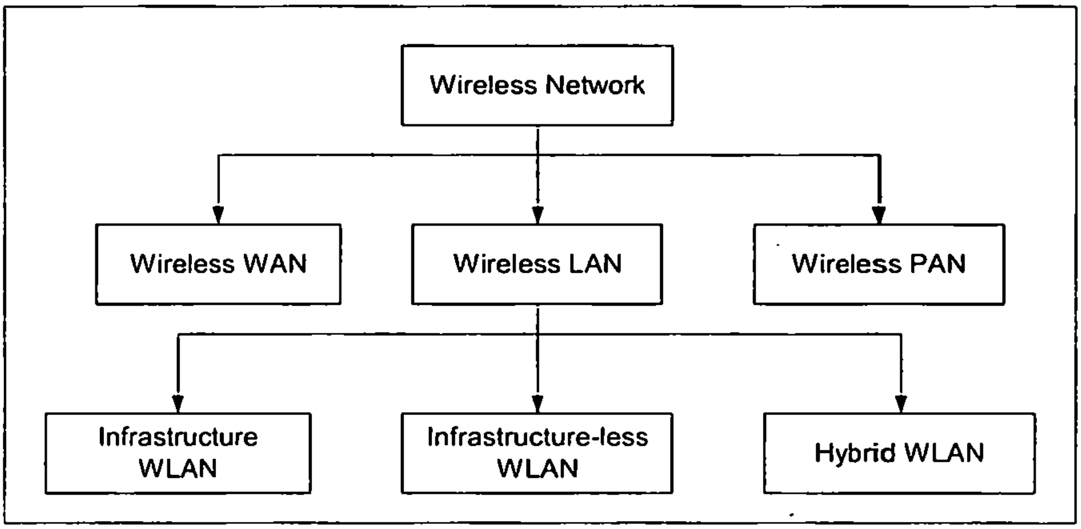


Figure 2- 1: Types of Wireless Network

Figure 2-1 illustrates types of wireless networks and the descriptions for each technology are as follows.

2.2.1 Wireless WAN (WWAN)

WWAN is a computer network using wireless networking devices to transfer data in a wide coverage area. Such technology is generally managed by a service provider and usually offers services to a large number of users. Examples of WWAN technology are CDMA, GSM, GPRS, CDPD and satellite networks (Chaplin, 2002). A simple implementation of WWAN is as illustrated in Figure 2-2.

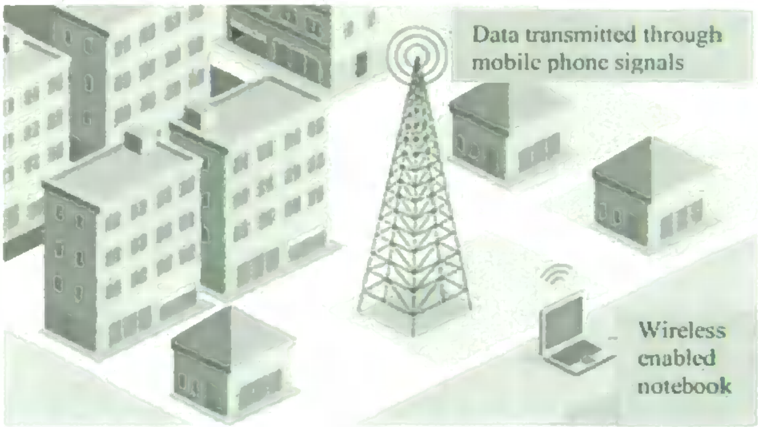

Data transmitted through mobile phone signals
Wireless enabled notebook

Figure 2- 2: An example of WWAN implementation (Dell US, 2006a)

2.2.2 Wireless PAN (WPAN)

WPAN is a collection of personal devices connecting to each other in a limited coverage area as illustrated in Figure 2-3. Technologies related to WPAN are IrDA, RFID and Bluetooth. Unlike in a WWAN, network connectivity in WPAN is completely controlled by the user who operates it, not by the service administrator. Another significant difference compared to a WWAN is that there is no charge for connection, as it uses free unlicensed frequency band. More explanation about licensed and unlicensed frequencies can be found in (CMP Media LLC, 2003).

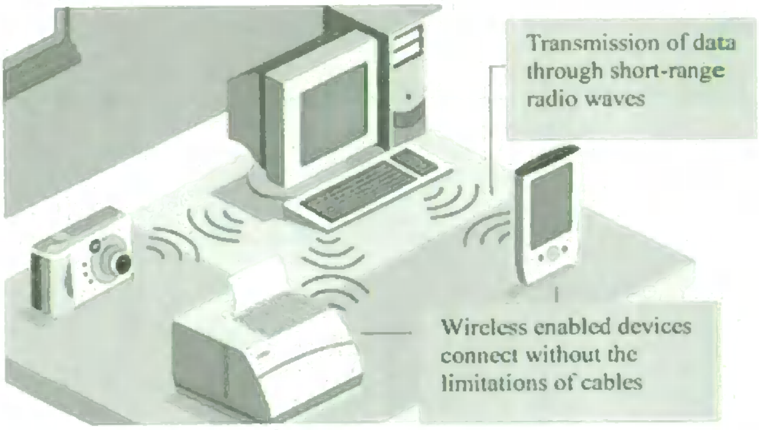

Transmission of data through short-range radio waves
Wireless enabled devices connect without the limitations of cables

Figure 2- 3: An Example of WPAN usage (Dell US, 2006b)

2.2.3 Wireless LAN (WLAN)

WLAN is a computer network designed to allow greater flexibility and mobility in a local area network connection. Similar to WPAN, this technology also uses an unlicensed frequency band to establish wireless connection. Since no service provider exists in the networks, the users must take responsibility to control and manage all the network operations by themselves. In general, WLAN can be divided into two main categories, namely infrastructure and infrastructure-less networks as described in (Al-Jaroodi, 2002). However, recent research in (CISE Wireless Project, 2002) has introduced the third type of WLAN technologies; the hybrid WLAN. The descriptions of each WLAN technology are as follow:

- *Infrastructure Wireless Network*

This kind of computer network consists of several mobile devices connected directly to an access point using wireless transmissions. An access point is a station that transmits and receives data from users within the network and can serve as the point of interconnection between the WLAN and a fixed wire network.

- *Infrastructure-less Wireless Network*

This type of WLAN which is also known as a MANET, is a network comprised only of mobile wireless devices. Nodes communicate directly with each other without the aid of any access points or wired backbone.

- *Hybrid Networks*

Hybrid networks can be used to ease the deployment of an infrastructure wireless network. The main problem in infrastructure wireless networks is the constraints in placing the access points. By exploiting the multi-hop capabilities in MANET, all nodes (including those in the outer range) are able to reach the access point to connect to the Internet.

Figure 2-4 illustrates an example of WLAN technologies consisting of infrastructure WLAN, infrastructure-less WLAN, and hybrid WLAN. White arrows show that mobile devices in MANET can connect to each other without an access point. On the other hand, each mobile node in an infrastructure WLAN needs to connect directly to the access point to establish connections amongst them. Black arrows show direct connections between nodes and access point. Nodes in the circle area connected to each other to establish connection between MANET and infrastructure WLAN. Wireless networks created in this way are also known as hybrid wireless networks.

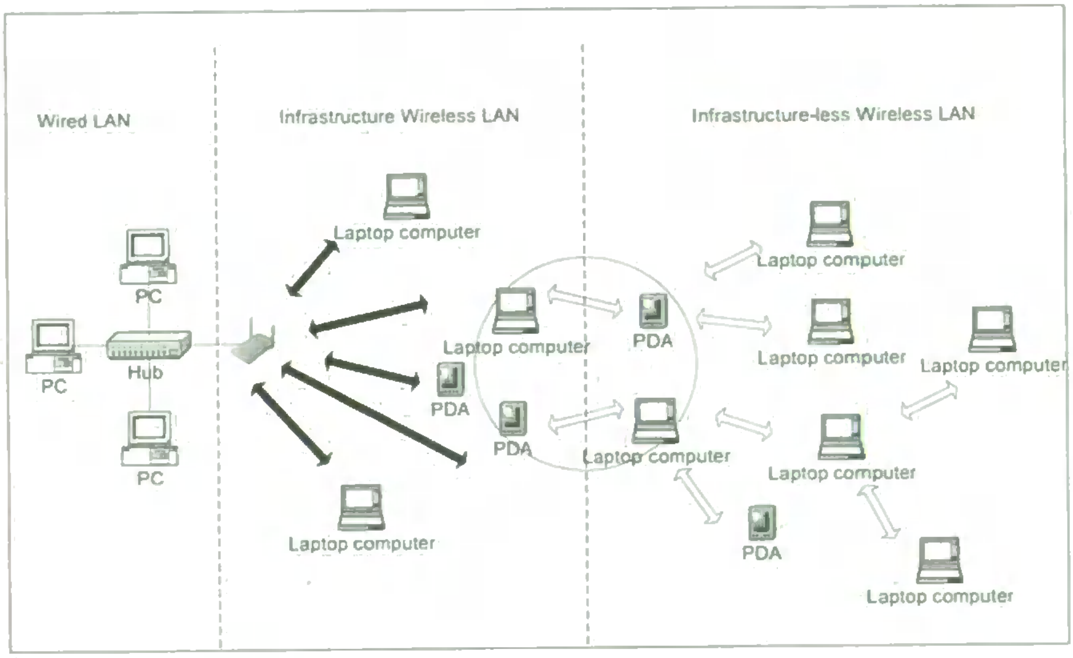


Figure 2- 4: Wireless LAN Technologies

2.3 MANET Characteristics

In general, MANET has a number of defining characteristics (Al-Jaroodi, 2002; Rafique, 2002) as described in the sub-sections that follow.

2.3.1 Dynamic Network Topology

MANET is highly dynamic in nature. Nodes in MANET are mobile and connected to each other via wireless links. Wireless connectivity allows nodes to join the network and dynamically associate to establish routing among themselves. The associations are often created and torn down without prior notice and thus make the ad hoc network topologies unpredictable. The topologies become more complex when nodes in MANET established a connection to any public infrastructure network.

2.3.2 Distributed Operations

Operations in MANET are performed in a distributed manner. Successful routing in MANET, for example, needs participation among nodes to collaborate in the route discovery process. In addition, since there is no central control for the networks, all the management processes in MANET must be carried out in a distributed manner.

2.3.3 Infrastructure-less

In a MANET, fixed infrastructure and specialised hardware that help in communication operations are necessarily absent. In addition, nodes participating in the network have not been given any specific roles such as servers, routers or gateways. These circumstances prevent the deployment of hierarchical node relationships and thus make security mechanisms that depend upon these relationships inappropriate.

2.3.4 Limited Resources

Generally, most ad hoc network enabled devices are small in size ranging from notebooks to PDAs including cellular phones and can be assumed to rely on batteries for their power supplies. Complex and frequent computational tasks must be avoided, as these operations will drain power quickly. Network bandwidth is another important resource. Usually MANET has a lower bandwidth capacity than a fixed network, and for that reason traffic used for connection and maintenance must be kept to a minimal. In addition, MANET also has limited CPU processing and limited data storage capabilities (Jung et al., 2005a).

2.3.5 Multi Hop Routing

MANET connectivity can be single hop based or multi-hop based depending on the distance between source and destination nodes (Brannstrom et al., 2006). Communications among nodes in MANET are generally within a short range. Nodes communicate directly using a single hop routing algorithm if they are close to each other. However, because of the geographical constraint and distance between source and destination nodes, data will usually traverse through the network via one or more intermediate nodes before it reaches the destination. In this situation, connectivity between sender and receiver is no longer in a single hop mode but now in a multi-hop mode. Figure 2-5 illustrates the difference between these two communication scenarios in MANET environments.

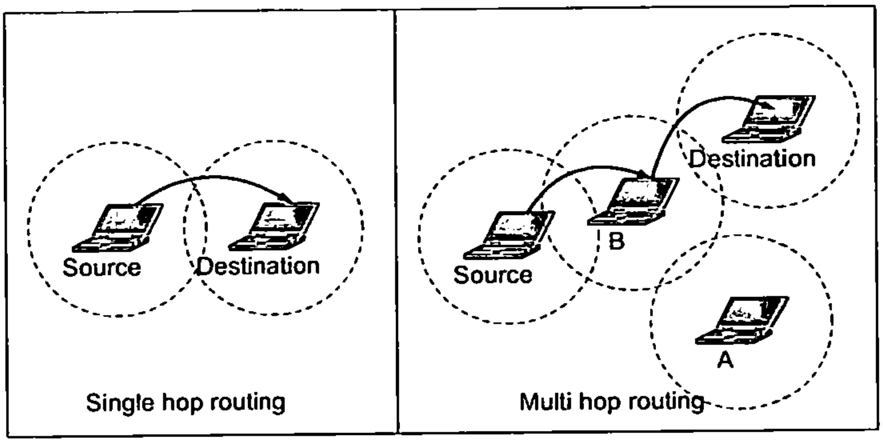


Figure 2- 5: Single and Multi Hop Routing in MANET Environments

2.3.6 Shared Transmission Media

The transmission medium used in MANET is not as stable as that used in a fixed network. Communication in MANET is subjected to noise, interference and even constraint to bandwidth limitation (Desilva & Boppana, 2004). Moreover, security requirements are usually higher in MANET than in the wired network because wireless links, which used for communication, are subjected to external attacks such as spoofing, eavesdropping and link jamming attacks.

2.4 MANET Environment

Various papers have given an example where mobile ad hoc network could be deployed. Meetings, conference, emergency, disaster relief, sensor network, wearable computing, and military applications are among the examples given in (CISE Wireless Project, 2002; Buttyan & Hubaux, 2003; Albers et al., 2002) to show the various environments where a MANET could exist. However, despite the diversity, MANET can be classified into three main categories: organised; localised; and open environments (Al-Jaroodi, 2002).

2.4.1 Organised Environment

Networks deployed for military in battlefield and disaster relief operation, as illustrated in Figure 2-6, are some examples of well-organised MANET. Normally, a well-organised MANET is needed to support a very well-defined purpose operation or application. Apart from the routing difficulties in this geographically constrained environment, enabling security measures in this challenging environment is also not an easy task. One of the security challenges in the organised MANET is on how to avoid a node's location from being disclosed to other unauthorised parties (e.g. in the military scenarios) (Khan, 2003).

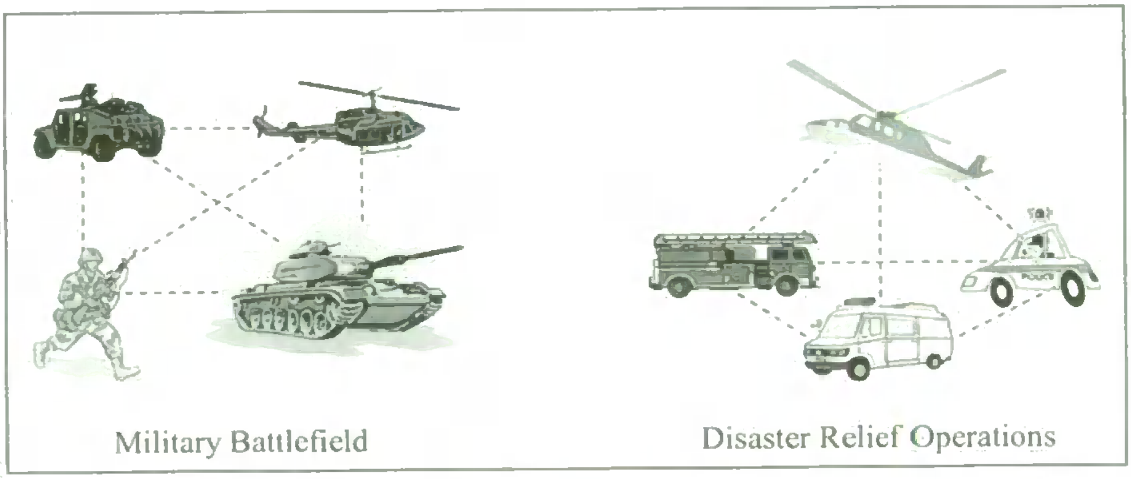


Figure 2- 6: Examples of Organised MANET Environment

2.4.2 Localised Environment

This type of network is the most common type of MANET. Ad hoc networks configured in conference rooms, lecture halls, and home appliances are examples of a localised network. All nodes in this network are in the proximity range thus can be physically contacted to establish any security measures. Unlike in organised MANET, where nodes are more vulnerable to external attacks, nodes communicate in localised MANET on the other hand are more vulnerable to internal attacks. This is because every node that participates in the localised network has the privilege to access network resources and secret information, which can be exploited once it has been hijacked by the external attackers. Examples of internal attacks in this scenario are password stealing and unauthorised access to secret documents. Figure 2-7 illustrates several MANET enabled devices connected to each other to create localised MANET environment.

Figure 2- 7: An Example of a Localised MANET Environment

2.4.3 Open Environment

Unstructured, vast nodes and the absence of *a priori* relations are some of the main characteristics of this network. The concept of this network is similar to a people in a city communicating with each other using personal wireless devices. This situation is quite similar to MANET in a localised environment, but the number of nodes participating and the coverage area in the open environment is bigger and wider than in the localised environment. The wider coverage area makes physical contacts for all the nodes inapplicable and thus results in the difficulty to deploy any security measures in an open MANET environment.

2.5 Research Challenges

All of the MANET characteristics discussed in the previous section have introduced many research issues in these networks. Some of them are quite similar to the issues faced in other networks, but others are more specific to the MANET environment. Figure 2-8 illustrates the main important issues related to MANET (Chlamtac et al., 2003).

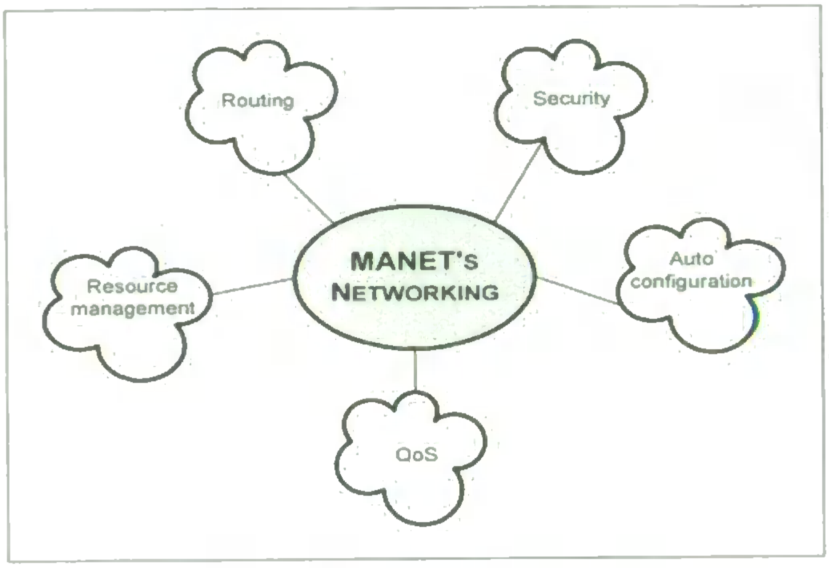


Figure 2- 8: Main Research Issues in MANET Networking

2.5.1 Routing

Providing a robust and reliable routing mechanism is very important in MANET. Proactive and reactive protocols are two approaches considered in uni-cast routing algorithms (Clausen, 2003). Both of these protocols have their own advantages and weaknesses, but the most important thing is that both rely upon the cooperation of all nodes in the networks. Other concerns include multicast routing algorithms and broadcast routing algorithms. Because of the random movement of nodes in MANET, providing efficient multicast and broadcast routing algorithms becomes more complex and challenging than in a wired network. Various protocols have been proposed to solve all the routing complexities and challenges (Perkins et al., 2003; Paul & Westhoff, 2002). However, despite much effort in this area, none of the proposed approaches became a standard protocol for all MANET configurations (Albers et al., 2002).

2.5.2 Auto-Configuration

MANET operates in a self-organised manner. Each node in the network is responsible to configure itself including all the services and applications required such as routing protocols, security mechanisms and IP address allocations. However, this configuration is often too complex to be done by end users. Providing auto-configuration mechanisms would be very useful and may help in attracting more people to use MANET (Clausen, 2006).

2.5.3 Resource Management

Resource management is crucial in MANET. Battery power, bandwidth, CPU processing capability and storage capacity are the most important resources, and thus proper management of them is required. All of these resources are limited because of the devices physical constraints. For that reason, communication algorithms as well as services offered in a MANET must be optimised to meet the minimum level of bandwidth usage, CPU processing, power utilisation and data storage of the ad hoc network enabled devices (Rafique, 2002).

2.5.4 Quality of Service (QoS)

QoS is another challenging issue in MANET (Chlamtac et al., 2003). QoS for any network is always related to the characteristics of that particular network. Wireless links used for communication in MANET have a fluctuating link capacity and connectivity, thus making it more difficult to guarantee the QoS in the network (Jain et al., 2005). In addition, there are many other unique characteristics in MANET as described earlier, which add the difficulties in providing reliable QoS in such network.

2.5.5 Security

Providing a robust and reliable security mechanism in MANET is not an easy task because of the unique characteristics described earlier. Although many security mechanisms (e.g. public key cryptography and firewalls) are found to work well in wired networks, such mechanisms are impractical in MANET because of the infrastructure constraint (Haas et al., 2002). In MANET, all nodes are expected to operate in a self-organised manner, thus the existence of a central authority to manage the public key infrastructure cannot be assumed. In addition, the nature of instability in the network connections and unpredictable node movements add to the difficulty in differentiating between malicious activities and 'natural' network problems.

2.6 MANET Security Requirements

Open MANET, as well as other MANET environments, has a number of requirements that need to be fulfilled in order to ensure network security and reliability. An overview of the security requirements in MANET are described in the following subsections.

2.6.1 Authentication

In MANET, nodes can communicate with each other as long as they are in the communication range, and at one moment in time, there is a possibility to have more than two nodes in the communication range. For that reason, when two nodes communicate to each other there is a need to verify the identities of each other. Authentication can be used to provide evidence of own identity and at the same time enable the node to ensure the identity of the other nodes it is communicating with. Additionally, authentication can also be used to ensure legitimate access to the network (Hafslund & Anderson, 2006). Identity verification ensures that an adversary cannot masquerade as a trusted entity thus gaining unauthorised access to the network. Authentication can be considered as the most important service, because without proper authentication, security services such as integrity, confidentiality and non-repudiation cannot be maintained. The relationship between integrity, confidentiality, non-repudiation, and authentication services has been discussed in (Maki, 2000).

2.6.2 Confidentiality

Confidentiality in network communications is always related to privacy. Both services ensure that certain information is never disclosed to unauthorised users but in a different context. Confidentiality ensures the secrecy of data in the network, whilst privacy concerns protecting the identity of nodes and sometimes hiding the location of nodes for security reasons. Privacy is a crucial concept in a military environment because an enemy can use unprotected information to launch attacks on particular nodes. Data to be secured by the confidentiality service can be divided into two categories, namely data forwarded within the network and data stored in the ad hoc network enabled devices. Researchers are always concerned about the secrecy of data forwarded through the network, but unaware about the security of data stored in the devices memory (Oltsik & Biggar, 2006). Routing information also must be kept confidential because exposure of such data can lead to many routing mechanism attacks. Confidentiality can be achieved by using encryption techniques once secure authentication systems have been deployed (Stajano & Anderson, 1999).

2.6.3 Integrity

When transferring data over the network, people want to make sure that the data received by the receiver is the same as what they have transferred earlier. Integrity is the security service, which can be used to guarantee that a message being transferred is not altered or modified by a malicious node. One must take into account that the integrity of message transferred can also be affected by benign failure such as radio propagation impairments. Ensuring integrity can be as simple as adding hashing functions to the encrypted message once the strong authentication mechanism has been deployed. Integrity is not only used to ensure that the data exchanged between nodes has not been altered, but also can be used to ensure the integrity of the network connection. In this situation, integrity can be used to guarantee no messages are removed, replayed or unlawfully inserted during the transmission (Rafique, 2002).

2.6.4 Non-Repudiation

Non-repudiation is a security service to ensure that a sender cannot later deny having sent a message. This service can be very important especially in the existence of an adversary. One way to enable this service is by using a public key cryptography system where a receiver can use the sender public key to prove its identity. In addition, the receiver also cannot deny the reception of the sent message since such message can only be decrypted using the receiver's private key. By employing this security service, each node in the network is enforced to be responsible for its actions. In addition, this service also can be very useful for detection and isolation of malicious nodes. Again, this service also seems to rely upon the authentication service to maintain the service. Once the authentication service takes place, a user can employ suitable cryptography techniques to enable non-repudiation service. However, if no authentication schemes exist or weak authentication schemes being used, the non-repudiation service cannot be guaranteed to work in a proper manner (Al-Jaroodi, 2002).

2.6.5 Availability

If confidentiality is related to privacy, availability on the other hand is related to survivability. In a communication, high availability of network resources and services are always desirable. Availability means that any nodes can get access to all the services provided by the network even in the presence of an adversary. Survivability seems to be always supporting the availability service in the network (Chen et al., 2002). Attacks from internal or external nodes are more difficult to detect in an MANET environment. Additionally, malfunction of network services can also occur in this unpredictable network topology. Because of that, survivability is needed to effectively return the network services to their normal conditions after such malfunction or attacks against the networks. Accompanied by the strong survivability mechanisms, high availability in network services can be achieved. Denial of service is an example of attacks that can be launched by attackers to bring down the availability service. Consequences of such attacks can lead to physical channel interference, disruption of the routing protocol and even total disconnection of the entire network. In addition, one must also consider the existence of selfish nodes in the networks. Such selfish nodes add the difficulties in ensuring high availability and survivability of the ad hoc network (Buttyan & Hubaux, 2003).

2.6.6 Authorisation

Authorisation is a security service used to set up rules that define what operations or actions each node is or is not allowed to do in the network. These predefined rules can also be used to determine which resources or information across the network each node can access. Nodes in the network need to be authorised to access the shared resources in the network, especially in MANET. Most of the mobile devices carry private and personal information that need to be kept secure when communicating with other nodes. Since each node in a MANET is mobile and communicates with various types of users, different types of restriction are needed for different types of users. For that reason, an authorisation service is required and needs to be maintained. In addition to that, as mentioned earlier, resource management is among the key issues being researched in an ad hoc network. In this case, authorisation can be used to facilitate proper utilisation of network resources to ensure only legitimate users can use such resources based upon the restriction policies that are set up (Al-Jaroodi, 2002).

2.7 Conclusion

Mobile ad hoc network (MANET) is one of the current emerging technologies in the computer industry. Since it is part of the wireless network family, it inherits all of the advantages offered by wireless communications. In addition, with characteristics such as, easy set up, infrastructure-independent, scalable, and dynamic topology, MANET has become a preferred network to replace an infrastructure network especially in a geographically constrained environment such as a military battlefield or a disaster relief operation. However, introducing this network to the public is not as easy as its sounds. Several issues need to be considered before the deployment of such network especially issues related to the routing and security mechanisms. In the next chapter, a review of several common attacks against MANET will be presented. The review includes the variation of attacks against MANET, their characteristics, and some examples of common attacks that threatening such network.

Chapter 3

Attacks against MANET Routing Protocols

3.1 Introduction

Similar to other networks, MANET is vulnerable to many security attacks. MANET not only inherits all the security threats faced in both wired and wireless networks, but it also introduces security attacks that are unique to itself (Karygiannis & Owens, 2002). As people will be encouraged to use a secured network, it is important to provide MANET with reliable security mechanisms that will make this exciting technology become more widely used in the next few years. Prior to the development of any measures to secure MANET, it is important to study the variety of attacks that might be related to such a network. With the knowledge of some common attack issues, researchers will have a better understanding of how MANET could be threatened and compromised by the attackers, and thus should lead to the development of more reliable security measures.

This chapter presents important issues that related to attacks against MANET. In the first place, variations of attacks depending upon in which environment they are launched, what communication layer they are targeting, and what type of ad hoc network mechanisms are targeted will be detailed. Following that, some of the important characteristics of those attacks also will be presented. In this research, special attention has been given to attacks that could be launched against MANET routing mechanism. This is because the failure to defend such an important mechanism could jeopardise the whole network operations. For that reason, this chapter also outlines some examples of attacks falling into this category.

3.2 Attack Variations

MANET is threatened by a variety of attacks, which require different protection strategies. This variation is as a result of several factors as detailed in the sub-sections that follow.

3.2.1 Ad Hoc Networks Environments

Ad hoc network can exist in one of three environments; organised, localised, and open environments. Nodes in all of these environments are generally threatened by the same security problems. However, there are some security problems that are unique to one environment and need more attention in that environment than the others need. Vast numbers of unstructured nodes and the absence of *a priori* relations are some of the main characteristics of the open environment ad hoc networks. Such networks are quite similar to the localised environment networks, but the larger amount of nodes, and the wider coverage area, renders nodes in the open environment vulnerable to more sophisticated security attacks than the localised networks. For instance, nodes in both open and localised environments suffer from the absence of a central authority. However, this is not a big issue in a localised environment, because nodes in that environment might have physical contact with each other to employ any security measures. Security could also be easily enforced in the organised environment because nodes in that environment are usually pre-employed with appropriate security measures before they participate in any specific tasks such as in a military operation.

3.2.2 Communication Layers

Each layer in the ad hoc networks communication protocols has its own vulnerabilities. In a physical layer, mobile nodes as well as the communication links are vulnerable to both passive and active attacks. Passive eavesdropping, signal jamming, denial of service (DoS) attacks, and physical hardware tampering are among the most popular attacks in this layer (Maki, 2000). Such attacks could be made less harmful by encrypting the communication signal, employing spread-spectrum communication technology, and using tamper-resistant hardware.

Similar link jamming and DoS attacks are also threatening MANET at the data link layer. At this layer, adversaries might jam the communication links by sending huge amounts of data to the network, or by replaying unnecessary packets to exhaust the networks' resources. Complex cryptography algorithms and more sophisticated security measures could be very useful at this layer to protect the networks and to distinguish between valid and invalid packets transiting in the networks (Al-Jaroodi, 2002).

The network layer provides the most critical service in the ad hoc network, which is the routing protocol. Several routing protocols have been introduced to provide reliable communication among nodes, but less attention to the security aspects when designing such protocols has opened many security holes at this layer (Sanzgiri et al., 2002). In addition, because of its important function, the network layer has always been a target from external as well as internal attackers. An external attacker could launch many attacks against the routing mechanism such as creating a DoS attack or compromising an internal node to capture confidential information of the network. Misbehaving internal nodes on the other hand could jeopardise the network operations when they refuse to cooperate in the packet forwarding process to conserve their limited resources.

Attackers are also threatening MANET at both the transport and the application layers. At the transport layer, messages are exchanged on the end-to-end basis using secured routes established in the network layer. For that reason, ensuring security at the network layer is very important to provide reliable communication at the transport layer. Similar to the other types of networks, attackers can always find a loophole in the ad hoc networks' applications and might use this vulnerability to launch attacks at the application layer (Al-Jaroodi, 2002). However, since similar attacks also occur in the other types of networks, regular solutions used in wired networks could be reused to defend the ad hoc networks against attacks at this layer.

3.2.3 Attack Level

There are two main targets of attacks in MANET; attacks targeting at the basic mechanisms, and attacks targeting at the security mechanisms (Hubaux et al., 2001). MANET has its own unique basic mechanisms, such as the use of wireless links for communications, employing its own routing strategies, and operating in a distributed manner. Attackers might launch many attacks against these basic mechanisms. For instance, attackers could launch passive eavesdropping attacks against the wireless links, drain off a node's limited resources, and launch active attacks to interrupt the routing mechanisms.

Responding to the many attacks against the ad hoc networks basic mechanisms, researchers have introduced a number of security measures. However, the majority of these security measures are also vulnerable to attacks and need to be secured. Examples of attacks against MANET security mechanisms are stealing usernames and passwords to obtain unauthorised access to the networks, and modifying public key databases to disrupt authentication, confidentiality, and integrity services (Hubaux et al., 2001).

3.3 Attack Characteristics

Dynamic topology, distributed operation, and resource constraints are some of the unique characteristics that exist in the ad hoc networks, which inevitably increase the vulnerability of such networks. Many characteristics might be used to classify attacks against MANET. Examples would include looking at the behaviour of the attacks (passive vs. active), the source of the attacks (external vs. internal), the processing capability of the attackers (mobile vs. wired), and the number of the attackers (single vs. multiple). These are considered in the sub-sections that follow.

3.3.1 Passive vs. Active Attacks

Passive attacks are launched to steal valuable information in the targeted networks. Examples of passive attacks in MANET are eavesdropping and traffic analysis. Detecting this kind of attack is difficult because neither the system resources nor the critical network functions are physically affected making it problematic to detect (Bouam & Othman, 2003).

Whilst passive attacks do not intend to disrupt the network operation, active attacks on the other hand actively alter the data with the intention to obstruct the operation of the targeted networks. Examples of active attacks comprise of actions such as message modifications, message replays, message fabrications, and denial of service attacks.

3.3.2 External vs. Internal Attacks

External attacks are attacks launched by adversaries who are not initially authorised to participate in the network operations (Yang et al., 2006). These attacks usually aim to cause network congestion, denying access to specific network functions or disrupting the whole network operations. Bogus packet injection, denial of service, and impersonation are some of the attacks that are usually initiated by the external attackers. More severe attacks in MANET might come from internal attacks, which are initiated by authorised nodes in the networks. This type of attack might come from both compromised and misbehaving nodes. Internal nodes are identified as compromised nodes if the external attackers hijacked the authorised internal nodes and are then using them to launch attacks against the other nodes in the network (Chan & Perrig, 2003). Security requirements such as authentication, confidentiality and integrity are severely vulnerable in ad hoc networks with compromised internal nodes, because communication keys used by these nodes might be stolen and passed to the other colluding attackers.

On the other hand, nodes will be classified as misbehaving if they are authorised to access the system resources, but fail to use these resources in a way they should be (Ghazizadeh et al., 2002). Internal nodes might misbehave to save their limited resources, such as the battery power, the processing capabilities, and the communication bandwidth. Attacks that are caused by the misbehaving internal nodes are difficult to detect because to distinguish between normal network failures and misbehaviour activities in MANET is not a simple task (Contos, 2006).

3.3.3 Mobile vs. Wired Attackers

Mobile attackers are attackers that have the same capabilities as the other nodes in the ad hoc networks. Since they have the same resource limitations, their capabilities to harm the network operations are also limited. For instance, with the limited transmitting capabilities and battery power, mobile attackers could only jam the wireless links within their vicinity. They are not capable of launching the network jamming attacks that disrupt the whole network operation.

On the other hand, wired attackers are attackers that are capable of gaining access to external resources such as electricity and extra processing power. Since they have more resources, they could launch more severe attacks in the networks, such as jamming the complete networks or launch a brute force attack to obtain other users' authentication credentials. Existence of the wired attackers in MANET (especially in the open environment networks) is always possible as long as the wired attackers are able to locate themselves in the communication range and have access to the wired infrastructures.

3.3.4 Single vs. Multiple Attackers

Attackers might choose to launch attacks against the ad hoc networks independently or by colluding with the other attackers. Single attackers usually generate a moderate traffic load as long as they are not capable of reaching any wired facilities. Since they also have similar abilities to the other nodes in the networks, their limited resources become their weak points (Schafer, 2002). For instance, complex cryptography algorithms could be used to help in defending the authentication, integrity, and the confidentiality services from a single attacker. As it becomes very expensive for the single attackers to break the encrypted messages, nodes in the networks could share the expensive cryptography workloads with each others by exploiting the distributed operations and the multiple connections they have amongst them. However, if several attackers are colluding to launch attacks, defending the ad hoc networks against them will be much harder.

Colluding attackers could easily shut down any single node in the network and are capable of degrading the effectiveness of the network's distributed operations, including the security mechanisms. Adding to the severity, colluding attackers could be widely distributed or reside at a certain area where they presumed a high communication rate in the network exists. If no suitable security measures are employed, nodes in that targeted area are susceptible to any kind of denial of service (DoS) attacks that could be launched by the colluding attackers (Gosh et al., 2005).

3.4 Attacks against Routing Messages

Routing is one of the most vital mechanisms in MANET. Improper and insecure routing mechanisms will not only degrade the performance of the MANET, but will also render such networks vulnerable to many security attacks. One of the basic elements in the routing mechanism is the routing message, which is used to establish and maintain relationships between nodes in the networks. The importance of the routing message has made it a principal target for attackers to launch attacks against MANET (Sanzgiri et al., 2002; Li et al., 2003).

Attacks against MANET routing messages could be launched in many forms and may include all the attacks characteristics described earlier. In this study, attacks against routing messages are classified based on the approach suggested by Stallings (Stallings, 1999). In such classification, information or messages could be deviated from normal operation flow using modification, interception, interruption or fabrication attacks. In a more severe case, attackers might also use any combination of these attacks to disrupt the normal information flow. Examples of attacks against MANET routing messages are discussed in the sub-sections that follow.

3.4.1 Modification

In a message modification attack, adversaries make changes to the routing messages, and thus endanger the integrity of the packets in the networks. Since nodes in MANET are free to move and self-organise, relationships among nodes at some moments of time might include the malicious nodes. These malicious nodes might exploit these sporadic relationships in the network to participate in the packet forwarding process, and later launch the message modification attacks. Examples of attacks that can be classified under the message modification attacks are packet misrouting, impersonation, and The Sybil attacks (Douceur, 2002).

3.4.1.1 Packet Misrouting Attacks

Routing protocols are always susceptible to the packet misrouting attacks (Srinivasan et al., 2003). In this attack, malicious nodes reroute traffic from their original path to make them reach the wrong destination. Attackers might launch the packet misrouting attacks to achieve several malicious goals. In a general case, attackers might misroute packet to make it stay in the network longer than its lifetime and thus cause it to be dropped from the network. As a result, the source node needs to retransmit the lost packets and this will consume more bandwidth as well as increase the overhead in the networks. Attackers could also misroute several packets from several different paths to flood one targeted victim or to congest a certain area in the network. Additionally, attackers also might reroute the routing packets to another colluding attacker, as described in the wormhole attacks (see section 3.4.2.1). Figure 3-1 illustrates an example of the packet misrouting attacks in MANET.

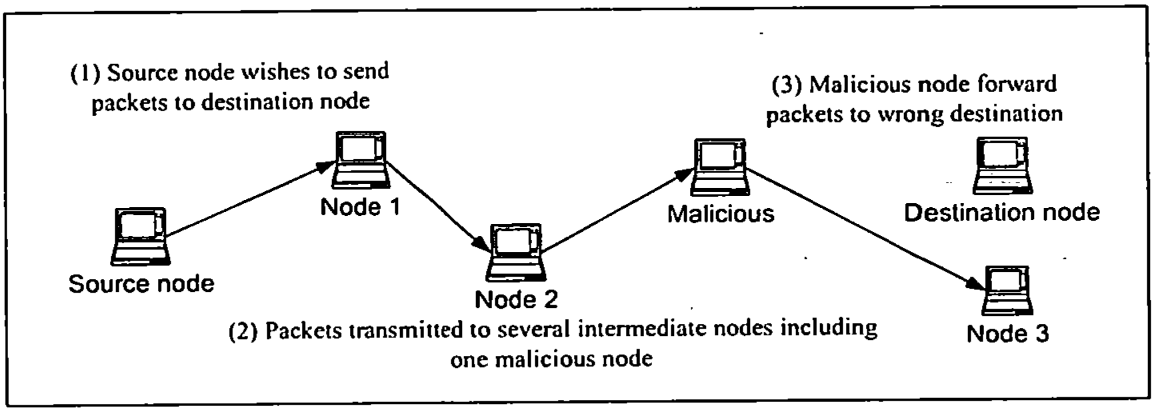

(1) Source node wishes to send packets to destination node
(3) Malicious node forward packets to wrong destination
Node 1
Malicious
Destination node
Source node
Node 2
(2) Packets transmitted to several intermediate nodes including one malicious node
Node 3

Figure 3- 1: Packet Misrouting Attack

3.4.1.2 Impersonation Attacks

Impersonation attacks, also called spoofing attacks, are attacks where the malicious node assumes the identity of another node in the network (Burg, 2003). By impersonating another node, attackers are able to receive routing messages that are directed to the nodes they faked. Impersonation attacks are possible in MANET because most of the current ad hoc routing protocols do not authenticate the routing packets (Choi, 2003). As a result, malicious nodes might exploit this loophole to masquerade as another node by modifying the contents of the packets. Attackers may launch an impersonation attack to achieve various malicious goals. Attackers might impersonate as either a sender or a destination node to intercept secret information in the networks. Attackers also might launch the impersonation attacks against the intermediate nodes to disrupt normal routing operations such as to launch packet dropping, black hole, and packet misrouting attacks. Figure 3-2 illustrates an example of impersonation attacks in MANET.

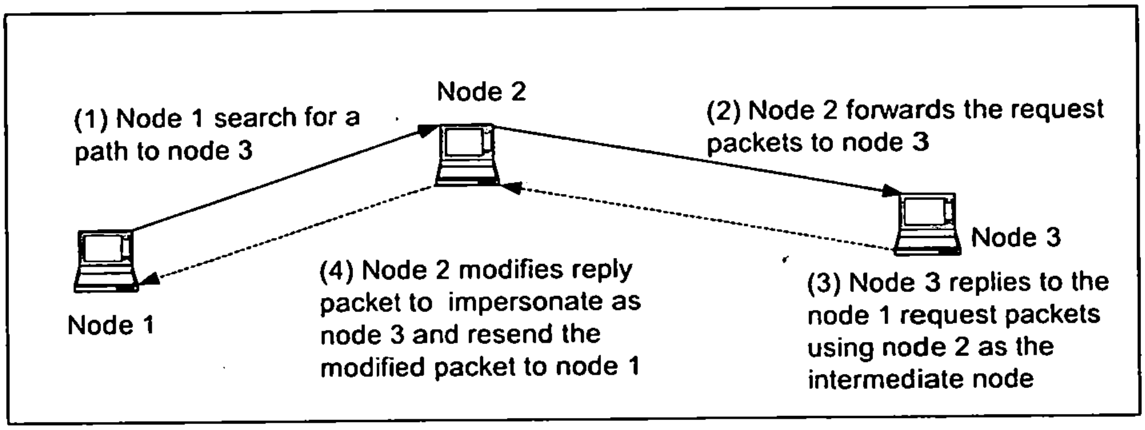

Node 2
(1) Node 1 search for a path to node 3
(2) Node 2 forwards the request packets to node 3
Node 3
Node 1
(4) Node 2 modifies reply packet to impersonate as node 3 and resend the modified packet to node 1
(3) Node 3 replies to the node 1 request packets using node 2 as the intermediate node

Figure 3- 2: Impersonation Attack

3.4.1.3 The Sybil Attacks

Impersonation attacks might be launched in a more severe way, known as The Sybil attacks (named after the subject of the book Sybil, a case study of a woman with multiple personality disorder), as described in (Douceur, 2002). In such attacks, instead of masquerading under the identity of a single node, adversaries launch an impersonation attack to masquerade as several nodes' identities in the network. In a route discovery process, several different paths from source to the destination node might be revealed. These extra paths have been exploited in several routing protocols to mitigate the effects of the impersonation attacks against the ad hoc networks. In such protocols, packets from the source node were duplicated and redundantly sent through several different paths to ensure their survivability to reach the destination node. However, this strategy is not suitable in the presence of the Sybil attackers. Since such attackers are able to masquerade as several nodes, they are also capable to compromise several routes in the networks thus degrade the effectiveness of the packet redundancy strategy.

3.4.2 Interception

Attackers might launch interception attacks to get unauthorised access to the routing messages that are not intentionally sent to them. This kind of attack jeopardises the integrity of the packets because such packets might be modified before being forwarded to the next hop. Additionally, the intercepted packets might also be analysed before being passed to the destination thus violating the confidentiality. Examples of attacks that can be classified under the interception attacks are wormhole attacks, black hole attacks, and routing packet analysis attacks.

3.4.2.1 Wormhole Attacks

In wormhole attacks, two malicious nodes are colluding to create a shortcut to reach the destination node. By creating this shortcut, they could trick the source node and win in the route discovery process (Hu et al., 2003). Packets in these two colluding attackers are usually transmitted using a wired connection to create the fastest route from source to the destination node. Figure 3-3 illustrates how adversaries could launch the wormhole attacks. If the wormhole nodes consistently maintain the bogus routes, they could permanently deny other routes from being established. As a result, the intermediate nodes reside along that denied routes are unable to participate in network operations.

Wormhole attacks are usually difficult to detect because such attacks actually follow the nature of routing mechanisms where the shortest path will always be chosen to forward packets from source to the destination node. In addition, detecting wormhole attacks will be more difficult if such attacks are only used to launch passive attacks like traffic analysis or password cracking attacks, rather than any active disrupting attacks such as misrouting or denial of service. At some point, if the wormhole nodes did not launch any disruptive attacks, the shortcut created by them might help in lowering the times to forward packets from source to the destination node (Burg, 2003). However, this shortcut will confuse the connections between nodes in the networks and might damage the routing protocols, especially when the wormhole nodes stop their operations.

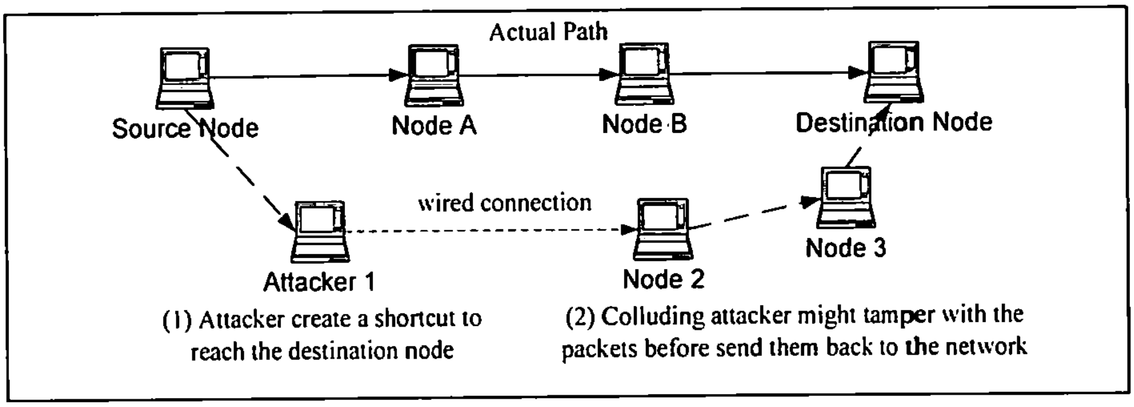

Actual Path
Source Node
Node A
Node B
Destination Node
wired connection
Attacker 1
Node 2
Node 3
(1) Attacker create a shortcut to reach the destination node
(2) Colluding attacker might tamper with the packets before send them back to the network

Figure 3- 3: Wormhole Attack

3.4.2.2 Routing Packet Analysis Attacks

Since no disruptive actions occurred, routing packet analysis could be classified as one of the passive attacks against MANET (Qian et al., 2006). One way to launch such an attack is by exploiting the *promiscuous* mode employed in MANET. In *promiscuous* mode, if node A is the neighbour of node B and node C at a particular time, node A can always hear the transmissions between node B and node C. By exploiting this property, node A is able to analyse the overheard packets transmitted between node B and node C. More explanation regarding the *promiscuous* mode in MANET could be found in (Marti, 2000).

In addition, malicious nodes also could launch this attack by exploiting the nature in a multi hop routing.

In a multi hop routing, packets need to be forwarded through several intermediate nodes before they reach the final destination. Malicious nodes might exploit this opportunity by locating themselves in any location along the route to participate in the message forwarding process and later launch the routing packet analysis attacks. Attackers could use this attack as a first step to launch further attacks against MANET, such as to launch password cracking and location disclosure attacks. Routing packet analysis is very difficult to detect because of the nature of such attack, which does not directly disrupt the normal routing behaviour. An example of routing packet analysis attack is as illustrated in Figure 3-4.

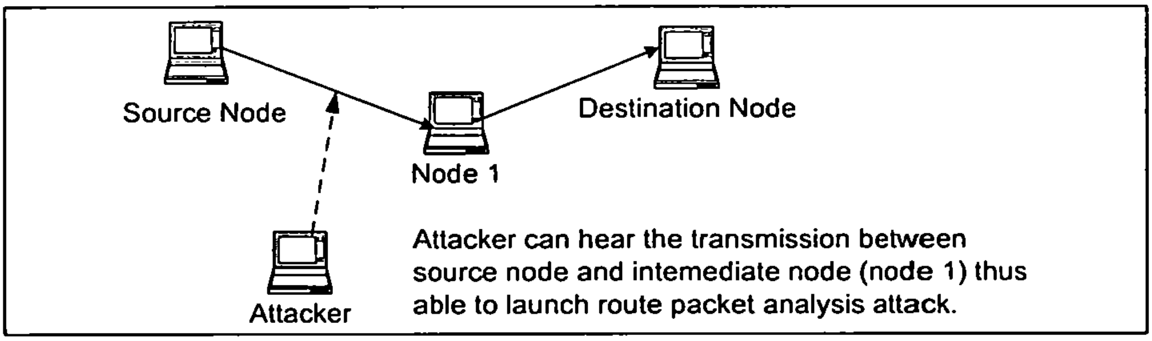

Source Node
Destination Node
Node 1
Attacker
Attacker can hear the transmission between source node and intemediate node (node 1) thus able to launch route packet analysis attack.

Figure 3- 4: Routing Packet Analysis Attack

3.4.2.3 Black Hole Attacks

In this attack, malicious nodes trick all their neighbouring nodes to attract all the routing packets to them. As in the wormhole attacks, malicious nodes could launch the black hole attacks by advertising themselves to the neighbouring nodes as having the optimal route to the requested destinations. However, unlike the wormhole attacks where multiple attackers colluded to attack one neighbouring node, in the black hole attacks, only one attacker is involved and it threatens all its neighbouring nodes (Lundberg, 2000). Figure 3-5 illustrates an example of the black hole attacks in MANET.

Detecting black hole attacks in the network is much easier than detecting the wormhole attacks because all the neighbouring nodes could collaborate with each other to report any malicious activity done by the black hole node. Attackers could do anything they desire to the captured packets. They could route the packets to the wrong destinations, modify the packets to interrupt the integrity service in the network or they could launch a denial of service attack by maliciously dropping all the packets. The effects of the black hole attacks might be more severe when several attackers collude to launch the collaborative black hole attacks (Ramaswamy et al., 2003).

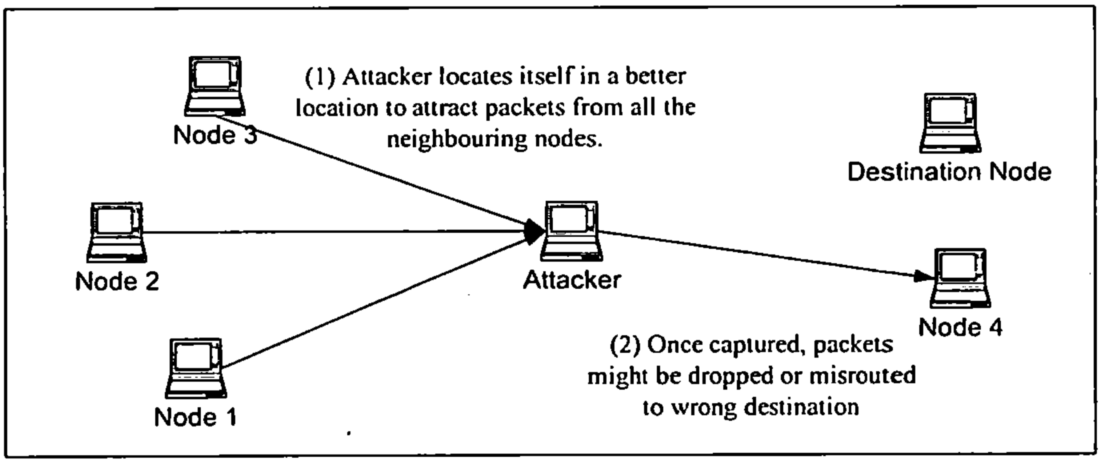

(1) Attacker locates itself in a better location to attract packets from all the neighbouring nodes.
Node 3
Destination Node
Node 2
Attacker
Node 4
(2) Once captured, packets might be dropped or misrouted to wrong destination
Node 1

Figure 3- 5: Black Hole Attack

3.4.2.4 Grey Hole Attacks

Similar to the black hole attacks, the grey hole attacks are also launched by malicious nodes to attract routing messages from all their neighbouring nodes. Therefore, similar strategies to attract packets in the black hole attacks might be reused in this attack. As mentioned in (Lundberg, 2000), one of the attacks that could be launched by black hole attackers is dropping all the intercepted routing messages. However, by consistently dropping all the packets, such a black hole node could be easily detected as being malicious to the network. Therefore, in order to make their malicious activity harder to detect, attackers in grey hole attack vary the packet dropping strategies like randomly dropping the packets or selectively forwarding the packets. Such strategies might confuse nodes in the networks to distinguish between the packet dropping attacks and the benign failure of the ad hoc network. Until now, no special attention has been given to overcome this attack. The only attempt that further discusses this attack can be found in (Ramaswamy et al., 2003).

3.4.3 Fabrication

Instead of modifying or interrupting the existing routing packets in the networks, malicious nodes could also fabricate their own packets to cause disruption and chaos in the network operation. They could launch message fabrication attacks by injecting huge packets into the networks such as in the sleep deprivation attacks. However, message fabrication attacks are not only launched by the malicious nodes. Such attacks also might come from the internal misbehaving nodes such as in route salvaging attacks.

3.4.3.1 Sleep Deprivation Attacks

This kind of attack is actually more specific to the MANET than the other types of network (Burg, 2003). The aim of this attack is to drain off limited resources in the mobile ad hoc nodes (e.g. the battery power), by constantly keeping them busy with processing unnecessary packets. In a routing protocol, sleep deprivation attacks might be launched by flooding the targeted node with unnecessary routing packets. For instance, attackers could flood any node in the network by sending a huge number of route requests, route replies or route error packets to the targeted node. As a result, that particular node is unable to participate in the routing mechanisms and denied from being reached by other nodes in the network. Figure 3-6 illustrates an example of sleep deprivation attack in MANET.

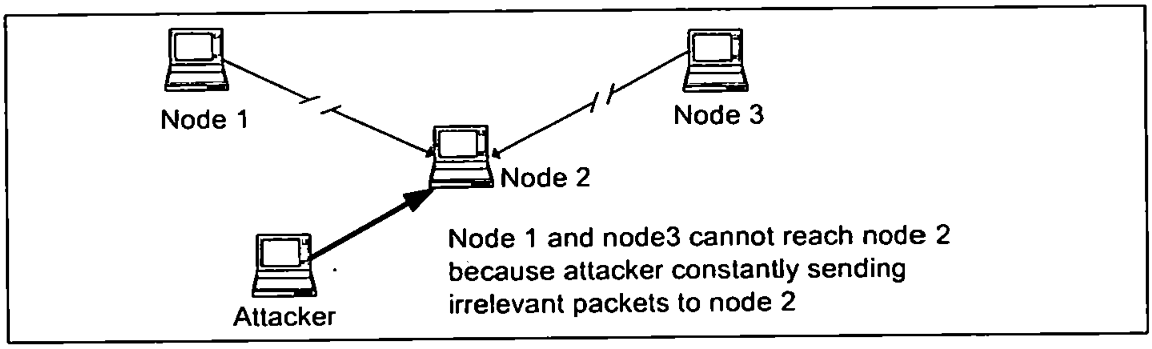

Node 1
Node 3
Node 2
Attacker
Node 1 and node3 cannot reach node 2 because attacker constantly sending irrelevant packets to node 2

Figure 3- 6: Sleep Deprivation Attack

3.4.3.2 Route Salvaging Attacks

Internal nodes might not only refuse to cooperate in network operations but they also might be greedy in using network resources. In a computer network, there is no guarantee that each transmitted packet will successfully reach the desired destination node (Ni et al., 1999). Packets might not reach the destination node because of natural network failures or might be under attack by adversaries. Therefore, to salvage their packets from such failures, internal nodes might duplicate and retransmit their packets even when not having received error messages. The effects of the route salvaging attacks might be more severe if there are large numbers of greedy nodes in the network. Additionally, it drains off more resources in intermediate and destination nodes, this attack also might cause the consumption of unnecessary network communication bandwidth. Figure 3-7 illustrates an example of route salvaging attack in MANET.

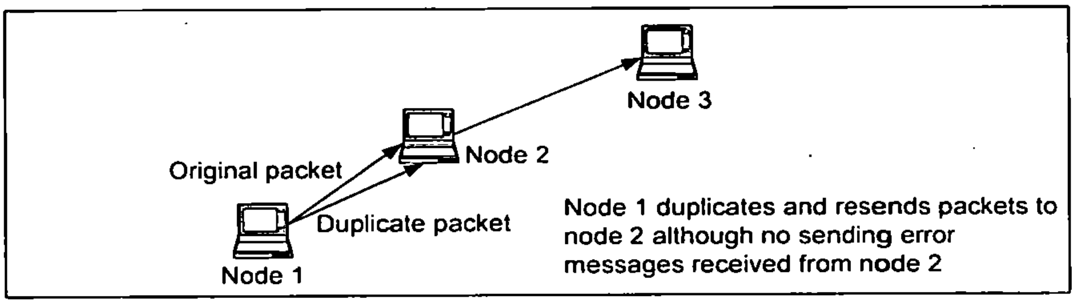

Node 3
Node 2
Original packet
Duplicate packet
Node 1
Node 1 duplicates and resends packets to node 2 although no sending error messages received from node 2

Figure 3- 7: Route Salvaging Attack

3.4.4 Interruption

Interruption attacks are launched to deny routing messages from reaching the destination nodes. Adversaries could do this by either attacking the routing messages or attacking the mobile nodes in the network. Actually, most of the attacks launched in the modification, interception, and fabrication attacks are aimed at interrupting the normal operations of the ad hoc network. For instance, adversaries aiming to interrupt the availability service in the network might destroy all paths to a particular victim node by using the message modification attacks. In a message fabrication attack, adversaries could overload the networks by injecting large numbers of unnecessary packets. Examples of attacks that could be classified under the interruption attacks category are packet dropping attacks, flooding attacks, and lack of cooperation attacks.

3.4.4.1 Packet Dropping Attacks

In a normal packet dropping attack, the adversary collaborates normally in the route discovery process and launches the packet dropping attacks if it is included as one of the intermediate nodes. Unlike in the black hole or in the grey hole attacks where packet dropping attacks are solely initiated by the malicious nodes, the adversary in the normal packet dropping attacks might as well come from the misbehaving internal nodes. Internal nodes are discouraged to participate in the packet forwarding process because such process will waste some of their own limited resources. Figure 3-8 illustrates an example of packet dropping attack in MANET.

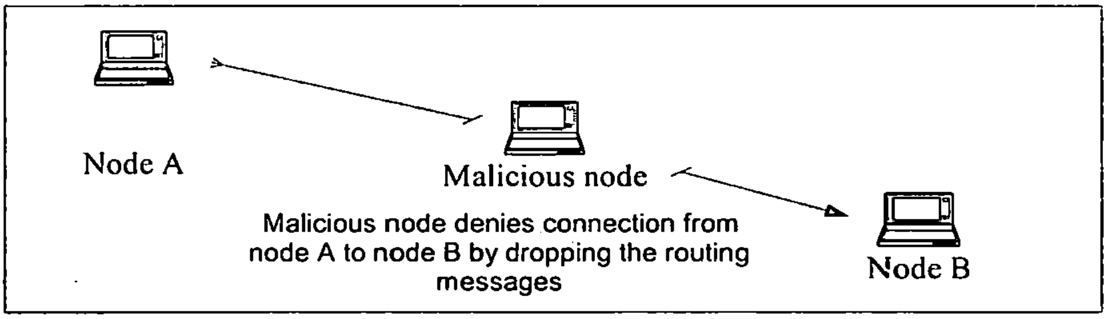

Node A
Malicious node
Malicious node denies connection from node A to node B by dropping the routing messages
Node B

Figure 3- 8: Packet Dropping Attack

There are four type of packet dropping attacks (Huang et al., 2003b), the explanations of which are as follows:

- *Constant Packet Dropping Attack*

  This is the very basic type of the packet dropping attacks. In this technique, all the incoming and outgoing packets from the targeted nodes will be dropped from the network.

- *Periodic Packet Dropping Attack*

  In this attack, adversary might not simply drop all the packets but will occasionally drop the packets in a periodic manner to avoid being easily detected.

- *Random Packet Dropping Attack*

  This attack is quite similar to the periodic packet dropping attack. However, since no specific time intervals used in this attack, the adversary will drop packets in a random time fashion.

- *Selective Packet Dropping Attack*

In this kind of attack, the adversary might use certain characteristic to select which packet it wants to drop. For instance, the adversary might choose to drop every packet destined to a particular node in the network.

3.4.4.2 Flooding Attacks

Adversaries also might interrupt the normal operations in the packet forwarding process by flooding the targeted destination nodes with large numbers of unnecessary packets. Nodes under the flooding attacks are unable to receive or forward any packet thus all the packets directed to them will be discarded from the network. An example of a flooding attack in MANET is as illustrated in Figure 3-9.

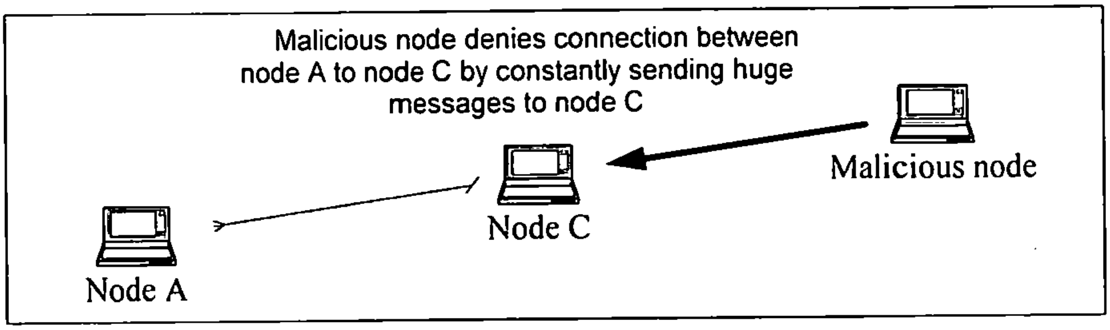

Malicious node denies connection between node A to node C by constantly sending huge messages to node C
Malicious node
Node C
Node A

Figure 3- 9: Flooding Attack

3.4.4.3 Lack of Cooperation Attacks

The lack of cooperation from the internal nodes to participate in network operations is also known as the refusal of service attack. In such attacks, internal nodes are discouraged to cooperate in network operations that do not benefit them because participating in such operations will drain their resources. Internal nodes might use different strategies to save their limited resources. They might refuse to forward the other nodes' packets, not send back the route error report to the sender when failing to forward packets, or might turn off their devices when not sending any packets in the networks. It is true that users in the open environment ad hoc networks are usually incapable to make their devices less cooperative in the network operations. However, this does not mean that this attack is not important in such an environment because users can always hire any commercial attacker to configure the devices to be less cooperative for them (Forristal et al., 2005). Figure 3-10 illustrates an example of lack of cooperation attack in MANET.

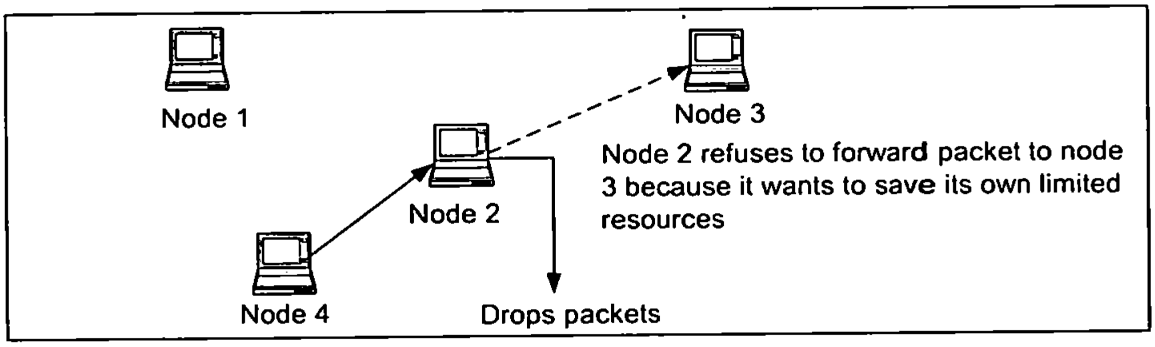

Node 1
Node 3
Node 2 refuses to forward packet to node 3 because it wants to save its own limited resources
Node 2
Node 4
Drops packets

Figure 3- 10: Lack of Cooperation Attack

3.5 Conclusion

This chapter has detailed several characteristics of attacks that targeting the ad hoc networks. Based on the investigation of the attacks patterns, one can make a conclusion that all of these common attacks are actually launched by exploiting the routing messages, which have been used for communication among nodes in MANET. Driven by that conclusion, further investigation on various techniques that could be used by the attackers to exploit the routing messages has been carried out and presented in this chapter. Responding to the threats and vulnerabilities, researchers have proposed several security measures to protect MANET, as discussed in the next chapter.

Chapter 4

MANET Security Schemes

4.1 Introduction

Responding to the security threats and vulnerabilities against MANET as detailed in previous chapter, researchers have proposed several security measures to protect such a network. In general, security measures proposed for MANET can be categorised into two groups: prevention and detection/response mechanisms. Figure 4-1 illustrates briefly how these two mechanisms can be used to protect MANET from security threats and vulnerabilities.

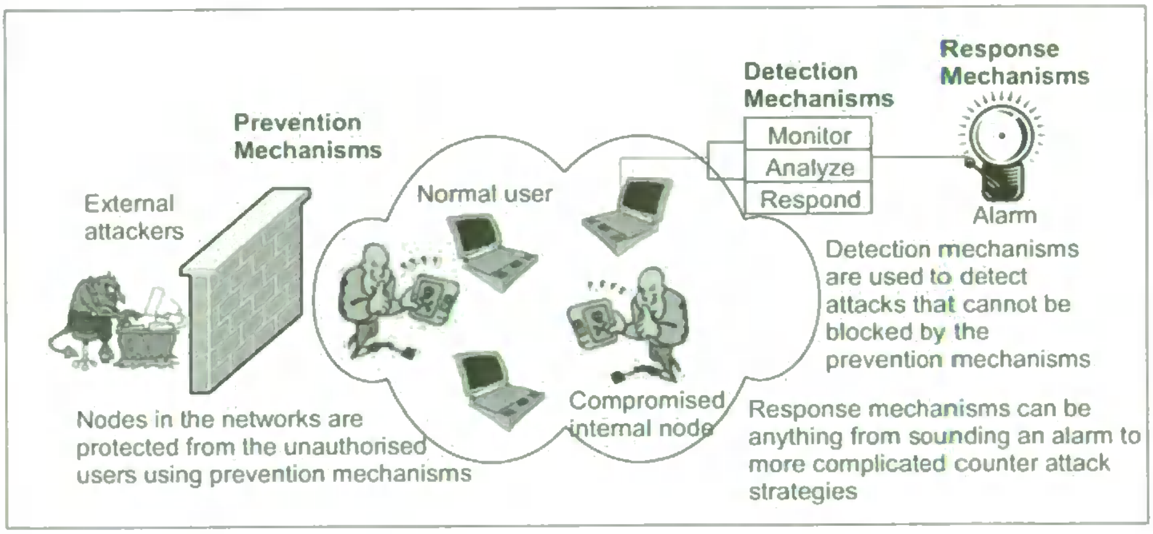

Detection Mechanisms
Response Mechanisms
Prevention Mechanisms
Monitor
Analyze
Respond
Alarm
External attackers
Normal user
Detection mechanisms are used to detect attacks that cannot be blocked by the prevention mechanisms
Compromised internal node
Nodes in the networks are protected from the unauthorised users using prevention mechanisms
Response mechanisms can be anything from sounding an alarm to more complicated counter attack strategies

Figure 4- 1: Security Solutions for MANET

4.2 Prevention Mechanisms

Prevention is one of the important phases in a security life cycle (King, 2002). Usually a prevention mechanism is deployed to secure network operations from malicious external attackers. A very basic way to prevent attacks is by employing an authentication mechanism. In authentication, techniques such as symmetric and public key system have been widely used and well operated in wired networks. However, the implementation of these techniques in MANET is not as straightforward as in the wired networks. The non-existence of dedicated nodes or central administrator in MANET to manage the authentication service makes the authentication mechanism one of the most challenging issues.

Responding to this problem, researchers have devised several mechanisms that are suited to the MANET environment, such as authentication schemes, secure routing protocols, and cooperation enforcement mechanisms as initial steps to defend from attacks. Examples of such mechanisms are considered in the subsections that follow.

4.2.1 Authentication

Authentication is a basic way to defend any network architecture from attacks. This mechanism usually needs security keys (private and public keys) to prove a node's identity in the network. Enabling this mechanism in MANET is very challenging because such a network operates in a self-organised manner with no central authority, which must exist to manage and distribute the security keys to all nodes in the network. However, despite the challenge, this mechanism is still very important as it can provide the first defensive wall to block external attackers from getting unauthorised access to the system (Zhu et al., 2006). Several techniques to enable authentication in MANET are as follows:

4.2.1.1 Self-Authentication / Distributed Authentication

Since it is unlikely to assume the existence of a Central Authority (CA) in MANET, the majority of researchers suggest that the authentication mechanism in such a network should be carried out in a distributed fashion. Each node in a MANET is responsible for authenticating other nodes in the network, as well as to collaborate in managing the authentication infrastructure. Capkun et al. (2003a) proposed two different self-organised authentication schemes for MANET where nodes can independently establish security associations amongst themselves in an offline mode. In the first method, each node creates its own private and public keys and exchanges the keys with other adjacent nodes through secure short-range connectivity channels such as infrared. In a second approach, they assume the existence of an offline CA to verify the identity of nodes that wish to participate in the network operations. The offline CA is only required at the initial stage when nodes wish to join the networks. This is to simplify the establishment of security associations among nodes because without the offline CA, each node needs to verify the identity of other nodes in the networks by itself before exchanging any public information. Both of these methods have the same drawback, which is that the establishment of security associations requires some time (nodes need close physical proximity to each other to exchange information). However, in their study, they did an experiment to prove that node mobility in MANET could help in establishing trust using their proposed methods.

Balfanz et al. (2002) proposed a two-phase authentication scheme for MANET, where in the first phase, each node independently exchanges their public keys by having a physical contact with each other. These pre-known public key sets will then be used in a second phase to authenticate users in a real multi hop ad hoc operations. However, as in (Capkun et al., 2003a), this approach also requires some time for all nodes in the networks to exchange public keys among themselves.

In another scenario, Hubaux et al. (2001) proposed a self-organised public key infrastructure for MANET. In their work, each node creates its own public/private key pairs and issues its own public certificate to other nodes. To eliminate the existence of CA, this work proposed a mechanism to store the public certificates for all nodes in a distributed fashion. Each node stores its own certificates as well as several other nodes certificates in its own repository. A description on their authentication mechanisms can be found in (Capkun et al., 2003b). However, this solution only provides probabilistic guarantees and is based on the assumption that all nodes are honest in the certificate issuing process. If any node issues more than one public certificate that corresponds to its identity, the integrity of the system will be jeopardised.

4.2.1.2 Imprinting

Imprinting is another way to establish secure transient associations among ad hoc network nodes in the absence of an online authentication server (Stajano & Anderson, 1999). In this approach, each node (slave) will be imprinted with a 'soul' that binds it to the other node (master) over a non-wireless channel. Once imprinted with a master soul, a slave node will only follow instructions that came from the master node throughout its participation in network operations. However, slave nodes are imprint-able, which means they can have different masters once the relations are revoked or expired. Whilst this approach seems to work well in the organised or localised environments, it is unlikely to be deployed in the open MANET environment where each node might have several connections simultaneously (not limited to one-to-one master/slave interaction). However, the author has proposed an extension of this approach in (Stajano, 2000), which covers the relationships between peers in an open environment. In the extension work, a slave node can act as a master node to its peer (after gaining permission from its master) so that trust chains can be established to support multi hop operations in MANET. However, whilst the author latter solution has successfully solved the problem of communications between peers, such solution was still not suitable to be used in the open MANET environment due to the requirement of master node to manage the slave nodes.

4.2.1.3 Central Authority (CA) Emulations

Central authority (CA) is much related to the authentication process. CA is used in the infrastructure networks to authenticate users as well as to manage the authentication infrastructure. However, in MANET, the existence of CA cannot be assumed. Therefore, each node is responsible to carry out the authentication process and to manage the authentication infrastructure. Researchers agree about the unsuitability to deploy CA in MANET, but they still believe that CA plays a critical role in the authentication process and it must be made available to support MANET operations. For that reason, several solutions have been proposed to make CA virtually exist in MANET.

For instance, Zhou and Haas (1999) proposed a security mechanism that emulates the CA role in authenticating users in MANET. In their work, $(t+1)$ nodes from the entire nodes in the networks are responsible to authenticate new nodes that wish to participate in the network operations. They introduced the concept of threshold cryptography to avoid single point attacks (which usually happen when employing one node to play the CA roles). $(t+1)$ nodes are pre-determined at an initial stage and each of them holds a partial share of the system private key. Any new node that wishes to join the network must send request packets to all the $(t+1)$ nodes. Upon received the requests, $(t+1)$ nodes will use their partial shares to sign certificate for that new node. They claimed that, for n period times, their system is immune to t compromised nodes because to compromise the system, attackers must gain all the $(t+1)$ shares. However, it is possible for the attackers to gradually break into all the $(t+1)$ nodes for a certain n times. To overcome this problem, they proposed a share refreshing scheme where $(t+1)$ will generate new sub shares over time and distributed among them through a secured side channel. So that, to compromise the system, adversaries need to compromise all the $(t+1)$ nodes before the shares expired.

Kong et al. (2001) also suggest the same approach. However, they outlined some problems if the static $(t+1)$ nodes are used to hold the system private shares. In open MANET environments, a huge number of users can join and leave the networks. As a result, it might be difficult or it will take longer for one node to contact all the $(t+1)$ nodes to join the networks. To solve this problem, they proposed a solution where $(t+1)$ nodes (which are called k nodes in their system) can be replicated depending on the density of the networks. Any node that has been authenticated can request to copy the shares and play the same roles as the existing k nodes. As a result, the time required for new nodes to collect all the system shares is reduced in this system. However, this approach is more vulnerable to security attacks than the previous one. Replication of the k nodes makes the chances for adversaries to capture the system shares broader. Unlike in (Zhou & Haas, 1999) where adversaries need to find all the partial shares in $(t+1)$ before the shares expired, in this system, the time needed will be smaller because adversaries have more *k* nodes as the targets. Therefore, a new mechanism has been suggested to overcome this problem, which can be found in (Luo et al., 2002). In that work, the authors suggest to select *k* nodes in a dynamic fashion instead of having fixed *k* nodes in the system. Since adversaries did not know how many shares they must obtain to compromise the system, it will at least make the attacking process harder. Another improvement in this expanded version is that *k* nodes can also collaboratively detect the misbehaving nodes in the networks. Any *k* node can broadcast the misbehaving activities of a certain user, but reports from all the *k* nodes are needed before a suspicious node can be penalised.

Yi and Kravets (2002) also proposed similar dynamic *k* nodes selection strategies. However, instead of randomly choosing the number of dynamic *k* nodes, they proposed a mechanism that will choose an appropriate number of *k* nodes dynamically depending on the density of the nodes in the networks. They also suggested that all the *k* nodes must be appropriately chosen based on the level of physical security offered by them (*k* nodes). The reason behind this is that they believed the *k* nodes with a high physical security levels are more difficult to be tampered or compromised by the attackers, thus could increase the reliability of their authentication mechanism.

All the solutions by Zhou and Haas (1999), Luo et al. (2002), and Yi and Kravets (2002) provide mechanisms to authenticate new users that wish to join the network. However, none of these studies consider appropriately the key distribution issues. Most of them assume that each node knows the other nodes' public keys. However, in practice, these public keys must be generated and advertised so that every node is aware of other nodes public keys. One solution for this problem is to use an ID-based cryptosystem as proposed in (Khalili et al., 2003). In this system, all nodes mutually decide an acceptable set of security parameters such as National Insurance number or combination of name and birth date to be used as user's public key. Since this public information can be derived from any existing database (e.g. advertise in newspaper), the need for public key distribution mechanism can be eliminated.

4.2.1.4 Friends Recommendations

There are also some efforts from researchers to introduce the concept of friends' recommendations in establishing trust among nodes in MANET. Weimerskirch and Thonet (2001) proposed a mechanism to establish trust based upon human interactions. In their solution, they suggest that one can only authenticate other users' identities if they are known to each other. If they have not met each other before, they can ask for recommendations from their friends that might know or have had any communication experience with the targeted node before. Recommendation also can come from the referee provided by the target node. This approach can be used when there is no friend that can help to verify the identity of the targeted node.

The concept of friends also has been introduced as an extension in (Capkun et al., 2003a) to speed up the establishment of security associations among nodes. Instead of only exchanging personal public information, two communicating nodes also can exchange their friends' public information over the secure channel (e.g. Infrared) so that many security associations can be establish at one time. The proposed IDS framework in this study utilises the same friendship concept, and its implementation is described later in the thesis.

4.2.1.5 Cluster-Based Approach

The cluster-based approach has been introduced in infrastructure networks to distribute CA roles into several cluster head nodes. A similar approach has also been introduced in MANET with few adjustments to fit with such networks requirements. Venkatraman and Agrawal (2000) proposed a novel authentication scheme, which employs a cluster-based approach for MANET. In their work, nodes are divided into several clusters, which were controlled by corresponding cluster heads. Each node has its own private key, and shares a cluster public key to enable them to authenticate and communicate locally with other nodes in the same cluster. Cluster head nodes are responsible to authenticate new nodes that wish to join in their own-managed cluster and to establish a cluster-to-cluster communication in case any child node wishes to make a cross-cluster communication.

Lu et al. (2001) also suggest the same approach, but with some improvements. The most significant improvement is that their solution supports the mobile nature of the ad hoc nodes. Unlike in (Venkatraman & Agrawal, 2000), where each node is assumed to be static and under control of one cluster head, this solution allows nodes to move from one cluster to another, thus enabling those nodes to communicate with other nodes in a different cluster without any help from cluster head. However, in self-organised MANET, the suitability of this cluster-based approach cannot be guaranteed because it is very challenging to choose a cluster head node in this autonomous and very dynamic network environment.

4.2.2 Secure Routing

MANET operates in a different manner from the other types of wireless networks. For instance, MANET employs multi hop packet forwarding process, which requires participation from all nodes in the networks. Since MANET nodes are autonomous, there is always a possibility for the authorised nodes (nodes that have been authenticated to use network resources) to misbehave during their participation in network operations. Some of the reasons for nodes misbehaving could be because they want to save some of their limited resources or simply because they are actually the adversaries who impersonate other nodes to get access to the network resources or secret information. Secure routing is a prevention mechanism that has been designed to protect MANET against this type of adversary. The sub-sections that follow present some of the secure routing mechanisms that have been proposed so far.

4.2.2.1 Solutions for Proactive Routing

A proactive routing protocol seems to be less efficient in terms of its performance and the ability to adapt to route changes in a highly mobile environment as claimed in (Hu et al., 2002b). This is because, unlike a reactive routing, where paths from source to destination nodes are established on fly, in proactive routing all routes are pre-determined before the packet forwarding process. However, proactive routing has its own advantages. Since routing paths are pre-determined and stored in a routing table, this information can be re-used to send packets to the same destination in future, so that redundant route discovery process can be eliminated. In addition, this information also can be used to add some security features in the ad hoc routing mechanisms. Hu et al. (2002a) proposed a secure routing protocol called SEAD: Secure efficient distance vector routing for MANET, which provides some security features to the existing DSDV routing protocol using efficient one-way hash functions. Standard DSDV is vulnerable to packet modification attacks where intermediate nodes can tamper with the packet's contents before forwarding it to the next intermediate nodes. Modifying the packet's content can cause many security attacks. In their work, they suggest using a hash chains method to protect the packets' integrity. In every communication hop, intermediate nodes will calculate the hash value of the packets and compare it with the hash value given by the previous node to check the integrity. However, for the system to work properly, they assume that every node use the same algorithm to generate the hash values of the packets, and the existence of a symmetric authentication mechanism to authenticate each user in the networks.

4.2.2.2 Solutions for Reactive Routing

The dynamic nature of reactive routing not only improves the performance of packet forwarding operation but also at the same time makes it more vulnerable to many security attacks. As a result, more security solutions have been proposed by researchers to solve security issues in reactive routing than those faced in proactive routing. For instance, Hu et al. (2002b) try to prevent unauthorised access and illegitimate modification to the routing packets by employing appropriate authentication and hashing mechanisms. Their solution called Ariadne (A secure on-demand routing protocol for ad hoc networks) is actually similar to SEAD in (Hu et al., 2002a), which also has been proposed by them. Since both SEAD and Ariadne are based on the same idea, their operations are not significantly different to each other.

Sanzgiri et al. (2002) proposed a mechanism called ARAN (A secure routing protocol for ad hoc networks) to protect a reactive routing protocol from security attacks. This solution employs both end-to-end and hop by hop authentication mechanisms to fulfil important security requirements such as integrity, confidentiality, authentication and non-repudiation services. Each message will be encrypted at the source node and can only be decrypted by the destination node to protect its integrity as well as its confidentiality. Messages also need to be authenticated at every communication hop by intermediate nodes to protect the networks against impersonation and message fabrication attacks. However, this solution assumes the existence of central server to manage and distribute the authentication keys for every user in the network thus limits its operation to the organised or localised MANET. It is also important to highlight here that this solution only protects the integrity of the messages, but does not protect the routing control information (e.g. hop counts) as has been proposed by Zapata and Asokan (2002).

Similar to (Sanzgiri et al., 2002), the secure routing protocol proposed in (Zapata & Asoka, 2002) also seems to be suitable for a MANET that operates in an organised environment. This is because this solution assumes the existence of a central authority to manage, as well as to distribute, the authentication keys to all nodes in the networks. This solution employs two mechanisms: a digital signature to authenticate users, and a hash chain to protect

AODV routing packets from being modified on route. By employing these two mechanisms, their solution is effective in preventing a number of attacks against the ad hoc network routing protocols such as an unauthorised access to the system resources, impersonation attacks, and message modification attacks.

Ghazizadeh et al. (2002) also proposed a digital signature authentication mechanism to protect routing packets in MANET. Similar to other proposals that employ the same technique, this solution also assumes the existence of CA to manage and distribute key pairs for every node in the networks. Since routing messages are authenticated at every hop from source to the destination nodes, this technique can be very useful to prevent both impersonation and fabrication attacks. What makes this solution different from the others is that the introduction of the path-rating mechanism to rate every different path from the source to the destination. In a path-rating mechanism, the destination node will provide an acknowledgment to the source node for every single packet that has been successfully received. This acknowledgement will give an idea to the source node about the trustworthiness of the intermediate nodes along the route to reach the destination node. Routes with a higher trustworthy value will be chosen to route packets to the destination nodes. As a result, routes with lower trustworthy values (usually because one or more nodes not operate properly or misbehave) will always being avoided. Since bad nodes are avoided from routes, network overhead because of packet loss can be reduced, thus will save some of nodes' limited resources.

Most of the secure routing protocols proposed above follow the standard of reactive routing protocol where every node in the network is responsible to participate in the packet forwarding process, however, Yi et al. (2001) proposed a different approach. In their solution, they proposed a mechanism called SAR: A Security-Aware Routing for a wireless ad hoc network, which employs two metrics to ensure the security of every packet traversing the network. In the first metric, they agree with other researchers in (Hu et al., 2002b; Zapata & Asokan, 2002; Ghazizadeh et al., 2002; Sanzgiri et al., 2002) that the security requirements in the network can be achieved by embedding some security features when forwarding the packets such as employing authentication mechanism to authenticate users and using a hash chain to protect packets' integrity. In addition to that, they suggested that the security of packets in the networks could be enhanced by using a second metric that they proposed in their system. They proposed to use a hierarchy system where only users with certain privilege or authority level can participate in the packet forwarding process. Users will be given a special privilege to participate in the network operations if they can fulfil a certain security level, which can ensure the security of packets handled by them. However, this hierarchical system is only suitable for an organised environment where each user in the network is expected to be a member of an organisation.

4.2.2.3 General Solutions for Both Proactive and Reactive Routing

All of the secure routing protocols discussed so far were designed to suit either proactive or reactive routing protocols. None of them can be used as a generic solution for both routing protocols. However, Papadimitratos and Haas (2002) have proposed a secure routing protocol that can achieve both of these routing protocol families. In general, their solution is quite similar to that proposed by Hu et al. (2002a) where they also employ the hashing method to .protect packets from being modified. However, instead of only protecting the MANET from modification attacks, their work also provides a solution for the message fabrication attacks. In their solution, extra packet header data (containing the hash value, the sequence number, and the id of the packet) will be appended to the original packet. Similar to the solution in (Hu et al., 2002a), the hash value will be used to check the integrity of the packet while the unique id will protect the packets from being maliciously created. In addition to that, a sequence number can be used to defend packets from being copied by adversaries that desire to launch a packet replay attack. Further explanation of this secure routing protocol can be found in (Papadimitratos & Haas, 2003).

4.2.3 Cooperation Enforcement

As mentioned earlier, MANET operations are much depending on node's willingness to cooperate in the network. However, some nodes in the network might refuse to cooperate, could be because they want to save their limited resources. To prevent nodes from being selfish, several cooperation enforcement mechanisms have been proposed and they are outlined in the sections that follow.

4.2.3.1 Charging and Rewarding Scheme

Providing incentives to stimulate nodes' cooperation can be used to prevent denial of service attacks that might come from the internal users. Zhong et al. (2003) proposed a credit-based mechanism to stimulate nodes cooperation in MANET operations. In their approach, each node has a certain amount of credit that can be used to send packets in the networks. Nodes will loose their credits when sending their own packets, but will gain some credits if forwarding other nodes packets. Nodes will not be able to send packets to the networks after utilising all their credits. As a result, each node will be self-enforced to cooperate in the network operations because in that way they can earn some credits for later use. However, this approach assumes the existence of central entity known as Credits Clearance System (CCS) to manage the charging and rewarding credits for every node in the networks. Each node will get a receipt for every packet that successfully reached the destination node. This receipt then needs to be presented to the CCS to claim the credits. Nodes with insufficient credits can also buy some credits from the CCS.

Blazevic et al. (2001) had introduced a virtual currency called 'Nuglet' to stimulate nodes' cooperation in MANET. In their approach, each user needs to pay a certain amount of 'Nuglets' when using network resources, but will gain some 'Nuglets' when participating in network operations. This concept is quite similar to the credit-based system in (Zhong et al., 2003). However, this approach suggests that each node needs to be charged and pay different values of 'Nuglet' depending on the cost of sending and forwarding the packets. This is because costs of sending and forwarding packets are varied depending upon the number of intermediate nodes involved, as well as the amount of resources consumed. The packet purse model and packet trade model are two mechanisms that are being used in this mechanism to deal with different values of 'Nuglets' in sending packets. In a packet purse model, the source node will estimate the number of 'Nuglets' needed to reach the destination. However, if the source node made a wrong estimation, the packet will be dropped because of insufficient 'Nuglets' to reach the destination. Nodes in the networks can use a packet trade model to avoid wrong estimation problem in a packet purse model. In a packet trade model, the source node is not charged for sending packets. Intermediate nodes will buy the packets using some amount of 'Nuglets' and sell it back to the next hop nodes with higher value until it reaches the destination. As a result, the destination node needs to pay the total costs of the packets requested by the last intermediate node. However, similar to the packet purse model, the packet trade model also has its own weaknesses. If the packet purse model is unsuitable because of the difficulty to estimate the number of 'Nuglets' needed to reach the destination node, MANET operations are vulnerable to denial of service attack in the packet trade model. This is because, in a packet trade model, the source node is not being charged to send packets thus adversaries can launch a denial of service attack by injecting bogus packets to the networks. As a result, they proposed a hybrid version of a packet purse model and a packet trade model to improve the performance. Their later work, as well as the simulation results, can be found in (Buttyan & Hubaux, 2001).

Both 'Nuglet' and credit-based mechanisms seem to be unfair for nodes that are located at the edge locations. Nodes located at the centre of the communications will have the opportunity to earn more credits because the opportunities to be included in the network operations are much greater than the nodes that are located outside the busy region. As a result, after utilising all the credits or 'Nuglets', nodes in the edge areas are incapable of sending packets in the networks.

Raghavan and Snoeren (2003) have a solution to this problem. In their work, they proposed a mechanism where nodes can choose to either use a policed best-effort method or priced priority-forwarding method when cooperating in the network operations. Nodes in the centre of the communications, which usually have more credits and have more opportunities to be included in the network operations might chose to use the priced priority-forwarding where they can earn more credits. In other hand, nodes that located outside the busy region might be interested to use a policed best-effort method where they do not need any credit to send packets. However, their activities will be monitored to detect any misuse.

4.2.3.2 Reputation Mechanism

Another way to prevent denial of service attacks initiated by the legitimate insider nodes is by employing a reputation mechanism. In such an approach as suggested by Michiardi and Molva (2002), each node has its own reputation rate that can be used by other nodes as indicator of its behaviour. Every node will try as hard as it can to avoid any communication with a node that has a lower reputation rate. In this approach, the reputation of each user is rated based on own experiences as well as reports from other nodes. Unlike in a self-observation method, where reputation of the suspicious node can be rated as positive or negative depending on the behaviour of that node, only positive reputations can be accepted from the other nodes. This is to avoid a denial of service attack, which could happen when the malicious nodes are broadcasting false negative reputations for other nodes. Nodes can rate a certain user as misbehaved or not by using a watchdog mechanism which is capable of detecting any abnormal activities in a packet forwarding process as proposed in (Marti, 2000).

4.3 Detection and Response Mechanisms

As discussed in the previous section, prevention mechanisms can be very useful as a first defensive wall to protect MANET from many security attacks. Various prevention methods have been proposed ranging from simple authentication architectures to more complicated secure routing and cooperation enforcement mechanisms. Regardless of the assumptions made, most of these prevention mechanisms seem to work well and can provide some levels of security for MANET. However, a prevention mechanism alone is not enough to protect MANET from attacks that might come from external and internal attackers. Security needs to be addressed as a continuous lifecycle to make it effective in protecting any network from attacks (King, 2002). A security lifecycle comprises of three elements: prevention, detection, and response mechanisms, which depend upon each other to provide a reliable security protection.

Responding to this issue, researchers have proposed several detection and response mechanisms to complement the existing prevention mechanisms. Most of the efforts have generally attempted to address one of the following vital issues in the intrusion detection and response mechanisms: how to collect the audit data; what is the appropriate method to detect an intrusion; how to minimise false alarms; and how to respond to the intrusion. The next sections will summarise some of the efforts proposed by researchers in addressing these issues.

4.3.1 Audit Data Source

The effectiveness of any intrusion detection scheme is often related to the quality of data, which has been collected and used to detect malicious activities in the network. In general, data can be gathered from one of these two sources; host-based data sources, which reside at each node in the network, and network-based data sources which are usually collected at a network concentration point by a dedicated node (Albers et al., 2002). Both audit data sources are important to give a better view of what is going on in the networks, and thus can help to detect any malicious activity. A host-based audit data source is an infrastructure-independent audit data source because it can exist in any network architecture. However, the same thing does not apply to the network-based audit data source. No such concentration point or dedicated node exists in MANET that can be used to collect the whole network information like in the wired networks. As a result, most of the researchers suggested that the only available data source that can be used in MANET is the host-based option. However, since both the host-based and the network-based audit data sources are important to detect any intrusion attempt, several strategies have been proposed to make the network-based audit data source exist virtually in MANET. Here are the examples of the data collection strategies proposed for MANET.

4.3.1.1 Host-Based Audit Data Collection

Host-based data collection is a method used to collect users' system behaviours that can be monitored by the node itself without the aid of any dedicated devices such as firewalls or monitoring servers (Innella & McMillan, 2001). The question here is why each node needs to monitor its own systems but not that of others that might be malicious. The main reason is that none of the nodes in the network have the privilege to monitor other nodes' system behaviour. In any network architecture, the only person that can be trusted is the node itself, thus granting access to an anonymous user to monitor a node's own activities can be very harmful. Actually, there is no need to snoop into other users' systems to detect anomalies. With an appropriate method, each node can capture its own system behaviours and use that information to detect any abnormal activities caused by other nodes in the networks. Various ways have been proposed to collect user's activities.

Albers et al. (2002) suggested that the use of Simple Network Monitoring Protocol (SNMP) could be very useful to monitor the status of each node's communication activities with other users in the networks. Using the SNMP data, which is located in Management Information Base (MIB) as an audit data source, each node can analyze its own system's behaviours and detect any deviation from the expected patterns. The same concept has also been applied in (Awerbuch et al., 2002) where SNMP is used to log each request made and acknowledgement received to detect packet-dropping attacks. SNMP usually logs all the standard information of the monitored operations and stores them into the database. By analysing the audit data logged in the database, most of the common attacks such as packet dropping, message replay, and denial of service attacks can be detected. However, this information is not always sufficient in MANET, especially when dealing with attacks that violate nodes' multi hops communication, such as the black hole and wormhole attacks. For instance, nodes use SNMP log data to detect packet dropping attacks launched by an intermediate node that is located one hop away from them but they cannot use the same information to detect if subsequent hops intermediate nodes (e.g. 2

hops away intermediate node) dropped the packets. To address this issue, nodes need to collect extra information that can complement the SNMP audit source.

Hu et al. (2003) suggested that by embedding some information into the packets, the integrity of the packets when traversed across the networks could be validated. In their approach, they used a node's geographical location and packet lifetime to defend against the wormhole attacks. Upon receiving packets from the network, embedded information will be extracted and used as the audit data to detect any deviation. As mentioned earlier, the more audit data that can be collected from the host the more reliable the decision can be made in detecting attacks. A good example of research work that addresses this issue can be found in (Zhang et al., 2003). In that work, the authors suggested that high false alarm rates could be reduced significantly by having multiple audit data sources collected at every communication stack layer. Whilst this idea could be very useful to enrich the audit data source with reliable information, the process of collecting this information from each layer is not an easy task. There was no specific work addressing this issue so far but the idea to use the mobile agents as the data collection tool could be very handy. The concept of mobile agent in MANET has been introduced in (Kachirski & Gupta, 2003) where three types of mobile agents (monitoring, detection, and response) collaborate in one intrusion detection system.

4.3.1.2 Emulation of the Network-Based Audit Data Collection

Whilst the host-based audit data source gives an idea about what is going on at every host in the network, the network-based audit data source on the other hand can provide each node in the network with useful information about the whole network activities (ISS, 1998). In wired networks, audit data from the network-based audit data source are usually gathered by deploying a dedicated device such as monitoring server or firewall at a strategic location (e.g. network concentration point). However, such server or firewall needs to be managed by the system administrator to ensure it operates properly. As a result, this approach seems to be infeasible in MANET environment because all the nodes in such a network operate autonomously, thus the existence of a system administrator cannot be assumed. However, this does not mean that the network information cannot be collected from the mobile ad hoc networks.

Researchers have proposed several mechanisms that emulate the role of monitoring servers and firewalls in collecting MANET network audit data. Listed here are some of the examples of such mechanisms. One of the most common assumptions made by researchers in their works is that each node in MANET is capable of hearing the transmission in and out from other nodes in the networks. This assumption, which known as node in a *promiscuous* mode (SearchSecurity.com, 2003), is also coupled with the assumption that each node has a bidirectional link to each other. Researchers claimed that by using this assumption, partial or localised network activities can be collected by each node, which later can be shared among them as a virtual network-based audit data source. This approach seems to be first applied in MANET by Marti (2000), but then has been widely used by researchers in (Yang et al., 2002; Stamouli, 2003; Buchegger & Boudec, 2001; Paul & Westhoff, 2002) as part of their research strategies or assumptions.

Whilst a node operating in *promiscuous* mode seems can provide a reliable network-based audit data, there is still another challenge that needs to be addressed, which is how the information can be shared amongst users in the network without being falsified by adversaries. Since each node in MANET possibly shares the audit data with the unknown users, which could be malicious, each node in MANET must have the capability to authenticate each other user's identity. However, the use of any authentication scheme not only consumes a node's limited resources, but also requires the existence of a CA, which is impractical in MANET operations. To overcome this problem, researchers have introduced the concept of friends to share the partial network audit data collected by each node in the network (Yang et al., 2002; Buchegger & Boudec, 2001; Paul & Westhoff, 2002). In such a concept, network audit data can only be shared if it comes from a friend that can be trusted. Since each node knows its own friends from the beginning, the deployment of any authentication scheme can be simplified. Similar works have been proposed by Weimerskirch and Thonet (2001) and Capkun et al. (2003a) as discussed in Section 4.2.1.4.

Another method proposed to share the partial localised network audit data is by using an agent technology (Kachirski & Gupta, 2002; Albers et al., 2002). In such an approach, each user has its own agent that will travel from one node to another to collect all the partial network audit data from each other node. The use of mobile agents in this approach can help in minimising the node's limited resources, as well as the network's bandwidth usage. However, this approach still requires an appropriate authentication scheme to authenticate each other node's mobile agents because it cannot be guaranteed that those mobile agents are originated from the legitimate users.

4.3.2 Method of Detection

As in wired networks, one can use either misuse or anomaly detection techniques to detect intrusions in MANET (Kazienko & Dorosz, 2004). Anomaly detection is a technique used to detect all the intrusive activities that deviate from the normal workflow of the system. It is a trainable system where patterns of the normal activities can be learned from time to time, even while the system is running. On the other hand, misuse detection is a technique used to detect all the intrusive activities that match to the attack signatures, which are stored in the database. However, it is difficult to train the system to detect new kind of attacks on its own, thus, the attack signatures need to be updated regularly by the system administrator. Anomaly and misuse detection techniques have their own capabilities and limitations as summarised in Table 4-1. Further explanation of these detection techniques can be found in (Vattikonda et al., 2003).

Misuse	Anomaly
Fast processing, no complex calculations	Requires more processing time
No training required	Requires training
Difficult to manage attack signatures	Requires minimum administrations
Fewer false alarms	High false alarms
Incapable to detect unknown attacks	Able to detect unknown attacks

Table 4- 1: Misuse vs. Anomaly Detections Capabilities and Limitations

However, between these two mechanisms, researchers claimed that the anomaly detection would perform better than the misuse detection in MANET (Huang et al., 2003b; Zhang et al., 2003). This is because MANET technology is still new and in fact until now there is still no standard protocol for a routing mechanism, thus making the process of compiling the attack signatures in MANET harder than in a matured technology (e.g. wired networks). There is also an issue of updating the attack signatures database. Unlike in an anomaly detection where patterns of normal activities can be trained autonomously, attack signatures used in a misuse detection mechanism need to be managed and updated by a system administrator (Meier, 2003). In addition, MANET also has its own characteristics, such as fluctuating link and random network topology that can adversely affect the performance of misuse detection mechanism. For instance, link breakage in MANET might be because of node's movement from one location to another, and not necessarily caused by the attackers. This is only an example of many other unexpected scenarios in MANET that can significantly increase the number of false positive alarms if the misuse detection mechanism is employed. However, this does not mean that the misuse detection mechanism is completely inappropriate in MANET. As mentioned earlier, both misuse and anomaly detection mechanisms have their own advantages and disadvantages, and perhaps the combination of these two mechanisms will improve the performance of intrusion detection mechanism in MANET (Wai et al., 2003; Vattikonda et al., 2003). Listed here are some of the detection mechanisms that have been proposed for MANET.

Zhang et al. (2003) proposed a detection mechanism that employs RIPPER classifiers, which are based on an IREP (incremental reduced error pruning) machine learning algorithm, introduced by Cohen (1995) and Light SVM (Support Vector Machine) (Joachims, 2004) classifiers to generate normal activity patterns derived from the selected audit data source. These auto-generated patterns will then be used by the detection engine to detect any deviations from normal MANET routing operations. From the experiments, the authors found that the Light SVM performed better that the RIPPER in generating normal behaviour patterns. This finding suggested that the traditional classifier such as RIPPER is not suitable to be used in MANET because of its inability to cope up with the node's high mobility.

Similar effort also has been undertaken in (Huang et al., 2003b). In that work, the authors suggested that in any MANET operation, several features can be extracted, and can then be used to generate any unclassified normal activities. For instance, in a packet forwarding process, several features can be identified such as (1) Is the destination node reachable? (2) Are there any packets successfully transmitted to the destination node before? (3) How many intermediate nodes are involved in order to reach the destination node? Correlations between these features can then be used to generate more normal activity patterns that have not been classified yet. The more normal behaviour patterns that can be generated without doubt will improve the accuracy of the anomaly detection mechanism (Abad et al., 2003).

In another effort, Stamouli (2003) has proposed a real time detection strategy to detect routing deviations in MANETs. Instead of analysing the whole network process using the statistical approach as in (Zhang et al., 2003; Huang et al., 2003b), the author suggested that an early detection of abnormal activity could be achieved by monitoring the current machine states. For instance, in a packet forwarding process, a machine (mobile node) can be in a several different states such as sending a route request packet, receiving an acknowledgement from the neighbour nodes, sending data, receiving reply packet from destination, etc. The validity of all these states usually can be examined. For instance, maximum time and threshold can be set to detect the route request packets that have been maliciously dropped. There are many other efforts, which attempt to provide reliable detection mechanisms in MANET such as in (Albers et al., 2002; Kachirski & Gupta, 2002; Wai et al., 2003; Paul & Westhoff, 2002). However, most of them are still in the early stages, thus no details about the detection architectures have yet been released.

In addition to the general anomaly detection techniques, researchers also have proposed several other techniques to detect attacks that are unique to MANET environment. Awerbuch et al. (2002) proposed a mechanism to detect the lack of cooperation attack, an attack where selfish intermediate nodes dropped all the packets forwarded through them in order to save their own limited resources (e.g. bandwidth, battery power). Their detection mechanism employs a very basic strategy by checking the acknowledgement packets to detect any packet loss. In normal packet send/request operations, the intermediate and destination nodes will send an acknowledgement packet for every successfully received packet. Thus, by monitoring these acknowledgement packets, the sender can immediately detect any packet loss caused by the malicious intermediate nodes or broken paths to the destination. However, this method cannot guarantee that the packets will go along the path and reach the destination node. Intermediate nodes can cheat this method by sending an acknowledgement packet to the sender but later drop the packets. This problem has been addressed in (Just et al., 2003), where the authors propose a mechanism called a distributed probing technique to solve this problem. In their work, they exploit redundant paths available in the networks to send probe packets to two hops away intermediate nodes. This probing technique can tell the sender node if the immediate (one hop away) intermediate nodes did send the acknowledgement packets but failed to forward packets to the next hop node.

Wormhole is another type of attack, which is unique to the MANET as described in previous chapter. Realising the effects that can be caused by this attack, Hu et al. (2003)

have proposed a mechanism to defend against it. In their work, they suggest to embed extra information (e.g. time and location of node) to the forwarding packets. For instance, a maximum time allowed for a packet to travel from one hop to another can be set to detect any packet that has travelled between two long distance wormhole nodes. In addition, the nodes' geographical location also could be very useful to detect any wormhole nodes but this requires additional tools (e.g. GPS).

The problem of black hole attacks also has been addressed in (Ramaswamy et al., 2003). In such work, the authors suggested that the black hole nodes in the networks could be identified by employing a cross checking strategy. The source node will collect some information (routing history, neighbouring nodes, etc) from the immediate intermediate node as well as the two hops away intermediate node. This information will be compared to each other and if there is a mismatch, both intermediate nodes (in this case, one hop and two hops away intermediate nodes) will be assumed as suspicious nodes. Further investigation will be carried out to check which one of these two intermediate nodes is malicious by doing the same cross checking procedure to both two hops and three hops away intermediate nodes. If the information received by the source node is still mismatched, confirmation can be made that the two hops away intermediate node is a black hole node, which falsified the routing information in order to intercept data from the source node. On the other hand, if the information received is matched to each other, the conclusion can be made that the one hop away intermediate node is the black hole node.

4.3.3 False Alarm Acceptance Level

False alarms are very common in intrusion detection systems that employ an anomaly detection mechanism. They happen when the system misjudges any normal activity as being abnormal. It is a very big problem in intrusion detection systems because if too many false alarms triggered in the system, users will start ignoring the alarms, and thus possibly overlook real intrusion attempts (Sekar et al., 2002). This problem becomes more acute in MANET because to classify what is normal and what is abnormal activity in such networks is not an easy task. Sometimes, nodes failed to forward packets in MANET because of natural network failure, not because of any malicious activity occurred in the networks. If detection architecture in MANET is built without considering this issue, there might be a lot of false alarms in the system and non-malicious nodes could be wrongly penalised. However, this problem is not left unseen. Several mechanisms have been proposed to tackle this issue and they are proven can reduce the number of false alarms in MANET intrusion detection system.

The very basic approach to this problem has been proposed in (Yang et al., 2002) where the authors employ a threshold mechanism to reduce a number of false alarms. To avoid misjudgement, upon detection of any anomaly behaviour, the misbehaving node will not be simply penalised. Each node maintains other nodes' bad behaviour table, which has a value that will increase every time that particular node misbehaves. When the value reaches the threshold, confirmation can be made that the particular node is malicious and will be avoided. Similar threshold techniques also have been applied in (Bhargava & Agrawal, 2001; Huang et al., 2003b). Besides threshold mechanism, a rating scheme also can be used to reduce the number of false alarm in the intrusion detection system. Rating scheme can be applied either to rate the reliability of route, or to rate the reputation of nodes to forward packets in the networks.

Ideas to rate the reliability of paths in MANET has been made in (Awerbuch et al., 2002; Marti, 2000). The difference between these two efforts is that in (Awerbuch et al., 2002), a positive value is used to rate a reliable path, thus when forwarding packets, source node will select a route with the highest value. On the other hand, in (Marti, 2000), a negative value is given to a path that contains a misbehaving nodes, thus a source node will try to avoid using the negative value path when forwarding packets to a destination node. As mentioned earlier, rating schemes also can be used to rate the reputation of each node in the networks. For instance, in (Buchegger & Boudec, 2002a), the authors proposed a rating mechanism called CONFIDANT to detect selfish nodes in MANET. Every node in the networks will detect its neighbour's behaviour when participating in a packet forwarding process. Nodes will receive a good reputation for every successful forwarded packet and on the other hand will receive a bad reputation when failed to do so. By maintaining every node's reputation rating, selfish nodes that refuse to forward packets in the networks can be detected and then further action can be taken, such as eliminating them from the networks or simply avoiding them in a future packet forwarding process. More harsh actions, like ignoring any packet forwarding requests from the selfish nodes, can also be used to motivate them to not being selfish in the networks.

In IDS, the more information gathered related to nodes activity means that the more accurate a conclusion about intrusive activity in the networks can be made. Several mechanisms have been proposed to collect observations made by other nodes in investigating one suspicious node. However, there is no doubt that the information provided might come from malicious nodes that try to blackmail a well-behaved node in the network. With that problem in mind, Zhang et al. (2003) have proposed a voting scheme to avoid misjudging a well-behaved node because of false accusations received from other nodes. The same voting scheme also has been applied in (Paul & Westhoff, 2002) where the authors used it to enable collaborative detection of a selfish node in MANET.

4.3.4 Response Behaviour

Another important characteristic of IDS is the method of response to the intrusion. Usually in wired networks, alarms will be triggered to alert the network administrator about the intrusions. Once alerted, the network administrator will take further actions, such as disconnecting the vulnerable nodes from the network, or initiate a re-authentication process to authenticate every user in the network and discard the intrusive nodes. However, a network administrator cannot be assumed to exist in a MANET environment. In MANET, each node is responsible to respond to any intrusive behaviour, which makes this issue more challenging than in the wired networks. Several response mechanisms have been proposed for MANET. A simple way to response to intrusion in MANET is by avoiding any communication with the intrusive nodes.

For instance, in (Awerbuch et al., 2002), malicious nodes are avoided from collaborating in a packet forwarding process by choosing a different path to reach the destination. Unlike in wired networks where usually one fixed connection is used to transfer data from one location to another, in MANET, the existence of redundant paths creates flexibility in network communication. Such flexibility allows a source node to choose a reliable path to forward packets and avoid using a path that contains malicious nodes in it. This mechanism also has been proposed in (Marti, 2000; Ramaswamy et al., 2003; Just et al., 2003) as a part of their intrusion detection and response mechanisms.

The main issue in response mechanism for MANET is on how to alert the other users about the detected intrusive activities. If an intrusion is confirmed to have occurred, an alarm can be triggered to inform other neighbouring nodes as suggested in (Buchegger & Boudec, 2001). In addition, the observed intrusive activities can be shared among the neighbouring nodes to initiate collaborative detection as in (Zhang et al., 2003; Paul & Westhoff, 2002). However, as mentioned in (Bhargava & Agrawal, 2001), this mechanism is vulnerable to blackmail attacks where a malicious node can sound a fake alarm to discard an innocent node from the networks. Methods that are more aggressive, such as eliminating the intrusive nodes from the networks, can also be used to respond to intrusion. This concept has been used in (Yang et al., 2002) where upon detection, malicious nodes are prohibited to renew their expired token (token used to join network communication), and thus they will be eliminated from the networks. The same elimination process also has been suggested in (Paul & Westhoff, 2002; Bhargava & Agrawal, 2001) but in a different approach. In such works, voting mechanisms have been used to detect the malicious nodes. Upon detection, the intrusive activities will be propagated to other nodes to isolate the malicious nodes from the networks.

4.4 Conclusion

It is unlikely for any approach to suit all the security requirements or be able to solve all the security problems that exist in MANET. Prevention mechanism can be very useful as a first defensive wall to guard MANET from external attackers. However, more dangerous attacks can come from the internal nodes (Buchegger & Boudec, 2002b). This is because the success of operations in MANET are very much dependent on the cooperation of all nodes to participate in the packet forwarding process. Detecting internal attackers is very challenging in MANET because such attackers might come from the outside nodes that compromised the internal nodes, or they also might come from the internal nodes that refused to collaborate in the network's operations. Defending MANET from the internal nodes is not as straightforward as defending the networks from the external attackers. Prevention mechanisms such as authentication and secure routing are not capable of defending against the internal attackers because the compromised internal nodes usually have the secret information or network private keys. The best way to defend against this problem is by deploying a detection mechanism. There is no doubt that the detection mechanism cannot prevent attacks from being launched, but it is capable of detecting the malicious activities in the networks once the prevention mechanism has been bypassed.

Several solutions have been proposed to detect intrusions in MANET environment. Some of them focused upon the detection method, some of them aimed to solve the problem of collecting the audit data source, and some others focused on how to deal with high false alarms and techniques to respond to intrusions. Since an IDS for MANETs are still new and immature, there are still many issues that need to be addressed and improved. Characteristics and strategies of the existing IDS proposed for MANET as discussed in this chapter have been used as guideline to the design of a novel IDS framework as presented in Chapter 5.

Chapter 5

A Two-tier Intrusion Detection Framework

5.1 Introduction

A two-tier hybrid IDS for MANET is a novel IDS architecture proposed to improve the efficiency of existing MANET IDS architectures with the help of friend nodes. The main idea of the proposed system is to provide a reliable IDS that can detect any intrusion attempts and at the same time reduce the number of false alarms raised in the system. An intrusion detection system has been chosen as the basis for the PhD work as it is capable of protecting MANET from both internal and external attackers. In addition, research works focusing on the detection mechanisms for such networks are also still few and immature when compared to the efforts put on the prevention mechanisms.

5.2 Conceptual Framework

The proposed IDS framework is designed to help users in detecting most of the active attacks that were discussed in Chapter 3, such as the modification, fabrication and interruption attacks. As mentioned in (Debar et al., 1998), different types of Intrusion Detection System (IDS) can be distinguished from each other by looking at their characteristics such as the location of the audit data source, the detection method, the behaviour on detection (response), and the usage frequency. For instance, an IDS that detects intrusions using a host-based detection strategy is different from the one that employs a network-based detection strategy, and both of them have their own advantages and disadvantages. Choosing the appropriate strategies for all the IDS characteristics is very important, especially when dealing with the challenging environment of MANET. A two-tier hybrid IDS for MANET is a novel IDS architecture proposed to improve the efficiency of existing MANET IDS architectures. The main idea of the proposed system is to provide a reliable intrusion detection mechanism that can detect any intrusion attempts and at the same time reduce the number of false alarms raised in the system. With the focus to improve the detection strategies, only a simple response mechanism will be deployed in the proposed system as a complement to the detection mechanisms. Table 5-1 summarises the problems and vulnerabilities of existing IDS that this study are intended to address.

IDS Requirements	**Techniques used in existing IDS**	**Problems/ Vulnerabilities**	**Proposed solutions in two-tier IDS**
Network-based audit data sources	Receive audit data gathered by other trusted nodes	Information might be altered by malicious nodes	**Self-experience:** Capture overheard audit data of adjacent nodes only
	Mobile agents to collect audit data for global detection	Mobile agents are vulnerable to attacks	**Friends-observation:** Audit data for global detection will be captured and analysed by friends
Detection methods	Misuse detection	Difficult to manage/update attack signatures	**Signature management:** A platform for nodes to exchange attack signatures
		Unable to detect novel attacks	**Hybrid detection:** Misuse/ anomaly detection
	Anomaly detection	High false alarms	**2nd tier global detection:** Request further investigation and votes from trusted friends before making any decision
Global detections	Receive single report from trusted friends/ neighbours	Vulnerable to blackmail attacks	**Filtered Reports:** Accept intrusion reports from friends and drop reports from anonymous users
	Voting threshold for several reports received	Vulnerable to colluding blackmail attackers	

Table 5- 1: Objectives of the Proposed IDS

5.3 Architecture Consideration

This study is not the first attempt to secure MANET operations using intrusion detection mechanisms. Several proposals have been made in (Zhang et al., 2003; Rajavaram et al., 2002; Huang et al., 2003b). The proposed IDS framework, as illustrated in Figure 5-1, has been designed after considering several issues identified from the investigation of previous proposals. The sections that follow present some of the considerations that have been made when designing the proposed IDS framework.

5.3.1 Two-Tier Detection Architecture

The idea of having two-tier detection architecture is to provide a faster detection mechanism that is capable of detecting intrusive activities at their initial stages. Relying upon a local-based detection method alone is not sufficient to detect intrusion at its initial stage. Since information gathered in a local-based IDS is limited to the local activities, each node in the network must have enough experiences before any suspicious activities can be confirmed as intrusive or not. This limitation will slow down the detection process, which can be made faster if the host-based detection method is combined with the network-based detection method. For that reason, a global detection mechanism, which emulates the roles of a network-based detection method, has been proposed in this new IDS architecture.

A local-based IDS is located in the first tier and will be triggered first to investigate any suspicious activity before being passed to the global detection mechanism, which is located at the second tier. This is because the information gathered in a local-based IDS is the first hand information collected from a self local audit data source, which can be trusted and can be made available in an instant (King et al., 2004). On the other hand, the detection process in the global detection mechanism will require more time to be completed because the information supplied by the other nodes will require more time to reach the requested node and must be validated to ensure the integrity.

5.3.2 Real Time Audit Data Source

Two important issues have been considered when deciding appropriate audit data source strategies for the proposed architecture. The first one is on the behaviour of the data collection strategy, whether to use a real time or a periodic technique. Since the objective of the proposed architecture is to detect intrusion at the early stage of its appearance, a real time data collection strategy is proposed in the architecture. In addition to that, a periodic technique might not be suitable in a high mobility MANET environment as the collected information might be valid for a short period only. For instance, the attackers might have already left the network when the IDS detects their intrusive activities. Another important issue that needs to be considered is the location of the audit data source. Since MANET operates in a self-organised manner and the identity of other nodes is difficult to verify (because of the absence of a third party authentication server), the only audit data sources that can be trusted are the ones that come from nodes self-experience and self-observation. Details explanations about these data collection strategies will be discussed in section 5.4.

5.3.3 Hybrid Detection Method

The core element of an intrusion detection system is the detection method, which is used to investigate any suspicious activities that have occurred in the network. In general, there are two detection methods that can be used and they are misuse detection and anomaly detection methods as previously discussed in section 4.2.2. Although researchers have made clear the advantages and disadvantages of both detection strategies (Kachirski & Gupta, 2002), choosing between the two methods is still not an easy task especially in a MANET environment. Misuse signatures are difficult to build in MANET because of its immaturity and the unique characteristics (e.g. high mobility, transient connection, fluctuate wireless links) it exhibits. This situation has driven most of the researchers working on MANET IDS to choose an anomaly detection method for their proposed architectures (Zhang et al., 2003; Rajavaram et al., 2002). However, the capability of a misuse detection method cannot be simply ignored. Misuse detection can give results that are more accurate in term of detecting true intrusion attempts and therefore able to reduce the number of false alarms compare to the anomaly detection method (Kachirski & Gupta, 2002). Considering both misuse and anomaly detection capabilities, the proposed framework tries to combine these two detection methods into a hybrid system with the aim to study its performance compared to the most frequently used (an anomaly detection strategy), in MANET environment. However, the difficulty to build the attack database is not the only reason that makes researchers choose to employ anomaly detection rather than a misuse detection method. Another reason is that the absence of a system administrator in MANET makes the process of updating the attack database more difficult compared to the infrastructure networks. To ease this problem, a signature management mechanism has been deployed in the proposed framework. The proposed signature mechanism provides a platform for nodes to exchange their attacks signatures. This, if not eliminated, could limit the involvement of CA in the network.

5.3.4 Using Friends for Global Detection

The main issue to clarify here is the reasons of using friends to assist in a global detection mechanism. As mentioned earlier, MANET operates in self-organised manner without the existence of any authentication server to authorise each user in the network. Without a reliable identity verification measure, information about any intrusive activities gathered by each node cannot be shared with other nodes as the information might be falsified to blackmail other users. Zhang et al. (2003) have proposed a voting mechanism to overcome this problem. Similar approaches have also been proposed in (Paul & Westhoff, 2002; Bhargava & Agrawal, 2001). Voting mechanisms can be very useful to defend against a single blackmail attacker, but it is not immune against multiple colluding blackmail attackers. However, this voting mechanism can be improved by filtering the votes. For instance, only votes from friends can be counted to judge any intrusive activity. In the proposed IDS framework, the concept of friendship has been introduced as an alternative to the existing voting mechanism. For further explanation of the friendship concept, please refer to Chapter 6.

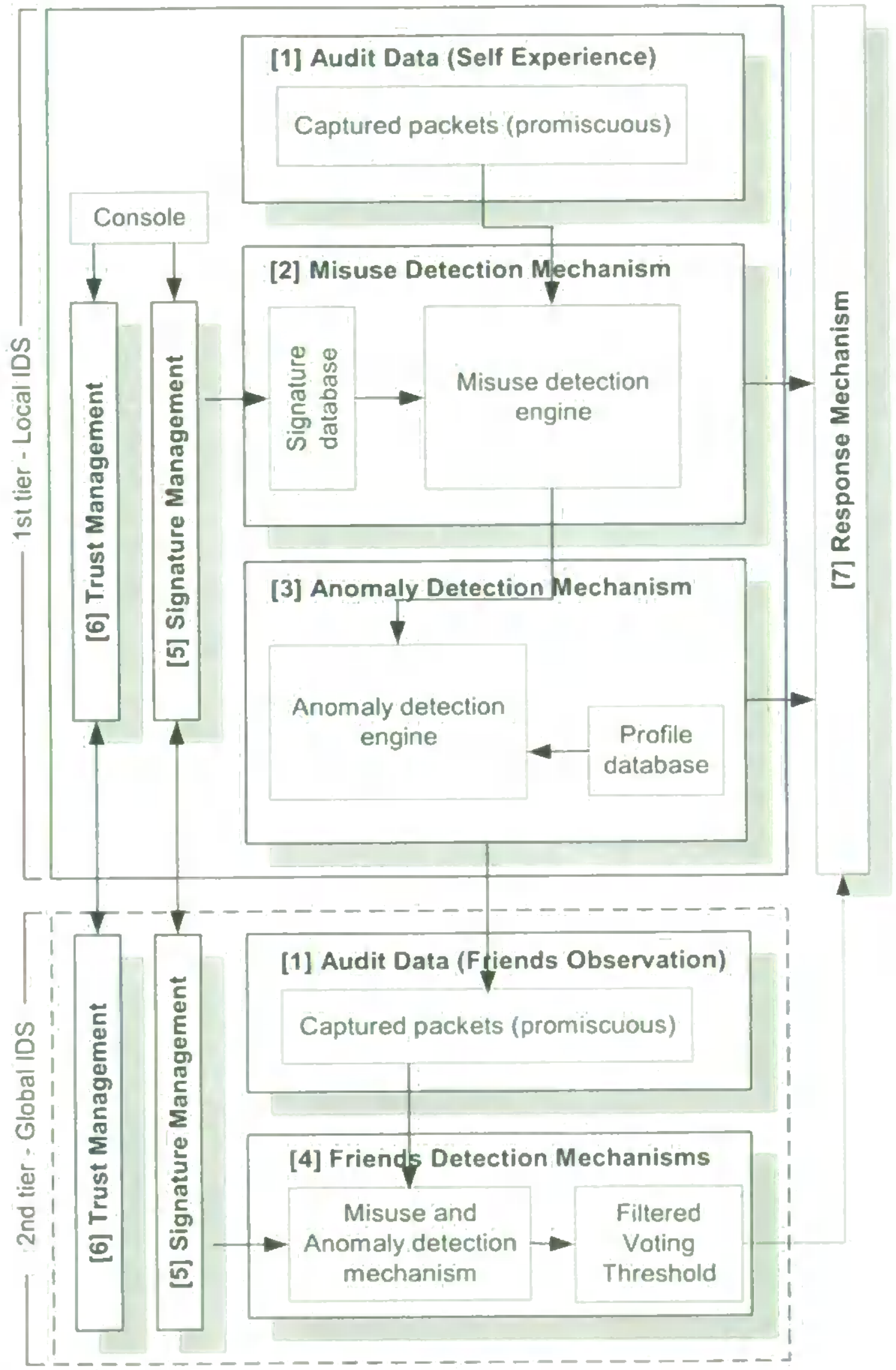

[1] Audit Data (Self Experience)
Captured packets (promiscuous)
Console
[2] Misuse Detection Mechanism
Signature database
Misuse detection engine
[3] Anomaly Detection Mechanism
Anomaly detection engine
Profile database
[6] Trust Management
[5] Signature Management
1st tier - Local IDS
[7] Response Mechanism
[1] Audit Data (Friends Observation)
Captured packets (promiscuous)
[4] Friends Detection Mechanisms
Misuse and Anomaly detection mechanism
Filtered Voting Threshold
[6] Trust Management
[5] Signature Management
2nd tier - Global IDS

Figure 5- 1: Conceptual Framework of the Two-tier Hybrid IDS for MANET

5.4 System Components

As illustrated in Figure 5-1, the proposed IDS framework has seven main modules, covering the audit data source to the response mechanism for alerting other nodes. Details of each module and their roles are described below.

5.4.1 Audit Data Source

This is where the audit data will be gathered for further investigation. In the proposed architecture, two audit data sources have been identified as appropriate to help detecting intrusive activities in the networks.

- *Self-Experience Audit Data*

  Any network operation, which has been initiated or having a direct connection with the user itself is classified as a self-experience audit data. For instance, in a packet forwarding process, the source, destination, and all the intermediate nodes will have a direct experience of such process and are capable of logging the related activities of the process for further investigation if something suspicious is detected.

- *Neighbours/Friends Observation Audit Data*

  Data are not restricted to be gathered by direct participation nodes (source, destination, and all the intermediate nodes). Neighbours that are physically close to the participating nodes are also capable to capture the overheard network activities using a *promiscuous* mode. This kind of audit data is known as the neighbours/friends observation audit data in the proposed framework and can make the detection process faster compared to the intrusion detection system that only relies upon the self-experience audit data.

Details about the type of data being captured and analysed in this study is discussed in Chapter 7.

5.4.2 Misuse Detection Mechanism

The misuse detection mechanism detects intrusions in the network by comparing the audit data with a set of attacks signatures that have been stored in the database. At the initial stage, the attack database might only cover a few attack signatures, but as the time goes by, with help from friends and the signature management module, the attack signature database will reach its maturity level and thus capable to detect more attacks.

- *Misuse Detection Engine*

  This is where the captured audit data will be compared to the attack signatures stored in the signature database. An existing misuse detection engine (e.g. pattern matching) will be applied here.

- *Signature Database*

  This is where the set of pre-known attacks against MANET routing protocols are stored. The signatures are semi-dynamic, where each node could receive updates from its friends, thus eliminates the need of CA to distribute attack signatures updates.

5.4.3 Anomaly Detection Mechanism

Attacks that cannot be detected by a misuse detection mechanism will be passed to the anomaly detection mechanism for further investigation. The failure of detecting the attacks could be because of the attack signature database is still immature or could be because of the insufficient evidence. The anomaly detection mechanism applied in this study is similar to the existing techniques proposed by previous researchers and the main components are as follows:

- *Anomaly Detection Engine*

  This is where the captured audit data will be compared to the user/network profiles stored in the profile database.

- *Profile Database*

This is where the normal profiles of user and network behaviours are stored. It is quite difficult to build a complete set of user/network profiles for a MANET, due to its unique characteristics. Common practice in a MANET research is to build the user/network profiles of a MANET based on certain specification applied in such network, such as routing protocol or security mechanism specifications. Similar practice is applied in this study. A detail explanation of users/networks behaviour profiling is discussed in Chapter 7.

5.4.4 Friends Detection Mechanism

The aim of the proposed architecture is to detect attacks at their initial stages so that the implications of the attacks can be minimised. For that reason, any suspicious activity that has not been detected as intrusive by a local detection mechanism must be sent to the global detection module for further investigation. This global detection mechanism requires cooperation from all nodes in the networks to detect intrusions. However, since MANET operates without the aid of a network administrator or third party authentication server, not a single node in the network can be trusted except the node itself. Receiving intrusion reports or alerts from anonymous nodes in MANET could expose the entire network from the impact of blackmail attackers. For that reason, a friend detection mechanism has been proposed to overcome this node's trustworthiness problem. Detailed explanations on this concept will be discussed further in Chapter 6.

5.4.5 Signature Management

This module will enable a dynamic update to the misuse detection mechanism of the proposed architecture. It might be impossible for each node to create its own attack signatures, thus a minor involvement of CA could be expected here. CA could insert new attack signatures to several nodes in the networks via a console, and later the updates will be passed to all nodes via a signature management module.

5.4.6 Trust Management

The proposed IDS framework utilises the concept of friendship for global detection and response mechanisms. Each node needs to build its own friend lists so that it could be included in as many global detection and response action as possible. This module provides a platform for each node to build its own trusted friend lists.

5.4.7 Response Mechanism

This module is responsible for reacting to any intrusive activity detected by the misuse, anomaly, or friend detection mechanisms. However, since the focus of the proposed architecture is on the detection strategies, only basic response strategies will be deployed here. Such strategies are as follow:

- *Local Response*

  A local response unit in this module will add the misbehaving nodes to the bad node table for further action. Once identified, several punishment steps could be done against the bad nodes such as excluding them from participating in a packet forwarding process, or refusing to forward their packets.

- *Global Response*

  A global response unit in this module will alert other nodes in the network by broadcasting the intrusion alarms. Neighbour nodes receiving these alarms will add the intrusive nodes to their bad node tables to avoid using them in a future packet-forwarding process. However, to avoid false accusations, only alarms received from friends can be accepted in the proposed framework.

5.5 Conclusion

The proposed system is a novel IDS architecture, which aims to detect intrusions in MANET at the early stages of such intrusive activities, and at the same time to improve the detection accuracy by reducing the number of false alarms rate. The system combines misuse and anomaly detection methods to provide each node in the network with a better local intrusion detection mechanism. This hybrid detection mechanism is supported with trust and signature management mechanisms, which are useful to ease the task of updating the attack database. The proposed system is also equipped with a global detection mechanism to increase the chances of detecting attacks at their initial stages. However, as mentioned in previous works, global detection and response mechanisms in MANET IDS are always vulnerable to false accusation and blackmail attacks. MANET's characteristics such as nodes are anonymous to each other, operate in a distributed fashion, and without a fixed network topology, are amongst the reasons that caused the problem. A friendship concept has been proposed in the system to ease this global IDS problem. It is envisaged that this concept will not only motivate nodes to cooperate in global IDS mechanisms, but also ease the problems of false accusations and blackmail attacks in MANET environments. The next chapter will discuss the details behind this friendship concept. The discussions include the reasons why such a relationship between nodes is important, how it could be made available in MANET environments, and how it is being implemented in this study.

Chapter 6

Friendship and a Trusted Community in a MANET

6.1 Introduction

As discussed in previous chapters, a MANET has its own characteristics that create more challenges during its operations compared to other types of wireless networks (e.g. WLAN, and WPAN). Since each node in a MANET is autonomous and has its own interests in the network, there is always a possibility that some of the nodes might refuse to cooperate in network operations to save their own limited resources. This kind of node misbehaviour, if not mitigated, could jeopardise the whole network operation, which heavily rely upon nodes' participations. This chapter discusses how friendship relations could motivate nodes to participate in the network operations. This chapter also discusses how the proposed friendship mechanism can provide solutions to solve several of MANET's security issues as mentioned in the proposed IDS framework.

6.2. Trust Relationship in MANET Environments

Having reputation or credit-based mechanisms deployed in a MANET environment as proposed in previous works might encourage nodes' cooperation in network operations (Michiardi & Molva, 2002). However, similar to other security mechanisms (e.g. authentication, secure routing protocols, and intrusion detection systems), these cooperative enforcement mechanisms are often proposed with an assumption that there also exists some level of mutual trust to ease the problem of node anonymity in the network. Without this assumption, the reliability of such security mechanisms could not be justified. Mutual trust provides a basis for each node to establish a security association with another node during network operations (Pirzada & McDonald, 2006). For instance, two nodes that have established a mutual trust between them in an offline mode (e.g. via secure side channel) could agree to become each other's trusted entity when participating in MANET operations. The more nodes that have established mutual trust between them in an offline mode results in more security associations that could be made available in MANET operations. Security associations will ensure that only the authorised nodes are allowed to participate in network operations, which could minimise the problems of misbehaving nodes. Security association also eliminates the problem of anonymous autonomous nodes in the system, thus providing capability to punish any misbehaving nodes in the network.

Building a trust relationship is not a new research field in MANET environments. Several solutions (as discussed in Chapter 4) have been proposed to solve this issue. However, most of them are associated to the authentication mechanisms, which usually require expensive cryptography, an assumption of a Central Authority (CA), and in some cases require several nodes to play the administrator roles (Zhou & Haas, 1999; Khalili et al., 2003; Stajano & Anderson, 1999; Stajano, 2000). In addition, almost all the existing work lack one important feature, which is, no collaborative effort among nodes to create a trusted community. As mentioned earlier, the creation of a trusted community is important to ensure the success of MANET operations. A special mechanism needs to be deployed to enable nodes to exchange security associations between them. This study proposes a friendship mechanism as an alternative solution to the problem. A pair of friend nodes, which are assumed to have a mutual trust between them before joining the network are capable of creating a security association between them to participate in MANET operations. In addition, the friendship mechanism is able to speed up the creation process of a trusted community in the network. If each security association that exists in the network is exclusively owned by any two nodes that created it, the development pace of the trusted community will be very slow. Each node needs to meet and establish mutual trust with other nodes, which requires a lot of time and effort. The friendship concept proposed in this study makes this process simpler and faster by providing a secure platform for nodes to exchange their security associations. This ongoing trust exchange process between nodes without doubt could lessen the number of anonymous communication, and thus lead to the creation of a trusted community in the network.

Although there are some existing studies suggesting a similar friendship concept, the way such a concept is used and interpreted is different from the one suggested in this thesis. For instance, Weimerskirch and Thonet (2001) proposed the same concept to authenticate anonymous nodes in MANET environments. In their system, two nodes are considered as friends to each other if they have physically met in the real world before participating in MANET operations. If a node, lets say node A, wishes to have a trust relation with node B, which it never physically met before, node A needs to have at least one node in node B's friends list, lets say node C, to authenticate its identity. If there is no node in B's friend list that has physically met node A before, the recommendation request will then be forwarded to the next hop in the same manner. Once a node that knows the identity of node A is found, the information is sent back to node B to complete the authentication process. However, if no one in the chain knows about node A's identity, node A then must name at least one node, lets say node D, that it has met before to act as a reference node. Node B then will do the same process to authenticate node D's identity. If the identity of node D is known by any of node B's friends in the chains, the identity of node A then is considered authenticated. The introduction of referee nodes in their framework is very useful to speed up the security association's establishment process especially in a situation where only a few trust relationships exist in the system. However, their proposal requires strong encryption to be deployed in the system to avoid identity thefts when the recommendation packets travel across the networks. Their proposed framework also seems to cause extra overhead in the networks because of the complicated process in searching recommendations and referees.

Capkun et al. (2003a) also proposed the same approach, but they dropped out the reference mechanism to minimise overhead caused by their proposed framework. They suggested that friendships in the real world could be used to establish trust between two or more mobile nodes that have never met each other in MANET environments. They divided the process to establish trust relationships in their work into two phases. In a first phase, two nodes will be friends to each other when they establish mutual trust between themselves by providing their personal information via a secure side channel (e.g. Infrared). Those two nodes will then exchange appropriate security keys, to enable them to communicate with each other using encrypted messages over radio links. With the encryption facility installed, those two nodes will be able to recommend their friends to each other via radio communication, which will be much faster than communication via a secure side channel as they are not required to get close to each other to ensure secure communication. They claimed that a mobile characteristic of MANET nodes help in their proposed friendship concept, which is important for the overall performance of their authentication mechanism. However, there is one thing missing in their proposed friendship concept. There is no collaborative effort from each node to create a trusted community in MANET environments. Since the recommendation process will only take place when there is a need to authenticate an anonymous node, the process of creating a trusted community is a responsibility of each node itself.

The trust framework proposed in this research is based on the two earlier works mentioned above. It is designed to best suit the MANET environment by considering several aspects such as resources constraints, self-organisation, security, scalability, and the simplicity of the process. It also provides a platform for nodes to exchange their security associations with other trusted nodes in the system, which then leads to the creation of a trusted community. The detailed design of the proposed framework is discussed in the next section.

6.3 Trust Framework

The main focus of the proposed framework is to provide a platform for nodes to exchange their security associations with other trusted nodes in the network. By providing such platform, it is hope that more security associations could be established, especially between anonymous nodes, which then lead to the creation of a trusted community. Security association in autonomous networks such as MANET could be established based upon nodes' initial trusts. Initial trusts between nodes exist via several ways, including based on the friendships of the bearer (i.e. human) in a real world, or based on the good reputation of other nodes through experiences (Walsh & Sirer, 2006). Each method has its own advantages and limitations. For instance, initial trust based on a real world friendship is more relevant than that established based on nodes' experiences at the early stages of the proposed framework implementation. This is because in such situation, each node is very unlikely to have sufficient knowledge/experience about other nodes, thus will not be able to rate other nodes' reputations. Initial trust based on reputation is more suitable at the later stages when sufficient experiences have been gathered. Perhaps the combination of the two methods could result in a better performance. However, for simplicity, only initial trust based on a real world friendship is implemented in this study to show how a trusted community could be created in MANET environments. This section discusses how initial trust could be exchanged in MANET environment, as well as the important concepts behind the proposed friendship framework.

6.3.1 Initial Trust

In most existing MANET's trust frameworks, researchers claimed that security associations between nodes could be established based on the initial trusts that have been setup beforehand. Although they mentioned about how the setup could take place (e.g. via a secure side channel when two nodes are adjacent to each other), in most cases they did not address what motivates the nodes to create such relationships. The proposed trust framework in this research suggests a human (node's bearer) relationship is one of the factors that could motivate initial trust establishment between nodes.

People do not live in this world alone. They socialise, make friends, live in a neighbourhood, and have family. Some people find that their family members are the group of people that they can trust the most. Some others might think differently. In some cases, friends could be the ones that are more trustworthy than a family member. The issue of trustworthiness is very subjective and it depends upon how the relationship is developed (Castelfranchi & Falcone, 1998). However, the issue of how the trust relationship is being developed between individuals is not the main focus of this study. What needs to be highlighted here is that everyone has their own sets of friends. Within a set of friends, there might be a few of them that could be trusted, and vice versa. The above statements are supported by a series of surveys conducted in 1986 and 1995 by a group of researchers in Great Britain (Britsocat, 1995). The surveys, which were conducted for the British Social Attitudes Survey Series (Britsocat), revealed that the average Briton has 14 close friends. One might thinks that the figure is obsolete because the latest survey was conducted 10 years ago. However, the number might be slightly higher than 14 as communication nowadays are much simpler with e-mail, instant messenger, and mobile phones technologies (Frean, 2003). This real world friendship could be a very good basis to setup initial trust between nodes. In this study, it is assumed that each mobile node inherits all the friendship relations established by its bearer and uses them as a basis to establish initial trust with other nodes. Figure 6-1 illustrates how this process could happen in MANET environments.

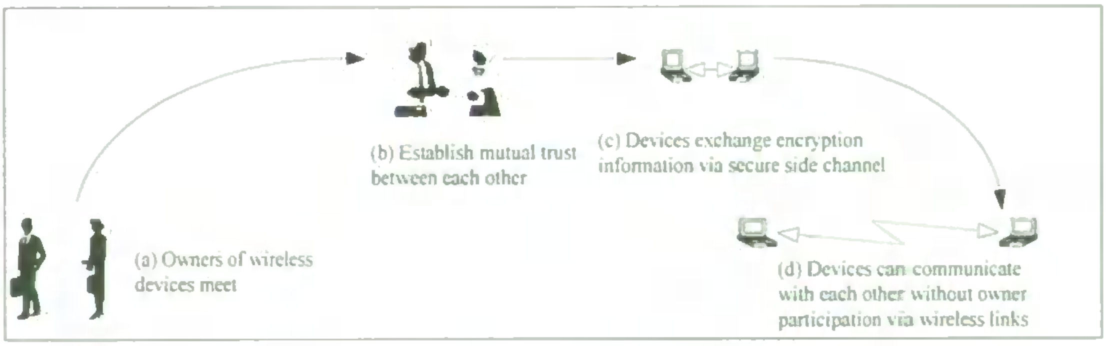

(a) Owners of wireless devices meet
(b) Establish mutual trust between each other
(c) Devices exchange encryption information via secure side channel
(d) Devices can communicate with each other without owner participation via wireless links

Figure 6- 1: Trust Establishment between Nodes Based on Real World Friendship

In Figure 6-1, it is shown that initial trust is established bidirectionally between two nodes. However this does not always happen in a real world. Initial trust is not always necessarily to be established bidirectionally. In some cases, initial trust is established unidirectionally. For instance, a person called A might be a friend to a person called B, and vice versa. In a unidirectional trust relationship, A might believe that B is trustworthy, but not the other way round. This is because trustworthiness is a very subjective issue and it depends upon each individual consideration whether to trust or not to trust any other person. This subjective trust relationship creates an advantage in the proposed trust framework as it increases the number of initial trusts that could exist in the networks. For instance, as shown in Table 6-1, node A could establish a unidirectional trust relationship with node B without node B's approval, as it is its right to trust node B. However, such a scenario will not happen in a bidirectional trust establishment as both node A and node B need to agree on the relationship prior to its establishment.

Nodes	Two Trusted Individuals	Unidirectional Initial Trust	Bidirectional Initial Trust
A	B & C	A → B, A → C	A ↔ C
B	C & D	B → C, B → D	B ↔ D
C	D & A	C → D, C → A	C ↔ A
D	A & B	D → A, D → B	D ↔ B
	Total Relationships	**8**	**4**

Table 6- 1: Unidirectional vs. Bidirectional Trust Establishment

For the case in Table 6-1, each node is assumed to have two initial trusted friends. However, in a real MANET implementation, it is difficult to estimate the number of trusted friends owned by each. One of the reasons is because MANET could exist in several environments. Each environment has its own characteristics (i.e. different kind of users, varies in network's density and coverage), which lead to various number of trusted friends each node could have.

In the proposed framework, it is assumed that this number could vary between 0 and 14 (i.e. the average number suggested in Britsocat survey (Britsocat, 1995)) depending upon which environment the network is deployed. For instance, users in a university campus environment might have more trusted friends than users operating in a city environment due to the fact that more friendships could be established between course mates.

6.3.2 Trust Chain and Recommendation Concepts

Considering a MANET with 4 nodes and having unidirectional trust relationships as shown in Table 6-1, there is a possibility for each node to add another node to its trusted lists. In such case, node A could add node D to its trusted lists, node B could add node A, node C could add node B, and node D could add node C to its trusted lists. Node A might not consider to add node D to its trusted list in the first place because it needs more time to ensure node D's trustworthiness. This is a case for 4 nodes, which does not require much time and efforts for nodes to build their own trusted lists. However, in a wider and/or denser MANET environment, each node might require a little help from other nodes in the networks to build its own trusted list.

The proposed framework in this research suggests the concept of a trust chain via friends' recommendations to help each node build its own trusted list. Based on a scenario as illustrated in Table 6-1, with a trust chain concept in place, node A this time could rely on a recommendation from node B and/or node C to establish a security association with node D. This without doubt could save node A's time and effort in the process of building its own trusted list. Table 6-2 shows how more initial trust between nodes could be established via the trust chain and recommendation concepts.

Nodes	Two Trusted Individuals	Unidirectional Initial Trust	Self + Shared Established Initial Trust
A	B & C	A → B, A → C	A→ B, A→ C, A →B/C →D
B	C & D	B → C, B → D	B→ C, B→ D, B →C/D →A
C	D & A	C → D, C → A	C→ D, C→ A, C →D/A →B
D	A & B	D → A, D → B	D→ A, D→ B, D →A/B →C
	Total Relationships	**8**	**12**

Table 6- 2: Trust Sharing between MANET Nodes

The concept of trust sharing in this paper is motivated by a research finding published by Milgram (1967). The author introduced a small world phenomenon concept, which suggests any two individuals selected randomly from almost anywhere in this world, are connected via a chain of no more than six acquaintances (often referred to as six degrees of separation (Guare, 1990)). The author brought this concept into a discussion in 1967 with an experiment in which he sent 60 letters to various recruits in Wichita, Kansas who were asked to forward the letter to the wife of a divinity student living at Cambridge, Massachusetts. The letters could only be forwarded by hand to personal acquaintances (directly or through a friend of a friend) who they thought might be able to reach the recipient. He claimed that he has proved the concept when 3 out of 60 letters that he sent reached the recipients but neglected to say about the low (i.e. 5%) chain completion percentage. However, his experiment has motivated other researchers to investigate more on this concept, such as in the Internet context, as observed by Adamic (1999). In that study, the author suggested that the World Wide Web is a 'small world' in a sense that all the sites are highly clustered yet the path length between them is small.

The concept of small world phenomenon has been brought into discussion in a wireless network by Helmy (2003). His study was based on findings from Watts and Strogatz (1998), where the authors proposed that by adding a few random links in the system, the average path length between nodes could be reduced dramatically. These few random links could be made available in the ad hoc networks by adding a few 'short cut' nodes in the system. Simulation results from his study proved this hypothesis. One question emerging from this study is how to select the few 'short cut' nodes in an autonomous, fully distributed, and self-organised ad hoc network. The author proposed the concept of *contacts*, which will act as short cuts to transform the wireless network into a small world. However, the author did not discuss how these *contacts* can be made available in the system, and this problem remains an open issue. That is the reason the friendship concept is being introduced in this study. It acts as a useful *contact* to create a relationship between two or more anonymous nodes, thus enables more interactions/communications in the networks.

The deployment of a friend as a useful contact is also supported by a research finding in (Capkun et al., 2003a), where the authors claimed that a mobility characteristic of MANET could create more interactions between nodes. Table 6-3, 6-4, and Figure 6-2 illustrate an example how this could happen in the proposed trust framework. In this example, all nodes in the network have a set of initial trust that they have established offline via physical meetings. This kind of relationship is known as a direct trust between nodes and the relationships are as listed in Table 6.3.

Nodes	Initial Trust / Direct Friend
A	(A-B) and (A-G)
B	(B-F) and (B-D)
C	(C-H) and (C-B)
D	(D-J) and (D-C)
E	(E-G) and (E-H)
F	(F-I) and (F-D)
G	(G-J) and (G-A)
H	(H-C) and (H-I)
I	(I-F) and (I-E)
J	(J-E) and (J-A)
Total Relationships	**20**

Table 6- 3: Initial Trust Relationships between Nodes

Each node carries and exchanges its direct friend list between them to create a new set of friend list, namely the indirect friend. For instance, with a first nodes' movement as illustrated in Figure 6-2(a), the newly created indirect friend list is as illustrated in Table 6-4(a).

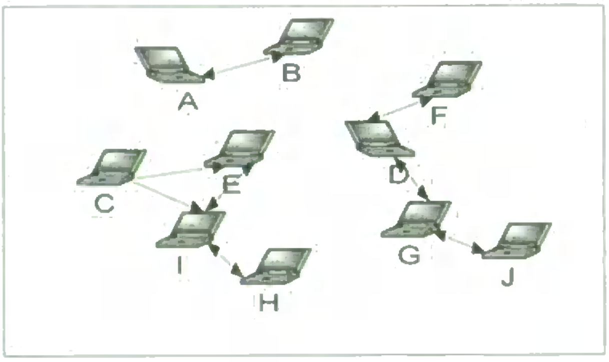


Figure 6- 2(a): Node's Locations and Interactions in 1st Movement

Nodes	Initial Trust	Additional Trust Relationships in 1st Movement	Trust Relationships after 1st Movement
A	(A-B)(A-G)	(A-B-F)(A-B-D)	(A-B)(A-G)(A-F)(A-D)
B	(B-F)(B-D)		(B-F)(B-D)
C	(C-H)(C-B)		(C-H)(C-B)
D	(D-J)(D-C)		(D-J)(D-C)
E	(E-G)(E-H)		(E-G)(E-H)
F	(F-I)(F-D)	(F-D-J)(F-D-C)	(F-I)(F-D)(F-J)(F-C)
G	(G-J)(G-A)	(G-J-E)	(G-J)(G-A)(G-E)
H	(H-C)(H-I)	(H-I-F)(H-I-E)	(H-C)(H-I)(H-F)(H-E)
I	(I-F)(I-E)	(I-E-G)(I-E-H)	(I-F)(I-E)(I-G)(I-H)
J	(J-E)(J-A)		(J-E)(J-A)
Total Relationships	**20**	**+ 9**	**29**

Table 6- 4(a): Trust Relationships Establishment in First Movement Scenario

The second column in Table 6-4(a) represents the initial trust relationships owned by each node at the early stage of the implementation. Nodes in a MANET move randomly and communicate with the neighbouring nodes. If the neighbouring node is one of its direct friends, it will receive recommendations from the friends to add other trustworthy nodes to its own trustworthy lists. The third column in Table 6-4(a) shows additional trust relations (+9 from initial 20) that have been established after the first mobility scenario. The number of trust relationships between nodes keeps growing (10 more relationships) after the second mobility scenario as shown in Figure 6-2(b) and Table 6-4(b).

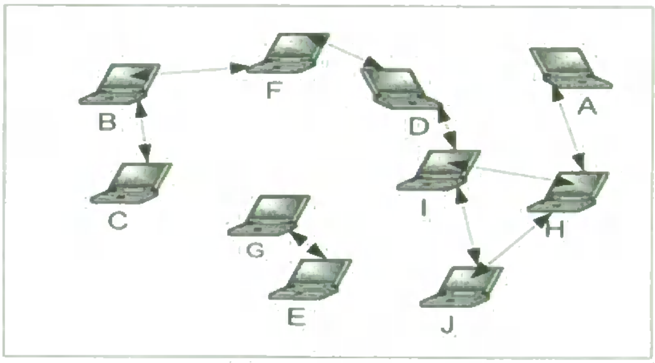


Figure 6- 2(b): Node's Locations and Interactions in 2nd Movement

Nodes	Trust Relationships after 1st Movement	Additional Trust Relationships after 2nd Movement	Trust Relationships after 2nd Movement
A	(A-B)(A-G)(A-F)(A-D)		(A-B)(A-G)(A-F)(A-D)
B	(B-F)(B-D)	(B-F-I)(B-F-J)(B-F-C)	(B-F)(B-D)(B-I)(B-J)(B-C)
C	(C-H)(C-B)	(C-B-F)(C-B-D)	(C-H)(C-B)(C-F)(C-D)
D	(D-J)(D-C)		(D-J)(D-C)
E	(E-G)(E-H)	(E-G-J)(E-G-A)	(E-G)(E-H)(E-J)(E-A)
F	(F-I)(F-D)(F-J)(F-C)		(F-I)(F-D)(F-J)(F-C)
G	(G-J)(G-A)(G-E)	(G-E-H)	(G-J)(G-A)(G-E)(G-H)
H	(H-C)(H-I)(H-F)(H-E)	(H-I-G)	(H-C)(H-I)(H-F)(H-E)(H-G)
I	(I-F)(I-E)(I-G)(I-H)	(I-H-C)	(I-F)(I-E)(I-G)(I-H)(I-C)
J	(J-E)(J-A)		(J-E)(J-A)
Total Relationships	**29**	**+10**	**39**

Table 6- 4(b): Trust Relationships Establishment in Second Movement Scenario

6.3.3 Assumptions

The reliability of the proposed trust framework is based upon the following assumptions.

- *Bearer's Unique ID*

This study assumes that node's bearer has a unique ID, which is used by other nodes to identify his/her identity. Such an ID could be in form of a smart card, biometrics devices, or any tamper resistant hardware to minimise the effects of identity theft (Hubaux et al., 2001). This study also assumes that an authentication mechanism has been deployed in the first place to assist in the authentication process. Several authentication mechanisms for MANET environments are available to make the above assumptions valid (as reviewed in the earlier chapter). One of the good examples is an ID-based cryptosystem as proposed by Khalili et al. (2003).

- *Spoof-proofed ID*

This study assumes that the bearer's ID is resistant from being masqueraded or impersonated by the attackers. Identity impersonation attack is not a MANET-only issue, and several approaches have been proposed to ease the problem in other types of wireless networks as well as in wired networks (Huang et al., 2003a; Barbeau et al., 2006). In the MANET case, Gwalani et al. (2004) have proposed a unique way to detect MAC address spoofing attacks. In this approach, the authors suggested that by identifying anomalous changes in the generation of sequence numbers for a specific MAC address, it is possible to detect MAC spoofing. IEEE 802.11 sequence numbers can be modified only by changing the firmware of the wireless card, thus making such an approach fairly reliable. If a unique MAC address of the wireless card together with the sequence numbers could be used to detect MAC spoofing, a similar approach also could be used to detect bearer's identity spoofing as illustrated in Figure 6-3. In such a case, each node records the sequence number and the sender ID pair for every RREQ (Route Request). Any significant difference between current and previous pairs will trigger an alert.

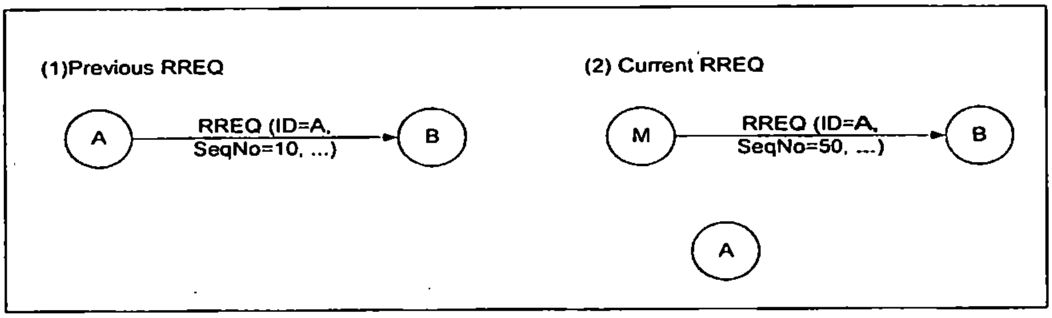


Figure 6- 3: ID Spoofing Detection

6.4 Key Features of the Proposed Framework

As mentioned earlier, this study is not the first to suggest the concept of friend's recommendations to establish trust in a MANET environment. A similar concept has been proposed in (Weimerskirch & Thonet, 2001; Capkun et al., 2003a) as discussed earlier in this chapter. The aims of this study are to provide solutions to several unaddressed issues in previous works, as well as to suggest some enhancements that could be made to provide a reliable trust relationship framework for a MANET.

6.4.1 Light Weight

One of the main concerns in MANET operations is the node's limited resources (Salem et al., 2003). The proposed framework in this study minimises the problem by not incorporating heavy computational mechanisms that might increase the network's operational overhead. For instance, friend lists could only be exchanged between nodes in a single hop communication, which eliminates the need for complex authentication mechanisms if exchange is permitted over multihop links. Complex authentication mechanisms utilise more network resources, especially for computational purposes. In another case, although it is true that a reference concept as suggested in (Weimerskirch & Thonet, 2001) could speed up the trust establishment process, such a concept is being avoided in this study as its complexity would increase the network's operational overhead. On the other hand, this study proposes trust chain and recommendation concepts, which it is believed could offer a better trade off between trust establishment speeds and network operational overhead.

6.4.2 Self-organisation

Capkun et al. (2003a) used a CA to improve the performance of their trust framework. However, as MANET operates in a self-organised manner without any central point to perform the administration, it is very useful if the assumption of CA existence is avoided. This study introduces a constant friend recommendations concept, which could lead to a trust chain establishment to improve the performance of the proposed trust framework. A pair of friend nodes will exchange their trusted friend list whenever they are in range to each other (i.e. one hop away). Although this approach could not match the performance of a trust framework assisted by a CA, the results from simulations presented later in this chapter show that the number of trust relationships could be significantly increased.

6.4.3 Security Issues

Operating without any CA creates challenges for each node in MANET to deal with security issues. Each node is responsible to protect itself from various passive and active attacks (Al-Jaroodi, 2002). The trust framework proposed in this study is designed with this issue in mind to ease the burden for each node in dealing with security issues. Since two nodes are required to have a physical meeting before they could establish trust relationships with each other, an issue about identity theft could be eliminated. Another important security issue that could be eased by the friend concept introduced in this study is the problem of a blackmail attacker. With an assumption that all the trusted nodes are behaving properly, the friendship concept could act as a filter for nodes to avoid receiving false accusations from the blackmail attackers.

6.4.4 Scalable

Scalability is not an issue in the proposed trust framework. Unlike in (Capkun et al., 2003a), no meeting point is required in the proposed trust framework. A meeting point might boost up the establishment of trust relationships, but with a large size of networks, this might cause some problems. For instance, in a large network environment, nodes might need to travel a long distance to reach the meeting point, and in a dense environment, the meeting point might be crammed full. Besides, a CA also might be required to ensure security at the meeting point, which if not taken seriously could be abused by the attackers.

The proposed friendship concept is expandable to a worldwide scale. For instance, considering each node has 50 trusted friends, which were established directly on indirectly via recommendations, as discussed in previous sections. With the concept of six degrees of separation as discussed in Section 6.3.2, each node could have up to 15 943 877 550 (i.e. 50 + 2500 + 125000 + 6250000 + 312500000 + 15625000000) trusted individuals worldwide. The relationships are as illustrated in Figure 6-4.

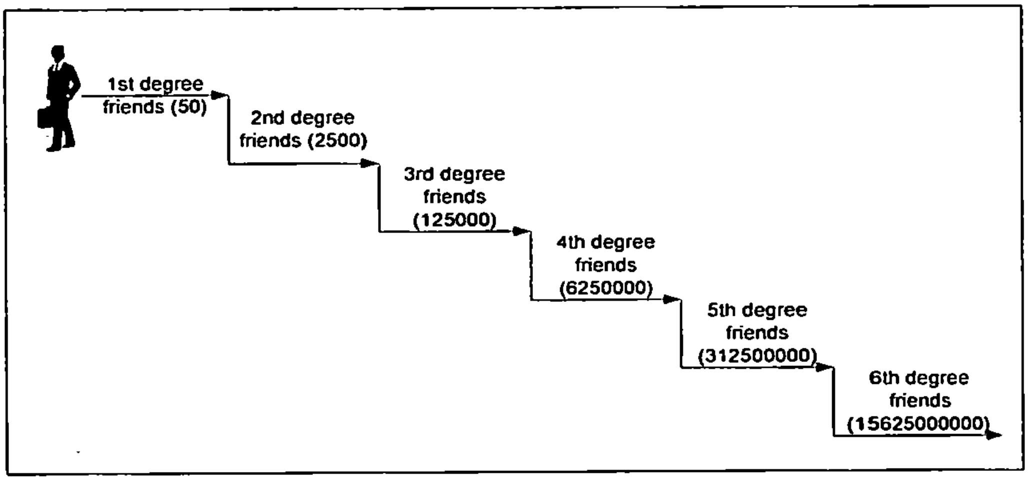

1st degree friends (50)
2nd degree friends (2500)
3rd degree friends (125000)
4th degree friends (6250000)
5th degree friends (312500000)
6th degree friends (15625000000)

Figure 6- 4: Virtual Trust Relationships in Six Degrees of Separation Concept for the case of 50 Initial Friends

6.5 Evaluating the Performance of the Proposed Framework

Evaluating a new MANET theoretical system or framework is not easy. One of the big challenges, especially in the case of this study, is to set up a huge amount of mobile nodes to represent a standard MANET environment. The applicability of the trust framework cannot be seen with the interactions of a small number of mobile nodes. With such requirement, although it is always a desire to use an implementation-based approach, a simulation-based approach is often more practical. Moreover, a simulation has been chosen as a preferred approach by previous researchers as it is useful to support and demonstrate node's mobility patterns in MANET environments (Kurkowski et al., 2005).

There is a concern in the research community about the credibility of MANET simulations. Andel and Yasinac (2006) brought this issue into a discussion and suggested that errors in simulation models or improper data analysis often produce incorrect or misleading results. However, this is not a big issue in this study, as no critical simulation attributes (e.g. traffic patterns, data transfer rate, or communication disruptions) are directly involved in the simulations, which if not carefully selected could lead to a misleading result. In fact the main reason for using a simulation to evaluate the performance of the proposed framework is because it is more practical considering the huge amount of nodes that are involved in this study.

The following sub-sections describe simulation experiments, which aim to evaluate the performance of the proposed trust framework in creating a trusted community in MANET environments.

6.5.1 Network Simulator 2 (NS-2)

There are several simulation tools available to evaluate new MANET theories or frameworks (Andel & Yasinac, 2006). One of them is the NS-2, which is used in this study. NS-2 is a DARPA-supported discrete event simulator that is suitable for both wired and wireless networks. Since its first introduction in 1989, several versions of the software have been released, with substantial contributions from the research community. This study uses the NS-2 simulator version 2.29, where its core MANET module (i.e. AODV routing protocols) has been developed by researchers from Carnegie Mellon University (CMU) Monarch project (Monarch Project, 2000). The decision to use NS-2 as the simulation tool has been made due to the fact that it has become the most popular simulation tool within MANET research community (Kurkowski et al., 2005), thus is providing an increased level of confidence in the reliability of the simulation results. However, this does not mean that other simulation tools (e.g. GloMoSim (Bajaj et al., 1999) and OPNET (OPNET, 2006)) are not suitable for this study. Simulation metrics evaluated in this study are independent from the design of any simulation tools, thus the end results from any simulation tool are not expected to differ considerably from one another. An overview on NS-2 and its main components can be found in (Chung & Claypool, 2005).

6.5.2 Simulation Setup

This study investigates the performance of the proposed trust framework in 3 different MANET settings, which differ from each other based on the network's density level. The differences in network density and coverage provide some means to investigate the applicability of the proposed framework in several MANET environments. For better understanding, these 3 MANET settings are classified into 2 open MANET environments, namely university campus and city network. The first setting, which is the densest setting, is represented by the university campus environment. The other 2 settings with 2 times and 4 times less dense than the first settings are represented by the city-1 and city-2 network environments, which usually have a wider network terrain. Example scenarios for both environments are as follow.

- *University Campus*

This setup represents an open MANET environment with high network density level within a small coverage area. In this environment, nodes are expected to interact more often with each other because of the limited space in the university campus. An example of such scenarios is where students carry their laptops to the library, lecture halls, or use them in their halls of residence. A high number of direct trust relationships are expected in this environment as the students make friends with their course mates, trust their lecturers and have contacts with university authorities.

- *City Network*

This setup represents an open MANET environment with a lesser density level within a bigger network coverage area than the university campus. The situation can be seen as the city community can communicate with each other when they are shopping in the city centre, meeting at coffee shops, having drinks at pubs, or even communicating with relatives and neighbours etc. The number of mobile nodes, as well as the direct trust relationships in this environment, could be higher or lower than the university environment depending on the real word relationships within the community. However, for better comparison, the number of node's relationships used in this setup is set as similar as the number being used in the university campus setup. Through this setting, the applicability of the proposed framework in different network density level could be investigated.

The differences in network density level represent one of the factors that might have an impact on the proposed framework's overall performance. This factor is combined with the other two factors, namely the size of nodes' initial trust relationships and simulation time (representing network's age) in two separate simulation setups, as illustrated in Table 6-5(a) and 6-5(b), to evaluate the overall performance of the proposed trust framework. The maximum number of possible initial trust relationships owned by each node in the simulation experiments is set to 14, as suggested in the Britsocat survey (Britsocat, 1995).

University Campus	Set 1	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	1	0.5km^2	200.0s
	Set 2	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	3	0.5km^2	200.0s
	Set 3	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	0.5km^2	200.0s
	Set 4	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	10	0.5km^2	200.0s
	Set 5	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	14	0.5km^2	200.0s
City Network (2x less)	Set 1	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	1	1km^2	200.0s
	Set 2	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	3	1km^2	200.0s
	Set 3	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	1km^2	200.0s
	Set 4	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	10	1km^2	200.0s
	Set 5	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	14	1km^2	200.0s
City Network (4x less)	Set 1	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	1	2km^2	200.0s
	Set 2	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	3	2km^2	200.0s
	Set 3	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	2km^2	200.0s
	Set 4	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	10	2km^2	200.0s
	Set 5	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	14	2km^2	200.0s

Table 6- 5(a): Simulation Setup A – to Investigate Nodes' Initial Trusts Influence in Various Network Density Level

University Campus	Set 1	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	0.5km^2	50.0s
	Set 2	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	0.5km^2	100.0s
	Set 3	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	0.5km^2	150.0s
	Set 4	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	0.5km^2	200.0s
	Set 5	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	0.5km^2	250.0s
City Network (2x less)	Set 1	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	1km^2	50.0s
	Set 2	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	1km^2	100.0s
	Set 3	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	1km^2	150.0s
	Set 4	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	1km^2	200.0s
	Set 5	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	1km^2	250.0s
City Network (4x less)	Set 1	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	2km^2	50.0s
	Set 2	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	2km^2	100.0s
	Set 3	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	2km^2	150.0s
	Set 4	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	2km^2	200.0s
	Set 5	Total Nodes	Friend Nodes	Terrain Size	Simulation Time
		100	5	2km^2	250.0s

Table 6-5(b): Simulation Setup B – to Investigate Network's Age Influence in Various Network Density Level

Apart from that, both simulation sets also follow general MANET simulation settings, as usually found in previous works (Kurkowski et al., 2005; Andel & Yasinac, 2006). Those settings are as follows:

- *Mobility Pattern*

Nodes are set to move from one location to another using a *random waypoint model* (Hyytia, 2005) (i.e. generated using *setdest* node-movement generator, produced by CMU research) with *10.0 seconds pause time* and *30.0 seconds maximum speed.* A copy of complete nodes' movements can be found in a CD supplied with this thesis.

- *Transmission Range*

Nodes are restricted to share their friendship relations with other nodes within *100 metres single hop communication range.* This could be done by setting up the RXThresh_ of each mobile node to *1.42681e-08.* Steps to change mobile node's transmission range can be found in (Ke, 2006). The use of a multi hop trust sharing is avoided in the proposed framework to minimise the impact of a *man in the middle attack* (Ornaghi & Valleri, 2003).

- *Simulation Runs*

The reliability of the end results is achieved by repeating each simulation set with *5* different sets of direct friends on each run. These direct friends are equally assigned to each node using a *random number generator.* A complete list of nodes' initial relationships for each simulation run can be found on the CD.

6.5.3 Simulation Metrics

The aim of the proposed trust framework is to expand the initial trusts that are owned by each node to create a trusted community in MANET environments as soon as possible. Whilst moving from one location to another, nodes exchange their initial trust relationships with their trusted friends, thus creating a set of indirect trust relationships. This study investigates how many indirect trust relationships could be established in the proposed trust framework whilst taking into account several factors that could affect its performance. Such factors include:

1. The effects of the network's age towards the overall percentage of the established trust relationships. This could be done by investigating the number of established indirect trust relationships against various simulation times.

2. If different size in nodes' initial trusts will have an impact on the trust framework's overall performance.

3. Will the proposed trust framework be suitable for any kind of MANET environment, which differ in network density and coverage area?

All the above factors formed several simulations sets as illustrated in Table 6-5(a) and 6-5(b), which are then used to measure the overall performance (*T*) of the proposed trust framework. The overall performance of the proposed framework is based upon the percentage of a trusted community created at the end of each simulation set, which is calculated using the following formula:

$$T = \frac{\sum_{i=1}^{n}(x+y)}{n(n-1)} \text{ (100\%), where } x = \text{direct trust; } y = \text{indirect trust; } n = \text{total nodes}$$

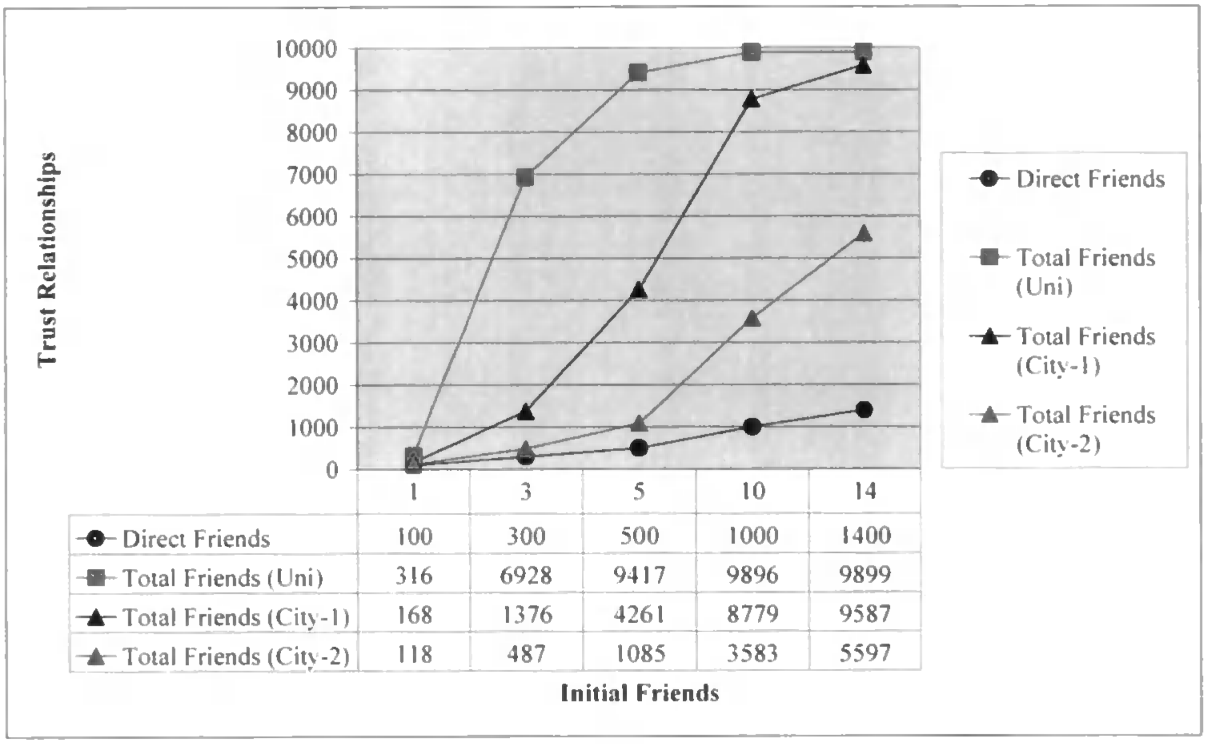
Formula 6- 1: Formula to Calculate the Proposed Trust Framework Performance

6.5.4 Results

This section presents the results and observations obtained from the simulation experiments, which were carried out to investigate the effects of the following factors towards the proposed trust framework's overall performance.

6.5.4.1 The Effects of Nodes' Initial Friendships

Figure 6-5 presents the results from simulation experiments conducted to investigate the effects of nodes' initial friendships towards the proposed friendship framework's overall performance. As described earlier, the result is an average of 5 simulation runs, which were conducted to improve the statistical validity of the experiments (Andel & Yasinac, 2006). The complete results of each simulation run for this experiment can be found in Appendix A. From the results, it can be seen that the number of trust relationships between nodes is increased significantly according to the initial friends owned by each node. For instance, the number of trust relationships established with the help of 1 initial friend is increased from 100 to 316 in the university campus environment. A similar scenario also occurs in the city network environment, where the trust relationships were increased from 100 to 168 and 100 to 118 in both 2 times and 4 times less dense settings. The significant increase of trust relationships in 3, 5, 10 and 14 initial trusts simulation sets confirmed the effects nodes' initial friendships towards the overall performance of the proposed framework



	1	3	5	10	14
● Direct Friends	100	300	500	1000	1400
■ Total Friends (Uni)	316	6928	9417	9896	9899
▲ Total Friends (City-1)	168	1376	4261	8779	9587
▲ Total Friends (City-2)	118	487	1085	3583	5597

Figure 6- 5: The Effects of Node's Initial Trust

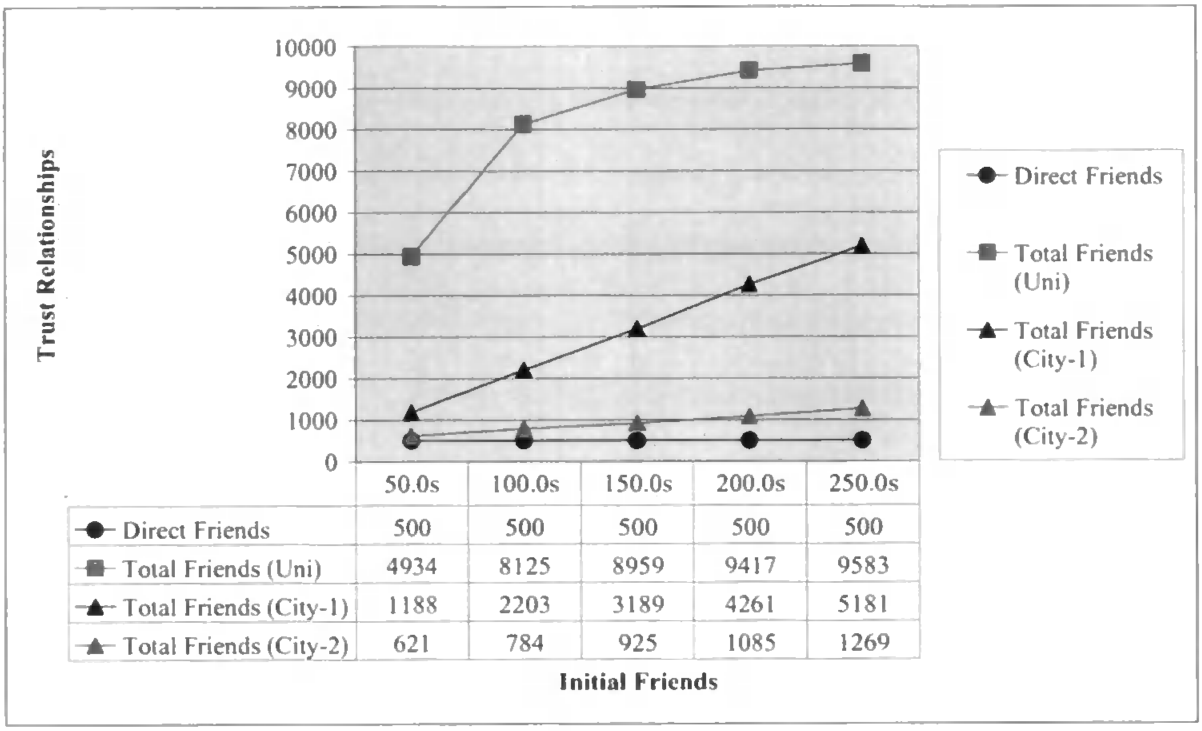
Apart from providing a platform for nodes to expand their trust relationships, the proposed friendship framework is also capable of virtually creating a trusted community in the network as described earlier in this chapter. Results from the experiments justify this claim. As illustrated in Figure 6-5, the percentage of a trusted community in the network could be increased significantly with the help of friends. For instance, in a case of 5 initial friends in university environment, the percentage of a trusted community (*T*) is increased from 5% (without help of friends) to 95% (with the help of friends). The results are even better with 10 and 14 initial friends, where the percentage of a trusted community established in the university environment almost reaches the total 9900 trust pairs (i.e. 100%). In the case of the city environment, although the number of trust relationships created is not as high as in the university environment, the percentage of a trusted community established in both city-1 and city-2 settings are much better than solely depending upon a direct trust establishment.

It is important to mention here that the results in this experiment sets were obtained based on a fixed 200.0 seconds simulation time. The results are expected to be better (especially in the city environment settings) if a longer simulation time were used in the experiments. The next section discusses this issue.

6.5.4.2 The Effects of Network's Age (Simulation Time)

Figure 6-6 justifies that the network's age, which in this study is represented by the simulation time, is one of the important factors that could affect the overall performance of the proposed trust framework. From the results, it can be seen that the number of trust relationships established between nodes were gradually increased over a period of time. By using a fixed number of initial friends (i.e. 5), the total trust relationships established in a university campus environment were gradually increased from 500 to 9583 initial trusts within 250 seconds simulation time – more than 19 times increase. A similar scenario also occurs in the city network environment, where the total trust relationships were increased to 5181 and 1269 in both 2 times and 4 times less dense settings.

In terms of the overall performance (i.e. *T* % trusted community), results from the experiments shown that with a sufficient amount of time, the percentage of a trusted community established in the network could be gradually increased towards the 9900 total trust relationships. For instance, the percentages of a trusted community created in both university and city network environments were increased from 5% (without any help from friend) to approximately 96% in university environment, and 52% in city-1 environment (with the help of friends) within just 250.0 seconds simulation time. In addition, a higher *T* % value could be expected if more initial friendships were used in the experiment sets. For instance, 10 initial friends instead of 5 are very likely to increase the percentage of a trusted community for both university and city network environments. Similar to the previous experiment, results from this experiment is based upon an average of 5 simulation runs. The complete results of each simulation run can be found in Appendix B.



	50.0s	100.0s	150.0s	200.0s	250.0s
● Direct Friends	500	500	500	500	500
■ Total Friends (Uni)	4934	8125	8959	9417	9583
▲ Total Friends (City-1)	1188	2203	3189	4261	5181
▲ Total Friends (City-2)	621	784	925	1085	1269

Figure 6- 6: The Effects of Network's Age

6.5.4.3 The Effects of Network's Density

The reason of having three different MANET settings (i.e. represented by a university campus and 2 city network environments) in the two previous experiments is to investigate if the proposed trust framework could perform well in various MANET environments. Each setup represents a different network density level based on the number of nodes and the size of the terrain. The network density level (*D*) for each environment is obtained using the following formula:

$$D = \frac{n}{km^2}\text{, where } n = \text{total nodes; } km^2 = \text{terrain size}$$

Formula 6- 2: Network's Density Level

Therefore, network's density levels for the university and the 2 city network environments are 0.2n/m2, 0.1n/m2 and 0.05n/m2 respectively. Although with such a difference (i.e. 2 times and 4 times less dense), it can be seen that the total trust relationships between nodes in the two city network environments are still much higher than the trust relationships that have been established solely on each node's efforts (i.e. direct trust). For instance, in the experiment to investigate the effects of nodes' initial trust as discussed in the previous section, although the percentage of a trusted community in the city-1 network is approximately 40% less than the percentage of a university environment (i.e. in a case of 5 initial friends), an approximately 40% increase from initial 5% direct trust is still a good performance.

6.5.5 Discussion

Based on the observations of the simulation results, it is apparent that the proposed friendship mechanism is capable to expand node's initial trust towards the creation of a trusted community. Although the true potential of the proposed trust framework is dependent upon various factors (such as network's age, the number of nodes' initial trust, and network's density level), its performance is still very good in the least optimum scenarios. For instance, although with only a single trusted friend for each node, the number of trust relationships in the network still could be increased to at least 18% for the case of city-2 environment, (refer to Figure 6-5). Moreover, this 18% is achieved from a very immature network (i.e. only 200.0 seconds lifetime) in a less dense environment. The percentage is expected to be higher if the network's lifetime is longer than 200.0 seconds, as shown in Figure 6-6. It is important to mention here that the simulation time (e.g. 50.0, 100.0, 200.0 seconds, etc.) used in the simulation do not represent a real time in real MANET implementation. Various simulation times were used in this study to show that a longer MANET operational time will have an impact towards the overall performance of the proposed framework.

From the observation, the number of direct friends possessed by each individual node has a significant impact towards the overall performance of the proposed trust framework. The more direct friends owned by each node means the better the performance of the proposed trust framework will be. However, as mentioned earlier, this direct friendship relation is a very subjective issue. Some people might have as many as 5, 10, or even more individuals or friends that they could trust. On the other hand, some people might only have 1 or no friend that they could rely on. However, the later case is almost never true since all people in this world live with friends (Britsocat, 1995), especially with the current advancements in todays telecommunication technology (Frean, 2003).

There is also another important factor that was not investigated in the experiment but has a major impact towards the overall performance of the proposed trust framework. Such factor is known as the inconsistent value of nodes' initial trust. It is always possible for each node to establish a new mutual trust with another node in the network. This situation mimics a real world friendship scenario, in which each person could establish a new friendship as he/she desires with any newly met person. Such a situation gives an advantage towards the overall performance of the proposed framework. This factor's influence towards the proposed framework's overall performance was not experimentally investigated in this study because the nature of direct trust establishment process is very subjective. Apart from that, the inclusion of this factor will not decrease any performance of the proposed trust framework, but will absolutely add to the number of indirect trusts in the obtained results.

Results from the simulations also suggested that the proposed trust framework is applicable in various network density levels. Compared with a denser environment (i.e. university campus), the trust growths in the city network environments (i.e. 2x and 4x less dense) are still significant. This suggests that the proposed trust framework is scalable although with a slight decrease on the overall framework performance.

6.6 Conclusions

This chapter has discussed the concept behind the proposed friendship mechanism, which includes discussion on its background and how it could help in MANET operations, particularly in security mechanisms. The applicability of the proposed framework in various MANET environments and scenarios is also justified via a set of simulation experiments. The main conclusion derived from the experiments is that the maximum potential of the proposed trust framework is dependent upon the number of direct friends owned by each node. The more direct relationships established in the network the better the performance the proposed friendship mechanism. In addition, this chapter also addressed the issue of network maturity and density levels toward the overall performance of the proposed trust framework.

Having justified the applicability of the proposed trust framework in MANET environments, the next chapter outlines the detailed architecture of the intrusion detection and response mechanisms. This includes the implementation of the proposed friendship concept, which aims to assist in the global detection and response mechanisms of the proposed two-tier IDS framework.

Chapter 7

Intrusion Detection and Response Mechanisms

7.1 Introduction

As presented in Chapter 5, the proposed IDS framework is divided into two tiers of detection and response mechanisms. In a first tier, each node independently captures and analyses its own audit data to detect any misuses or anomalies in the system. The outputs from the first tier detection will determine the next step to be taken in the second tier mechanisms. If an intrusion is successfully detected in a first tier detection, a local response as well as the global response mechanisms will be triggered. Otherwise, further investigation will be carried out to analyse the suspicious activities, and this time with the help of friends via a global detection mechanism.

This chapter describes a simulation-based implementation of the proposed detection and response mechanisms in both tiers. The aims of the simulations are to demonstrate the applicability of the friendship concept in global detection and response mechanisms, as well as to evaluate the performance of the proposed IDS framework in defending against colluding blackmail attackers. The implementations have been undertaken using the NS-2 simulation tools, as it is the most suitable way to prove the concepts of the proposed IDS framework, in various MANET environments.

7.2 Scope of the Implementation

It is important to mention that this research is not intended to design a comprehensive and fully operational misuse and anomaly IDS for MANET. The aim is rather to propose an IDS framework to show that a friendship relation between nodes could play a major role in improving global detection and response mechanisms for MANET IDS. With that in mind, only simple operational misuse and anomaly detection engines based on existing works are implemented in this study.

The capability of the detection and response mechanisms implemented in this study are confined within the following boundaries.

7.2.1 Detection

Although there are many variations of attacks that could be launched against a MANET as discussed in Chapter 3, only attacks that associated with an AODV routing protocol are further analysed in this study. The decision was based upon the following reasons:

- The routing protocol is the most important basic mechanism in a MANET, thus it is very important to address its security threats (Hubaux et al., 2001).

- An AODV is one of the most popular routing protocols in MANET research field (Perkins & Royer, 1999). Its architecture is constantly reviewed and updated by researchers with aims to improve its capability in terms of the performance as well as the reliability to cope with security issues (Jung et al., 2005b; Zapata & Asokan, 2002; Bhargava & Agrawal, 2001). However, this does not mean that the proposed IDS framework is not suitable for other routing protocols (e.g. DSR, DSDV). This is because the reliability of the global detection and response mechanisms (i.e. the focus of this study) is independent from the routing protocol implemented in the networks. A complete specification of an AODV routing protocol can be found in (Perkins et al., 2003).

7.2.2 Response

In the proposed IDS framework, each node responds to intrusions in two steps. At first, each node triggers a local response mechanism every time its local detection mechanisms detect an intrusion. Consecutively, the local alerts are transmitted to other nodes in the networks via a global response mechanism so that they are aware about the existence of an intrusive node, and take subsequently necessary actions to deal with it. Responsive behaviour of each node in a local response mechanism implemented in this study is set to be very minimal. This is because although it is a desire to provide a mechanism for each node to punish the misbehaving nodes (e.g. exclude them from participating in networks' operations), such an issue is not a main focus of this study. The focus of this study is on the more prominent problem of response mechanisms for MANET IDS, which is to propose a reliable method that enables each node to securely share intrusion alerts with other nodes, especially in the presence of blackmail attackers. Responsive behaviours implemented in this study are carried out in two ways, namely a local response and a global response.

- *Local Response*

  Once received an alert from its local detection mechanisms, each node will add the intrusive node to its own set of bad node list. In an ideal situation, nodes are supposed to avoid any communication with the bad nodes. This feature will be addressed in a future work.

- *Global Response*

  Each node shares its own bad node list with its friends. Sharing is only allowed within a single hop communication range to avoid the information being modified by the malicious intermediate nodes. To prevent false accusations caused by blackmail attackers, intrusion alerts are restricted to be shared between friend nodes.

In addition, not every module of the proposed IDS framework is fully designed and implemented in the simulation software. For instance, misuse and anomaly detection mechanisms are partly implemented as the architectures of those modules have been designed and revised in several existing works. The signature management mechanism module on the other hand has not been implemented as the impacts of the proposed friendship concept towards the performance of such mechanism are similar with the performance of the trust management mechanism module. Figure 7-1 illustrates which modules were largely implemented, partly implemented, and which were not.

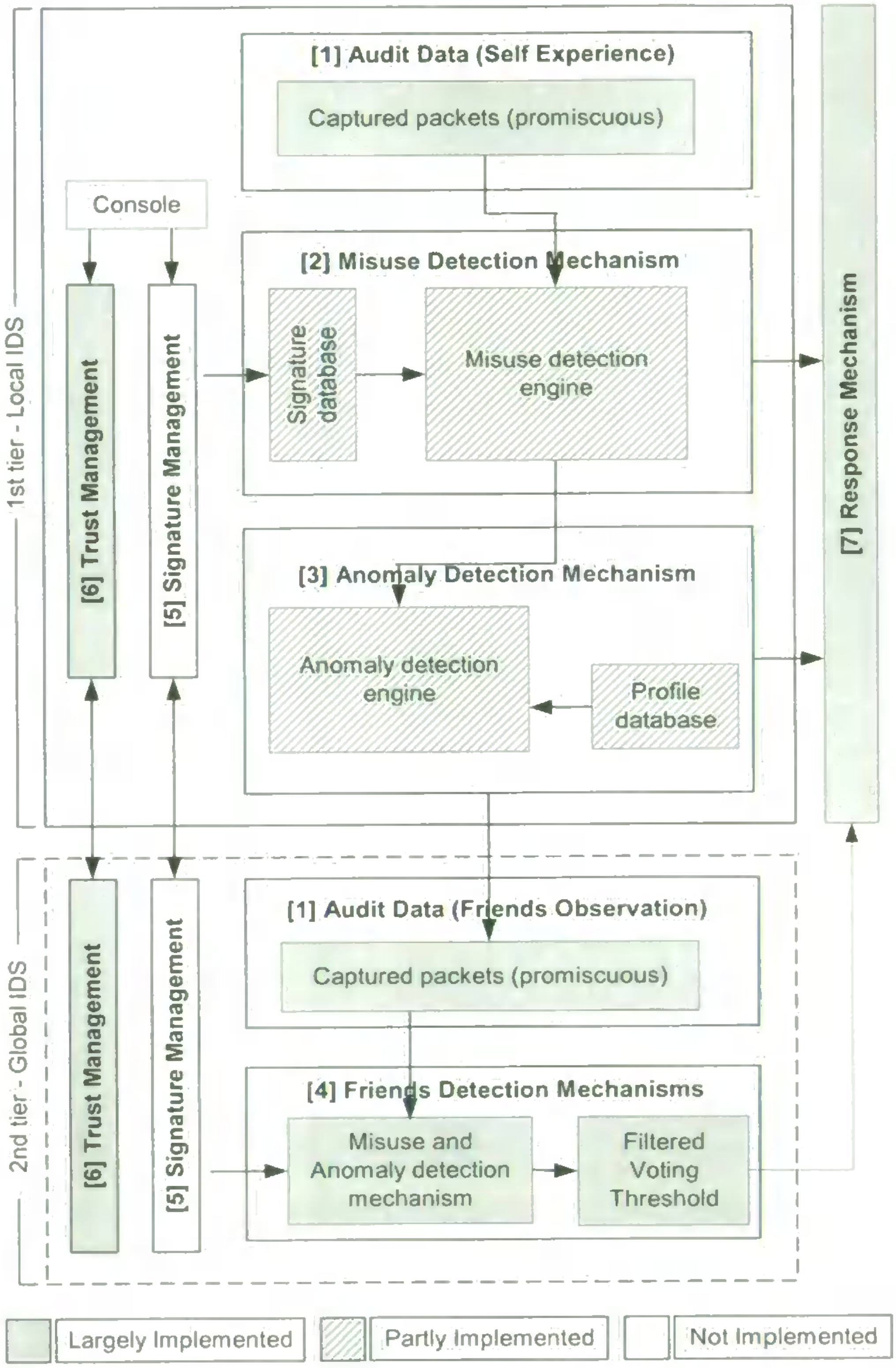


Figure 7- 1: Implementation of the Proposed IDS Framework

7.3 Local Detection and Response

MANET's characteristics demand each node in the network to play its role in detecting intrusion attempts. The proposed IDS framework in this study provides a platform for each node to fulfil such needs via the local detection and response mechanisms. This section discusses a simulation-based implementation of each module associated with the local detection and response mechanisms.

7.3.1 Audit Data Sources

Audit data is the first module to be called in the local detection and response mechanisms. It is classified into two categories. The first category is real time audit data, which is captured promiscuously and used by the misuse detection engine. The second category of the audit data is a set of recorded inbound and outbound packets, which also being captured promiscuously but used by the anomaly detection engine. The audit data is pre-processed into these two categories to suit the needs of both misuse and anomaly detection strategies, which will be explained further in the next section. Figure 7-2 illustrates how the audit data is promiscuously captured and simultaneously passed to the misuse and anomaly detection engines.

Although the proposed IDS framework categorised the audit data into two categories, both of the audit data types are actually coming from the same source. As illustrated in Figure 7-2, each node logs the overheard inbound/outbound packets of its neighbouring nodes, so that the information could be used later by its anomaly detection engine. Simultaneously, the overheard packets are fed into the misuse detection engine to detect any misuse activities.

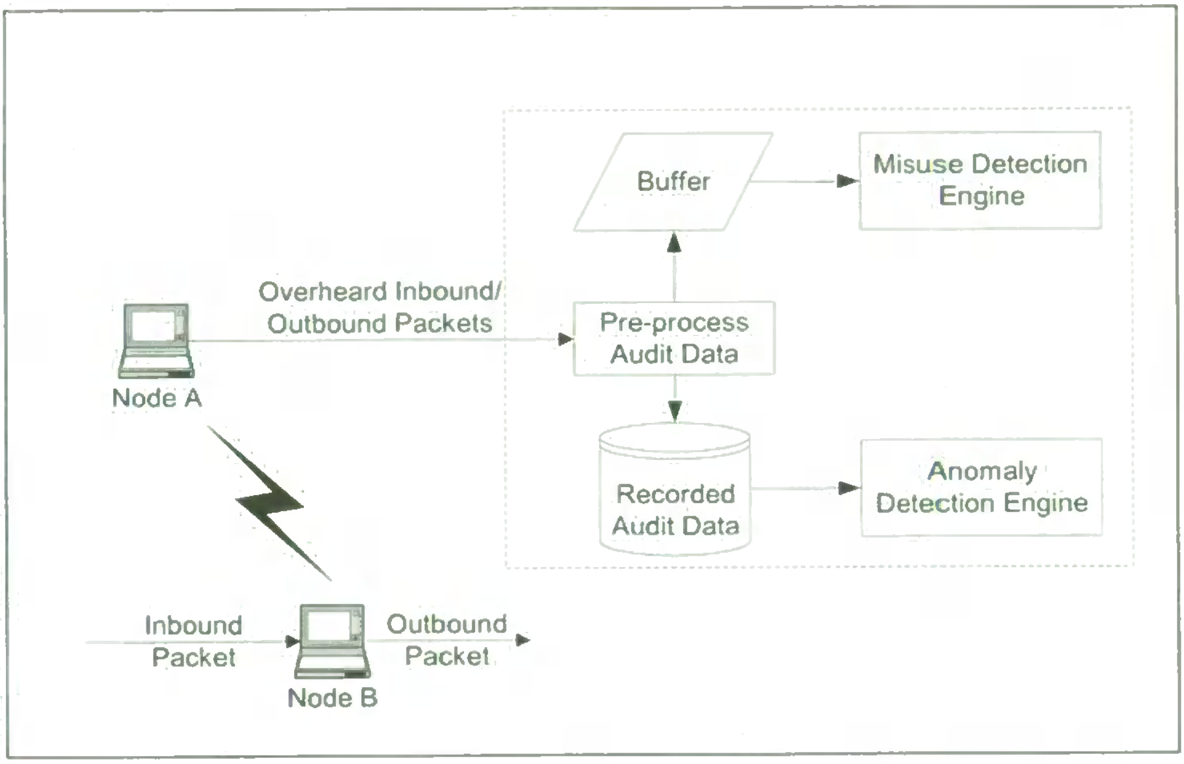


Figure 7- 2: Audit Data being Captured and Passed to the Detection Engines

A module called *logaodvpacket* is added to the NS-2 to capture the real time audit data. It captures RREQ and RREP packets that originated and/or are relayed from one hop away neighbouring nodes and stores them in a log file. Each raw packet entry is pre-processed before recorded in the log file so that it could be used straight away by the misuse and anomaly detection mechanisms. A pseudo code as illustrated in Figure 7-3 explains the process involved in the *logaodvpacket* module. An example of log packet entries is as illustrated in Table 7-1.

```
If PT_AODV
{
 If AODVTYPE_RREQ
 {
  Pre-process RREQ packet:
 }
 Else IF AODVTYPE_RREP
 {
  Pre-process RREP packet:
 }
 Else
 Exit():
 Open Log File:
 Record   pktTime.   src.   dst.   prev hop.
 next_hop.   pktType.   pktID.   BcastID.
 HopCount.SeqNumber:
 Close Log File:
}
Else
Exit():
```

Figure 7- 3: Audit Data Capture Pseudo Code

Time	Prev_ Hop	Next_ Hop	Src	Dst	Pkt Type	PktID	Bcast ID	Hop Count	Seq No
10.000000	0	1	0	3	RREQ	0-3-10	1	1	1
10.000000	0	2	0	3	RREQ	0-3-10	1	1	1
10.000000	1	2	0	3	RREQ	0-3-10	1	2	1
10.000000	2	3	0	3	RREQ	0-3-10	1	3	1
10.500000	3	2	3	0	RREP	0-3-10	-1	1	1
10.500000	2	1	3	0	RREP	0-3-10	-1	2	1
10.500000	1	0	3	0	RREP	0-3-10	-1	3	1

Table 7- 1: An Example of Log File Entries

7.3.2 Misuse and Anomaly Detection Mechanisms

As discussed earlier, both misuse and anomaly detection techniques are proposed to be used in this study to make the most of the detection capabilities offered by both techniques, as well as to counter balance the limitations associated with each approach (Fell, 2002). This research utilises a specification based technique for both misuse and anomaly detection engines. This research uses an AODV routing protocol as the specification model to define normal and abnormal activities in the networks.

An investigation of the captured inbound and outbound packets follows the Finite State Machine (FSM) technique with extended features, as suggested by Huang and Lee (2004). In such work, it is suggested that the FSM could also carry extra attributes or variables alongside the state conditions to provide extra inputs for the IDS in the detection process. Figure 7-4 illustrates the components of the detection architecture for both misuse and anomaly detection mechanisms, as proposed in this study. Several FSM attributes based on the AODV specifications have been constructed to be compared with the audit data streams. Each time an inbound or outbound packet is overheard from the neighbouring node, the detection engines try to find the corresponding state condition in the FSM that match the current state of the packet. The detection engines then investigate the logical process as well as the attributes of the packet to detect intrusion based on the FSM specifications.

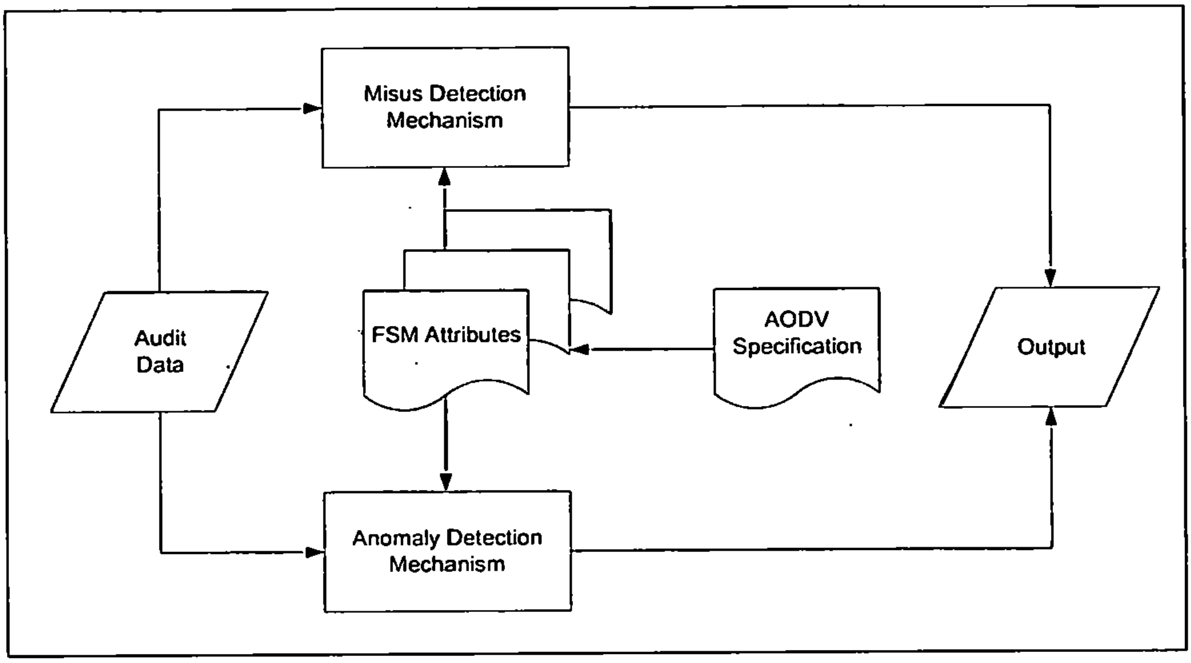


Figure 7- 4: IDS Detection Engines Architecture

7.3.2.1 Modelling the Normal AODV Route Discovery Specification

Normal operations of an AODV routing protocol have been extensively modelled by Bhargavan et al. (2002). Their specification includes normal operations in both data and routing messages transmissions. This study follows their specification model but set the focus on the route discovery operation. This is because a route discovery process involves different states and logical processes, thus it is sufficient to design a simple but realistic IDS as proposed in this study.

In a route discovery process, a source node broadcasts route request (RREQ) packets to its neighbouring nodes to find a valid route to the destination node. Neighbouring nodes that receive the RREQ packets will send back a route reply (RREP) packet if they have a valid route to the destination node. Otherwise, they will re-broadcast the RREQ packet to the next hop nodes where a similar process will be repeated until a valid route to the destination node is found. In some cases, the RREQ packet will reach its time out limit without discovering any valid route to the destination node. In such situation, the source node will have to re-initiate another route discovery process if the path to the destination node is still required.

Apart from RREQ and RREP packets, there is another type of routing messages, which is called a route error (RERR) packet. The RERR packets are used to maintain the availability of the established routes as well as to inform neighbouring nodes about any broken links, which often happen in a MANET environment due to the node's mobility. However, since the RERR packets are not directly associated with the route discovery process, its specification is not included in the FSM specification as discussed in this thesis. A complete version of the AODV FSM specification as modelled in (Huang & Lee, 2004) can be found in Appendix C.

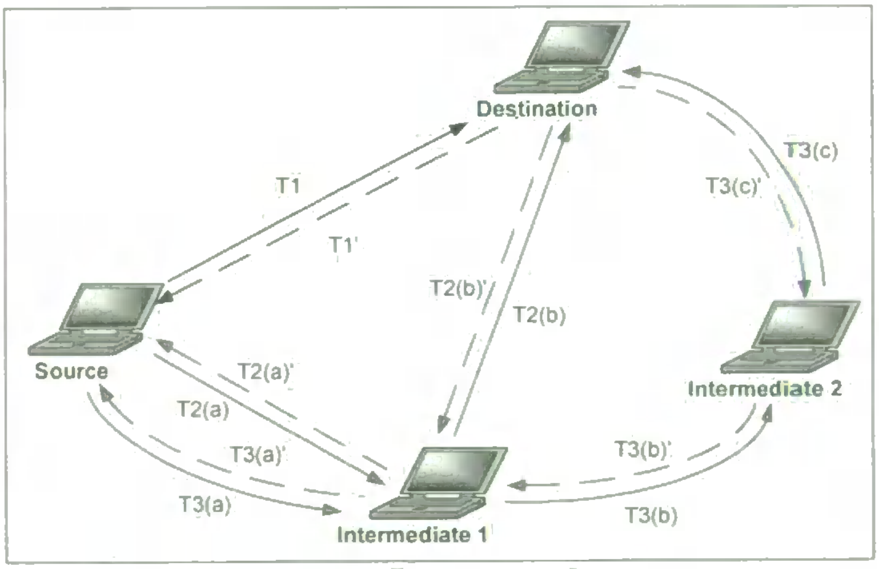


Figure 7- 5: Simplified Route Discovery States

State	Attributes
T1	RREQ(Src,Dst,Hc,BcastID,SeqNo)
T1'	RREP(Dst,Src,Hc,SeqNo)
T2(a) – T2(b)	RREQ(Src,Dst,NxtHp,Hc,BcastID,SeqNo)
T2(b)' – T2(a)'	RREP(Dst,Src,NxtHp,Hc,SeqNo)
T3(a) – T3(b) – T3(c)	RREQ(Src,Dst,NxtHp,PrvHp,Hc,BcastID,SeqNo)
T3(c)' – T3(b)'– T3(a)'	RREP(Dst,Src,NxtHp,PrvHp,Hc,SeqNo)

Table 7- 2: Simplified Route Discovery Attributes

Figure 7-5 and Table 7-2 illustrate 3 scenarios of a simplified AODV route discovery process, along with its associated attributes. States {T1-T1'} represent the first scenario, which involves only the source and the destination nodes. In such a scenario, a source node sends a RREQ packet, which is received directly by the destination node (i.e. the state marked as T1). The destination node then replies with a RREP packet (i.e. marked as T1'). The second scenario is represented by states {T2(a)-T2(b)-T2(b)'-T2(a)'}. In this scenario, source and destination nodes are not within each others wireless range, and thus require an intermediate node to relay the RREQ and RREP packets. In some cases, RREQ and RREP packets need to be relayed by more than one intermediate node. Such a scenario is represented by states {T3(a)-T3(b)-T3(c)-T3(c)'-T3(b)'-T3(a)'}, as shown in Figure 7-5. Any missing state or unmatched attributes in those 3 scenarios will be detected by the misuse and anomaly detection mechanisms, as discussed in the next sub-sections.

7.3.2.2 Misuse Signatures

Ideally, a misuse signatures database should contain as many attack signatures as possible. However, because of time and resource constraints for this research there is sufficient time only to prove the concept. As such only a few misuse signatures have been implemented in this study. Misuse signatures designed in this study are based on the following attacks scenarios.

- *Modification Attack: Changes to Hop Count Value*

  Each time a node received a RREQ/RREP packet, and it is not the destination, it needs to forward the packet to the next hop node. There should be no changes to the packet apart from the hop count field, which needs to be increased by 1 on each hop (please refer to (Perkins et al., 2003) for the AODV route discovery specification). An attacker might change the hop count to the lower value to make sure it is included in the packet forwarding process. Once it has direct access to the packet, it could launch several kinds of attacks as described in Chapter 3. On the other hand, a misbehaving node might change the hop count to the Infinity value to avoid being included in the packet forwarding process, as the involvement would utilise its limited resources. Figure 7-6 illustrates an example of the scenarios when a misbehaving node modifies the hop count value to Infinity.

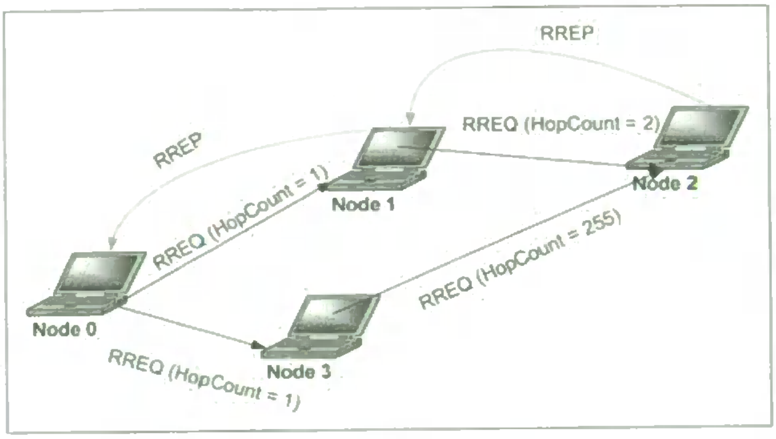
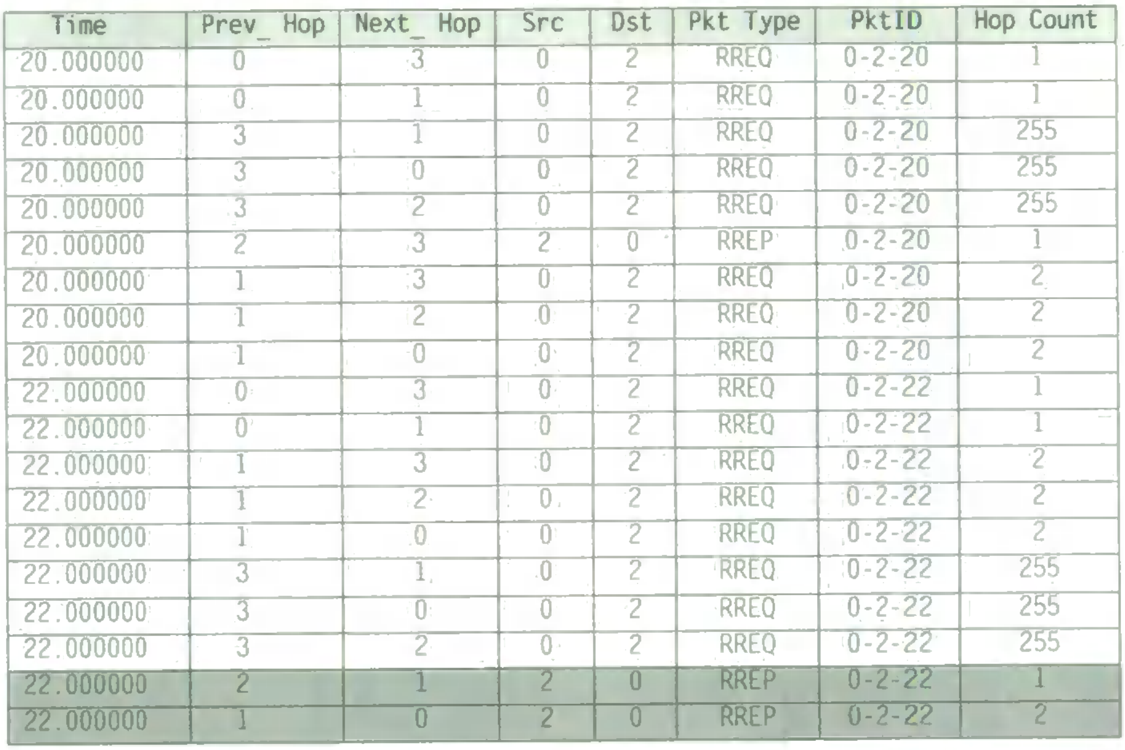


Figure 7- 6: Hop Count Modification Attack

In this example, node 3 is the misbehaving node. Actually, the first discovered route from source node 0 to destination node 2 is (0-3-2), as the first RREP send out from node 2 is routed via node 3. This is shown in the log file as illustrated in Table 7-3. However, because node 3 has changed the hop count for the path to Infinity (i.e. 255), that route has become invalid. As a result, node 2 has to choose the next available route, which is (0-1-2). The complete path from node 0 to node 2, via node 1 is as shown in Table 7-3 (i.e. highlighted cells). In this scenario, node 3 has successfully excluded itself from the packet forwarding process to save its limited resources.



Time	Prev_ Hop	Next_ Hop	Src	Dst	Pkt Type	PktID	Hop Count
20.000000	0	3	0	2	RREQ	0-2-20	1
20.000000	0	1	0	2	RREQ	0-2-20	1
20.000000	3	1	0	2	RREQ	0-2-20	255
20.000000	3	0	0	2	RREQ	0-2-20	255
20.000000	3	2	0	2	RREQ	0-2-20	255
20.000000	2	3	2	0	RREP	0-2-20	1
20.000000	1	3	0	2	RREQ	0-2-20	2
20.000000	1	2	0	2	RREQ	0-2-20	2
20.000000	1	0	0	2	RREQ	0-2-20	2
22.000000	0	3	0	2	RREQ	0-2-22	1
22.000000	0	1	0	2	RREQ	0-2-22	1
22.000000	1	3	0	2	RREQ	0-2-22	2
22.000000	1	2	0	2	RREQ	0-2-22	2
22.000000	1	0	0	2	RREQ	0-2-22	2
22.000000	3	1	0	2	RREQ	0-2-22	255
22.000000	3	0	0	2	RREQ	0-2-22	255
22.000000	3	2	0	2	RREQ	0-2-22	255
22.000000	2	1	2	0	RREP	0-2-22	1
22.000000	1	0	2	0	RREP	0-2-22	2

Table 7- 3: An Example of Log File with the Hop Count Modification

- *Interception Attack: Modifying Sequence Number to Win the Route Discovery Process*

In an AODV routing, a RREQ sequence number is used by each node to determine the freshness of the request, so that no resources are wasted processing an old or identical request. Any request to the same destination node that has a lower sequence number than the previous one will be dropped. Attackers might exploit this scenario by always advertising fresher route requests so that all the packets will be routed via them. Similar to the scenario in a hop count modification attack, once the attackers gain a physical access to the packets, they could launch a more severe attack such as drop the packet or analyse the content of the packet. As illustrated in Figure 7-7, an attacker (i.e. node 3) constantly adds 10 to the sequence number of the victim node's (i.e. node 0) RREQ. As a result, it will always be selected to route node's 0 packets, thus denies other nodes' participation.

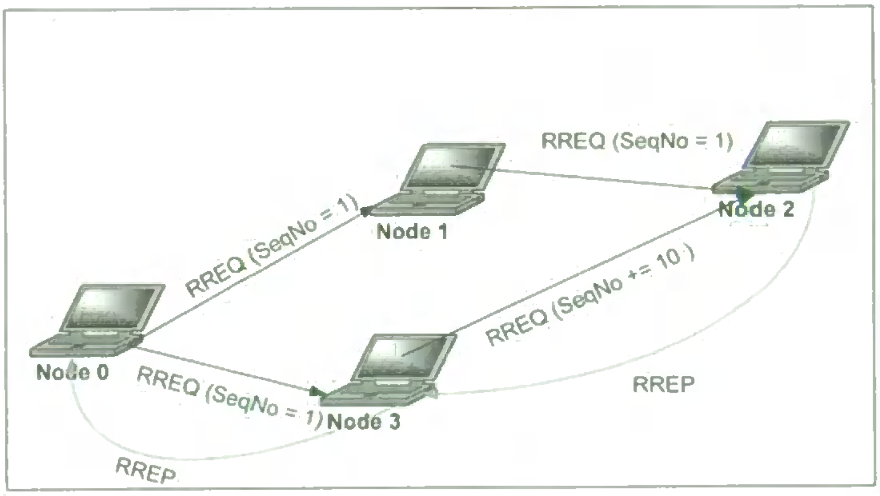
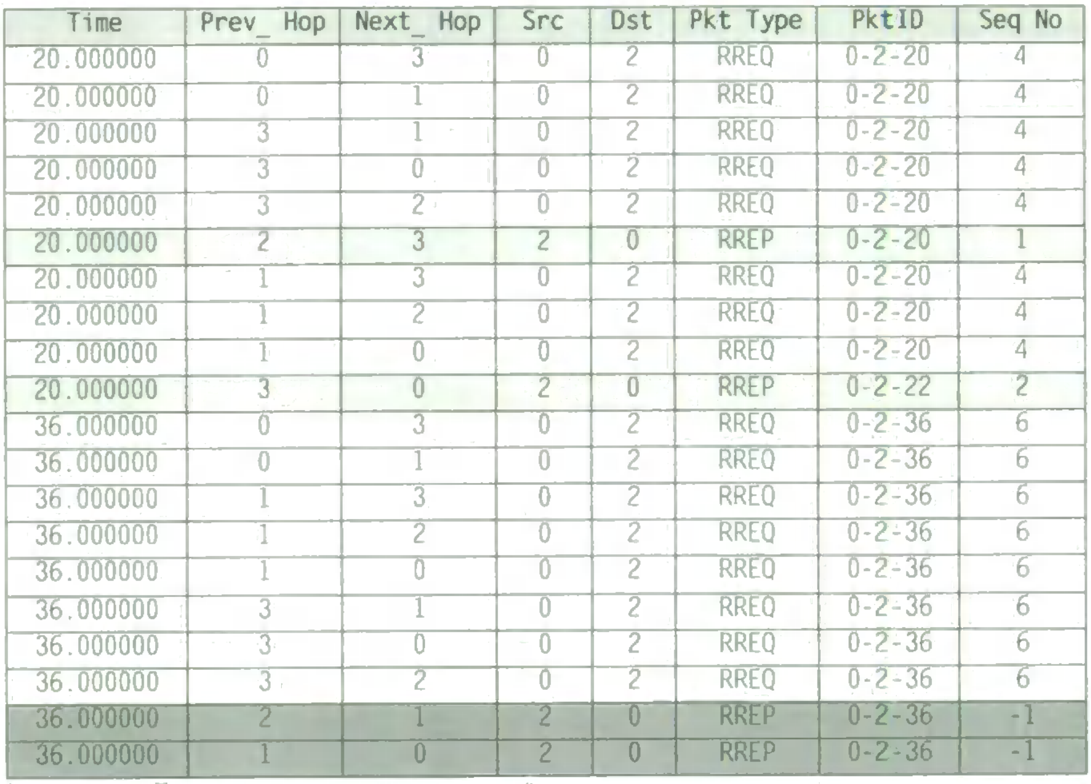


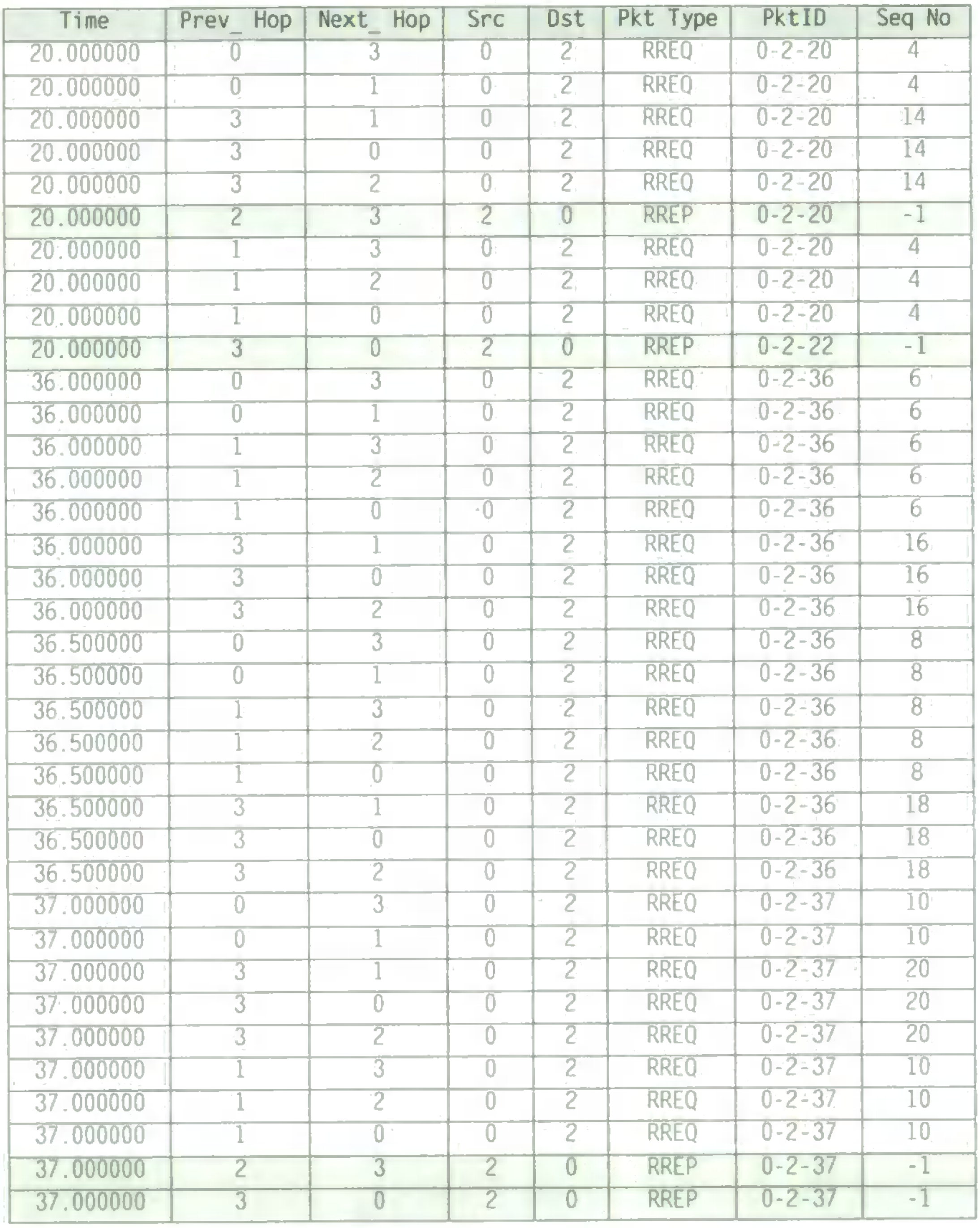
Figure 7- 7: Sequence Number Modification Attack

Log files for normal and under attack conditions, as illustrated in Table 7-4 and 7-5, show that node 3 is always selected in the route discovery process, when the scenario as illustrated in Figure 7-7 is simulated in the NS-2.



Time	Prev_ Hop	Next_ Hop	Src	Dst	Pkt Type	PktID	Seq No
20.000000	0	3	0	2	RREQ	0-2-20	4
20.000000	0	1	0	2	RREQ	0-2-20	4
20.000000	3	1	0	2	RREQ	0-2-20	4
20.000000	3	0	0	2	RREQ	0-2-20	4
20.000000	3	2	0	2	RREQ	0-2-20	4
20.000000	2	3	2	0	RREP	0-2-20	1
20.000000	1	3	0	2	RREQ	0-2-20	4
20.000000	1	2	0	2	RREQ	0-2-20	4
20.000000	1	0	0	2	RREQ	0-2-20	4
20.000000	3	0	2	0	RREP	0-2-22	2
36.000000	0	3	0	2	RREQ	0-2-36	6
36.000000	0	1	0	2	RREQ	0-2-36	6
36.000000	1	3	0	2	RREQ	0-2-36	6
36.000000	1	2	0	2	RREQ	0-2-36	6
36.000000	1	0	0	2	RREQ	0-2-36	6
36.000000	3	1	0	2	RREQ	0-2-36	6
36.000000	3	0	0	2	RREQ	0-2-36	6
36.000000	3	2	0	2	RREQ	0-2-36	6
36.000000	2	1	2	0	RREP	0-2-36	-1
36.000000	1	0	2	0	RREP	0-2-36	-1

Table 7- 4: An Example of Log File without the Sequence Number Modification Attack



Time	Prev_ Hop	Next_ Hop	Src	Dst	Pkt Type	PktID	Seq No
20.000000	0	3	0	2	RREQ	0-2-20	4
20.000000	0	1	0	2	RREQ	0-2-20	4
20.000000	3	1	0	2	RREQ	0-2-20	14
20.000000	3	0	0	2	RREQ	0-2-20	14
20.000000	3	2	0	2	RREQ	0-2-20	14
20.000000	2	3	2	0	RREP	0-2-20	-1
20.000000	1	3	0	2	RREQ	0-2-20	4
20.000000	1	2	0	2	RREQ	0-2-20	4
20.000000	1	0	0	2	RREQ	0-2-20	4
20.000000	3	0	2	0	RREP	0-2-22	-1
36.000000	0	3	0	2	RREQ	0-2-36	6
36.000000	0	1	0	2	RREQ	0-2-36	6
36.000000	1	3	0	2	RREQ	0-2-36	6
36.000000	1	2	0	2	RREQ	0-2-36	6
36.000000	1	0	0	2	RREQ	0-2-36	6
36.000000	3	1	0	2	RREQ	0-2-36	16
36.000000	3	0	0	2	RREQ	0-2-36	16
36.000000	3	2	0	2	RREQ	0-2-36	16
36.500000	0	3	0	2	RREQ	0-2-36	8
36.500000	0	1	0	2	RREQ	0-2-36	8
36.500000	1	3	0	2	RREQ	0-2-36	8
36.500000	1	2	0	2	RREQ	0-2-36	8
36.500000	1	0	0	2	RREQ	0-2-36	8
36.500000	3	1	0	2	RREQ	0-2-36	18
36.500000	3	0	0	2	RREQ	0-2-36	18
36.500000	3	2	0	2	RREQ	0-2-36	18
37.000000	0	3	0	2	RREQ	0-2-37	10
37.000000	0	1	0	2	RREQ	0-2-37	10
37.000000	3	1	0	2	RREQ	0-2-37	20
37.000000	3	0	0	2	RREQ	0-2-37	20
37.000000	3	2	0	2	RREQ	0-2-37	20
37.000000	1	3	0	2	RREQ	0-2-37	10
37.000000	1	2	0	2	RREQ	0-2-37	10
37.000000	1	0	0	2	RREQ	0-2-37	10
37.000000	2	3	2	0	RREP	0-2-37	-1
37.000000	3	0	2	0	RREP	0-2-37	-1

Table 7- 5: An Example of Log File with the Sequence Number Modification Attack

- *Fabrication Attacks: Selfish Node*

Another kind of attack implemented in this study is a selfish attack, which happens when a misbehaving node sends multiple RREQ packets in an inappropriate way. This kind of attack is an example of a route salvaging attack as discussed earlier in Chapter 3. The phenomenon of broken links always happens in MANET environments due to node's mobility and the instability of wireless links. However, the AODV routing protocol provides a mechanism for nodes to recover the lost links. When there is a broken link, a source node is permitted to re-initiate a route discovery process up to *RREQ_RETRIES* times but must wait until the *MAX_RREQ_TIMEOUT* is reached before the next try. A selfish node (i.e. a node that does not want to wait a little longer to complete its packet transmissions) sends multiple RREQ packets to search and maintain routes to the desired destination. In case the current link is lost, the established backup routes will be used to complete the packet forwarding process. This kind of behaviour is not fair to other nodes as it could rapidly drain off their resources. Such behaviour could also increase network activity, which could cause bottlenecks and worsen the packet collision problems within the network. An example of audit data when this attack is simulated in the NS-2 is as illustrated in Table 7-6. In such example, node 0 sends 2 duplicate packets for each of its RREQ packet and causes more activity in the network.

Time	Prev_ Hop	Next_ Hop	Src	Dst	Pkt Type	PktID	BCast ID
20.000000	0	3	0	2	RREQ	0-2-20	1
20.000000	0	1	0	2	RREQ	0-2-20	1
20.000000	0	3	0	2	RREQ	0-2-20	2
20.000000	0	1	0	2	RREQ	0-2-20	2
20.000000	1	3	0	2	RREQ	0-2-20	2
20.000000	1	2	0	2	RREQ	0-2-20	2
20.000000	1	0	0	2	RREQ	0-2-20	2
20.000000	3	1	0	2	RREQ	0-2-20	1
20.000000	3	0	0	2	RREQ	0-2-20	1
20.000000	3	2	0	2	RREQ	0-2-20	1
20.000000	1	3	0	2	RREQ	0-2-20	1
20.000000	1	2	0	2	RREQ	0-2-20	1
20.000000	1	0	0	2	RREQ	0-2-20	1
20.000000	2	1	2	0	RREP	0-2-20	-1
20.000000	3	1	0	2	RREQ	0-2-20	2
20.000000	3	0	0	2	RREQ	0-2-20	2
20.000000	3	2	0	2	RREQ	0-2-20	2
20.000000	2	3	2	0	RREP	0-2-20	-1
20.000000	3	0	2	0	RREP	0-2-20	-1
36.000000	1	0	2	0	RREP	0-2-36	-1
36.000000	0	3	0	2	RREQ	0-2-36	3
36.000000	0	1	0	2	RREQ	0-2-36	3
36.000000	0	3	0	2	RREQ	0-2-36	4
36.000000	0	1	0	2	RREQ	0-2-36	4
36.000000	1	3	0	2	RREQ	0-2-36	3
36.000000	1	2	0	2	RREQ	0-2-36	3
36.000000	1	0	0	2	RREQ	0-2-36	3
36.000000	3	1	0	2	RREQ	0-2-36	3
36.000000	3	0	0	2	RREQ	0-2-36	3
36.000000	3	2	0	2	RREQ	0-2-36	3
36.000000	3	1	0	2	RREQ	0-2-36	4
36.000000	3	0	0	2	RREQ	0-2-36	4
36.000000	3	2	0	2	RREQ	0-2-36	4
36.000000	2	3	2	0	RREP	0-2-36	-1
36.000000	3	0	2	0	RREP	0-2-36	-1
36.000000	1	3	0	2	RREQ	0-2-36	4
36.000000	1	2	0	2	RREQ	0-2-36	4
36.000000	1	0	0	2	RREQ	0-2-36	4

Table 7- 6: An Example of Selfish Attack Audit Data

7.3.2.3 Normal Profiles

In an actual system implementation, all the FSM states and attributes as outlined in previous sections should be formalised to create a list of normal system and user profiles to detect anomalies attacks. However, in this study, only some of the states and attributes are formalised to define a set of normal profiles. Those selected states and attributes are chosen because they are essential to detect anomalies in the packet forwarding process, as designed in this study. One of the good examples to show how the proposed IDS framework could benefit from the anomaly detection mechanism is by testing the system with a packet dropping attack. Detecting a packet dropping attack in MANET environments is not a straightforward procedure. This is because it is difficult to distinguish between a genuine packet dropping attack and a benign failure (i.e. lost links) in ad hoc networks. Deploying a misuse detection mechanism to detect this kind of attack could lead to many false alarms. Some of the scenarios that follow are examples where an anomaly detection mechanism should be deployed instead of a misuse detection mechanism, to avoid producing many false alarms in detecting a packet dropping attack.

- *Collision*

  Packet collision is a common problem in a computer network. It happens when two or more nodes attempt to transmit a packet across the network at the same time (Stathopoulos et al., 2004). In a case of a MANET, the packet will be discarded and the source nodes need to retransmit the packets. In this case, the receiving node is not intended to drop the packet, thus cannot be punished for not forwarding or replying to the packets.

- *Wireless Link Failure*

In ad hoc networks, nodes move from one location to another even when they are in the middle of a packet forwarding process. As a result, packets are sometime sent or forwarded to an unreachable node. As illustrated in Figure 7-8, node 1 is in range when the packet is sent out from the source node. However, before the packet arrives, node 1 has moved to a different location, which is out of the source node's communication range. The packet will be discarded and the source node needs to establish an alternative route to reach the destination node. Node 1 in this case cannot be punished because it has not intended to drop the packet.

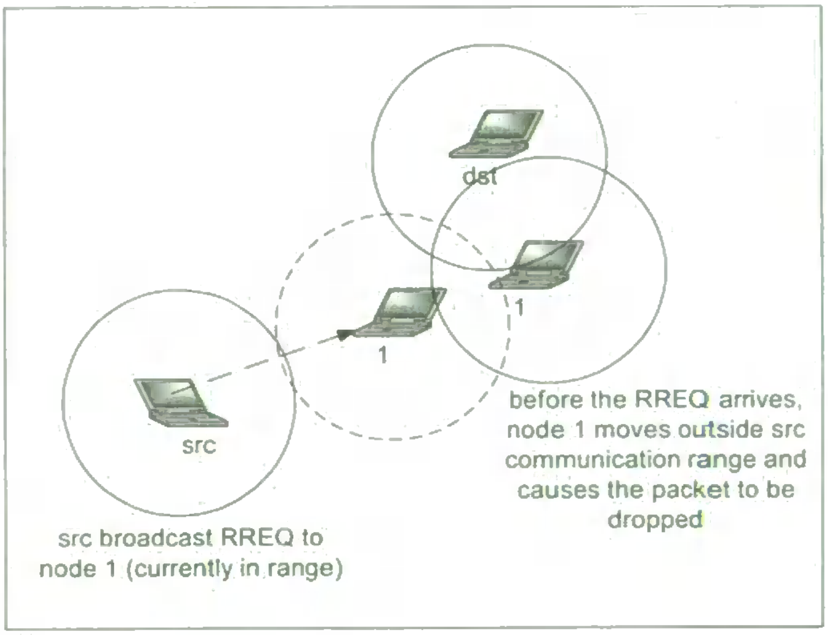

dst
1
1
src
before the RREQ arrives, node 1 moves outside src communication range and causes the packet to be dropped
src broadcast RREQ to node 1 (currently in range)

Figure 7- 8: Wireless Link Failure in MANET Environments

A pseudo code for the neighbouring nodes to detect a packet dropping attack is as presented in Figure 7-9. It consists of 3 parts as different code is needed to analyse the source, intermediate, and destination nodes behaviour. However, relying solely on this analysis is not sufficient as it could lead to many false alarms. The analysis needs to be accompanied with a set of tests (i.e. cross-feature test) to confirm the attack, as discussed in Section 7.3.2.5.

```
For each neighbouring node {
 If ((neighbour == source) && (pktType == RREQ))
 {
  No action;
 }
 Else if ((neighbour != source) && (pktType == RREQ))
 {
  If (!destination)
  {
   If ((RREQ. RREQ')==1)
   {
    Log as normal;
   }
   Else
   {
    Go to test sets;
   }
  }
  Else if (destination)
  {
   If ((RREQ. RREP)==1)
   {
    Log as normal;
   }
   Else
   {
    Go to test sets;
   }
  }
 }}
```

Figure 7- 9: Pseudo Code to Detect Packet Dropping Attack

7.3.2.4 Misuse Detection Engine

The misuse detection engine implemented in this study follows a pattern matching technique. As illustrated in Figure 7-10, the engine compares real time audit data with the misuse signatures, which have been extracted from the normal AODV route discovery FSM attributes. Each time the audit data matches one of the misuse signatures in the database, a local response, followed by a global response mechanism will be triggered.

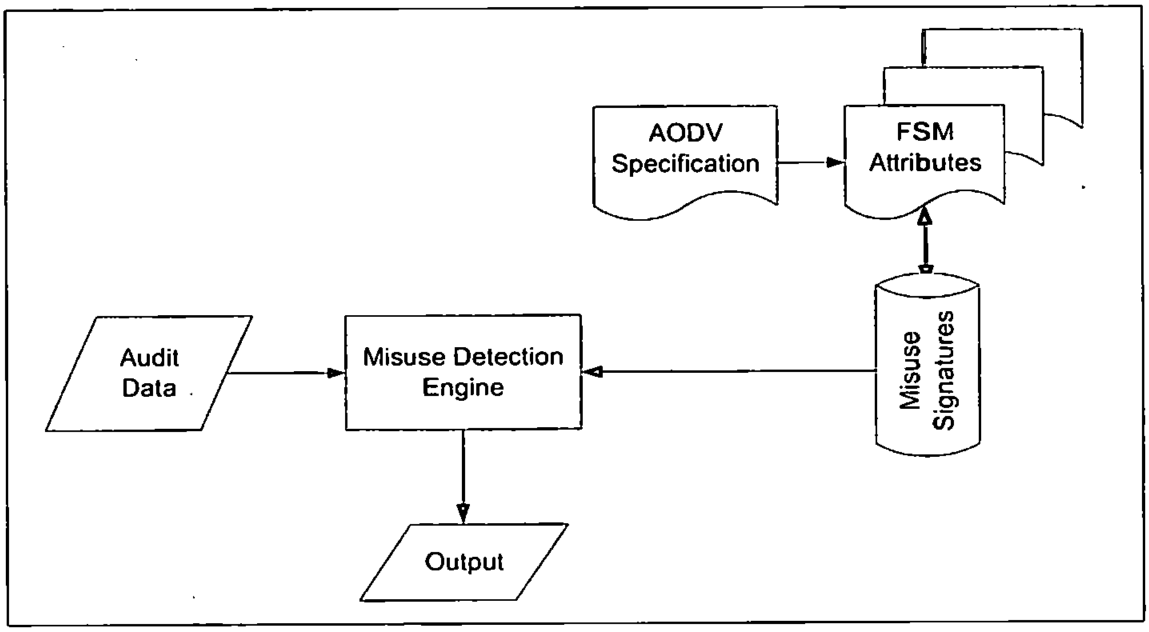


Figure 7- 10: Misuse Detection Mechanism

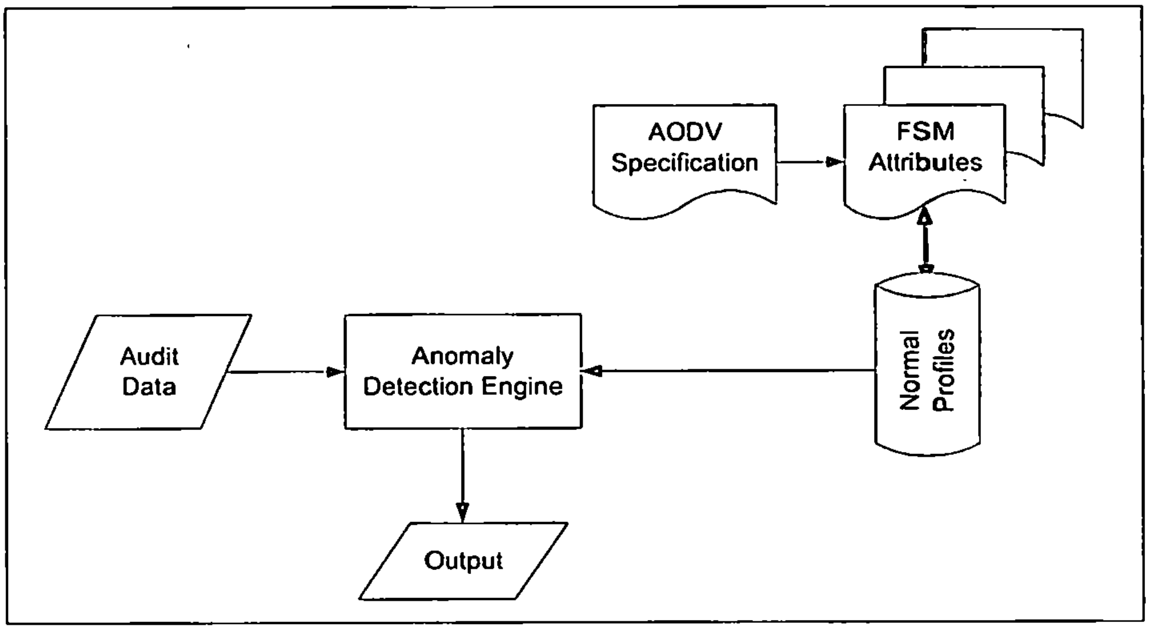
As for the purpose of proving the concept, only a few attacks signatures/rules have been implemented in this study. Those rules are designed to detect all the attacks scenarios as described in section 7.3.2.2. A pseudo code as illustrated in Figure 7-11 describes the procedure to detect misuse activities in the network.

```
For each current pre-processed packet
{
 For each attack signature
 {
  Check curr_packet state:
  Read prev_packet state:
  If (curr_packet state != prev_packet state)
  {
   If (match signature)
   {
    Trigger local response:
    Trigger global response:
   }
  }
  Check curr_packet attributes:
  Read prev_packet attributes:
  If (curr_packet attributes != prev_packet attributes)
  {
   If (match signature)
   {
    Trigger local response:
    Trigger global response:
   }
  }
 }
}
```

Figure 7- 11: Pseudo Code to Detect Misuse Activity in the Network

7.3.2.5 Anomaly Detection Engine

As discussed in previous chapters, the process to detect anomaly attacks in MANET environments is not as straightforward as in a wired network. In some cases, attacks might be confused with benign failures, which always occur in MANET environments. For that reason, in addition to the pattern matching, as implemented in the misuse detection engine, the anomaly detection engine in this study also applies a cross-feature test to reduce the number of false alarms triggered in the network. The components of the anomaly detection mechanism are as illustrated in Figure 7-12.



Figurc 7- 12: Anomaly Detection Mechanism

The anomaly detection engine in this study is designed to detect attacks scenarios discussed in section 7.3.2.3. A pseudo code that explains procedures of the anomaly detection engine is as illustrated in Figure 7-13.

```
For each current pre-processed packet
{
 For each normal profile
 {
  Check curr_packet state/attribute:
  Read prev_packet state/attribute:
  If (curr_packet state/attribute != prev_packet state/attribute)
  {
   If (match profile)
   {
    No action:
   }
   Else if not (match profile)
   {
    Cross_Feature_Test_1(source. suspect):
    Cross_Feature_Test_2(source. suspect):
   }
  }
 }
}
```

```
Cross_Feature_Test_1(int source. int suspect)
{
 If (forward packet == yes)
 {
  Suspect node not malicious:
  Halt All Test:
 }
 Else
 {
  Continue next test:
 }
}
```

```
Cross_Feature_Test_2(int source, int
suspect)
{
 If (dropped packets >= threshold)
 {
  Suspect node malicious;
  Halt All Test;
 }
 Else
 {
  Send global investigation request;
 }
}
```

Figure 7- 13: Pseudo Code to Detect Anomalies in the Network

7.3.3 Response Mechanism

A local response mechanism is triggered each time a misuse or an anomaly detection mechanism detected an intrusion. As mentioned earlier, it is not an interest of this study to focus on the local response mechanism, thus only a simple responsive behaviour is implemented to complete the IDS cycle. In the proposed IDS framework, each node reacts to an intrusion by adding the malicious node's identity into its own bad nodes table. Figure 7-14 illustrates a pseudo code of the local response module.

```
For each detected attack
{
    Display alert on screen;
    Open BadNode file
    Add malicious node to BadNode
    file;
    Record time of the malicious
    activity;
    Close BadNode file
}
```

Figure 7- 14: Local Response Pseudo Code

An example of on screen alert is as illustrated in Figure 7-15.

```
Node 1 detects node 51 sent a fake RREQ packet at time =
22.000000
```

Figure 7- 15: An Example of On Screen Alert

Node	Bad Node	Time
1	50	22.000000
1	2	40.000000
3	10	10.000000
5	50	40.000000
7	2	40.000000
7	10	20.000000
7	50	30.000000

Table 7- 7: An Example of BadNode File Entries

Each node maintains its own bad nodes table. For instance, as illustrated in Table 7-7, node 1 has 2 entries in its BadNode table (i.e. node 50 and 2). Later, these lists will be shared between friends via the global response mechanism, when they meet each other. The next section explains how this could be done.

7.4 Global Detection and Response

Information sharing between nodes is very useful as it could benefit an IDS in many ways. For instance, nodes are able to gather more evidence if they share their audit data sources with other nodes in the network. More information/evidence in the audit data sources could improve the accuracy of the detection results. Information sharing also is very useful to alert other nodes about the existence of an intrusive node in the network. However, without a proper implementation, information sharing might be exploited by the attackers to launch several attacks against a MANET (refer to Chapter 3 for the list of attacks). This section presents the implementation of the global detection and response mechanisms of the proposed IDS, and explains how they could minimise the risk of receiving false information/alerts from the attackers.

7.4.1 Audit Data Source

Although it is mentioned earlier that the accuracy of the detection results could be improved by sharing each node's audit data, the proposed IDS framework in this study did not utilise such technique. This is because of the following reasons:

- The integrity of the audit data that traversed multi hop from source to destination nodes is vulnerable to modification, interruption, and interception attacks.
- Malicious nodes might fabricate fake audit data to poison other nodes audit data sources.
- Audit data sharing might utilise a lot of the network's bandwidth, as well as increasing the nodes' activity. High utilisation of network's bandwidth might cause bottlenecks in communication, whilst an increase in nodes' activities does not help in preserving their limited resources.

As an alternative, this study proposes that the audit data sources are only to be used by the nodes that own them. Each node investigates its own audit data and simply presents the results to the friend nodes when needed in the global detection mechanisms. The next sub-sections explain this in more detail.

7.4.2 Friend Detection Mechanism

A friend detection mechanism is proposed in this study to speed up the detection process as well as to reduce the number of false alarms that usually occurs in an anomaly IDS. Each time a local anomaly detection mechanism suspects an intrusion with a lower confidence level (i.e. without concrete evidence) a collaborative friend detection mechanism will be triggered to support the IDS decision. The process involved in this mechanism is somewhat similar to the traditional voting mechanism, where the source node (i.e. node that detects the suspicious activity) requests its neighbouring nodes votes/opinions regarding the suspicious activity. Having collected the votes, the source node will make a decision and inform the participated neighbouring nodes about the voting results. Although the traditional voting mechanism is proven to speed up the detection process, as discussed in (Zhang et al., 2003; Kachirski & Gupta, 2002), such a collaborative mechanism is exposed to a colluding blackmail attack.

Blackmail attack occurs when a malicious node sends a false accusation/bad vote to make the victim node look bad in the eyes of other nodes. A traditional voting mechanism is able to protect the network against such attack as long as the number of blackmailer nodes that present in the network is less than the value of the voting threshold. For instance, a voting mechanism with the voting threshold value set to 3 is reliable against a single blackmail attacker. Such situation is illustrated in Figure 7-16.

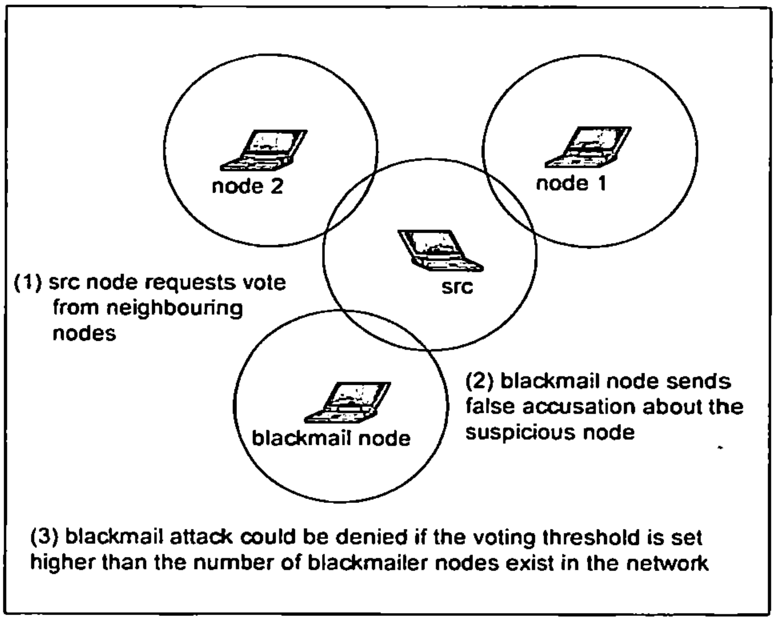

node 2
node 1
src
(1) src node requests vote from neighbouring nodes
blackmail node
(2) blackmail node sends false accusation about the suspicious node
(3) blackmail attack could be denied if the voting threshold is set higher than the number of blackmailer nodes exist in the network

Figure 7- 16: False Accusation from a single blackmail attacker

However, the reliability of the voting mechanism could be jeopardised in the presence of colluding blackmail attackers. In such a scenario, several blackmail attackers work together to blackmail the victim node. This kind of attack is very difficult to defend because of the following reasons:

- It is difficult to know the actual number of the colluding blackmail attackers that are present in the network. Without such information, the number of votes required to make the voting threshold immune against the colluding blackmailer, need to be guessed. This without doubt affects the reliability of the voting mechanism. The blackmail attackers could also always add new nodes to their blackmailing team to match the minimum voting threshold value preset in the network.

- It is true that by setting the minimum vote's value to the higher number will lessen the effects of the colluding blackmail attackers. However, such action could also affect the performance of the voting mechanism. More votes are required to reach the minimum votes count, thus requires a longer time for the source node to collect all the votes to make a decision about any suspicious activity.

A friend detection mechanism proposed in this study is designed to provide an alternative solution for the above issues. The problem in the voting mechanism is that there is no method for the source node to distinguish between votes that come from legitimate users and false votes that come from the attackers. Mutual trust between friends as discussed earlier in Chapter 6 could eliminate this node's anonymity problem. By filtering all the incoming votes using the established mutual trusts, the source node would be able to avoid receiving votes from the colluding blackmail attackers, and thus the reliability of the collaborative detection mechanism is ensured.

```
(1)   For each neighbouring node
(2)   {
(3)    If (adjacent node == friend)
(4)    {
(5)     While (!feof(FRequest)
(6)     {
(7)      Request friend detection;
(8)      If (Test1)
(9)      {
(10)      Suspicious node = Malicous:
(11)      Add to MalNode table:
(12)      Break:
(13)     }
(14)     If (Test2)
(15)     {
(16)      Suspicious node = Good:
(17)      Add to GoodNode table:
(18)      Break:
(19)     }
(20)    }
(21)   }
(22)   Else
(23)   {
(24)    Continue:
(25)   }
(26)  }
```

```
(8a)  Test1()
(8b)  {
(8c)   While (!feof(MalNode))
(8d)   {
(8e)    If ((suspicious.malnode)==1))
(8f)     Return 1:
(8g)    Else
(8h)     Return 0:
(8i)   }
(8j)  }
```

```
Test2()
{
 While (!feof(GoodNode))
 {
  If ((suspicious.goodnode)==1))
   Return 1;
  Else
   Return 0;
 }
}
```

Figure 7- 17: Pseudo Code for Friend Detection Mechanism

The pseudo code in Figure 7-17 explains the process of a collaborative friend detection mechanism as implemented in this study. The process begins when a source node sends a request to its neighbouring friends to investigate if the suspicious node is malicious or not is based upon their own audit data sources and local detection mechanisms. Results from the friends' detection mechanisms are classified into 3 categories, namely malicious, good, and neutral. If there is enough evidence, friend nodes will be able to suggest whether the suspicious node is malicious or a good node based on their packet forwarding histories. On the other hand, a neutral result will be returned to the source node if the friend nodes could not decide whether the suspicious node is malicious or not. This situation happens when the friend nodes do not have enough experience communicating with the suspicious node, thus will not be able to gather sufficient evidence to make a concrete decision. Each request returned with a neutral result will be investigated by other friends when they meet the source node in the future.

The time needed for a source node to receive confirmation from friends whether the suspicious nodes in the request table are malicious or not depends upon several factors, such as the initial number of friend relationships established between the source nodes and other nodes in the network, the density of the network, and the node's mobility patterns. The next chapter evaluates the performances of the proposed friend detection mechanism and compares the results with the performance of the traditional voting mechanism in the presence of colluding blackmail attackers.

7.4.3 Global Response Mechanism

Apart from ensuring the reliability of the global detection mechanism, the proposed friendship concept also plays an important role in facilitating the secured global response mechanism of the proposed IDS framework. A global response mechanism could significantly improve the performance of an IDS by informing other nodes about the malicious nodes that they have yet to encounter. By receiving such alerts, intrusive nodes could be identified at the early stages of their existence. In addition, a lot of time and node's limited resources could be saved. However, without a proper implementation, the global response mechanism is vulnerable to attack, and thus could jeopardise the reliability of an IDS. The problems become worse in MANET environments due to the node's anonymity issue, which could be exploited by the malicious nodes to launch a blackmail attack, as discussed in the previous sections.

In the proposed IDS framework, a friend node is used as a filter to help the receiving node to distinguish between genuine and fake alerts. Each alert received by the friend nodes will be accepted and shared with other friends, whilst intrusion alerts received from other anonymous nodes will be dropped as their reliability could not be ensured. With an assumption that all the friend nodes are behaving correctly, and none of them have been compromised by the attackers, the problem of blackmail attacks could be eliminated. A pseudo code of the global response mechanism as implemented in this study is presented in Figure 7-18.

```
For each neighbouring node
{
 If (adjacent node == friend)
 {
  Exchange Local();
  Exchange Global();
 }
 Else
 {
  Continue;
 }
}
```

```
Exchange Local()
{
 While (!feof(MalNode)
 {
  If ((MalNode.Local) != 1))//not exist
  {
   Copy    (Local.MalNode)//copy    entry    to
   local list
  }
  Else
  {
   Duplicate entry; //not copy
  }
 }
}
```

```
Exchange GLobal()
{
 While (!feof(MalNode)
 {
  If ((MalNode.Global) != 1))//not
  exist
  {
   Copy (Global.MalNode)//copy entry to
   global list
  }
  Else
  {
   Duplicate entry: //not copy
  }
 }
}
```

Figure 7- 18: Pseudo Code of the Global Response Mechanism

7.5 Trust Management

The proposed IDS framework also includes a trust management mechanism for nodes to manage their friend lists. This mechanism is executed each time two friend nodes are adjacent to each other. Each node maintains two sets of friend lists, namely direct friends and indirect friends. Every time a node meets its friend, both of them will exchange their direct and indirect friend lists. A pseudo code as illustrated in Figure 7-14 describes the process in trust management mechanism. The implementation of this mechanism in the simulation package is as described earlier in Chapter 6. Programming codes for this module, as well as other important modules of the proposed IDS framework as discussed earlier, can be found in Appendix D.

```
If (Nodel adjacent to Node2)
{
 If (Node 2 is a friend to Node 1)
 {
  Copy (Node 1 indirect friends <- Node 2 direct friends):
  Copy (Node 1 indirect friends <- Node 2 indirect friends):
 }
 Else
  Proceed with intrusion detection:
 If (Node 1 is a friend to Node 2)
 {
  Copy (Node 2 indirect friends <- Node 1 direct friends):
  Copy (Node 2 indirect friends <- Node 1 indirect friends):
 }
 Else
  Proceed with intrusion detection:
}
```

Figure 7- 19: Pseudo Code for Friends Exchange between Two Nodes

7.6 Signature Management

As mentioned earlier, one of the misuse detection mechanism limitations is that the process to create attack/misuse signatures requires a lot of time and effort. This is because attackers could launch attacks in many ways (refer to Chapter 3) and always find new techniques to breach security holes. The process to create the attack signatures also usually carried out by an expert or a trusted third party, because average users do not have sufficient knowledge to do so.

In general, the problems of maintaining attack signatures for misuse IDS can be classified into 2 categories, as follow:

- *Creating new attack signatures based on the pre-known attacks*

  As mentioned above, this problem is quite difficult to address, especially in MANET environments. Since MANET nodes are anonymous, one might have a problem to add new attack signatures that have been created by other nodes because of the accuracy issue. Adding an attack signature that has been created by another user without a confirmation from a trusted third party or a CA might cause an inaccurate detection result, which then could lead to many false alarms triggered in the network.

- *Advertising/updating attack signature database for all nodes in the networks*

  Another problem faced by MANET nodes is a difficulty to reach the trusted third party to get its own attack signatures updated. Since MANET nodes move from one location to another and sometimes station in an area that is unreachable by the third party, their attack signature database might not be able to protect them against the most recent threats.

Solving the first problem (i.e. enabling nodes to create an accurate attack signature) is not of interest to this study, and could be a possible extension for future work. On the other hand, this study proposes a solution for the latter problem, which is to enable remote nodes to obtain their updated attack signatures. In addition, by using the friendship mechanism as proposed in this study, the remote nodes also could be protected by obtaining false updates from malicious nodes. A pseudo code that illustrates the process to exchange attack signatures is shown in Figure 7-20.

```
If (Node1 adjacent to Node2)
{
 If (Node 2 is a friend to Node 1)
 {
  Copy (Node 1 attack signatures <- Node 2 attack signatures);
 }
 Else
  Proceed with intrusion detection;
 If (Node 1 is a friend to Node 2)
 {
  Copy (Node 2 attack signatures <- Node 1 attack signatures);
 }
 Else
  Proceed with intrusion detection;
}
```

Figure 7- 20: Pseudo Code for Nodes to Exchange Attack Signatures

However, this mechanism has not been implemented in this study. This is because the effects of friendship concept towards the performance of this mechanism should not be different from the results obtained from the trust management evaluation, as the procedures in both mechanisms are similar. Please refer to Chapter 6 for the results of trust management mechanism evaluation.

7.7 Conclusion

This chapter has extensively presented the concept behind the simulation design of the proposed IDS framework. The design has been made using the NS-2 network simulation tools because of the hardware limitation, as well as because it is the most appropriate way to show the applicability of the proposed IDS framework in various MANET environments. Since the purpose of the simulations is to prove that the friendship concept is capable of facilitating the global detection and response mechanisms, only some of the IDS modules have been fully implemented. Apart from the IDS modules, several attack scenarios have also been designed to emulate malicious nodes activity in the network.

The next chapter investigates the performances of the proposed IDS framework in various MANET settings. The investigation focuses on the global detection and response mechanisms to evaluate the capabilities of the proposed friendship concept. For the comparison purposes (e.g. number of false alarms, number of detected malicious nodes), performance of an IDS without the proposed friendship concept also has been investigated.

Chapter 8

Evaluating the Proposed Two-tier IDS Framework Performance

8.1 Introduction

The novelty of this study is on the introduction of the friendship concept to provide reliable global detection and response mechanisms for MANET IDS, especially in the presence of blackmail attackers. Having presented the conceptual design as well as the implementation of the proposed IDS framework in previous chapters, this chapter presents results from a series of simulation experiments, carried out to evaluate the performances of the proposed global detection and response mechanisms in several MANET environments.

8.2 Simulation Setup

Similar to the experiments carried out in Chapter 6, the proposed global detection and response mechanisms also have been evaluated in several MANET environments to investigate the effects of node's initial friendships, simulation times, and network's density towards the overall performance of the proposed IDS framework. Details on each environment (i.e. university campus, city network-1 and city network-2) can be found in Chapter 6. In addition, several bad nodes have been added into the system to emulate the malicious nodes activities. Such malicious activities are as discussed in Chapter 7. Table 8-1 lists all the bad nodes that have been added into the system along with their associated malicious activities.

Simulation Experiment	Bad Node	Attack Type	Malicious Activity
Global Detection	96	Interruption	Drop every packet that does not destined to or sourced from itself
Global Response	97	Modification	Modify the hop count value of every packet that does not destined to or sourced from itself
	98	Interception	Advertise packet with lower sequence number so that it is included in the packet forwarding process to gain access to the packet
	99	Fabrication	Fabricate duplicate packets to ensure stability of own route

Table 8- 1: Bad Nodes and their associated Malicious Activities in Global Detection and Response Mechanisms Simulations

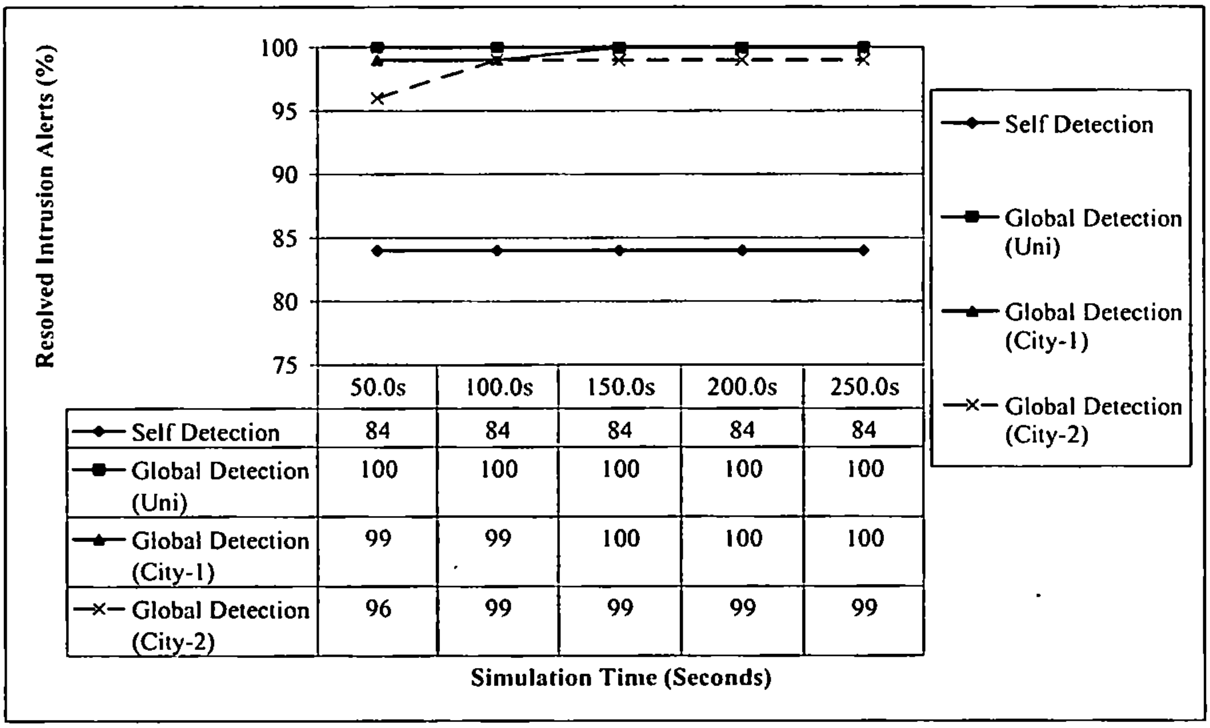
The bad nodes are assumed to not have any trust relationships with other nodes in the network to ensure the reliability of friends' detection reports/alarms. For simplicity, only one attack scenario is simulated in each simulation run. For instance, in an experiment to investigate the performance of the global detection mechanism, only the packet dropping attack is added into the system. This is because, by having several attack scenarios inserted into the system at one simulation run, the malicious node's activities will become too apparent, thus could demote the impact that could be brought by the global detection and response mechanisms towards the overall performance of the proposed IDS framework.

8.3 Global Detection Performance

Having a global detection mechanism in the proposed IDS framework has its own advantages and drawbacks. One of the advantages is that it provides a platform for nodes to share their intrusion detection results to help other nodes in making a decision regarding any suspicious activities. As a result, such intrusive activity could be detected and advertised to as many nodes as possible in its early appearance. However, without an appropriate security measure, malicious nodes could exploit the global detection mechanism to launch a blackmail attack against the other well-behaved nodes in the network. The following sub-sections discuss this issue by presenting the performance comparison between an IDS that utilises a global detection mechanism and an IDS that relies solely on the self-detection (i.e. local) mechanism. Results from the simulations should provide evidence that the overall performance of an IDS could be improved by implementing a global detection mechanism. The effects of blackmail attackers in the system also will be discussed in this section. First, results from experiments will be presented to show how a voting mechanism could be very useful to defend an IDS from an independent blackmail attacker. Following that, a result from an experiment that has been carried out to show how such a voting mechanism is vulnerable to a colluding blackmail attacker, will be presented. Finally, the performance of the proposed friend-assisted global detection mechanism will be presented to prove that such a mechanism is capable to minimise the problem of the colluding blackmail attacker, as faced in the voting mechanism.

8.3.1 Unfiltered Global Detection vs. Self-Detection

This section presents the results and observations obtained from the simulation experiments, which were carried out to investigate the performance comparison between an unfiltered global detection mechanism and an IDS that relies solely upon a self-detection mechanism. Unfiltered global detection mechanism here means every node could participate in the global detection mechanism and no voting mechanism is deployed to reduce the risk of a blackmail attacker. Results from the simulations are as illustrated in Figure 8-1.



	50.0s	100.0s	150.0s	200.0s	250.0s
—◆— Self Detection	84	84	84	84	84
—■— Global Detection (Uni)	100	100	100	100	100
—▲— Global Detection (City-1)	99	99	100	100	100
—×— Global Detection (City-2)	96	99	99	99	99

Figure 8- 1: Performance Comparisons between Unfiltered Global Detection and Self-Detection Mechanisms

From the simulation results as presented in Figure 8-1, it can be seen that the percentage of resolved intrusion alerts in an IDS that utilises a global detection mechanism is higher than the percentage of a resolved intrusion alerts in an IDS that solely relies on the self-detection mechanism. More decisions can be made about the suspicious activities in an IDS that employs a global detection mechanism because more evidence can be shared between nodes. In a self-detection mechanism, each node is responsible for analysing all the suspicious activities that have been detected by its own local detection mechanism and make a decision whether the suspicious activity is malicious or not. The analysis is based upon each node's self experience, which is gathered during its participation in the packet forwarding process. A lack of experience prohibits each node to make a prompt decision about the suspicious activities, and thus leaves the detection mechanism with many unresolved suspicious activities. Until sufficient experiences (i.e. evidence) are gathered, the status of the suspicious activities will remain unknown. The waiting time to make a decision regarding the suspicious activities can be shortened by deploying a global detection mechanism. As illustrated in Figure 8-1, the numbers of the resolved suspicious activities in all three different MANET environments (i.e. university, city-1 and city-2 environments) that utilise a global detection mechanism have increased significantly from their initial value in the self-detection mechanism. For instance, within a same simulation time (i.e. 50 seconds) in the city-2 environment, the percentage of a resolved intrusion alert have increased to 96% by utilising the unfiltered global detection mechanism from its initial percentage (i.e. 84%) in the self-detection mechanism. A better result can be seen in the university environment, where in such a dense network, an unfiltered global detection mechanism was able to help nodes in making decision regarding all the unresolved suspicious activities that have been detected by their local detection engines.

The formula used to calculate the percentage of a resolved intrusion alert is as presented below:

$$P = \frac{m+g}{s}\ (100\%),\ \text{where } m = \text{resolved malicious node},\ g = \text{resolved good node},\ s = \text{suspicious activity}$$

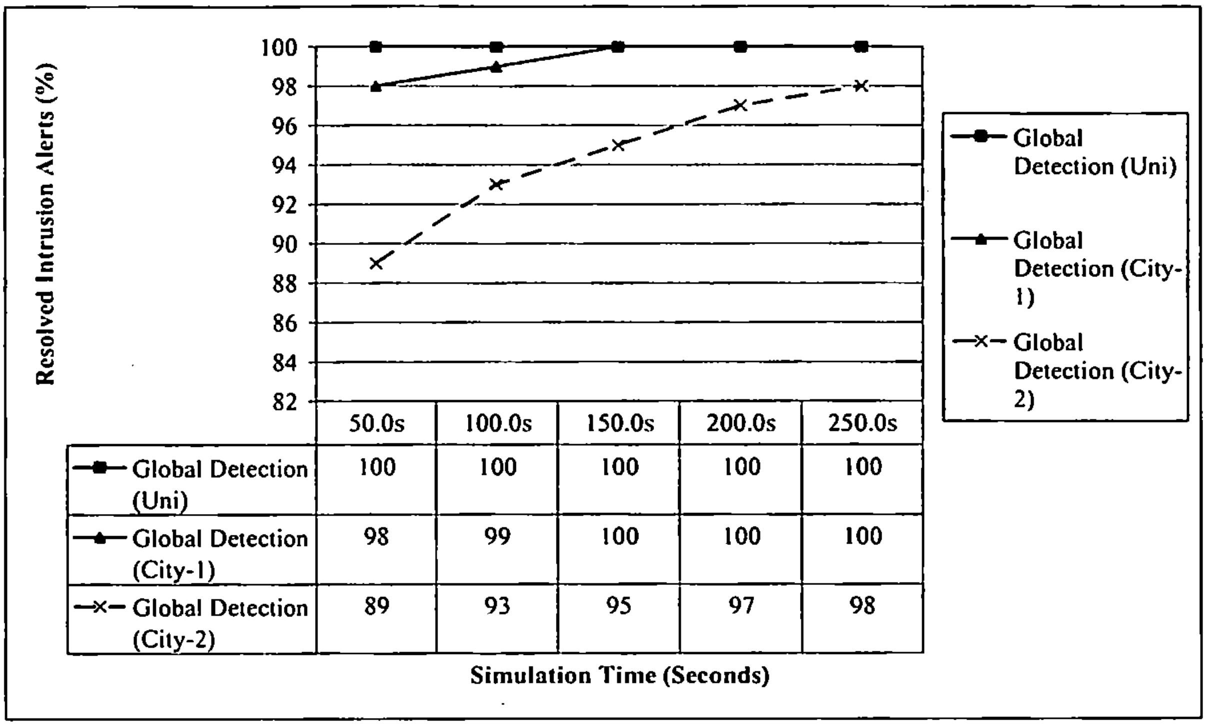
Formula 8- 1: Percentage of Intrusion Alerts

Another observation that can be made is that the unfiltered global detection mechanism is capable to assist nodes in sharing their detection intrusion analysis with others in a very short time. Since there is no threshold value that needs to be matched, as required in a voting mechanism, the process of intrusion information sharing could be achieved in no time. Each node requests evidence or results from other nodes about a particular suspicious activity and gets instantaneous results from the neighbouring nodes. The percentage of resolved intrusion alerts in the university environment is higher than the percentages in the two city network environments because more interactions amongst nodes are expected in such a dense environment. However, despite the high detection rate, the unfiltered global detection mechanism is vulnerable to blackmail attackers, a similar problem that also happens in the global response mechanism. The next subsection discusses this issue.

The TCP transmissions used in the simulations have been generated using an automated script (i.e. cbrgen.tcl), which is included in the NS-2 package. A copy of the script as well as the examples of TCP connections used in the simulation can be found in Appendix E.

8.3.2 Unfiltered Global Detection vs. Vote-filtered Global Detection

This subsection presents a result from simulation experiments carried out to investigate the performance comparison between unfiltered and vote-filtered global detection mechanisms. In the previous subsection, it can be seen that the unfiltered global detection mechanism could provide a platform for nodes to share their intrusion evidence and help others in making a decision about any suspicious activities. The results were very good, as almost all the suspicious activities could be resolved at the early stage of their appearance. However, one significant problem with such a mechanism is that it is vulnerable to a blackmail attack. Since report or evidence of intrusions can be received from any nodes, there is a possibility that the reports were received from a blackmail attacker, who aims to falsely accuse other well-behaved nodes in the network. Deploying a voting threshold could minimise the impact of such a problem. In the vote-filtered global detection mechanism, any intrusion reports/evidence from anonymous nodes will not be considered as genuine until sufficient numbers of votes (i.e. the voting threshold) are gathered. As a result, any fake intrusion reports/evidence from independent blackmail attackers will be denied. However, since each node needs to gather sufficient reports from other nodes about the suspicious activities, it is expected that the performance of the vote-filtered global detection mechanism will be affected. Figure 8-2 illustrates results from simulation experiment which has been carried out to investigate the performance of the vote-filtered detection mechanism in various MANET settings.



	50.0s	100.0s	150.0s	200.0s	250.0s
—■— Global Detection (Uni)	100	100	100	100	100
—▲— Global Detection (City-1)	98	99	100	100	100
—×— Global Detection (City-2)	89	93	95	97	98

Figure 8- 2: Performance of the Vote-filtered Global Detection Mechanism in Various MANET Environments

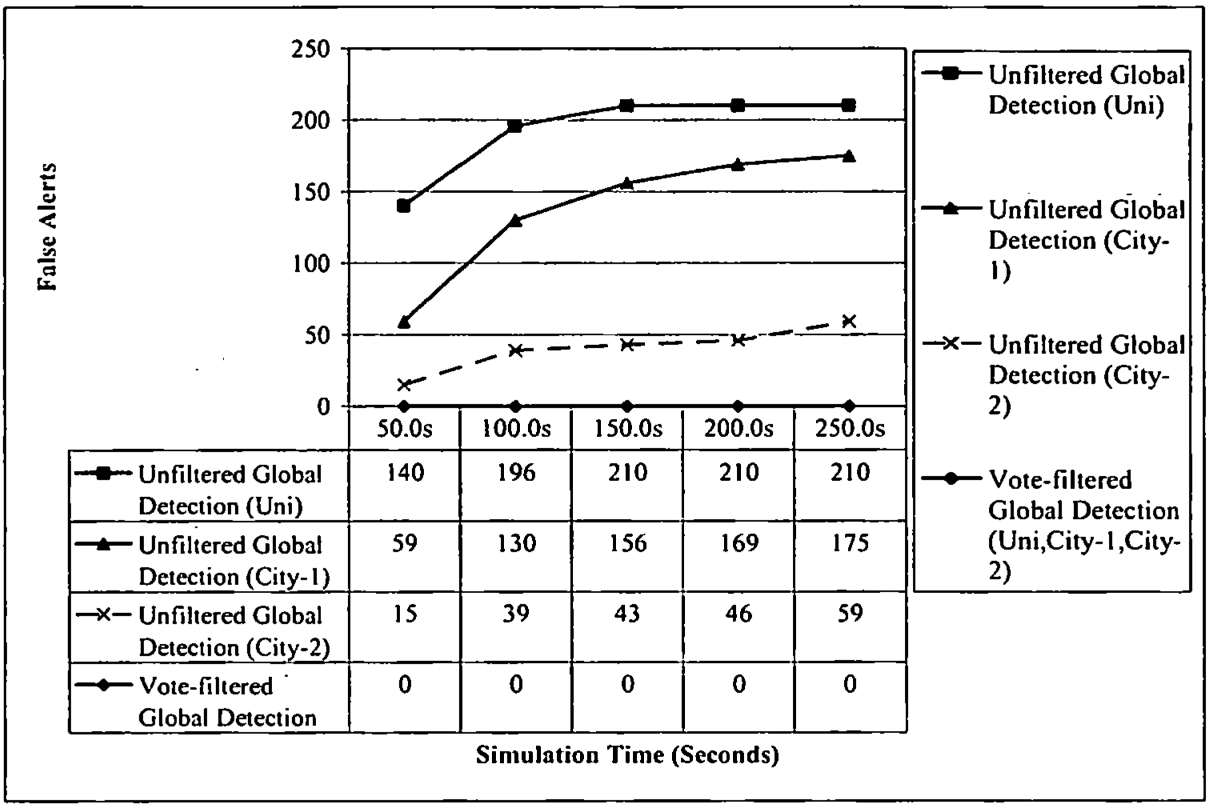
From the simulation results, it can be seen that the performance of the vote-filtered global detection mechanism is slightly lower at the beginning of the simulation compared to the unfiltered global detection mechanism (refer to Figure 8-1) especially in the city-2 environment. This situation happens because in the vote-filtered detection mechanism, each node requires extra time to reach any conclusion regarding the suspicious activities. However, despite the longer detection time, the vote-filtered detection mechanism was still able to match the performance of the unfiltered global detection mechanism as the simulation progressed. For instance, as illustrated in Figure 8-2, the performance of the vote-filtered global detection mechanism in the city-2 environment for 50 seconds simulation time is decreased by 7% compared to the unfiltered global detection. However, as the experiment progressed, and more votes have been gathered by nodes in the network, the vote-filtered global detection mechanism was able to match the performance of the unfiltered global detection mechanism.

For the purpose of investigating the effect of a blackmail attack, several nodes have been selected to play the role of the blackmail attackers in the simulation setups. Those nodes are as illustrated in Table 8-2. There are two types of blackmail attacker, namely an independent blackmail attacker and a colluding blackmail attacker. An independent blackmail attacker does not cooperate with other blackmail attackers in sending false accusations about other well-behaved nodes in the network. Malicious activity from this kind of blackmail attacker could be easily denied by the voting mechanism.

Blackmailer Node	Type of Attack
90	Independent
91	Independent
92	Colluding
93	Colluding
94	Colluding
95	Colluding

Table 8- 2: Blackmail Attackers

The second type of the blackmail attacker, the colluding blackmail attacker, is more difficult to defend against because the attackers are capable of matching the threshold value of the voting mechanism by cooperating with other blackmail attackers. In this subsection, only the effect of an independent blackmail attacker is being investigated. The effect of a colluding blackmail attacker will be investigated and discussed in the next subsection.



	50.0s	100.0s	150.0s	200.0s	250.0s
■ Unfiltered Global Detection (Uni)	140	196	210	210	210
▲ Unfiltered Global Detection (City-1)	59	130	156	169	175
× Unfiltered Global Detection (City-2)	15	39	43	46	59
◆ Vote-filtered Global Detection	0	0	0	0	0

Figure 8- 3: The Effect of an Independent Blackmail Attacker in an Unfiltered and Vote-filtered Global Detection Mechanisms

As illustrated in Figure 8-3, many false alarms have been triggered as a result of false accusations from the blackmail attackers. Within 50.0 seconds simulation time, 15 false alerts have been triggered in the city-2 unfiltered detection mechanism as a result of blackmail attackers' actions, and the number continues to increase as the simulation progressed. The situation is even worse in the denser environments (i.e. city-1 and university campus environments) as more interactions between nodes provide more opportunities for the blackmail attackers to advertise their false alerts. This is not the case in the vote-filtered detection mechanism. In the simulation experiment, as illustrated in Figure 8-3, the voting threshold value has been set to 3, which is higher than the number of independent blackmail attackers that exist in the network (i.e. 2 blackmail nodes). Since the threshold value for nodes to accept any intrusion reports/evidence is higher than the number of independent blackmail attackers, the vote-filtered global detection mechanism is immune against such an attack. However, there is another problem that could jeopardise the reliability of the vote-filtered global detection mechanism. Such problem is that the vote-filtered global detection mechanism is not immune against the second type of the blackmail attacker, which is the colluding blackmail attacker. The next subsection discusses this issue.

8.3.3 Vote-filtered Global Detection vs. Friend-filtered Global Detection

Having presented the potential of global detection mechanism and discussed the capability of the voting mechanism in defending against an independent blackmail attacker, this subsection discusses how the proposed friend-filtered global detection mechanism could minimise the effect of the second type of the blackmail attacker, which is the colluding blackmail attacker. As mentioned earlier, a voting mechanism is capable of denying false accusation attacks from an independent blackmail attacker, but is not immune against the colluding blackmail attacker. The immunity of the voting mechanism is dependent upon 2 factors, namely the voting threshold value, and the number of colluding blackmail attackers in the network. If the number of colluding blackmail attacker is less than the value of the voting threshold, then the voting mechanism is immune against such attack. On the other hand, if the number of colluding blackmail attackers that exist in the network is equal to or more than the number set for the voting threshold, then the reliability of the vote-filtered global detection mechanism is in jeopardy. The higher the threshold value means the global detection mechanism is more resistant against the colluding blackmail attackers. Figure 8-4 and 8-5 illustrate this situation.

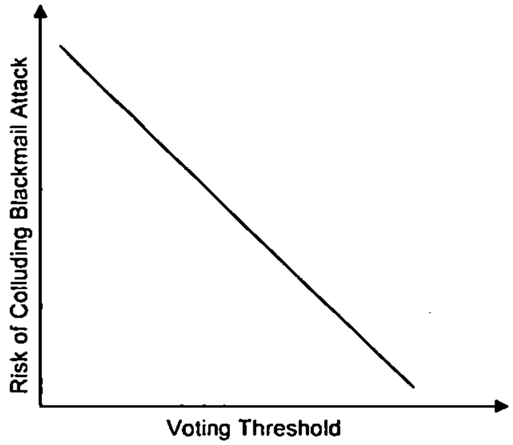


Figure 8- 4: Relation between Voting Threshold Value and the Risk of Colluding Blackmail Attackers

However, if the value of the voting threshold is set too high, the efficiency of the global detection mechanism could decrease significantly. This relationship is represented in Figure 8-5.

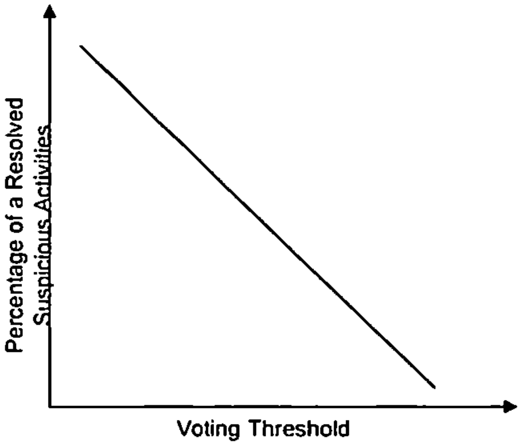
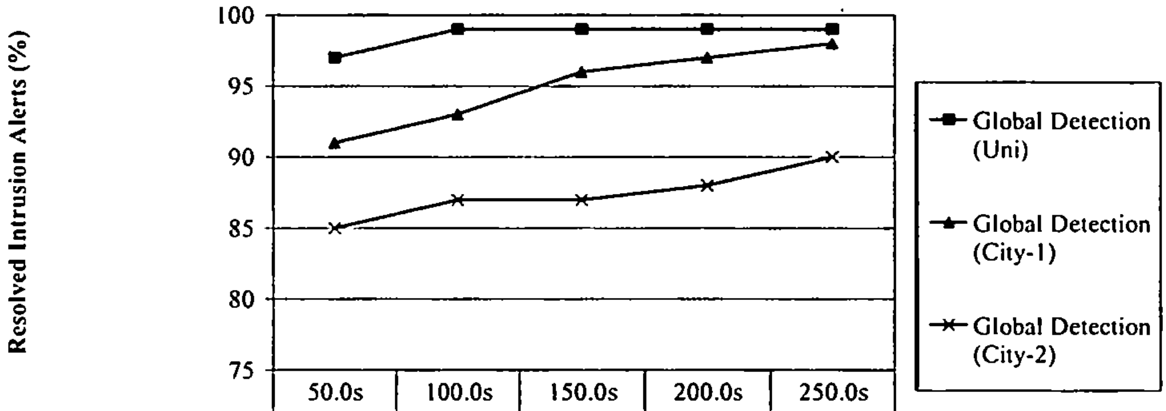


Figure 8- 5: Relation between Voting Threshold Value and the Performance of the Vote-filtered Global Detection Mechanism

Friend-filtered global detection mechanism could solve the problem of determining a suitable voting threshold value as discussed above. Instead of using a voting threshold to minimise the effect of false accusations from anonymous nodes in the network, a friend-filtered detection mechanism could distinguish between genuine and fake intrusion reports based upon the trust relationships that have been established between nodes. However, as mentioned earlier, the reliability of this mechanism could only be ensured with an assumption that each node does not have a relationship with the blackmail attackers. Detailed discussion on the friendship and trust relations issues can be found in Chapter 6. The performance of the friend-filtered global detection mechanisms is as illustrated in Figure 8-6.



	50.0s	100.0s	150.0s	200.0s	250.0s
■ Global Detection (Uni)	97	99	99	99	99
▲ Global Detection (City-1)	91	93	96	97	98
× Global Detection (City-2)	85	87	87	88	90

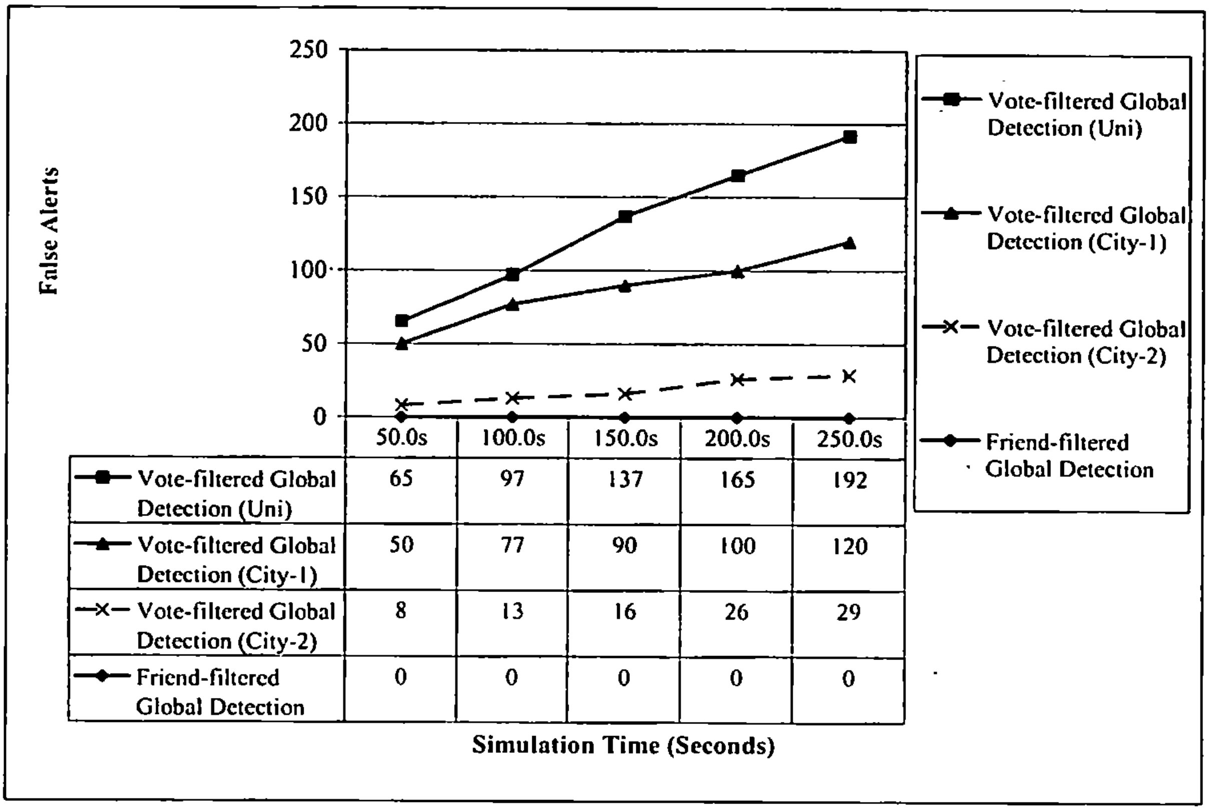
Simulation Time (Seconds)

Figure 8- 6: Performance of the Friend-filtered Global Detection Mechanism

The performance has been measured based upon one-to-five friend relationships. That means each node is assumed to have 5 initial trusted friends. There is no specific reason of choosing 5 initial trusted friends in the simulation as different number of initial trust relationships between nodes (i.e. 1, 3, 7, 10, 14, etc) also could be used to prove the concept proposed in this study. The effect of various initial friendships towards the overall performance of the detection mechanism will not be discussed here as the results are identical as those presented in Chapter 6.

As illustrated in Figure 8-6, the performance of the friend-filtered global detection mechanism is lower than the performance of the vote-filtered global detection mechanism in terms of its effectiveness in helping other nodes to distinguish between normal and anomaly activities. However, this situation only occurs at the early stages of the simulation. For instance, the performance of the friend-filtered global detection mechanism almost matches the performance of the vote-filtered global detection mechanism in both university and city-1 environments at the final stage of the simulation. The result is expected to be much higher if longer simulation times are used in the simulation. Additionally, the number of initial trust relationships (i.e. 5) that has been used in the experiment also had an impact towards the overall results. As proved in Chapter 6, when higher node's initial relationships is being used in the simulation, more interactions between friends can be expected, which could lead to more information about intrusions being shared between themselves.

Another advantage of using a friend-filtered global detection mechanism (although with a slight decrease in performance than the vote-filtered global detection mechanism) is that such a mechanism is capable of protecting the network against the colluding blackmail attacker, a kind of blackmail attack that could not be defended by the vote-filtered global detection mechanism. Figure 8-7 illustrates the problem of a colluding blackmail attacker in the vote-filtered global detection mechanism and shows how the friend-filtered mechanism is immune against such attack.



	50.0s	100.0s	150.0s	200.0s	250.0s
—■— Vote-filtered Global Detection (Uni)	65	97	137	165	192
—▲— Vote-filtered Global Detection (City-1)	50	77	90	100	120
—×— Vote-filtered Global Detection (City-2)	8	13	16	26	29
—◆— Friend-filtered Global Detection	0	0	0	0	0

Figure 8- 7: The Effect of a Colluding Blackmail Attacker in the Vote-filtered and Friend-filtered Global Detection Mechanisms

The result illustrated in Figure 8-7 suggests that the problem of a colluding blackmail attacker in the vote-filtered global detection mechanism is not too apparent in a less dense MANET environment (i.e. city-2 environment) as node's interactions in such an environment are not as many as in the denser environment. However, as the simulation progressed, more and more false alarms as a result of a colluding blackmail attack are detected in the network. On the other hand, the problem of a colluding blackmail attack in the denser environment (i.e. university and city-1 environments) can be seen from the very beginning of the simulation. This is because the blackmail attackers do not have to travel a far distance in such environments to advertise their false alarms to other nodes. The result in Figure 8-7 also shows that the problem of a colluding blackmail attacker does not exist in the friend-filtered global detection mechanism, assuming that no friendship has been established with the blackmail attackers. This result shows that although with a slight decrease in node's performance to globally detect an intrusion, a friend-filtered global detection mechanism is very useful to defend the network against a colluding blackmail attack.

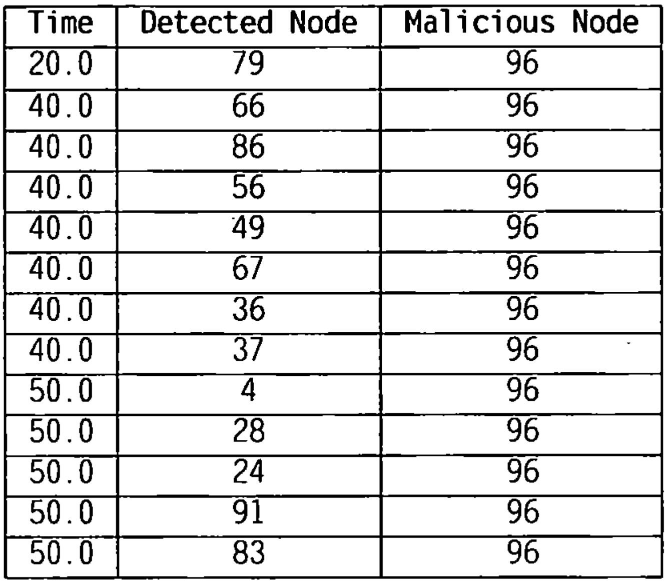
8.4 Global Response Performance

The global response mechanism receives alerts from a local response mechanism and tries to inform as many nodes as possible about the existence of a malicious node in the network. By doing so, more nodes will become aware about the existence of the intrusive nodes, even if they have no experience communicating with those malicious nodes. This section presents results from a series of simulation experiments, which have been carried out to evaluate the performance of the global response mechanism in terms of its efficiency in advertising intrusion alerts to all nodes in the network. Apart from that the existing global response mechanism (i.e. unfiltered and vote-filtered global response mechanisms) as well as the proposed friend-filtered global response mechanism will be evaluated to investigate each of the mechanism's immunity against independent and colluding blackmail attackers. Simulation setups for all the experiments carried out in this section are similar to the one that have been used in the simulations to investigate the performance of the global detection mechanism as in section 8.3.

8.4.1 Unfiltered vs. Vote-filtered Global Response Mechanisms (in the Presence of a Single Blackmail Attacker)

The easiest way to enable global response is by allowing any nodes to broadcast their intrusion alerts to all nodes in the network. By doing so, more nodes could be made aware about the existence of an intrusive node in the network, thus necessary actions could be taken against the malicious nodes. However, similar to the scenario in the global detection mechanism, deploying an unfiltered global response mechanism could expose the network to the blackmail attackers, thus could jeopardise the reliability of the whole intrusion detection system. The most popular method to filter alerts that come from other anonymous nodes in the global response mechanism is by deploying a voting mechanism (Zhang et al., 2003; Rajavaram et al., 2002). By doing so, the problem of false alerts by blackmail attackers could be minimised whilst at the same time still maintain the high responsive rate of the response mechanism.

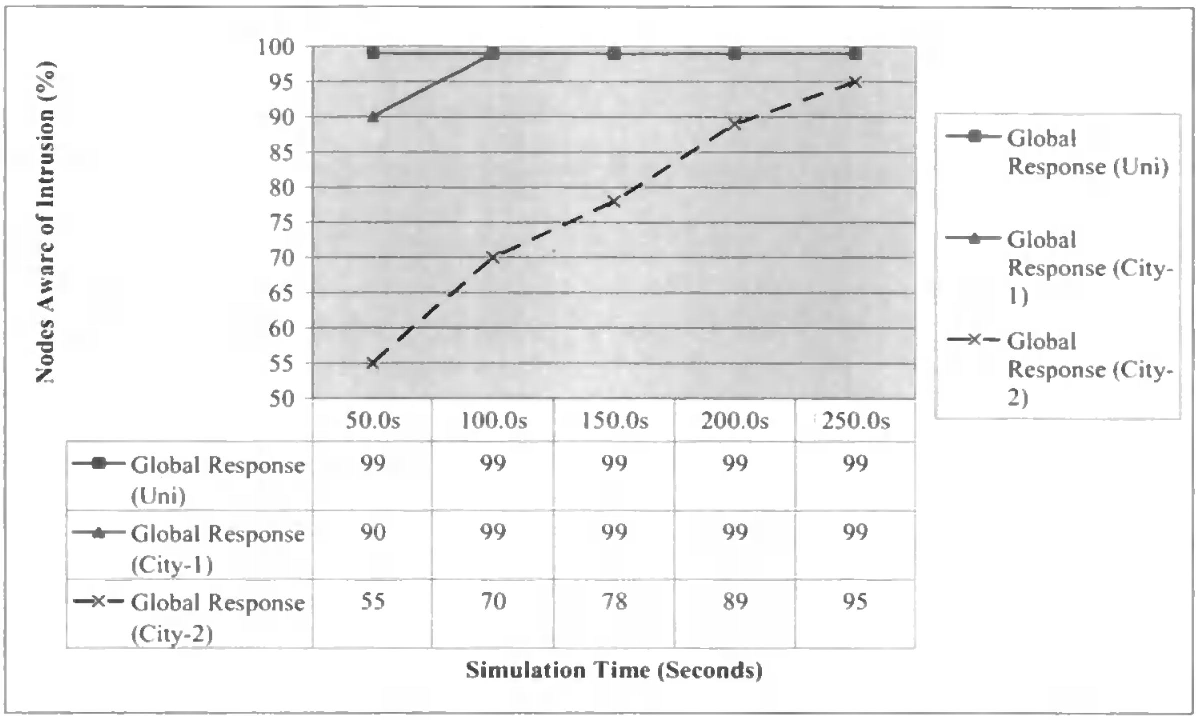
This section presents results from simulation experiments carried out to investigate the performance comparison between the global response mechanism that does not apply any filtering mechanism and the global response mechanism that apply a voting mechanism in defending against the blackmail attackers. Several nodes have been selected to play the role of the blackmail attackers, as illustrated in Table 8-2. Intrusion alerts advertised by nodes in this simulation have came from the list of alerts triggered by the local anomaly detection, as discussed in the previous section. An example of local anomaly detection intrusion alerts is as illustrated in Table 8-3.



Time	Detected Node	Malicious Node
20.0	79	96
40.0	66	96
40.0	86	96
40.0	56	96
40.0	49	96
40.0	67	96
40.0	36	96
40.0	37	96
50.0	4	96
50.0	28	96
50.0	24	96
50.0	91	96
50.0	83	96

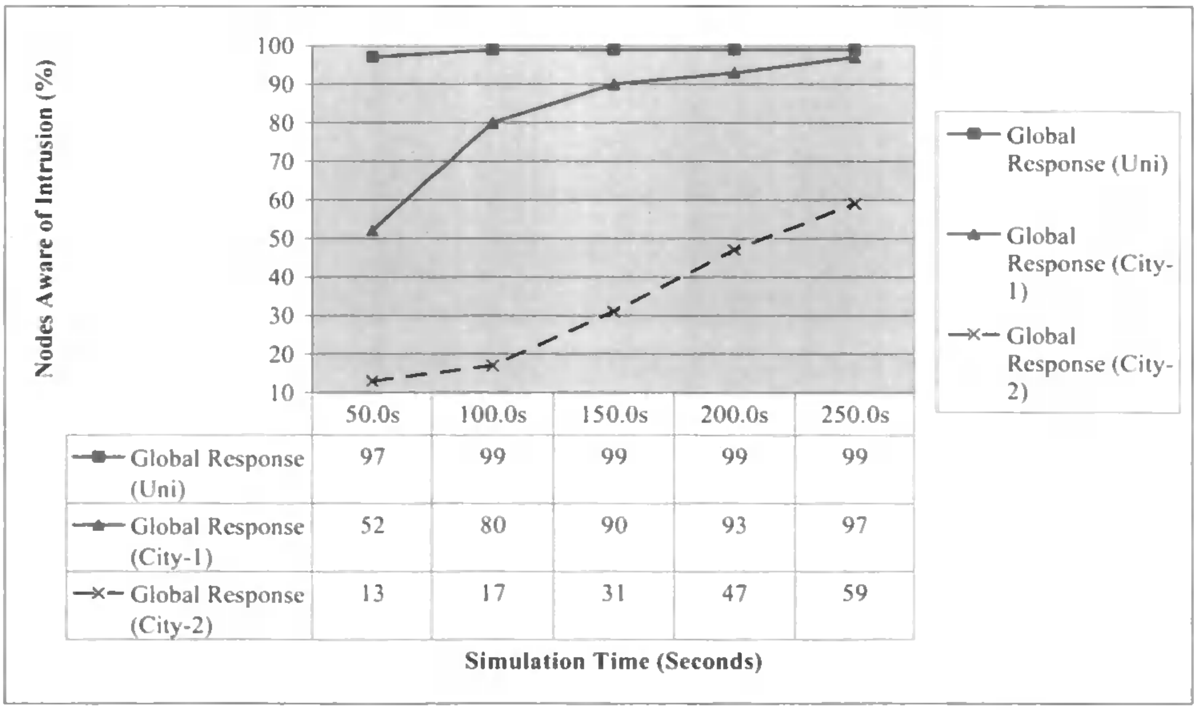
Table 8- 3: Intrusion Alerts from the Local Anomaly Detection Mechanism

Figure 8-8 and 8-9 illustrate the performance comparison between unfiltered and vote-filtered global response mechanisms in terms of their effectiveness in advertising intrusion alerts.



	50.0s	100.0s	150.0s	200.0s	250.0s
Global Response (Uni)	99	99	99	99	99
Global Response (City-1)	90	99	99	99	99
Global Response (City-2)	55	70	78	89	95

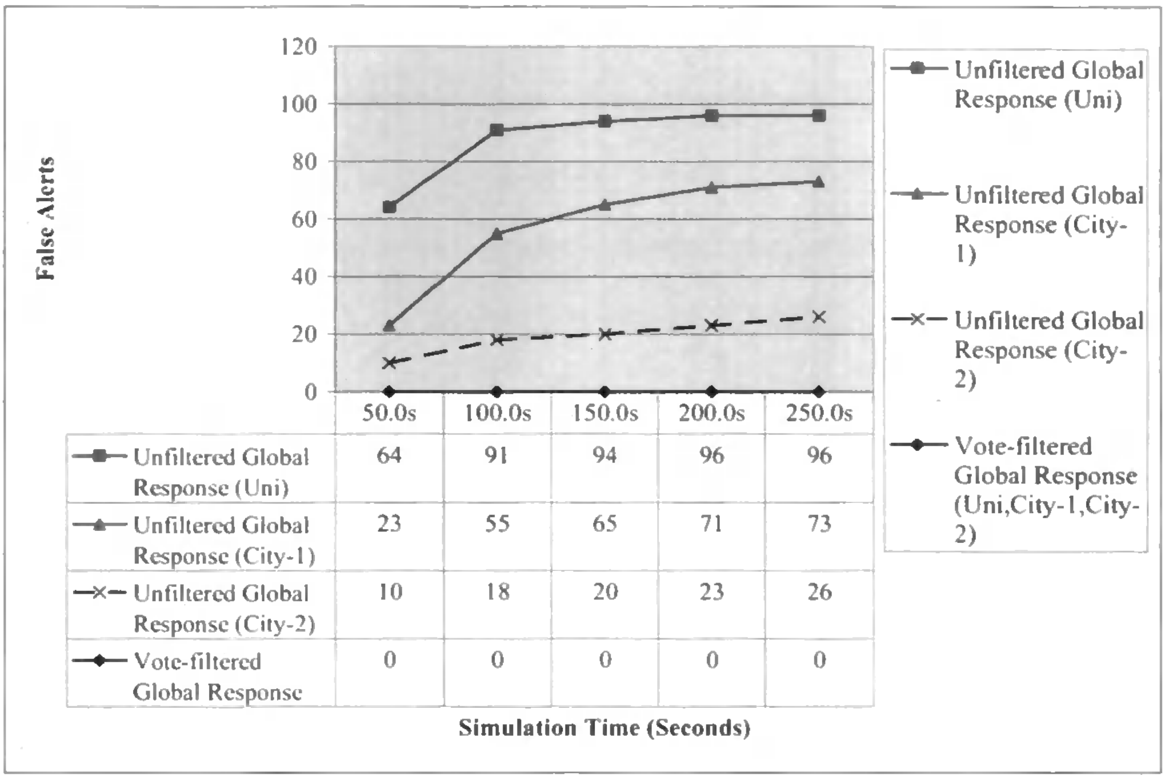
Figure 8- 8: Performance of the Unfiltered Global Response Mechanism



	50.0s	100.0s	150.0s	200.0s	250.0s
Global Response (Uni)	97	99	99	99	99
Global Response (City-1)	52	80	90	93	97
Global Response (City-2)	13	17	31	47	59

Figure 8- 9: Performance of the Vote-filtered Global Response Mechanism

As expected, there was a slight drop in the performance of the vote-filtered global response mechanism compared to the unfiltered global response mechanism, especially in the early stages of the simulation. The main reason for this is because as in the global detection mechanism, nodes in the vote-filtered global response mechanism need to gather enough votes from other nodes before accepting any intrusion alert that being sent to them. However, as the simulation progressed, the vote-filtered global response mechanism seems able to match the performance of the unfiltered global response mechanism especially in the denser MANET environment (i.e. university and city-1 environments). For the case of a less dense environment (i.e. city-2) although the performance of the vote-filtered global response mechanism seems dropped almost 50% the performance of the unfiltered global response mechanism, the total number of nodes that are aware about the intrusive node in the network is still much higher than the initial number of nodes that are aware about the intrusion when only a local response mechanism is being deployed. Moreover, the vote-filtered global response mechanism has an advantage over the unfiltered global response mechanism, which is its ability to defend the network against an independent blackmail attacker. Figure 8-10 summarises results from simulation experiments that support this statement.



	50.0s	100.0s	150.0s	200.0s	250.0s
Unfiltered Global Response (Uni)	64	91	94	96	96
Unfiltered Global Response (City-1)	23	55	65	71	73
Unfiltered Global Response (City-2)	10	18	20	23	26
Vote-filtered Global Response	0	0	0	0	0

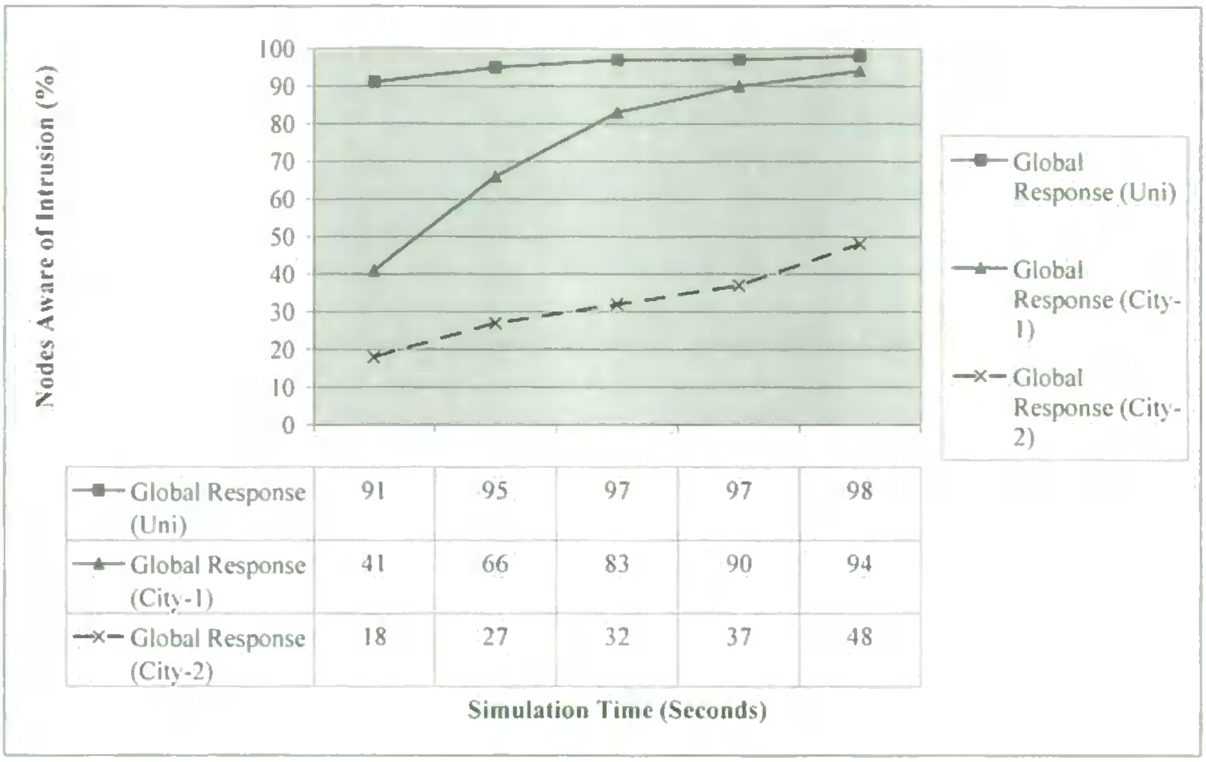
Figure 8- 10: The Effect of Independent Blackmail Attackers in the Unfiltered and Vote-filtered Global Response Mechanisms

It is true that in the unfiltered global response mechanism, any intrusion alert that has been triggered by the local misuse or anomaly detection mechanism could be sent across the network in a very quick time especially in an environment where nodes are located close to each other (e.g. university environment). However, as shown in Figure 8-10, such an advantage could bring a serious problem in the network. In the simulation experiment, two independent blackmail nodes (i.e. nodes 90 and 91) constantly send false alerts to the neighbouring nodes claiming that node 0 is malicious based upon their own local anomaly/misuse detection engine. Since there is no mechanism to distinguish between a genuine and a false alert, the fake alerts that have been advertised by those blackmail nodes are being sent across the network, thus more and more nodes are receiving and re-advertising the false alerts. As can be seen in Figure 8-10, the problem becomes worse and uncontrollable as the simulation progressed in all three MANET environments. Such a problem can be eased by having some sort of filtering mechanism that could help nodes to distinguish between fake and genuine alerts. The next subsection discusses this issue.

8.4.2 Vote-filtered vs. Friend-filtered Global Response Mechanisms (in the Presence of a Colluding Blackmail Attacker)

This subsection presents result from simulation experiments that have been carried out to investigate the effectiveness of a friend mechanism in preventing nodes from receiving false alerts in the global response mechanism. The simulation results also should provide an indication whether the introduction of such a filtering mechanism has an impact towards the overall performance of the global response mechanism (i.e. the percentage of nodes that aware about the existence of a malicious node).

As illustrated in Figure 8-11, the number of nodes that can be made aware about the existence of the malicious nodes in the friend-filtered global response mechanism is not much different from the vote-filtered global response mechanism. Although there was a slight decrease in all three simulated environments, the results were obtained with an assumption that each node has 5 initial trusted friends. The result is expected to be better if higher number of initial trusted friends is being used in the simulation sets. For the discussion on the effects of various numbers of initial trusted friends, please refer to Chapter 6.

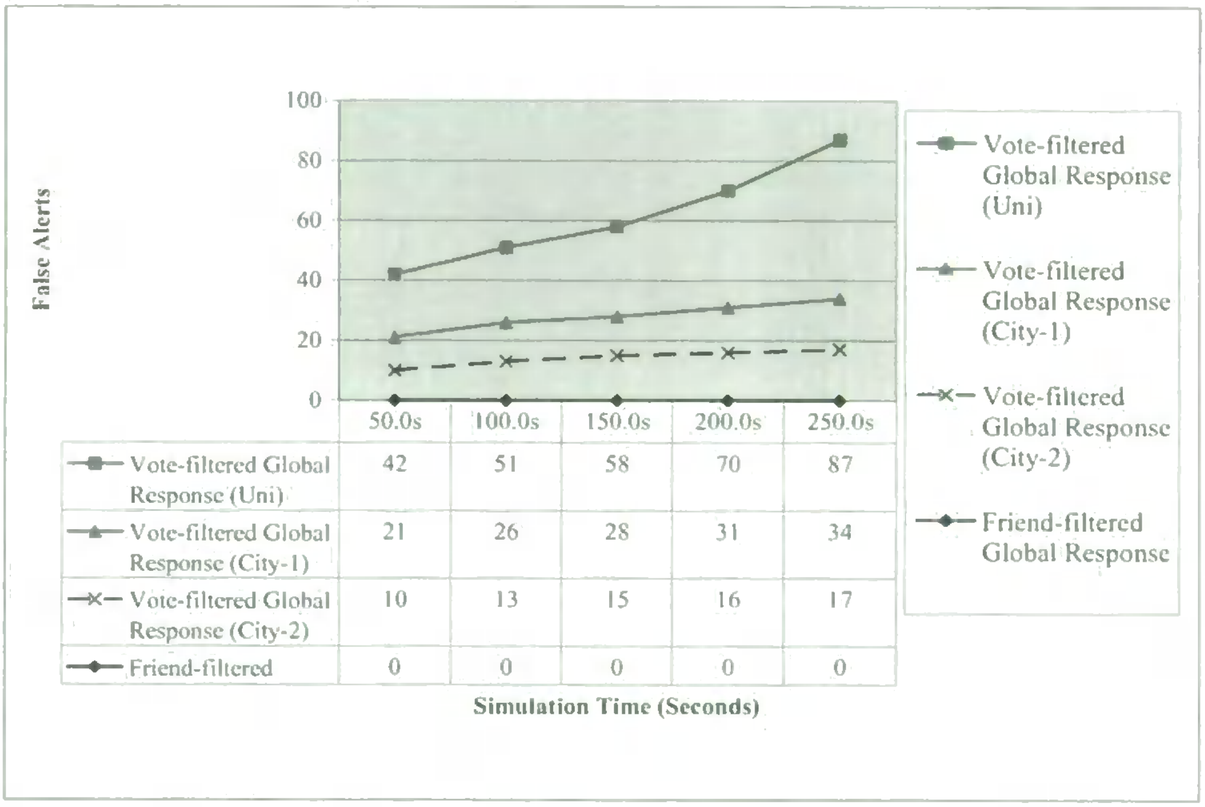


Global Response (Uni)	91	95	97	97	98
Global Response (City-1)	41	66	83	90	94
Global Response (City-2)	18	27	32	37	48

Figure 8- 11: Performance of the Friend-filtered Global Response Mechanism

Another significant advantage of having a friend-filtered instead of a vote-filtered global response mechanism is that it is capable to defend the network against a colluding blackmail attack. Figure 8-12 illustrates how serious the problem of a colluding blackmail attack could be in the vote-filtered global response mechanism. The trend of the graph shows that in all three simulated MANET environments, the problem of a colluding blackmail attacker is increasing over a period of time. The problem might not seem as a big issue in city-2 environment, especially at the early stages of the simulation. This is because, in such a less dense environment, the colluding blackmail attackers need to travel longer distances to collaboratively advertise their false alerts. In addition, large distances between nodes also lessen the chances of nodes exchanging false alerts between them.

However, in the denser MANET environments, such as in the university and city-1 environments, the effect of a colluding blackmail attacker can be seen problematic from the beginning of the simulations. For instance, in the university environment, more than half nodes in the network have received false alerts either directly from the colluding blackmail attackers or indirectly through the intrusion alerts exchange process with the neighbouring nodes. The problem becomes more evident as the simulation progressed as more and more nodes in the network receive the false alerts.



	50.0s	100.0s	150.0s	200.0s	250.0s
Vote-filtered Global Response (Uni)	42	51	58	70	87
Vote-filtered Global Response (City-1)	21	26	28	31	34
Vote-filtered Global Response (City-2)	10	13	15	16	17
Friend-filtered	0	0	0	0	0

Figure 8- 12: The Effect of a Colluding Blackmail Attacker in the Vote-filtered and Friend-filtered Global Response Mechanisms

The problem of a colluding blackmail attack is not an issue in the friend-filtered global response mechanism. Instead of accepting all the alerts that have been advertised by the neighbouring nodes, nodes in the friend-filtered global response mechanism only accept alerts that have been broadcasted by the trusted friends. However, this mechanism relies upon one assumption, which is all the trusted friend nodes are well behaved and not malicious. Figure 8-12 shows that the problem of a colluding blackmail attacker could be eliminated with the introduction of a friend-filtered mechanism in the network.

8.5 Conclusion

This chapter presented results from simulation experiments carried out to evaluate the performance of the proposed friend-filtered global detection and response mechanisms. Three MANET settings which differ from each other based on the network density level have been used to investigate if the proposed mechanism is suitable to be used in various MANET environments. The main objective of the simulation experiments is to show how such a friendship mechanism as proposed in Chapter 6, coupled with suitable intrusion detection mechanisms as presented in Chapter 7 could solve many issues faced by the MANET IDS. This study focuses on the node's anonymity problem faced by MANET IDS, which should be the main reason that causes a colluding blackmail attack to occur in the network. For a better understanding of the problem, this chapter has presented the performance comparisons between the proposed friend-filtered global detection/response mechanisms and other global detection/response techniques.

The first performance comparison has been made between unfiltered and vote-filtered global detection mechanisms. From this set of experiments, several observations have been made. Firstly, results from the experiment suggested that a global detection mechanism is indeed able to speed up the decision making process of the suspicious activity, which sometimes could not be decided by the local detection engine. The second observation is that a global detection mechanism without a proper filtering mechanism to differentiate between genuine and fake neighbouring nodes' opinions about the suspicious activities could lead to many false alarms being triggered by the global detection mechanism. This study refers this kind of problem as an independent blackmail attack, and the impact could be minimised by implementing a voting mechanism.

The second performance comparison has been made between the vote-filtered and friend-filtered global detection mechanisms. This experiment was carried out to show that although the vote-filtered global detection mechanism is able to minimise the impact of an independent blackmail attacker, such a mechanism is not immune against a colluding blackmail attacker. Once the number of the colluding blackmail nodes that exist in the network is equal or more than the value set for the voting threshold, the reliability of the vote-filtered global detection mechanism is in jeopardy. Results from the simulations in this experiment set proved that the friend-filtered global detection mechanism is capable to solve such a problem, whilst at the same time maintaining the high detection rate of the global detection mechanism as achieved in the vote-filtered detection mechanism. In addition, similar experiments also have been carried out to evaluate the performance of the global response mechanism, and as expected the results are not much different from the global detection mechanism.

The next chapter summarises the research that has been conducted in this study and discusses the achievements as well as the limitations of the proposed IDS framework. Unaddressed important issues and suggestions for future work extensions are also outlined.

Chapter 9

Conclusions and Future Work

9.1 Introduction

This chapter concludes the thesis by first highlighting the contributions and achievements that have been made in this study. This is followed by a discussion of the limitations and challenges experienced throughout the process of completing the study. Finally, the chapter outlines some suggestions for future development to extend the research areas as well as presenting the directions of the study.

9.2 Achievements

The aim of this research was to propose a novel IDS framework that is suitable for MANET environments and capable of protecting such a network against security threats. In the process of realising the idea, several achievements have been made that contribute to the area of the study. Specifically, these achievements are as follows:

- *In depth investigation of MANET characteristics that distinguish such a network from other wireless networks*

  Investigation of MANET characteristics provides useful information on how such a network operates, which is important to help in designing reliable security measures that are suitable for such a network. By identifying the characteristics as well as understanding the nature of its operations, knowledge of how such a network might be threatened by attackers were established. Such information is necessary to be fully understood in order to come out with a reliable security measure for a MANET.

- *A detailed investigation and analysis of attacks that could be launched against MANETs*

Attacks that could be launched against MANET have been identified, which include attacks that are unique in such a network environment, as well as attacks that were inherited from other types of wireless networks. Knowledge of the attacking scenarios and strategies in MANET environments is essential because it could help in determining the appropriate and efficient strategies to secure such a network. Information gathered in this investigation includes different kind of attackers that exist in a MANET environment, as well as their strategies in launching attacks. Additionally, the investigation has also identified the most important feature of MANET that need to be secured, which is the routing mechanism.

- *A comprehensive evaluation of the existing security measures proposed for MANET*

A thorough investigation on existing security measures proposed for MANET have been carried out to give an idea on what have been done and what still needs to be done to secure such a network from security threats. The investigation covers the whole of the computer security lifecycle as suggested by King (2002), which includes prevention, detection and response mechanisms. Such an investigation provides useful information in designing a novel security scheme for MANET that aims to improve deficiencies as well as to suggest solutions for unaddressed issues in the existing techniques.

- *The design of a novel IDS framework*

A novel IDS framework that aims to improve the deficiencies of existing solutions has been designed. The novelty of the framework is the introduction of a friendship concept to deny false information from being traversed across the network in the global detection and response mechanisms. In addition, the proposed framework also provides alternative solutions in enabling dynamic update to a misuse signature database via the trust and signature management mechanisms.

- *The evaluation of the proposed IDS framework through a series of simulation experiments*

The proposed IDS framework has been evaluated through a series of simulation experiments to establish its performance. Results from simulations suggested that the proposed IDS framework is capable of protecting nodes in the network from receiving false information/alerts that have been a major problem in global detection and response mechanisms. More importantly, the overall performance of the proposed IDS framework in speeding up the detection and response process has not been compromised as a result of the introduction of the friendship mechanism to filter the false information/alerts.

9.3 Limitations

There are several limitations and challenges that have been faced throughout the process of completing this study, and they are as follows:

- *Difficulty in assigning initial trusted friends for each MANET node*

As discussed in Chapter 6, the trust relationship between nodes is a subjective issue. Some individuals might have many people that they trust, whilst others might just have a few on their list. Trust relationships are created and torn down based on many reasons. For instance, it could be based on experience, or recommendation from other people. Since this study is carried out in a simulation manner, it is impossible to get an exact number of trust relationships possessed by each node. As a result, the number of initial trusted friends for each node has to be estimated to match as close as possible the real world relationships. This study follows the finding from the Britsocat survey (Britsocat, 1995), which suggested that on average a person has 14 close friends that they could trust. Throughout the simulations, various numbers of initial friends possessed by each node (i.e. between

1 and 14) have been used to see the impact of such a variant relationships towards the overall performance of the proposed IDS framework.

- *Simple operational misuse and anomaly detection engines*

The implementation of high performance misuse and anomaly detection engines is not the main focus of this study. A focus of this study is to prove that the friendship concept is capable to eliminate false accusations/alerts problem in the IDS global detection and response mechanisms. Misuse and anomaly detection engines are responsible to provide inputs that will trigger the global detection and response mechanisms. In a real world implementation, it is always a desire to have complete and efficient local misuse and anomaly detection engines that will provide better input to the global detection and response mechanisms. However, since the focus is not on this matter, simple operational misuse and anomaly detection engines are adequate in this study.

- *Limited selections of misuse signatures and normal network behaviour profiles*

Apart from having a simple operational local detection engine, this study also utilises a limited selection of misuse signatures and normal network behaviour profiles. This limitation is because of several reasons, outlined as follows:

- Research on a MANET misuse detection engine is still immature and researchers admitted that the process of creating the attack signatures is not an easy task due to MANET's unique characteristics.

- Similar to misuse detection engine, preparing a list of normal network behaviour for a MANET is not easy. The network behaves differently based upon the type of routing protocol being used. As a result, most researchers proposed that the network's normal behaviour profiles should be designed based upon the protocol being deployed in the system.

- A complete list of misuse signatures and normal network profiles is not essential in this study, as the use of only a few of them is adequate to prove the proposed friendship concept.

- *Difficulty in modelling general MANET environments*

MANET could exist in many forms. It does not have a fixed network topology where nodes can join and leave the network anytime and anywhere. Apart from that, there is also no fixed network boundary where nodes can communicate to each other miles away via multi hop communications. Network density is varying depending on how nodes are located. All the aforementioned situations have an impact upon how nodes communicate in the network and thus have an impact on the overall performance of the proposed IDS framework. It is impossible to evaluate the performance of the proposed IDS framework in all possible MANET environments. However, for better simulation results, the evaluation process in this study has been made in three predetermined MANET environments (i.e. university campus, city-1, and city-2) that represent different network density levels.

- *Limited evaluation of the proposed IDS framework*

The evaluation process of the proposed IDS framework has been made through a series of simulation experiments. There is a debate that questions the reliability of the simulation results compared to the physical implementation. However, the practicability of simulation techniques, especially in evaluating a new concept, should not be put aside. Moreover, conducting a physical evaluation in this study would not be practical considering the amount of physical MANET nodes that need to be made available.

9.4 Future Work

A number of suggestions for future work outside the scope of this study have been identified and they are as follows:

- *Implementation of fully dynamic real time trust relationship between nodes*

  In this study, trust relationships between nodes are based upon the number of initial direct friends preset for all nodes at the beginning of the simulation. The initial direct trusts are allowed to be shared between the direct friends to create a new set of a trusted friend, namely the indirect friend. This study has successfully proved that with an adequate number of initial direct friends, trust relationships between nodes could be expanded across the network to create a trusted community (as detailed in Chapter 6), which then could be used to solve many trust-related issues in MANET environments. However, the friendship concept introduced in this study did not take into consideration the dynamic nature of the trust relationships (i.e. the direct trust could be established and ended anytime). For instance, a new trust relationship could be established when there are a lot positive recommendations from others about the particular node. In another scenario, the existing trust could be ended when there are a lot negative opinions/feedbacks about the particular friend. This dynamic trust issue might have an impact towards the overall performance of the proposed framework, thus investigation on this matter in a future work is another possible contribution to MANET security research field. However, although this study did not include an observation on this dynamic relationship issue, the results obtained from the simulation experiments still could be considered valid. This is because neither new direct trusts nor the revoked friendships have been introduced in the system that could create bias to the end results.

- *Prototype implementation of the proposed IDS framework*

The proposed IDS framework in this study has been developed in simulation software (i.e. the NS-2) to prove the applicability of its concepts. The decision of having a simulation implementation was made because it is an inexpensive way in terms of money, time, and manpower compared to a physical testbed implementation to prove a new research idea that could not guarantee any promising result. In addition, the simulation has also been chosen to overcome the difficulties in preparing sufficient number of physical mobile nodes, as well as for simplicity to simulate the random movements of the mobile nodes. Since results obtained from the simulations have proved the applicability of the proposed friendship concept, the next step of the research is to develop a prototype of the proposed IDS framework and to further investigate its performances.

- *Real world evaluation via a case study*

Results from the simulation experiments carried out in this study suggested that the friendship concept is very practical to assist in the global detection and response mechanisms, especially in the presence of the colluding blackmail attackers. However, as claimed in (Andel & Yasinac, 2006), in most cases, results from simulation experiments are questionable as the controlled environment/attributes preset in the simulation are often different from the expected situation as it should be in a physical implementation. It might be difficult to make available 100 mobile nodes and investigate how the proposed IDS framework will perform in a physical implementation if the research has to be carried out in a laboratory with limited space and manpower (to create node's mobility scenario). A more practical solution is to select a group of students in a university campus to play the roles of the mobile nodes, which not only could solve the problems of limited manpower and space but also could bring real dynamic trust relationships into the system that they established between themselves. Carrying out this investigation is another potential extension for future work.

9.5 Conclusions

This study has presented a novel IDS framework that focuses upon the global detection and response mechanisms to minimise the effects of false accusations caused by the blackmail attackers. The important concept behind the proposed framework is the introduction of a trust chain, which could be established in the network through a friendship relation. Such a chain of trust created a trusted community in a network that not only could ensure a secure operation of the global detection and response mechanisms as proposed in this study, but also could solve many other trust-related issues in MANET environments.

The proposed IDS framework has been developed and evaluated in the NS-2 simulation software to prove its concepts as well as to investigate its performance, compared to the existing approaches. It utilises a simple operational misuse and anomaly detection techniques that have been proposed by previous researchers as the local detection mechanism is not the focus of the study. Besides such simple local misuse and anomaly detection engines also are sufficient to provide inputs for the global detection and response mechanisms.

Results from the simulation experiments in Chapter 6 proved that the proposed direct and indirect friendship concepts are capable to expand the limited trust relationships that initially exist in the network. With more and more trust relationships being established, the deployment of security measures that depend heavily on this relationship becomes more practical. Results from simulation experiments in Chapter 8 provide evidence on this matter as the proposed IDS framework that has been designed based upon the friendship concept performed very well.

Completing this study is just the beginning of more research studies focusing on MANET related issues, which will not be limited to the security issues. As emphasised at the beginning of the thesis, research studies on MANET-related issues are still few and immature, similar to the immaturity of the network itself (at least for the case of an open MANET). With more researchers working in addressing MANET-related issues it is hoped that this new exciting technology could become more popular and play an important role in supporting daily communication needs.

References

[1] Abad, C., Taylor, J., Sengul, C., Yurcik, W., Zhou, Y., Rowe, K. (2003). "Log Correlation for Intrusion Detection: A Proof of Concept". 19th Annual Computer Security Applications Conference (ACSAC), pp255-265. Las Vegas.

[2] Adamic, L. (1999). "The Small World Web". In Proc. of Eur. Conf. on Digital Libraries (ECDL), pp.443-452.

[3] Albers, P., Camp, O., Percher, J.-M., Jouga, B., Mé, L., Puttini, R. (2002). "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches". In Proc. of the First International Workshop on Wireless Information Systems (WIS-2002), pp.1-12.

[4] Al-Jaroodi, J. (2002). "Security Issues in Wireless Mobile Ad Hoc Networks at the Network Layer". Technical Report TR02-10-07. Computer Science and Engineering, University of Nebraska-Lincoln.

[5] Andel, T. R., Yasinac, A. (2006). "On the Credibility of Manet Simulations". IEEE Computer Magazine, Vol. 39, No. 7, pp.48-54.

[6] Awerbuch, B., Holmer, D., Rotaru, C. N., Rubens, H. (2002). "An on-demand secure routing protocol resilient to byzantine failures". In ACM Workshop on Wireless Security (WiSe'02), pp.21-30. Atlanta, Georgia.

[7] Bajaj, L., Takai, M., Ahuja, R., Tang, K., Bagrodia, R., Gerla, M. (1999). "GloMoSim: A scalable network simulation environment". Technical Report 990027. UCLA Computer Science Department.

[8] Balfanz, D., Smetters, D., Stewart, P., Wong, H. (2002). "Talking to Strangers-Authentication in Ad Hoc Wireless Networks". In Proc. of the 9th Annual Network and Distributed System Security Symposium (NDSS). San Diego, California.

[9] Barbeau, M., Hall, J., Kranakis, E. (2006). "Detecting Impersonation Attacks in Future Wireless and Mobile Networks". Mobile Ad-hoc Networks and Sensors workshop (MADNES). Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 4074, pp.80-95.

[10] Bhargava, S., Agrawal, D. P. (2001). "Security Enhancements in AODV protocol for Wireless Ad Hoc Networks". Vehicular Technology Conference, pp.2143-2147. Atlantic City.

[11] Bhargavan, K., Gunter, C. A., Kim, M., Lee, I., Obradovic, D., Sokolsky, O., Viswanathan, M. (2002). "Verisim: Formal analysis of network simulations". IEEE Transactions on Software Engineering.

[12] Blazevic, L., Buttyan, L., Capkun, S., Giordano, S., Hubaux, J. P., Le Boudec, J. Y. (2001). "Self-Organization in Mobile Ad-Hoc Networks: the Approach of Terminodes". IEEE Communications Magazine, Vol. 39, Issue 6, pp.166- 174.

[13] Bouam, S., Othman, J. B. (2003). "Data Security in Ad hoc Networks using MultiPath Routing". In Proc. of the 14th IEEE PIMRC, pp.1331-1335.

[14] Brannstrom, R., Kodikara E. R., Ahlund, C., Zaslavsky, A. (2006). "Implementing global connectivity and mobility support in a wireless multi-hop ad hoc network". In Proc. of the 4th Asian International Mobile Computing Conference (AMOC). India.

[15] Britsocat. (1995). "Number of close friends". http://www.britsocat.com/BodySecure.aspx?control=BritsocatMarginals&var=PALS&SurveyID=224.

[16] Buchegger, S., Boudec, J. –Y. L. (2001). "IBM Research Report: The Selfish Node - Increasing Routing Security for Mobile Ad Hoc Networks". RR No. 3354.

[17] Buchegger, S., Boudec, J. -Y. L. (2002a). "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes-Fairness In Dynamic Ad-hoc NeTworks)". In Proc. of 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02), pp.80-91. Lausanne, Switzerland.

[18] Buchegger, S., Boudec, J. –Y. L. (2002b) "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks". In Proc. of the Tenth Euromicro Workshop on Parallel, Distributed and Network-based Processing, pp.403-410. Canary Islands, Spain.

[19] Burg, A. (2003). "Ad hoc networks specific attacks". Seminar Paper. Seminar Ad Hoc Networking: concept, applications, and security. Technische Universität München, Institut für Informatik.

[20] Buttyan, L., Hubaux, J. -P. (2001). "Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks". Technical Report DSC/2001/001. Department of Communication Systems, Swiss Federal Institute of Technology - Lausanne.

[21] Buttyan, L., Hubaux, J.-P. (2003). "Report on a Working Session on Security in Wireless Ad Hoc Networks". Mobile Computing and Communications Review, Vol. 7, Number 1, pp.74-94.

[22] Capkun, S., Buttyan, L., Hubaux, J. –P. (2003a). "Self-Organized Public Key Management for Mobile Ad Hoc Networks". IEEE Transactions on Mobile Computing, pp.52-64.

[23] Capkun, S., Hubaux, J. –P., Buttyan, L. (2003b). "Mobility Helps Security in Mobile Ad Hoc Networks". In Proc. of MobiHoc'03, pp.46-56. Annapolis, Maryland, USA.

[24] Castelfranchi, C., Falcone, R. (1998). "Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification". Proceedings of the 3rd International Conference on Multi Agent Systems, pp72-79. Paris, France..

[25] CERIAS (2006). "Intrusion Detection". http://www.cerias.purdue.edu/about/history/coast_resources/intrusion_detection/.

[26] Chan. H., Perrig, A. (2003). "Security and Privacy in Sensor Networks". IEEE Computer, Vol. 36 No. 10, pp103-105.

[27] Chaplin, K. (2002) "Wireless LANs vs. Wireless WANs". http://www.sierra wireless.com/documents/corporate/2130273_WWAN_v_WLAN.pdf. Sierra Wireless White Paper.

[28] Chen, D., Garg, S., Trivedi, K. S. (2002). "Network survivability performance evaluation: a quantitative approach with applications in wireless ad-hoc networks". In Proceedings of the 5th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems, MSWiM '02, pp.61-68. USA.

[29] Chlamtac, I., Conti, M., Liu, J. J.-N. (2003). "Mobile ad Hoc Networking: imperatives and challenges". Ad Hoc Networks Journal, Vol. 1, Number 1, pp.13-64.

[30] Choi, J. Y. (2003). "Security problems for ad hoc routing protocols". Survey Paper. Computer Science Department, New York University.

[31] Chung, J., Claypool, M. (2005). "NS by Example". http://nile.wpi.edu/NS/.

[32] CISE Wireless Project. (2002). "A Brief Description of the Wireless Project". http://www.cs.ucsb.edu/projects/wireless/researchdesc/network/eroyer.shtml.

[33] Clausen, T. H. (2003). "Classification of MANET unicast routing protocols". http://www.soi.wide.ad.jp/class/20030000/slides/05/33.html.

[34] Clausen, T. H. (2006). "MANET Autoconfiguration Standardization". http://www.lix.polytechnique.fr/hipercom/index.php?option=com_content&task=view&id=126&Itemid=90.

[35] CMP Media LLC. (2003). "PCSes Are Coming". http://www.byte.com/art/9405/sec7/art2.htm.

[36] Cohen, W. W. (1995). "Fast Effective Rule Induction". In Proc. of 12th International Conference on Machine Learning, pp.115-123. Morgan Kaufmann.

[37] Contos, B. (2006). "Enemy at the Watercooler: Insider IT Threats Increasing". http://www.linuxinsider.com/story/security/49652.html.

[38] Debar, H., Dacier, M., Wespi, A. (1998). "Towards Taxonomy of Intrusion Detection Systems". IBM Zurich Research Laboratory. Ruschlikon, Switzerland.

[39] Dell US. (2006a). "Dell Wireless Solutions". http://dsnimg.dell.com/images/external/images/Applications/10898609_WWAN.jpg.

[40] Dell US. (2006b). "Dell Wireless Solutions". http://dsnimg.dell.com/ images/external/images/Network/10898608_WPAN.jpg.

[41] Desilva, S., Boppana, R. V. (2004). "On the impact of noise sensitivity on performance in 802.11 based ad hoc networks". IEEE Communications, Vol. 7, pp.4372-4376.

[42] Douceur, J. R. (2002). "The Sybil Attack". In Proc. of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), pp.251-260. MIT Faculty Club, Cambridge, MA, USA.

[43] Fell, J. (2002). "Combination of Misuse and Anomaly Network Intrusion Detection Systems". http://www.kaleton.com/infosec/idspaper.html.

[44] Forristal, J., Schuh, J., Shipley, G. (2005). "Think Like an Attacker". http://www.networkcomputing.com/showArticle.jhtml?articleID=163105313.

[45] Frean, A. (2003). "Friends like These: The relationships that shape our lives". ESRC Society Today - The Edge, Issue 13, pp.8-10.

[46] Ghazizadeh, S., Ilghami, O., Sirin, E., Yaman, F. (2002). "Security-Aware Adaptive Dynamic Source Routing Protocol". In Proc. of 27th Conference on Local Computer Networks, pp.751-760.

[47] Gosh, T., Pissinou, N., Makki, K. (2005). "Towards designing a trusted routing solution in mobile ad hoc networks". Mobile Networks and Applications, Vol. 10, Issue 6. Kluwer Acacemic Publisher.

[48] Guare, J. (1990). "Six Degrees of Separation: A Play". Vintage Book. New York.

[49] Gwalani, S., Srinivasan, K., Vigna, G., Beding-Royer, E. M., Kemmerer, R. (2004). "An Intrusion Detection Tool for AODV-based Ad hoc Wireless Networks". In Proceedings of the Annual Computer Security Applications Conference, pp.16-27. Tucson, AZ.

[50] Haas, Z. J., Deng, J., Liang, B., Papadimitratos, P., Sajama, S. (2002). "Wireless Ad Hoc Networks". Draft version of Encyclopaedia of Telecommunications, John Wiley.

[51] Hafslund, A., Anderson, J. (2006). "2-Level Authentication Mechanism in an Internet connected MANET". In 6th Scandinavian Workshop on Wireless Ad-hoc Networks. Stockholm.

[52] Helmy, A. (2003). "Small Worlds in Wireless Networks". IEEE Communications Letters, Vol. 7, No. 10.

[53] Hu, Y. -C., Johnson, D. B., Perrig, A. (2002a). "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks". In Fourth IEEE Workshop on Mobile Computing Systems and Applications, pp.3-13. Callicoon, New York.

[54] Hu, Y. –C., Perrig, A., Johnson, D. B. (2002b). "Ariadne: A secure on-demand routing protocol for ad hoc networks". In Proc. of the Eighth ACM International Conference on Mobile Computing and Networking (Mobicom 2002), pp.12-23. Georgia, USA.

[55] Hu, Y. -C., Perrig, A., Johnson, D. B. (2003). "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks". In Proc. of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), vol. 3, pp.1976-1986. San Francisco, CA.

[56] Huang, D., Sinha, A., Medhi, D. (2003a). "A Double Authentication Scheme To Detect Impersonation Attack In Link State Routing Protocols". In Proc. of IEEE International Conference on Communications (ICC'03), pp.1723-1727. Alaska.

[57] Huang, Y. A., Fan, W., Lee, W., Yu, P. S. (2003b). "Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies". In Proc. of the 23rd International Conference on Distributed Computing Systems (ICDCS), pp.478-489.

[58] Huang, Y. A., Lee, W. (2004). "Attack Analysis and Detection for Ad Hoc Routing Protocols". In Proc. of Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004. Sophia Antipolis, France.

[59] Hubaux, J. P., Buttyan, L., Capkun, S. (2001). "The Quest for Security in Mobile Ad Hoc Networks". In Proc. of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHOC 2001, pp.146-155.

[60] Hyytia, E. (2005). "Random Waypoint Model". http://www.netlab.tkk.fi/~esa/java/rwp/index.shtml, 2005.

[61] Innella, P., McMillan, O. (2001). "An Introduction to Intrusion Detection Systems". http://www.securityfocus.com/infocus/1520.

[62] ISS. (1998). "Network vs. Host-based Intrusion Detection - A Guide to Intrusion Detection Technology". http://documents.iss.net/whitepapers/nvh_ids.pdf, ISS White Papers.

[63] Jain, S., Lv, Y., Das, S. R. (2005). "Exploiting Path Diversity in the Link Layer in Wireless Ad Hoc Networks". In Proc. of the 6th IEEE WoWMoM Symposium. Taormina, Italy.

[64] Joachims, T. (2004). "SVM Light Support Vector Machine". http://svmlight.joachims.org.

[65] Jung, S., Hundewale, N., Zelikovsky, A. (2005a). "Energy efficiency of load balancing in MANET routing protocols". In Proc. of Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks, SNPD/SAWN, pp.476-483.

[66] Jung, S., Hundewale, N., Zelikovsky, A. (2005b). "Node Caching Enhancement of Reactive Ad Hoc Routing Protocols". In Proc. of IEEE Wireless Communication and Networking Conference (WCNC'05), Vol. 4, pp.1970-1975.

[67] Just, M., Kranakis, E., Wan, T. (2003). "Resisting Malicious Packet Dropping in Wireless Ad-Hoc Networks". In Proc. of 2nd Annual Conference on Ad hoc Networks and Wireless (ADHOCNOW'03), pp.151-163. Montreal, Canada.

[68] Kachirski, O., Gupta, R. (2002). "Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks". IEEE Workshop on Knowledge Media Networking (KMN'02), pp.153-160. Kyoto, Japan.

[69] Kachirski, O., Gupta, R. (2003). "Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks". In Proc. of 36th Annual Hawaii International Conference on System Sciences (HICSS'03), pp.57-64. Big Island, Hawaii.

[70] Karygiannis, T., Owens, L. (2002). "Wireless Network Security 802.11". Bluetooth and Handheld Devices. NIST Publication (800-48).

[71] Kazienko, P., Dorosz, P. (2004). "Intrusion Detection Systems (IDS) Part 2 - Classification; methods; techniques". http://www.windowsecurity.com/articles/IDS-Part2-Classification-methods-techniques.html

[72] Ke, C. –H. (2006). "Communication range". http://140.116.82.80/~smallko/ns2/range_en.htm.

[73] Khalili, A., Katz, J., Arbaugh, W. (2003). "Toward Secure Key Distribution in Truly Ad-Hoc Networks". In Proc. of IEEE Workshop on Security and Assurance in Ad hoc Networks, pp.342-346. Orlando, Florida.

[74] Khan, O. A. (2003). "A Survey of Secure Routing Techniques for MANET" Course Survey Report. National University of Computer and Emerging Sciences. Karachi.

[75] King, M. (2002). "Security Lifecycle- Managing the Threat". GSEC Practical, Vol. 1.

[76] King, S. T., Mao, Z. M., Chen, P. M. (2004). "CIDS: Causality Based Intrusion Detection System". http://www.eecs.umich.edu/techreports/cse/2004/CSE-TR-493-04.pdf. Tech Report, CSE-TR-493-04. University of Michigan.

[77] Kong, J., Zerfos, P., Luo, H., Lu, S., Zhang, L. (2001). "Providing robust and ubiquitous security support for mobile ad-hoc networks". In Proc. of IEEE International Conference on Network Protocols, pp.251-260. Riverside, CA, USA.

[78] Kurkowski, S., Camp, T., Colagrosso, M. (2005). "Manet Simulation Studies: The Incredibles". SIGMobile Mobile Comm. Rev., Vol. 9, No. 4, pp. 50-61.

[79] Li, H., Chen, Z., Qin, X. (2003). "Secure Routing in Wired Networks and Wireless Ad Hoc Networks". Term-paper. Department of Computer Science, Univ. of Kentucky.

[80] Liu, Z., Joy, T., Thompson, R. (2004). "A Dynamic Trust Model for Mobile Ad Hoc Networks". In the 10th IEEE International Workshop on Future Trends in Distributed Computing Systems (FTDCS 2004). Suzhou, China.

[81] Loo, F. Y. (2004). "Ad Hoc Network: Prospects and Challenges". Graduate School Research Paper (Rinkou). Department of Information and Communication Engineering, University of Tokyo.

[82] Lu, Y., Bhargava, B., Hefeeda, M. (2001) "An Architecture for Secure Wireless Networking". In Proc. of IEEE Workshop on Reliable and Secure Application in Mobile Environment.

[83] Lundberg, J. (2000). "Routing Security in Ad Hoc Networks". Seminar Paper. Seminar on Network Security. Helsinki University of Technology.

[84] Luo, H., Zerfos, P., Kong, J., Lu, S., Zhang, L. (2002). "Self-securing ad hoc wireless networks". In Proc. of 7th IEEE Symposium on Comp. and Communications (ISCC), pp.567-576. Taormina, Italy.

[85] Maki, S. (2000). "Security Fundamentals in Ad-hoc Networking". Seminar Paper. Seminar on Internetworking. Helsinki University of Technology.

[86] Marti, S., Giuli, T. J., Lai, K., Baker, M. (2000). "Mitigating routing misbehavior in mobile ad hoc networks". In Proc. of the 6th annual international conference on Mobile computing and networking, pp.255-265.

[87] Meier, A. V. (2003). "IDS in Ad Hoc Networks". Seminar Paper. Hauptseminar Ad Hoc Networks. Technische Universität München, Institut für Informatik.

[88] Michiardi, P., Molva, R. (2002). "Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation In Mobile Ad Hoc Networks". In Proc. of the 6th IFIP Communications and Multimedia Security Conference, pp.107-121. Portorosz, Slovenia.

[89] Milgram, S. (1967). "The Small World Problem". Psychology Today, Issue 1, pp.61–67.

[90] Monarch Project. (2000). "Rice Monarch Project Extensions to ns-2". http://www.monarch.cs.rice.edu/cmu-ns.html, Nov., 2000.

[91] Ni, S. Y., Tseng, Y. C., Chen, Y. S., Sheu, J. P. (1999). "The broadcast storm problem in a mobile ad hoc network". In Proc. of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp.151-162.

[92] Oltsik, J., Biggar, H. (2006). "Information-Centric Security and Data Erasure". http://securitypark.bitpipe.com/detail/RES/1159272195_648.html.

[93] OPNET. (2006). "OPNET Modeler". http://www.opnet.com/products/modeler/home.html, 2006.

[94] Ornaghi, A., Valleri, M. (2003). "Man in the middle attacks Demos". Blackhat Conference. USA. http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-ornaghi-valleri.pdf.

[95] Papadimitratos, P., Haas, Z. J. (2002). "Secure Routing for Mobile Ad Hoc Networks". SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), pp.27-31. San Antonio, TX.

[96] Papadimitratos, P., Haas, Z. J. (2003). "Securing Mobile Ad Hoc Networks". Handbook of Mobile Computing. CRC Press.

[97] Paul, K., Westhoff, D. (2002). "Context Aware Detection of Selfish Node in DSR based Ad-hoc Network". IEEE Global Telecommunications Conference, pp.178-182. Taipei, Taiwan.

[98] Perkins, C. E., Royer, E. M. (1999). "Ad hoc On-Demand Distance Vector Routing". In Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp.90-100. New Orleans, LA.

[99] Perkins, C., Belding-Royer, E., Das, S. (2003). "RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing". http://rfc.sunsite.dk/rfc/rfc3561.html.

[100] Pirzada, A. A., McDonald, C. (2006). "Trust Establishment in Pure Ad-hoc Networks". Wireless Personal Communications, Vol. 37, Numbers 1-2, pp139-168.

[101] Qian, L., Song, N., Li, X. (2006). "Secure Anonymous Routing in Clustered Multihop Wireless Ad Hoc Networks". In Proc. of Conference on Information Sciences and Systems (CISS).

[102] Rafique, K. (2002). "A Survey of Mobile Ad Hoc Networks". Columbia University Student Project Report.

[103] Raghavan, B., Snoeren, A. C. (2003). "Priority Forwarding in Ad Hoc Networks with Self-Interested Parties". Workshop on Economics of Peer-to-Peer Systems (P2PEcon '03). Berkeley, CA.

[104] Rajavaram, S., Shah, H., Shanbhag, V., Undercoffer, J., Joshi, A. (2002). "Neighborhood Watch: An Intrusion Detection and Response Protocol for Mobile Ad Hoc Networks". Student Research Conference. University of Maryland at Baltimore County (UMBC).

[105] Ramaswamy, S., Fu, H., Sreekantaradhya, M., Dixon. J., Nygard, K. (2003). "Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks". In Proc. of the International Conference on Wireless Networks, pp.570-575. Las Vegas.

[106] Salem, N. B., Buttyan, L., Hubaux, J. –P., Jakobsson, M. (2003). "A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks". In Proc. of 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03), pp.13-24. Annapolis, USA.

[107] Sanzgiri, K., Dahill, B., Levine, B. N., Shields, C., Royer, E. B. (2002). "A Secure Routing Protocol for Ad Hoc Networks". In Proc. of 2002 IEEE International Conference on Network Protocols (ICNP), pp.78-89.

[108] Schäfer, G. (2002). "Research Challenge in Security for Next Generation Mobile Networks". Position Papers PAMPAS '02 - Workshop on Requirements for Mobile Privacy & Security.

[109] SearchSecurity.com. (2003). "What is promiscuous mode?" http://searchsecurity.techtarget.com/sDefinition/0,290660,sid14_gci518283,00.html

[110] Sekar, R., Gupta, A., Frullo, J., Shanbhag, T., Tiwari, A., Yang, H., Zhou, S. (2002). "Specification-based anomaly detection: a new approach for detecting network intrusions". In Proc. of the 9th ACM conference on Computer and communications security, pp.265-274. Washington, DC, USA.

[111] Srinivasan, K., Gwalani, S., Royer, E. B., Vigna, G., Kemmerer, R. A. (2003). "AODVSTAT: An Intrusion Detection Tool for Ad Hoc Networks". Research Paper. Department of Computer Science, University of California Santa Barbara.

[112] Stajano, F., Anderson, R. (1999). "The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks". In Proc. of the 7th Int. Workshop on Security Protocols, LNCS, Vol. 1796, pp.172-194.

[113] Stajano, F. (2000). "The Resurrecting Duckling - what next?" In Proc of the 8th International Workshop on Security Protocols, pp.204-214. Cambridge, UK.

[114] Stallings, W. (1999). "Cryptography and network security, Principles and practice". Prentice Hall, Inc, 2nd ed., pp.6-9, 1999.

[115] Stamouli, I. (2003). "Real-time Intrusion Detection for Ad hoc Networks". Technical Report. Computer Science Department, University of Dublin, Trinity College.

[116] Stathopoulos, T., Kapur, R., Estrin, D., Heidemann, J., Zhang, L. (2004). "Application-Based Collision Avoidance in Wireless Sensor Networks". In Proceedings of the 29th IEEE International Conference on Local Computer Networks, pp.506-514. Tampa, Flordia, USA.

[117] Vattikonda, A., Gampa, R. K., Isukapalli, V. K., Kakarlapudi, V. R. (2003). "Intrusion Detection In Wireless Networks". Term Paper. Department of Computer Science, The University of Kentucky.

[118] Venkatraman, L., Agrawal, D. P. (2000). "A Novel Authentication Scheme for Ad Hoc Networks". IEEE Wireless Communications and Networking Conference (WCNC 2000), pp.1268-1273, Vol.3.

[119] Wai, F. H., Aye, Y. N., James, N. H. (2003). "Intrusion Detection in Wireless Ad-Hoc Networks". Term Paper. School of Computing, National University of Singapore.

[120] Walsh, K., Sirer, E. G. (2006). "Experience with an Object Reputation System for Peer-to-Peer Filesharing". In Proc. of NSDI '06: 3rd Symposium on Networked Systems Design & Implementation, pp1-14. CA.

[121] Watts, D. J., Strogatz, S. H. (1998). "Collective dynamics of 'small-world' networks". Nature, Vol. 393, pp.440-442.

[122] Weimerskirch, A., Thonet, G. (2001). "A Distributed Light-Weight Authentication Model for Ad-Hoc Networks". In Proc. of 4th International Conference on Information Security and Cryptology (ICISC 2001), pp.341-354. Seoul, South Korea.

[123] Yang, H., Meng, X., Lu, S. (2002). "Self-organized network-layer security in mobile ad hoc networks". In ACM MOBICOM Wireless Security Workshop (WiSe'02), pp.11-20. Atlanta.

[124] Yang, H., Ricciato, F., Lu, S., Zhang, L. (2006). "Securing a Wireless World". In Proc. of the IEEE, Vol. 94, No. 2.

[125] Yi, S., Naldurg, P., Kravets, R. (2001). "Security-aware ad hoc routing for wireless networks". In Proc. of the 2nd ACM Interational Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2001, pp.299-302. Long Beach, CA, USA.

[126] Yi, S., Kravets, R. (2002). "Key Management for Heterogeneous Ad Hoc Wireless Networks". Technical Report. Dept. of Computer Science, University of Illinois.

[127] Zapata, M. G., Asokan, N. (2002). "Securing ad hoc routing protocols". In Proc. of the ACM workshop on Wireless security, Wise'02, pp.1-10. Atlanta, Georgia, USA.

[128] Zhang, Y., Lee, W., Huang, Y.-A. (2003). "Intrusion Detection Techniques for Mobile Wireless Networks". Wireless Network, Vol. 9, Issue 5, pp.545-556.

[129] Zhong, S., Yang, Y. R., Chen, J. (2003). "Sprite: A Simple, CheatProof, Credit-Based System for Mobile Ad Hoc Networks". In Proc. of IEEE INFOCOM'03, pp.1987-1997. San Francisco.

[130] Zhou, L., Haas, Z. J. (1999). "Securing ad hoc networks". IEEE Network, Vol. 13, No. 6, pp.24-30.

[131] Zhu, S., Xu, S., Setia, S., Jajodia, S. (2006). "LHAP: A Lightweight Network Access Control Protocol for Ad-Hoc Networks". Elsevier Ad Hoc Networks Journal, Vol. 4, Issue 5, pp567-585.

Appendix A

The Effects of Nodes' Initial Friendships

Simulation A (Run = 1, Random seed = 1)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	0.5km^2	200.0s	100	228	328
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	0.5km^2	200.0s	300	6715	7015
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8898	9398
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	0.5km^2	200.0s	1000	8897	9897
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	0.5km^2	200.0s	1400	8500	9900

Simulation A (Run = 2, Random seed = 2)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	0.5km^2	200.0s	100	237	337
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	0.5km^2	200.0s	300	7011	7311
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	9060	9560
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	0.5km^2	200.0s	1000	8897	9897
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	0.5km^2	200.0s	1400	8500	9900

Simulation A (Run = 3, Random seed = 3)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	0.5km^2	200.0s	100	218	318
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	0.5km^2	200.0s	300	6659	6959
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8726	9226
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	0.5km^2	200.0s	1000	8897	9897
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	0.5km^2	200.0s	1400	8500	9900

Simulation A (Run = 4, Random seed = 4)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	0.5km^2	200.0s	100	203	303
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	0.5km^2	200.0s	300	6300	6600
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8897	9397
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	0.5km^2	200.0s	1000	8896	9896
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	0.5km^2	200.0s	1400	8499	9899

Simulation A (Run = 5; Random seed = 5)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	0.5km^2	200.0s	100	198	298
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	0.5km^2	200.0s	300	6455	6755
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	9004	9504
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	0.5km^2	200.0s	1000	8896	9896
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	0.5km^2	200.0s	1400	8500	9900

Simulation A (Run = 1, Random seed = 1)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	1km^2	200.0s	100	71	171
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	1km^2	200.0s	300	1086	1386
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3537	4037
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	1km^2	200.0s	1000	7848	8848
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	1km^2	200.0s	1400	8174	9574

Simulation A (Run = 2, Random seed = 2)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	1km^2	200.0s	100	65	165
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	1km^2	200.0s	300	1080	1380
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3832	4332
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	1km^2	200.0s	1000	7772	8772
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	1km^2	200.0s	1400	8134	9534

Simulation A (Run = 3, Random seed = 3)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	1km^2	200.0s	100	74	174
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	1km^2	200.0s	300	961	1261
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3838	4338
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	1km^2	200.0s	1000	7797	8797
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	1km^2	200.0s	1400	8297	9697

Simulation A (Run = 4, Random seed = 4)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	1km^2	200.0s	100	57	157
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	1km^2	200.0s	300	987	1287
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3763	4263
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	1km^2	200.0s	1000	7667	8667
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	1km^2	200.0s	1400	8119	9519

Simulation A (Run = 5, Random seed = 5)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	1km^2	200.0s	100	73	173
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	1km^2	200.0s	300	1270	1570
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3838	4338
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	1km^2	200.0s	1000	7812	8812
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	1km^2	200.0s	1400	8214	9614

Simulation A (Run = 1, Random seed = 1)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	2km^2	200.0s	100	18	118
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	2km^2	200.0s	300	202	502
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	549	1049
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	2km^2	200.0s	1000	2527	3527
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	2km^2	200.0s	1400	4397	5797

Simulation A (Run = 2, Random seed = 2)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	2km^2	200.0s	100	16	116
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	2km^2	200.0s	300	159	459
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	662	1162
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	2km^2	200.0s	1000	2662	3662
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	2km^2	200.0s	1400	3954	5354

Simulation A (Run = 3, Random seed = 3)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	2km^2	200.0s	100	26	126
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	2km^2	200.0s	300	199	499
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	582	1082
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	2km^2	200.0s	1000	2584	3584
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	2km^2	200.0s	1400	4188	5588

Simulation A (Run = 4, Random seed = 4)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	2km^2	200.0s	100	14	114
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	2km^2	200.0s	300	179	479
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	550	1050
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	2km^2	200.0s	1000	2583	3583
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	2km^2	200.0s	1400	4291	5691

Simulation A (Run = 5, Random seed = 5)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	2km^2	200.0s	100	16	116
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	2km^2	200.0s	300	196	496
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	583	1083
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	2km^2	200.0s	1000	2559	3559
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	2km^2	200.0s	1400	4158	5558

Simulation A (Average)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	0.5km^2	200.0s	100	216	316
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	0.5km^2	200.0s	300	6628	6298
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8917	9417
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	0.5km^2	200.0s	1000	8896	9896
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	0.5km^2	200.0s	1400	8499	9899

Simulation A (Average)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	1km^2	200.0s	100	68	168
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	1km^2	200.0s	300	1076	1376
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3761	4261
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	1km^2	200.0s	1000	7779	8779
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	1km^2	200.0s	1400	8187	9587

Simulation A (Average)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	1	2km^2	200.0s	100	18	118
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	3	2km^2	200.0s	300	187	487
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	585	1085
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	10	2km^2	200.0s	1000	2583	3583
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	14	2km^2	200.0s	1400	4197	5597

Appendix B

The Effects of Network's Age

Simulation B (Run = 1, Random seed = 1)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	50.0s	500	4607	5107
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	100.0s	500	7580	8080
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	150.0s	500	8395	8895
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8898	9398
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	250.0s	500	9092	9582

Simulation B (Run = 2, Random seed = 2)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	50.0s	500	4699	5199
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	100.0s	500	7849	8349
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	150.0s	500	8665	9165
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	9060	9560
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	250.0s	500	9217	9717

Simulation B (Run = 3, Random seed = 3)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	50.0s	500	4122	4622
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	100.0s	500	7378	7878
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	150.0s	500	8234	8734
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8726	9226
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	250.0s	500	8917	9417

Simulation B (Run = 4, Random seed = 4)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	50.0s	500	4131	4631
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	100.0s	500	7601	8101
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	150.0s	500	8412	8912
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8897	9397
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	250.0s	500	9046	9546

Simulation B (Run = 5, Random seed = 5)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	50.0s	500	4611	5111
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	100.0s	500	7721	8221
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	150.0s	500	8591	9091
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	9004	9504
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	250.0s	500	9147	9647

Simulation B (Run = 1, Random seed = 1)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	50.0s	500	696	1196
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	100.0s	500	1656	2156
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	150.0s	500	2566	3066
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3537	4037
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	250.0s	500	4526	5026

Simulation B (Run = 2, Random seed = 2)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	50.0s	500	784	1284
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	100.0s	500	1894	2394
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	150.0s	500	2811	3311
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3832	4332
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	250.0s	500	4657	5157

Simulation B (Run = 3, Random seed = 3)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	50.0s	500	677	1177
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	100.0s	500	1641	2141
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	150.0s	500	2648	3148
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3838	4338
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	250.0s	500	4754	5254

Simulation B (Run = 4, Random seed = 4)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	50.0s	500	639	1139
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	100.0s	500	1622	2122
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	150.0s	500	2652	3152
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3763	4263
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	250.0s	500	4697	5197

Simulation B (Run = 5, Random seed = 5)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	50.0s	500	645	1145
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	100.0s	500	1704	2204
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	150.0s	500	2770	3270
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3838	4338
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	250.0s	500	4771	5271

Simulation B (Run = 1, Random seed = 1)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	50.0s	500	126	626
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	100.0s	500	293	793
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	150.0s	500	427	927
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	549	1049
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	250.0s	500	774	1274

Simulation B (Run = 2, Random seed = 2)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	50.0s	500	166	666
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	100.0s	500	328	828
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	150.0s	500	475	975
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	662	1162
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	250.0s	500	830	1330

Simulation B (Run = 3, Random seed = 3)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	50.0s	500	127	627
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	100.0s	500	258	758
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	150.0s	500	407	907
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	200.0s	500	582	1082
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	250.0s	500	740	1240

Simulation B (Run = 4, Random seed = 4)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	50.0s	500	94	594
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	100.0s	500	249	749
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	150.0s	500	389	889
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	200.0s	500	550	1050
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	$2km^2$	250.0s	500	765	1265

Simulation B (Run = 5, Random seed = 5)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	50.0s	500	92	592
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	100.0s	500	296	796
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	150.0s	500	427	927
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	583	1083
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	250.0s	500	739	1239

Simulation B (Average)

University Campus	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	50.0s	500	4434	4934
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	100.0s	500	7625	8125
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	150.0s	500	8459	8959
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	200.0s	500	8917	9417
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	0.5km^2	250.0s	500	9083	9583

Simulation B (Average)

City Network (1)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	50.0s	500	688	1188
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	100.0s	500	1703	2203
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	150.0s	500	2689	3189
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	200.0s	500	3761	4261
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	1km^2	250.0s	500	4681	5181

Simulation B (Average)

City Network (2)	Set 1	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	50.0s	500	121	621
	Set 2	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	100.0s	500	284	784
	Set 3	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	150.0s	500	425	925
	Set 4	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	200.0s	500	585	1085
	Set 5	Total Nodes	Initial Friends	Terrain Size	Simulation Time	*Direct Friends*	*Indirect Friends*	*Total Friends*
		100	5	2km^2	250.0s	500	769	1269

Appendix C

AODV Finite State Machine Specification

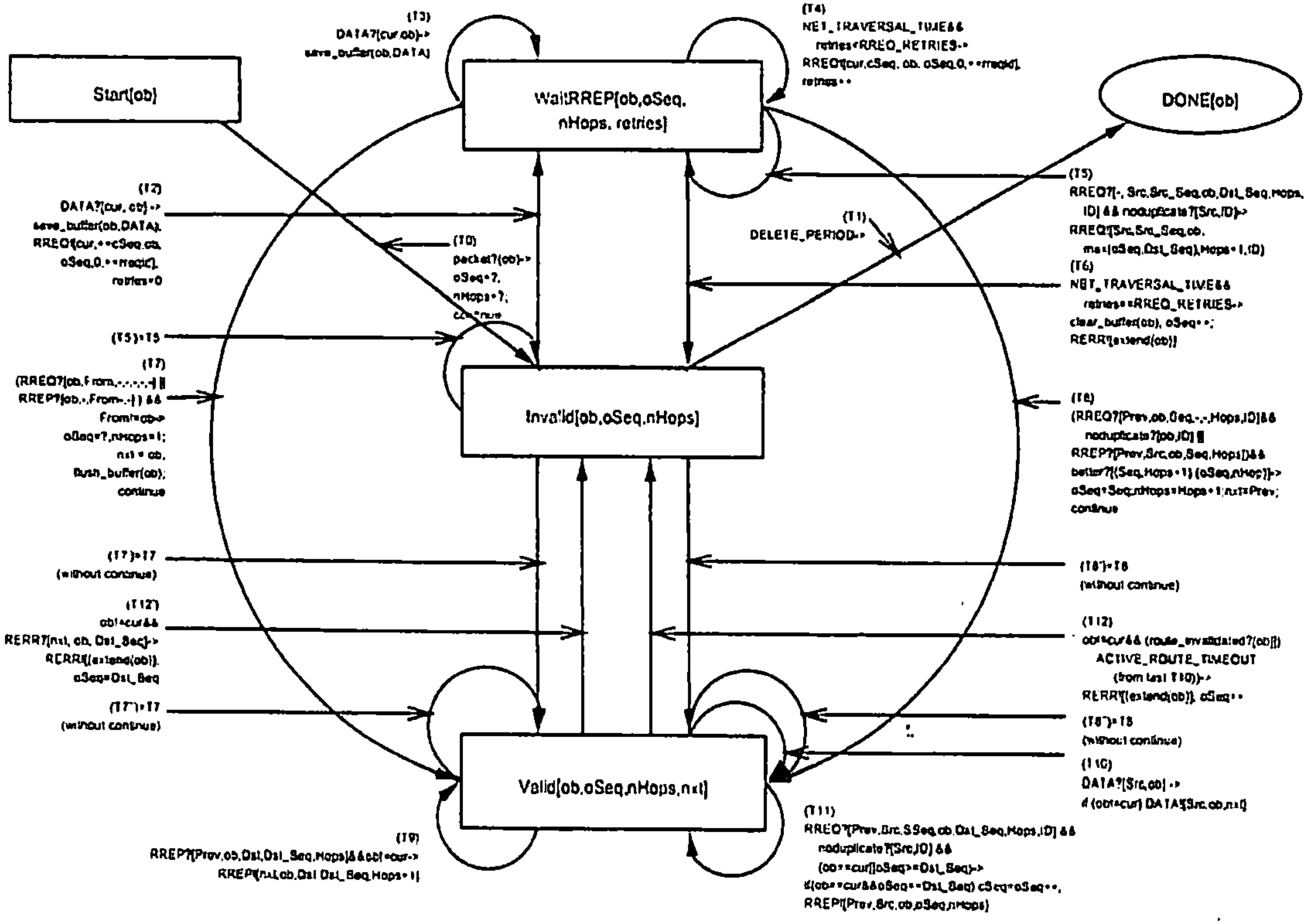


Fig. 1. AODV Extended Finite State machine (*ob*): In Normal Use

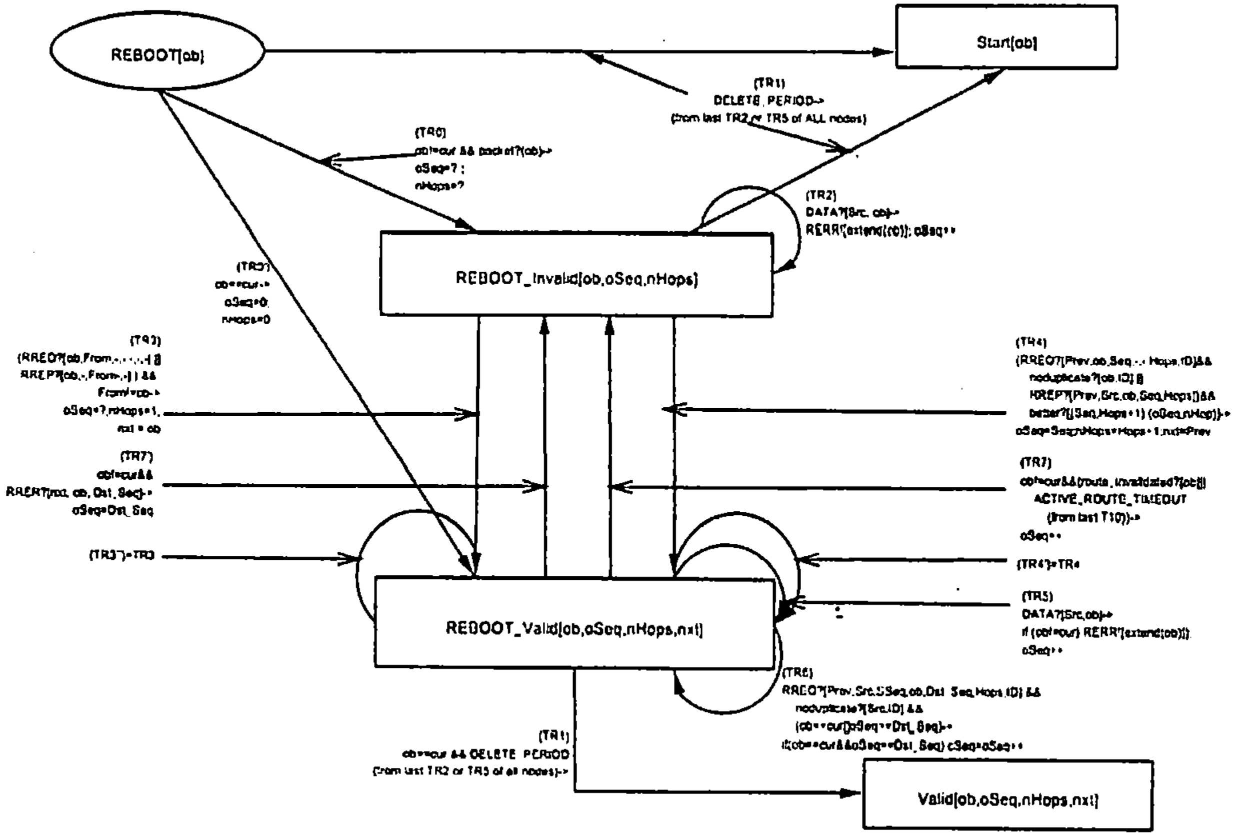

REBOOT[ob]
Start[ob]
(TR1)
DELETE_PERIOD->
(from last TR2 or TR5 of ALL nodes)
(TR0)
ob!=cur && packet?(ob)->
oSeq=? ;
nHops=?
(TR2)
DATA?[Src, ob]->
RERR![extend(ob)]; oSeq++
REBOOT_Invalid[ob,oSeq,nHops]
(TR3)
ob==cur->
oSeq=0;
nHops=0
(TR3)
(RREQ?[ob,From,-,-,-,-] ||
RREP?[ob,-,From,-,-]) &&
From!=ob->
oSeq=?,nHops=1,
nxt = ob
(TR7)
ob!=cur&&
RRERT[nxt, ob, Dst_Seq]->
oSeq=Dst_Seq
(TR3')=TR3
(TR4)
(RREQ?[Prev,ob,Seq,-,-,Hops,ID]&&
noduplicate?[ob,ID] ||
RREP?[Prev,Src,ob,Seq,Hops])&&
better?[(Seq,Hops+1),(oSeq,nHop)]->
oSeq=Seq;nHops=Hops+1;nxt=Prev
(TR7)
ob!=cur&&(route_invalidated?[ob] ||
ACTIVE_ROUTE_TIMEOUT
(from last T10))->
oSeq++
(TR4')=TR4
(TR5)
DATA?[Src,ob]->
if (ob!=cur) RERR![extend(ob)];
oSeq++
REBOOT_Valid[ob,oSeq,nHops,nxt]
(TR6)
RREQ?[Prev,Src,SSeq,ob,Dst_Seq,Hops,ID] &&
noduplicate?[Src,ID] &&
(ob==cur||oSeq>=Dst_Seq)->
if(ob==cur&&oSeq==Dst_Seq) oSeq=oSeq++
(TR1)
ob==cur && DELETE_PERIOD
(from last TR2 or TR5 of all nodes)->
Valid[ob,oSeq,nHops,nxt]

Fig. 2. AODV Extended Finite State machine (*ob*): After Reboot

Appendix D

Partial Programming Codes for the Proposed IDS Framework Modules

```
// Added to NS-2 to Capture Audit Data

void
AODV::logaodvpacket(Packet *p)
{
 struct hdr_cmn *ch = HDR_CMN(p);
 struct hdr_ip *ih = HDR_IP(p);
 struct hdr_mac802_11 *mh = HDR_MAC802_11(p);
 struct hdr_aodv *ah;
 struct hdr_aodv_request *rq;
 struct hdr_aodv_reply *rp;
 struct hdr_aodv_error *re;
 char pktType[10];
 char pktID[20];
 char pktT[20];
 float timest;
 char op = 'D';
 int bcastid = 0;
 int hopcount;
 int rpdst_seqno;
 int rqsrc_seqno;
 int rqdst_seqno;
 int dstseqno;
 int srcseqno;
 nsaddr_t neighbour, sender, src, dst, prevh;
 FILE *FLogAlla;
 FILE *FLogAllb;



 if (ch->ptype()==PT_AODV)
 {
  ah = HDR_AODV(p);
  switch(ah->ah_type)
  {
   case AODVTYPE_RREQ:
    rq = HDR_AODV_REQUEST(p);
    sender = ih->saddr();
    neighbour = index;
    src = rq->rq_src;
    dst = rq->rq_dst;
    hopcount = rq->rq_hop_count;
    srcseqno = rq->rq_src_seqno;
    dstseqno = rq->rq_dst_seqno;
    strcpy(pktType, "RREQ");
    timest = rq->rq_timestamp;
    sprintf(pktT,"%d",src);
    strcpy(pktID,pktT);
    strcat(pktID,"-");
    sprintf(pktT,"%d",dst);
    strcat(pktID,pktT);
    strcat(pktID,"-");
    sprintf(pktT,"%.0f",timest);
    strcat(pktID,pktT);
    bcastid = rq->rq_bcast_id;
   break;

   case AODVTYPE_RREP:
```

```
  rp = HDR_AODV_REPLY(p);
  sender = rp->rp_src;
  neighbour = index;
  src = ih->saddr();
  dst = ih->daddr();
  hopcount = rp->rp_hop_count;
  dstseqno = rp->rp_dst_seqno;
  srcseqno = -1;
  strcpy(pktType, "RREP");
  timest = rp->rp_timestamp;
  sprintf(pktT,"%d",dst);
  strcpy(pktID,pktT);
  strcat(pktID,"-");
  sprintf(pktT,"%d",src);
  strcat(pktID,pktT);
  strcat(pktID,"-");
  sprintf(pktT,"%.0f",timest);
  strcat(pktID,pktT);
  bcastid = -1;
 break;


 case AODVTYPE_RERR:
  neighbour = index;
  sender = ih->saddr();
  src = 1;
  dst = 1;
  hopcount = 1;
  dstseqno = 1;
  srcseqno = 1;
  strcpy(pktType, "RERR");
  timest = CURRENT_TIME;
  sprintf(pktT,"%d",dst);
  strcpy(pktID,pktT);
  strcat(pktID,"-");
  sprintf(pktT,"%d",src);
  strcat(pktID,pktT);
  strcat(pktID,"-");
  sprintf(pktT,"%.0f",timest);
  strcat(pktID,pktT);
  bcastid = 1;
 break;
}


 if (FLogAlla = fopen("/root/testing/LogAlla.txt","a+"))
 {
  fprintf(FLogAlla,"%d %.6f %d %d %d %d %s %s %d %d %d %d\n", 123, timest, \
  sender, neighbour, src, dst, pktType, pktID, bcastid, hopcount, srcseqno, dstseqno);
 }
 else
  printf("error opening LogAlla file\n");

 if (FLogAllb = fopen("/root/testing/LogAllb.txt","a+"))
 {
  fprintf(FLogAllb,"%d %.6f %d %d %d %d %s %s %d %d %d %d\n", 123, timest, \
  sender, neighbour, src, dst, pktType, pktID, bcastid, hopcount, srcseqno, dstseqno);
 }
```

```
   else
    printf("error opening LogAllb file\n");
 }
 fclose(FLogAlla);
 fclose(FLogAllb);
}

/****************** End of File ***************************/
```

```
// Misuse Detection Attacks Scenarios

void
AODV::rt_resolve(Packet *p)
{
 struct hdr_cmn *ch = HDR_CMN(p);
 struct hdr_ip *ih = HDR_IP(p);
 struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
 aodv_rt_entry *rt;
 /*
  *  Set the transmit failure callback.  That
  *  won't change.
  */
 ch->xmit_failure_ = aodv_rt_failed_callback;
 ch->xmit_failure_data_ = (void*) this;
 rt = rtable.rt_lookup(ih->daddr());
 if(rt == 0)
 {
  rt = rtable.rt_add(ih->daddr());
 }
 /*
  * If the route is up, forward the packet
  */

 if(rt->rt_flags == RTF_UP)
 {
  assert(rt->rt_hops != INFINITY2);
  forward(rt, p, NO_DELAY);
 }

 //SELFISH ATTACK
 else if((ih->saddr() == index) && (ih->saddr() == ATTACKER))
 {
  int cntduplicate;
  rqueue.enque(p);
  for (cntduplicate=1; cntduplicate<=2; cntduplicate++)
  {
   sendRequestFlood(rt->rt_dst);
  }
 }

 //FLOODING ATTACK
 else if((ih->saddr() == index) && (ih->saddr() == ATTACKER) \
 && (CURRENT_TIME == 20.0))
 {
  int cntflood;
  int fakesrc;
  int dst;
  srand(1);
  fakesrc = VICTIM;
  dst = 2;
  rqueue.enque(p);
  for (cntflood=1; cntflood<=10; cntflood++)
  {
   sendRequestFlood2(fakesrc,rt->rt_dst);
  }
 }
 //FAKE RREQ ATTACK
 else if((ih->saddr() == index) && (ih->saddr() == ATTACKER) \
 && (CURRENT_TIME == 20.0))
 {
  int index2;
  index2 = FAKE-DST;
  rqueue.enque(p);
  sendRequestFake(FAKE-SRC,index2);
 }

 /*
  *  if I am the source of the packet, then do a Route Request.
  */

 else if(ih->saddr() == index)
 {
  rqueue.enque(p);
  sendRequest(rt->rt_dst);
 }

 /*
  *     A local repair is in progress. Buffer the packet.
  */
 else if (rt->rt_flags == RTF_IN_REPAIR)
 {
  rqueue.enque(p);
 }

 /*
  * I am trying to forward a packet for someone else to which
  * I don't have a route.
  */
 else
 {
  Packet *rerr = Packet::alloc();
  struct hdr_aodv_error *re = HDR_AODV_ERROR(rerr);
  /*
   * For now, drop the packet and send error upstream.
   * Now the route errors are broadcast to upstream
   * neighbors - Mahesh 09/11/99
   */
  assert (rt->rt_flags == RTF_DOWN);
  re->DestCount = 0;
  re->unreachable_dst[re->DestCount] = rt->rt_dst;
  re->unreachable_dst_seqno[re->DestCount] = rt->rt_seqno;
  re->DestCount += 1;
  #ifdef DEBUG
  fprintf(stderr, "%s: sending RERR...\n", __FUNCTION__);
  #endif
  sendError(rerr, false);
  drop(p, DROP_RTR_NO_ROUTE);
 }
}


void
AODV::sendRequestFake(nsaddr_t dst, nsaddr_t index2)
{
```

```
// Allocate a RREQ packet
Packet *p = Packet::alloc();
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
aodv_rt_entry *rt = rtable.rt_lookup(dst);
assert(rt);

/*
 *  Rate limit sending of Route Requests. We are very conservative
 *  about sending out route requests. BUT NOT IN THIS ATTACK
 */

if (rt->rt_flags == RTF_UP) {
  assert(rt->rt_hops != INFINITY2);
  Packet::free((Packet *)p);
  return;
}

if (rt->rt_req_timeout > CURRENT_TIME) {
  Packet::free((Packet *)p);
  return;
}

// rt_req_cnt is the no. of times we did network-wide broadcast
// RREQ_RETRIES is the maximum number we will allow before
// going to a long timeout.

if (rt->rt_req_cnt > RREQ_RETRIES)
{
 rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
 rt->rt_req_cnt = 0;
 Packet *buf_pkt;
 while ((buf_pkt = rqueue.deque(rt->rt_dst)))
 {
  drop(buf_pkt, DROP_RTR_NO_ROUTE);
 }
 Packet::free((Packet *)p);
 return;
}

#ifdef DEBUG
fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d\n",
                   ++route_request, index, rt->rt_dst);
#endif // DEBUG

// Determine the TTL to be used this time.
// Dynamic TTL evaluation - SRD

rt->rt_req_last_ttl = max(rt->rt_req_last_ttl,rt->rt_last_hop_count);
if (0 == rt->rt_req_last_ttl)
{
 // first time query broadcast
 ih->ttl_ = TTL_START;
}
else
{
 // Expanding ring search.
```

```
  if (rt->rt_req_last_ttl < TTL_THRESHOLD)
   ih->ttl_ = rt->rt_req_last_ttl + TTL_INCREMENT;
  else
  {
   // network-wide broadcast
   ih->ttl_ = NETWORK_DIAMETER;
   rt->rt_req_cnt += 1;
  }
 }

 // remember the TTL used  for the next time
 rt->rt_req_last_ttl = ih->ttl_;
 // PerHopTime is the roundtrip time per hop for route requests.
 // The factor 2.0 is just to be safe .. SRD 5/22/99
 // Also note that we are making timeouts to be larger if we have
 // done network wide broadcast before.
 rt->rt_req_timeout = 2.0 * (double) ih->ttl_ * PerHopTime(rt);
 if (rt->rt_req_cnt > 0)
   rt->rt_req_timeout *= rt->rt_req_cnt;
 rt->rt_req_timeout += CURRENT_TIME;

 // Don't let the timeout to be too large, however .. SRD 6/8/99
 if (rt->rt_req_timeout > CURRENT_TIME + MAX_RREQ_TIMEOUT)
   rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
 rt->rt_expire = 0;

 #ifdef DEBUG
 fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d, tout %f ms\n",
                 ++route_request,
                 index, rt->rt_dst,
                 rt->rt_req_timeout - CURRENT_TIME);
 #endif // DEBUG

 // Fill out the RREQ packet
 // ch->uid() = 0;
 ch->ptype() = PT_AODV;
 ch->size() = IP_HDR_LEN + rq->size();
 ch->iface() = -2;
 ch->error() = 0;
 ch->addr_type() = NS_AF_NONE;
 ch->prev_hop_ = index;          // AODV hack
 ih->saddr() = index;
 ih->daddr() = IP_BROADCAST;
 ih->sport() = RT_PORT;
 ih->dport() = RT_PORT;
 // Fill up some more fields.
 rq->rq_type = AODVTYPE_RREQ;
 rq->rq_hop_count = 2;
 rq->rq_bcast_id = bid++;
 rq->rq_dst = dst;
 rq->rq_dst_seqno = (rt ? rt->rt_seqno : 0);
 rq->rq_src = index2;
 seqno += 2;
 assert ((seqno%2) == 0);
 rq->rq_src_seqno = seqno;
 rq->rq_timestamp = CURRENT_TIME;
 Scheduler::instance().schedule(target_, p, 0.);
}
```

```
void
AODV::sendRequestFlood(nsaddr_t dst)
{
 // Allocate a RREQ packet
 Packet *p = Packet::alloc();
 struct hdr_cmn *ch = HDR_CMN(p);
 struct hdr_ip *ih = HDR_IP(p);
 struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
 aodv_rt_entry *rt = rtable.rt_lookup(dst);
 assert(rt);

 #ifdef DEBUG
   fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d\n",
                    ++route_request, index, rt->rt_dst);
 #endif // DEBUG

 // Determine the TTL to be used this time.
 // Dynamic TTL evaluation - SRD

 rt->rt_req_last_ttl = max(rt->rt_req_last_ttl,rt->rt_last_hop_count);

 if (0 == rt->rt_req_last_ttl) {
 // first time query broadcast
   ih->ttl_ = TTL_START;
 }
 else
 {
  // Expanding ring search.
  if (rt->rt_req_last_ttl < TTL_THRESHOLD)
     ih->ttl_ = rt->rt_req_last_ttl + TTL_INCREMENT;
  else
  {
   // network-wide broadcast
   ih->ttl_ = NETWORK_DIAMETER;
   rt->rt_req_cnt += 1;
  }
 }

 // remember the TTL used  for the next time
 rt->rt_req_last_ttl = ih->ttl_;

 // PerHopTime is the roundtrip time per hop for route requests.
 // The factor 2.0 is just to be safe .. SRD 5/22/99
 // Also note that we are making timeouts to be larger if we have
 // done network wide broadcast before.

 rt->rt_req_timeout = 2.0 * (double) ih->ttl_ * PerHopTime(rt);
 if (rt->rt_req_cnt > 0)
   rt->rt_req_timeout *= rt->rt_req_cnt;
 rt->rt_req_timeout += CURRENT_TIME;

 // Don't let the timeout to be too large, however .. SRD 6/8/99
 if (rt->rt_req_timeout > CURRENT_TIME + MAX_RREQ_TIMEOUT)
   rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
 rt->rt_expire = 0;

 #ifdef DEBUG
 fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d, tout %f ms\n",
                 ++route_request,
                 index, rt->rt_dst,
                 rt->rt_req_timeout - CURRENT_TIME);
 #endif // DEBUG


 // Fill out the RREQ packet
 // ch->uid() = 0;
 ch->ptype() = PT_AODV;
 ch->size() = IP_HDR_LEN + rq->size();
 ch->iface() = -2;
 ch->error() = 0;
 ch->addr_type() = NS_AF_NONE;
 ch->prev_hop_ = index;          // AODV hack

 ih->saddr() = index;
 ih->daddr() = IP_BROADCAST;
 ih->sport() = RT_PORT;
 ih->dport() = RT_PORT;

 // Fill up some more fields.
 rq->rq_type = AODVTYPE_RREQ;
 rq->rq_hop_count = 1;
 rq->rq_bcast_id = bid++;
 rq->rq_dst = dst;
 rq->rq_dst_seqno = (rt ? rt->rt_seqno : 0);
 rq->rq_src = index;
 seqno += 2;
 assert ((seqno%2) == 0);
 rq->rq_src_seqno = seqno;
 rq->rq_timestamp = CURRENT_TIME;

 Scheduler::instance().schedule(target_, p, 0.);
}


void
AODV::sendRequestFlood2(nsaddr_t fakesrc, nsaddr_t dst)
{
 // Allocate a RREQ packet
 Packet *p = Packet::alloc();
 struct hdr_cmn *ch = HDR_CMN(p);
 struct hdr_ip *ih = HDR_IP(p);
 struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
 aodv_rt_entry *rt = rtable.rt_lookup(dst);
 assert(rt);

 #ifdef DEBUG
   fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d\n",
                    ++route_request, index, rt->rt_dst);
 #endif // DEBUG

 // Determine the TTL to be used this time.
 // Dynamic TTL evaluation - SRD

 rt->rt_req_last_ttl = max(rt->rt_req_last_ttl,rt->rt_last_hop_count);
```

```
if (0 == rt->rt_req_last_ttl)
{
 // first time query broadcast
 ih->ttl_ = TTL_START;
}
else
{
 // Expanding ring search.
 if (rt->rt_req_last_ttl < TTL_THRESHOLD)
    ih->ttl_ = rt->rt_req_last_ttl + TTL_INCREMENT;
 else
 {
  // network-wide broadcast
  ih->ttl_ = NETWORK_DIAMETER;
  rt->rt_req_cnt += 1;
 }
}

// remember the TTL used  for the next time
rt->rt_req_last_ttl = ih->ttl_;

// PerHopTime is the roundtrip time per hop for route requests.
// The factor 2.0 is just to be safe .. SRD 5/22/99
// Also note that we are making timeouts to be larger if we have
// done network wide broadcast before.

rt->rt_req_timeout = 2.0 * (double) ih->ttl_ * PerHopTime(rt);
if (rt->rt_req_cnt > 0)
  rt->rt_req_timeout *= rt->rt_req_cnt;
rt->rt_req_timeout += CURRENT_TIME;

// Don't let the timeout to be too large, however .. SRD 6/8/99
if (rt->rt_req_timeout > CURRENT_TIME + MAX_RREQ_TIMEOUT)
  rt->rt_req_timeout = CURRENT_TIME + MAX_RREQ_TIMEOUT;
rt->rt_expire = 0;

#ifdef DEBUG
fprintf(stderr, "(%2d) - %2d sending Route Request, dst: %d, tout %f ms\n",
                ++route_request,
                index, rt->rt_dst,
                rt->rt_req_timeout - CURRENT_TIME);
#endif // DEBUG

// Fill out the RREQ packet
// ch->uid() = 0;
ch->ptype() = PT_AODV;
ch->size() = IP_HDR_LEN + rq->size();
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_NONE;
ch->prev_hop_ = index;          // AODV hack

ih->saddr() = index;
ih->daddr() = IP_BROADCAST;
ih->sport() = RT_PORT;
ih->dport() = RT_PORT;

// Fill up some more fields.
```

```
 rq->rq_type = AODVTYPE_RREQ;
 rq->rq_hop_count = 2;
 rq->rq_bcast_id = bid++;
 rq->rq_dst = dst;
 rq->rq_dst_seqno = (rt ? rt->rt_seqno : 0);
 rq->rq_src = fakesrc;
 seqno += 2;
 assert ((seqno%2) == 0);
 rq->rq_src_seqno = seqno;
 rq->rq_timestamp = CURRENT_TIME;

 Scheduler::instance().schedule(target_, p, 0.);
}

void
AODV::recvRequest(Packet *p)
{
 struct hdr_ip *ih = HDR_IP(p);
 struct hdr_cmn *ch = HDR_CMN(p);
 struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
 aodv_rt_entry *rt;
 int bad;
 int tempn;

//Modification Attack#1: Hop Count
 if (index == ATTACKER)
 {
  rq->rq_hop_count = INFINITY2;
  ih->saddr() = index;
  ih->daddr() = IP_BROADCAST;
  rq->rq_hop_count = INFINITY2;
  if (rt) rq->rq_dst_seqno = max(rt->rt_seqno, rq->rq_dst_seqno);
  forward((aodv_rt_entry*) 0, p, DELAY);
 }


//Modification Attack#2: Seq No
 if (index == ATTACKER)
 {
  rq->rq_src_seqno += 10;
 }

}
```

```
void
AODV::recvRequest(Packet *p)
{
 struct hdr_ip *ih = HDR_IP(p);
 struct hdr_cmn *ch = HDR_CMN(p);
 struct hdr_aodv_request *rq = HDR_AODV_REQUEST(p);
 aodv_rt_entry *rt;

 if (index == ATTACKER)
 {
  printf("Node %d dropped packet\n",index);
  return;
 }
}
```

```
// Local Misuse Detection and Response

void
AODV::DetectMisRREQ(Packet *p)
{
 struct hdr_cmn *ch = HDR_CMN(p);
 struct hdr_ip *ih = HDR_IP(p);
 struct hdr_aodv *ah = HDR_AODV(p);
 struct hdr_aodv_request *rq;
 struct hdr_aodv_reply *rp;
 int bad;
 char cha;
 int hopcount;
 int bcastid;
 int srcseqno;
 char pktType[10];
 char pktID[20];
 char pktTypeM[10];
 char pktIDM[20];
 char pktT[20];
 char pktTypetemp[10];
 char pktIDtemp[20];
 float timest, timesttemp, timestM;
 int is_true1, is_true2, is_true3, is_true4;
 int tempx, tempx2, tempx3;
 int idtemp, bcastidtemp, hcounttemp, srcseqtemp, dstseqtemp;
 nsaddr_t neighbour, sender, src, dst, prevh, sendertemp, receivertemp, srctemp, dsttemp;
 FILE *FLogAlla;
 FILE *FMalNode;
 assert(HDR_IP (p)->sport() == RT_PORT);
 assert(HDR_IP (p)->dport() == RT_PORT);

 if (ch->ptype()==PT_AODV)
 {
  ah = HDR_AODV(p);
  switch(ah->ah_type)
  {
   case AODVTYPE_RREQ:
   rq = HDR_AODV_REQUEST(p);
   sender = ih->saddr();
   neighbour = index;
   src = rq->rq_src;
   dst = rq->rq_dst;
   hopcount = rq->rq_hop_count;
   bcastid = rq->rq_bcast_id;
   srcseqno = rq->rq_src_seqno;
   timest = rq->rq_timestamp;
   sprintf(pktT,"%d",src);
   strcpy(pktID,pktT);
   strcat(pktID,"-");
   sprintf(pktT,"%d",dst);
   strcat(pktID,pktT);
   strcat(pktID,"-");
   sprintf(pktT,"%.0f",timest);
   strcat(pktID,pktT);
   break;

   case AODVTYPE_RREP:
```

```
   rp = HDR_AODV_REPLY(p);
   sender = rp->rp_src;
   neighbour = index;
   src = ih->saddr();
   dst = ih->daddr();
   hopcount = rp->rp_hop_count;
   timest = rp->rp_timestamp;
   bcastid = -1;
   strcpy(pktType, "RREP");
   sprintf(pktT,"%d",dst);
   strcpy(pktID,pktT);
   strcat(pktID,"-");
   sprintf(pktT,"%d",src);
   strcat(pktID,pktT);
   strcat(pktID,"-");
   sprintf(pktT,"%.0f",timest);
   strcat(pktID,pktT);
   break;
  }
 }

 if (sender != src)
 {
  FLogAlla = fopen("/root/testing/LogAlla.txt","r+");
  FMalNode = fopen("/root/testing/MalNode.txt","a+");
  is_true1 = 0;
  is_true2 = 1;

  while ( ((cha = getc(FLogAlla)) != EOF) )
  {
   fscanf(FLogAlla,"%d %f %d %d %d %d %s %s %d %d %d %d
",&idtemp,&timesttemp,&sendertemp,&receivertemp, \
   &srctemp,&dsttemp,&pktTypetemp,&pktIDtemp,&bcastidtemp, &hcounttemp, &srcseqtemp,
&dstseqtemp);

   //Rule 1: Detecting RREQ Hop Count Modification
     if ((receivertemp == sender) && (strcmp(pktTypetemp,"RREQ") == 0) && (strcmp
(pktIDtemp,pktID) == 0) \
     && (timesttemp == timest))
     {
      if (hopcount == (hcounttemp + 1))
      {
       is_true1 = 1;
      }
     }

     //Rule 2: Detecting RREQ Sequence Number Modification
     if ((receivertemp == sender) && (strcmp(pktTypetemp,"RREQ") == 0) && (strcmp
(pktIDtemp,pktID) == 0) \
        && ((hopcount - hcounttemp) == 1) && (timesttemp == timest))
     {
      if (srcseqno != srcseqtemp)
      {
       is_true2 = 2;
       break;
      }
     }
     cha = getc(FLogAlla);
```

```
    }

    // Response to Rule 1 violation
    if (is_true1 != 1)
    {
     printf("\n Node:%d detects node:%d modified hop count at time:%f",
neighbour,sender,timest);
     getchar();
     fprintf(FMalNode,"\n%f %d %d %s %s", timestM,neighbour,sender,pktTypeM,pktIDM);
    }

    // Response to Rule 2 violation
    if (is_true2 != 1)
    {
     printf("\n Node:%d detects node:%d modified seq number at time:%f",
neighbour,sender,timest);
     getchar();
     fprintf(FMalNode,"\n%f %d %d %s %s", timest,neighbour,sender,pktTypetemp,pktIDtemp);
     //drop(p, DROP_RTR_TTL);
    }
    fclose(FLogAlla);
    fclose(FMalNode);
   }
   if (sender == src)
   {
    FLogAlla = fopen("/root/testing/LogAlla.txt","r+");
    FMalNode = fopen("/root/testing/MalNode.txt","a+");
    is_true3 = 1;
    while ( ((cha = getc(FLogAlla)) != EOF) )
    {
     fscanf(FLogAlla,"%d %f %d %d %d %d %s %s %d %d %d %d
",&idtemp,&timesttemp,&sendertemp,&receivertemp, \
     &srctemp,&dsttemp,&pktTypetemp,&pktIDtemp,&bcastidtemp, &hcounttemp, &srcseqtemp,
&dstseqtemp);

     //Rule3: Fabrication Attack 1
     if ((sender == srctemp) && (strcmp(pktTypetemp,"RREQ") == 0) && (strcmp
(pktIDtemp,pktID) == 0) \
        && (bcastid != bcastidtemp) && (timesttemp == timest))
     {
      is_true3 = 2;
      break;
     }
     cha = getc(FLogAlla);
    }

    // Respond to Rule 3 Violation
    if (is_true3 != 1)
    {
     printf("\n Node:%d detects node:%d sends duplicate RREQ at time:%f",
neighbour,sender,timest);
     getchar();
     fprintf(FMalNode,"\n%f %d %d %s %s", timest,neighbour,sender,pktTypetemp,pktIDtemp);
    }
    fclose(FLogAlla);
    fclose(FMalNode);
   }
}
```

```
// Local Anomaly Detection and Response

#include <stdio.h>
#include <string.h>
#include <aodv/aodv.h>
#include <aodv/aodv_packet.h>
#include "agent.h"

class DetectDrop : public Agent
{
 public:
        DetectDrop();
 protected:
        int command(int argc, const char*const* argv);
 private:
        int    my_var1;
        double my_var2;
        void   Anomaly(void);
        void   CountDrop(char pktTypetemp2[], char pktIDtemp2[], float timetemp, int
source, int suspect);
        void   TestForward(char pktTypetemp2[], char pktIDtemp2[], float timetemp, int
source, int suspect);
        void   CheckFriendRequest(char pktTypetemp2[], char pktIDtemp2[], float
timetemp, int source, int suspect);
};

static class DetectDropClass : public TclClass
{
 public:
        DetectDropClass() : TclClass("Agent/DetectDrop") {}
        TclObject* create(int, const char*const*) {
                return(new DetectDrop());
        }
} class_my_agent;

DetectDrop::DetectDrop() : Agent(PT_AODV)
{
}

int DetectDrop::command(int argc, const char*const* argv)
{
 if(argc == 2)
 {
  if(strcmp(argv[1], "call-detect-drop") == 0)
  {
   Anomaly();
   return(TCL_OK);
  }
 }
 return(Agent::command(argc, argv));
}

void DetectDrop::Anomaly(void)
{
 float timest, timesttemp;
 int sender, receiver, src, dst, receivertemp, sendertemp, srctemp, dsttemp;
 char pktType[15];
 char pktID[15];
 char pktTypetemp[15];
 char pktIDtemp[15];
 char ch;
 char ch2;
 char ch3;
 int tempn=0;
 int temp=0;
 int temp1=0;
 int hopcount, hopcounttemp, id, idtemp, srcseqtemp, dstseqtemp, srcseq, dstseq;
 int bcastid, bcastidtemp;

 int senderM, receiverM, senderS, receiverS;
 char pktTypeM[15];
 char pktIDM[15];
 float timestM,timestS;

 FILE *FLogAlla;
 FILE *FLogAllb;
 FILE *FActivity;
 FILE *FMalNode;
 FILE *FSusNodea;
 FILE *FSusNodeb;
 FILE *FSusNodea2;
 FLogAlla = fopen("/root/testing/LogAlla.txt","r+");
 FLogAllb = fopen("/root/testing/LogAllb.txt","r+");
 FActivity = fopen("/root/testing/Activity.txt","a+");
 float nstime, nsfiletime;
 nstime = Scheduler::instance().clock();
 nsfiletime = 0.0;
 FILE *FTime;
 FTime = fopen("/root/testing/Time.txt","r+");

 while ( !feof (FTime) )
 {
  fscanf(FTime,"%f",&nsfiletime);
 }
 fclose(FTime);

 FTime = fopen("/root/testing/Time.txt","w+");
 fprintf(FTime,"%f\n",nstime);
 fclose(FTime);

 printf("Performing Anomaly Detection Mechanism (Detecting suspicious nodes) ...\n");
 while ( ((ch = getc(FLogAlla)) != EOF) )
 {
  fscanf(FLogAlla,"%d %f %d %d %d %d %s %s %d %d %d %
d",&idtemp,&timesttemp,&sendertemp,&receivertemp, \

&srctemp,&dsttemp,&pktTypetemp,&pktIDtemp,&bcastidtemp,&hopcounttemp,&srcseqtemp,&dstseqtemp);
  if ((receivertemp != srctemp) && (strcmp(pktTypetemp,"RREQ")==0))
  {
   rewind(FLogAllb);
   while ( ((ch2 = getc(FLogAllb)) != (EOF)) )
   {
    fscanf(FLogAllb,"%d %f %d %d %d %d %s %s %d %d %d %d", &id, &timest, &sender,
&receiver, &src, &dst, \
    &pktType, &pktID, &bcastid, &hopcount,&srcseq,&dstseq);
    if ((sender == receivertemp) && (src == srctemp) && (dst == dsttemp) && (strcmp
(pktType,"RREQ")==0))
    {
     fprintf(FActivity,"%d %f %d %d %d %d %s %s %d %d %d %d
\n",idtemp,timesttemp,sendertemp,receivertemp, \

srctemp,dsttemp,pktTypetemp,pktIDtemp,bcastidtemp,hopcounttemp,srcseqtemp,dstseqtemp);//
Log normal activity
     tempn = 1;
     break;
    }
    else if ((sender == receivertemp) && (srctemp == dst) && (dsttemp == src) && (strcmp
(pktType,"RREP")==0))
    {
     fprintf(FActivity,"%d %f %d %d %d %d %s %s %d %d %d %d
\n",idtemp,timesttemp,sendertemp,receivertemp, \

srctemp,dsttemp,pktTypetemp,pktIDtemp,bcastidtemp,hopcounttemp,srcseqtemp,dstseqtemp);//
Log normal activity
     tempn = 1;
     break;
    }
    else
    {
     tempn = 2;
    }
   }
   if (tempn == 2)
   {
    FMalNode = fopen("/root/testing/MalNode.txt","r+");
    rewind(FMalNode);
    while ( !feof (FMalNode) )
    {
     fscanf(FMalNode,"%f %d %d %s %s",&timestM,&senderM,&receiverM,&pktTypeM,&pktIDM);
     if ((sendertemp == senderM) && (receivertemp == receiverM))
     {
      temp=0;
      break;
     }
     else
     {
      temp = 1;
     }
    }
    fclose(FMalNode);
    if (temp == 1)
    {
     printf("Suspicious activity detected ... Node:%d suspects Node:%d malicious
\n",sendertemp,receivertemp);
     FSusNodea = fopen("/root/testing/SusNodea.txt","r+");
     rewind(FSusNodea);
     temp1 = 0;
     while ( !feof (FSusNodea) )
     {
      fscanf(FSusNodea,"%f %d %d",&timestS,&senderS,&receiverS);
      if (((sendertemp == senderS) && (receivertemp == receiverS)) && ((timesttemp -
timestS)<1))
      {
       temp1=0;
       break;
      }
      else
      {
       temp1 = 1;
      }
     }
     fclose(FSusNodea);
     if (temp1==1)
     {
      FSusNodea2 = fopen("/root/testing/SusNodea.txt","a+");
      FSusNodeb = fopen("/root/testing/SusNodeb.txt","a+");
      printf("%f %d %d\n", timesttemp,sendertemp,receivertemp);
      fprintf(FSusNodea2,"%f %d %d\n", timesttemp,sendertemp,receivertemp);
      fprintf(FSusNodeb,"%f %d %d\n", timesttemp,sendertemp,receivertemp);
      fclose(FSusNodea2);
      fclose(FSusNodeb);
     }
     TestForward(pktTypetemp,pktIDtemp,timesttemp,sendertemp,receivertemp);
    }
   }
  }
  else if ((receivertemp == srctemp) && (strcmp(pktTypetemp,"RREQ")==0))
  {
   fprintf(FActivity,"%d %f %d %d %d %d %s %s %d %d %d %d
\n",idtemp,timesttemp,sendertemp,receivertemp, \

srctemp,dsttemp,pktTypetemp,pktIDtemp,bcastidtemp,hopcounttemp,srcseqtemp,dstseqtemp);//
Log normal activity
  }
  ch = getc(FLogAlla);
 }
 fclose(FLogAlla);
 fclose(FLogAllb);
 fclose(FActivity);
}

void DetectDrop::TestForward(char pktTypetemp2[], char pktIDtemp2[],float timetemp, int
source, int suspect)
{
 printf("Anomaly Test 1: Node:%d checks Node:%d activity history\n",source,suspect);
 FILE *FActivity2;
 FILE *FGood;
 FILE *FGood2;
 int idtemp, receivertemp, sendertemp, srctemp, dsttemp;
 float timestM, timesttemp;
 char pktTypetemp[15];
 char pktIDtemp[15];
 char pktTypeM[15];
 char pktIDM[15];
 int count;
 int temp = 0;
 int senderM, receiverM;
 int hopcounttemp,srcseqtemp,dstseqtemp,bcastidtemp;
 FActivity2 = fopen("/root/testing/Activity.txt","r+");

 while ( !feof (FActivity2) )
 {
  count = 0;
  fscanf(FActivity2,"%d %f %d %d %d %d %s %s %d %d %d %
d",&idtemp,&timesttemp,&sendertemp,&receivertemp, \

&srctemp,&dsttemp,&pktTypetemp,&pktIDtemp,&bcastidtemp,&hopcounttemp,&srcseqtemp,&dstseqtemp);
  if ((suspect == sendertemp) && (suspect != srctemp) && (receivertemp == source))
  {
   count = 1;
   break;
  }
  else
  {
   count = 0;
  }
 }
 if (count == 1)
 {
  printf("Test 1 Result: Node:%d detects Node:%d not malicious\n",source,suspect);
  FGood = fopen("/root/testing/Good.txt","a+");
  rewind(FGood);
  while ( !feof (FGood) )
  {
   fscanf(FGood,"%f %d %d %s %s",&timestM,&senderM,&receiverM,&pktTypeM,&pktIDM);
   if ((source == senderM) && (suspect == receiverM))
   {
    temp=0;
    break;
   }
   else
    temp = 1;
  }
  fclose(FGood);
  if (temp==1)
  {
   FGood2 = fopen("/root/testing/Good.txt","a+");
   fprintf(FGood2,"%f %d %d %s %s\n", timetemp,source,suspect,pktTypetemp2,pktIDtemp2);
   fclose(FGood2);
  }
 }
 else if (count != 1)
 {
  printf("Test 1 Result: Can't make decision -> Proceed to Test 2\n");
  CountDrop(pktTypetemp2,pktIDtemp2,timetemp,source,suspect);
 }
 fclose(FActivity2);
}

void DetectDrop::CountDrop(char pktTypetemp2[], char pktIDtemp2[], float timetemp, int
source, int suspect)
{
printf("Anomaly Test 2: Node:%d checks Node:%d packet drop history\n",source,suspect);

 FILE *FSusNodea;
 FILE *FMalNode;
 FILE *FMalNode2;
 FILE *FRequest;
 FILE *FRequest2;
 char ch, ch2, ch3;
 int sender, receiver, src, dst, receivertemp, sendertemp, srctemp, dsttemp;
 int senderM, receiverM, senderR, receiverR;
 char pktType[15];
 char pktID[15];
 char pktTypetemp[15];
 char pktIDtemp[15];
 char pktTypeM[15];
 char pktIDM[15];
 int count;
 int temp=0;
 int temp3=0;
 int bcastid,bcastidtemp;
 float timestM, timesttemp, timestR;
 float timeM = 11.111111;
 float nstime;
 nstime = Scheduler::instance().clock();
 FSusNodea = fopen("/root/testing/SusNodea.txt","r+");
 count =0;
 rewind(FSusNodea);

 while ( !feof (FSusNodea) )
 {
  fscanf(FSusNodea,"%f %d %d",&timesttemp,&sendertemp,&receivertemp);
  if ((source == sendertemp) && (suspect == receivertemp))
  {
   count = count + 1;
  }
 }
 if (count >=6)
 {
  printf("Test 2 Result: Node:%d detects Node:%d malicious\n", source,suspect);
  FMalNode = fopen("/root/testing/MalNode.txt","r+");
  rewind(FMalNode);
  while ( !feof (FMalNode) )
  {
   fscanf(FMalNode,"%f %d %d %s %s",&timestM,&senderM,&receiverM,&pktTypeM,&pktIDM);
   if ((source == senderM) && (suspect == receiverM))
   {
    temp=0;
    break;
   }
   else
    temp = 1;
  }
  fclose(FMalNode);
  if (temp==1)
  {
   FMalNode2 = fopen("/root/testing/MalNode.txt","a+");
   fprintf(FMalNode2,"%f %d %d %s %s\n", nstime,source,suspect,pktTypetemp2,pktIDtemp2);
   fclose(FMalNode2);
  }
 }
 else if (count < 6)
 {
  printf("Local anomaly detection failed to make a decision -> Send global detection
request\n");
  FRequest = fopen("/root/testing/Request.txt","r+");
  rewind(FRequest);
  while ( !feof (FRequest) )
  {
   fscanf(FRequest,"%f %d %d",&timestR,&senderR,&receiverR);
   if ((source == senderR) && (suspect == receiverR))
   {
    temp3=0;
    break;
   }
   else
    temp3 = 1;
  }
  fclose(FRequest);
  if (temp3==1)
  {
   FRequest2 = fopen("/root/testing/Request.txt","a+");
   fprintf(FRequest2,"%f %d %d\n",nstime,source,suspect);
   fclose(FRequest2);
  }
 }
 fclose(FSusNodea);
}
```

```
//Global Detection

int Friend::command(int argc, const char*const* argv)
{
 MobileNode *mn;
 MobileNode *mn2;
 if (mn->address() != mn2->address())
 {
  int distance = 0;
  distance = sqrt(((grid_x-grid2_x)*(grid_x-grid2_x)) + ((grid_y-grid2_y)*(grid_y-
grid2_y)));
  if (distance <= mn->radius())
  {
   while ( !feof (FFriend1) )
   {
    fscanf(FFriend1,"%d %d",&firstnode,&secondnode);
    if ((firstnode == mn->address()) && (secondnode == mn2->address()))
    {
     ExchangeRequest(mn2->address(), mn->address());
    }
   }

   while ( !feof (FFriend2) )
   {
    fscanf(FFriend1,"%d %d",&firstnode,&secondnode);
    if ((firstnode == mn->address()) && (secondnode == mn2->address()))
    {
     ExchangeRequest(mn2->address(), mn->address());
    }
   }

  }
 }
}

void Friend::ExchangeRequest(int second, int first)
{
 FILE *FRequest;
 FILE *FFriendRequest;
 FILE *FMalNode;
 FILE *FMalNode2;
 FILE *FMalNode3;
 FILE *FActivity2;
 FILE *FGood;
 FILE *FGood2;
 int fnode, snode, lfnode, lsnode, idnode,id, idA, bcastidA, hopcountA,srcseqA,dstseqA;
 int senderM,receiverM,senderG,receiverG,senderA,receiverA,srcA,dstA;
 char pktTypeM[15];
 char pktIDM[15];
 char pktTypeA[15];
 char pktIDA[15];
 char pktTypeG[15];
 char pktIDG[15];
 int temp1 = 0;
 int temp2 = 0;
 int temp4 = 0;
 int count, tempa;
 int input1, input2, input3;
 float time, ltime, timestM,timeA,timeG;
 float nstime;
 FRequest = fopen("/root/testing/Request.txt","r+");
 FFriendRequest = fopen("/root/testing/FriendRequest.txt","a+");
 nstime = Scheduler::instance().clock();
 while ( !feof (FRequest) )
 {
  fscanf(FRequest,"%f %d %d",&time,&fnode,&snode);
  if (fnode == first)
  {
   FMalNode = fopen("/root/testing/MalNode.txt","r+");
   rewind(FMalNode);
   temp1 = 0;
   while ( !feof (FMalNode) )
   {
    fscanf(FMalNode,"%f %d %d %s %s",&timestM,&senderM,&receiverM,&pktTypeM,&pktIDM);
    if ((first == senderM) && (snode == receiverM))
    {
     temp1 = 1;
     break;
    }
   }
   fclose(FMalNode);

   if (temp1 != 1)
   {
    FMalNode3 = fopen("/root/testing/MalNode.txt","r+");
    rewind(FMalNode3);
    temp2 = 0;
    printf("Global Detection Test 1 -> Check MalNode List\n");
    while ( !feof (FMalNode3) )
    {
     fscanf(FMalNode3,"%f %d %d %s %s",&timestM,&senderM,&receiverM,&pktTypeM,&pktIDM);
     if ((second == senderM) && (snode == receiverM))
     {
      temp2 = 1;
      break;
     }
    }
    fclose(FMalNode3);
    if (temp2 == 1)
    {
     printf("Global Detection Result (Test 1) -> Node:%d detect Node:%d malicious with
the help of \
     Node:%d\n", first,snode,second);
     FMalNode2 = fopen("/root/testing/MalNode.txt","a+");
     fprintf(FMalNode2,"%f %d %d %s %s\n",nstime,first,snode,pktTypeM,pktIDM);
     fclose(FMalNode2);
    }
    else if (temp2 != 1)
    {
     printf("Global Detection Test 2 -> Check Packet Forwarding History\n");
     //getchar();
     FActivity2 = fopen("/root/testing/Activity.txt","r+");
     count = 0;
     while ( !feof (FActivity2) )
     {
      fscanf(FActivity2,"%d %f %d %d %d %d %s %s %d %d %d %
d",&idA,&timeA,&senderA,&receiverA,&srcA,&dstA, \
      &pktTypeA,&pktIDA,&bcastidA,&hopcountA,&srcseqA,&dstseqA);
      if ((snode == senderA) && (snode != srcA) && (receiverA == second))
      {
       count = 1;
       break;
      }
      else
      {
       count = 0;
      }
     }
     fclose(FActivity2);
     if (count == 1)
     {
      printf("Global Detection Result (Test 2) -> Node:%d detect Node:%d not malicious
with the help of \
      Node:%d\n", first,snode,second);
      FGood = fopen("/root/testing/Good.txt","r+");
      rewind(FGood);
      tempa = 0;
      while ( !feof (FGood) )
      {
       fscanf(FGood,"%f %d %d %s %s",&timeG,&senderG,&receiverG,&pktTypeG,&pktIDG);
       if ((first == senderG) && (snode == receiverG))
       {
        tempa=1;
        break;
       }
       else
        tempa = 0;
      }
      fclose(FGood);
      if (tempa!=1)
      {
       FGood2 = fopen("/root/testing/Good.txt","a+");
       fprintf(FGood2,"%f %d %d %s %s\n", nstime,first,snode,pktTypeA,pktIDA);
       fclose(FGood2);
      }
     }
     else if (count != 1)
     {
      printf("Node:%d can't help Node:%d making decision -> Add the suspicious Node:%d
to its \
      FriendRequest list\n", second, first, snode);
      rewind(FFriendRequest);
      temp4 = 0;
      while ( !feof (FFriendRequest) )
      {
       fscanf(FFriendRequest,"%d %d %d %f",&idnode,&lsnode,&lfnode,&ltime);
       if ((second == idnode) && (lsnode == snode) && (lfnode == fnode))
       {
        temp4 = 1;
        break;
       }
      }
      if (temp4 != 1)
      {
       fprintf(FFriendRequest,"\n%d %d %f",second,snode,first,nstime);
      }
     }
    }
   }
  }
 }
 fclose(FRequest);
 fclose(FFriendRequest);
}
```

```
//Global Response

int Friend::command(int argc, const char*const* argv)
{
 MobileNode *mn;
 MobileNode *mn2;
 if (mn->address() != mn2->address())
 {
  int distance = 0;
  distance = sqrt(((grid_x-grid2_x)*(grid_x-grid2_x)) + ((grid_y-grid2_y)*(grid_y-
grid2_y)));
  if (distance <= mn->radius())
  {
   while ( !feof (FFriend1) )
   {
    fscanf(FFriend1,"%d %d",&firstnode,&secondnode);
    if ((firstnode == mn->address()) && (secondnode == mn2->address()))
    {
     exchangelocal(mn2->address(), mn->address());
     exchangeglobal(mn2->address(), mn->address());
    }
   }

   while ( !feof (FFriend2) )
   {
    fscanf(FFriend1,"%d %d",&firstnode,&secondnode);
    if ((firstnode == mn->address()) && (secondnode == mn2->address()))
    {
     exchangelocal(mn2->address(), mn->address());
     exchangeglobal(mn2->address(), mn->address());
    }
   }

  }
 }
}

void Friend::exchangelocal(int second, int first)
{
 FILE *FMalNode;
 FILE *FGlobal;
 int fnode, snode, lfnode, lsnode, idnode,id;
 int temp = 0;
 int dum, dum2;
 char ch, ch2;
 char pktTypeM[15];
 char pktIDM[15];
 float time, ltime;
 FMalNode = fopen("/root/testing/MalNode.txt","r");
 FGlobal = fopen("/root/testing/GlobalMis.txt","a+");
 while ( !feof (FMalNode) )
 {
  fscanf(FMalNode,"%f %d %d %s %s",&time,&fnode,&snode,&pktTypeM,&pktIDM);
  if (fnode == second)
  {
   rewind(FGlobal);
   temp = 0;
   while ( !feof (FGlobal) )
   {
    fscanf(FGlobal,"%d %d %d %f",&idnode,&lsnode,&lfnode,&ltime);
    if ((first == idnode) && (lsnode == snode) && (lfnode == fnode))
    {
     temp = 2;
    }
   }
   if (temp != 2)
   {
    printf("Node:%d sends its local malicious node list to Node:%d global malicious list
\n", second, first);
    fprintf(FGlobal,"\n%d %d %d %f",first,snode,second,time);
   }
  }
 }
 fclose(FMalNode);
 fclose(FGlobal);
}

void Friend::exchangeglobal(int second, int first)
{
 FILE *FGlobal;
 FILE *FGlobalEx;
 int gfnode, gsnode, gfnode2, gsnode2, idnode, idnode2,id;
 int temp2 = 0;
 int dum, dum2;
 char ch, ch2;
 float gtime, gtime2;
 FGlobalEx = fopen("/root/testing/GlobalMisEx.txt","a+");
 FGlobal = fopen("/root/testing/GlobalMis.txt","r");
 while ( !feof (FGlobal) )
 {
  fscanf(FGlobal,"%d %d %d %f",&idnode,&gsnode,&gfnode,&gtime);
  if ((idnode == second) && (gfnode != first))
  {
   rewind(FGlobalEx);
   temp2 = 0;
   while ( !feof (FGlobalEx) )
   {
    fscanf(FGlobalEx,"%d %d %d %f",&idnode2,&gsnode2,&gfnode2,&gtime2);
    if ((idnode2 == first) && (gfnode2 == second) && (gsnode2 == gsnode))
    {
     temp2 = 2;
    }
   }
   if (temp2 != 2)
   {
    printf("Node:%d sends its global malicious node list to Node:%d global malicious
(ex) list\n", second, first);
    fprintf(FGlobalEx,"\n%d %d %d %f",first,gsnode,second,gtime);
   }
  }
 }
 fclose(FGlobal);
 fclose(FGlobalEx);
}
```

Appendix E

Cbrgen.tcl Script and TCP Transmission Example

```
#
#  Copyright (c) 1999 by the University of Southern California
#  All rights reserved.
#
#  This program is free software; you can redistribute it and/or
#  modify it under the terms of the GNU General Public License,
#  version 2, as published by the Free Software Foundation.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License along
#  with this program; if not, write to the Free Software Foundation, Inc.,
#  59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.
#
#  The copyright of this module includes the following
#  linking-with-specific-other-licenses addition:
#
#  In addition, as a special exception, the copyright holders of
#  this module give you permission to combine (via static or
#  dynamic linking) this module with free software programs or
#  libraries that are released under the GNU LGPL and with code
#  included in the standard release of ns-2 under the Apache 2.0
#  license or under otherwise-compatible licenses with advertising
#  requirements (or modified versions of such code, with unchanged
#  license).  You may copy and distribute such a system following the
#  terms of the GNU GPL for this module and the licenses of the
#  other code concerned, provided that you include the source code of
#  that other code when and as the GNU GPL requires distribution of
#  source code.
#
#  Note that people who make modified versions of this module
#  are not obligated to grant this special exception for their
#  modified versions; it is their choice whether to do so.  The GNU
#  General Public License gives permission to release a modified
#  version without this exception; this exception also makes it
#  possible to release a modified version which carries forward this
#  exception.

# Traffic source generator from CMU's mobile code.
#
# $Header: /nfs/jade/vint/CVSROOT/ns-2/indep-utils/cmu-scen-gen/cbrgen.tcl,v 1.4
2005/09/16 03:05:39 tomh Exp $

# ======================================================================
# Default Script Options
# ======================================================================
set opt(nn)             0               ;# Number of Nodes
set opt(seed)           0.0
set opt(mc)             0
set opt(pktsize)        512

set opt(rate)           0
set opt(interval)       0.0             ;# inverse of rate
set opt(type)           ""
```

```
# ======================================================================

proc usage {} {
    global argv0

    puts "\nusage: $argv0 \[-type cbr|tcp\] \[-nn nodes\] \[-seed seed\] \[-mc
connections\] \[-rate rate\]\n"
}

proc getopt {argc argv} {
        global opt
        lappend optlist nn seed mc rate type

        for {set i 0} {$i < $argc} {incr i} {
                set arg [lindex $argv $i]
                if {[string range $arg 0 0] != "-"} continue

                set name [string range $arg 1 end]
                set opt($name) [lindex $argv [expr $i+1]]
        }
}

proc create-cbr-connection { src dst } {
        global rng cbr_cnt opt

        set stime [$rng uniform 0.0 180.0]

        puts "#\n# $src connecting to $dst at time $stime\n#"

        ##puts "set cbr_($cbr_cnt) \[\$ns_ create-connection \
                ##CBR \$node_($src) CBR \$node_($dst) 0\]";
        puts "set udp_($cbr_cnt) \[new Agent/UDP\]"
        puts "\$ns_ attach-agent \$node_($src) \$udp_($cbr_cnt)"
        puts "set null_($cbr_cnt) \[new Agent/Null\]"
        puts "\$ns_ attach-agent \$node_($dst) \$null_($cbr_cnt)"
        puts "set cbr_($cbr_cnt) \[new Application/Traffic/CBR\]"
        puts "\$cbr_($cbr_cnt) set packetSize_ $opt(pktsize)"
        puts "\$cbr_($cbr_cnt) set interval_ $opt(interval)"
        puts "\$cbr_($cbr_cnt) set random_ 1"
        puts "\$cbr_($cbr_cnt) set maxpkts_ 10000"
        puts "\$cbr_($cbr_cnt) attach-agent \$udp_($cbr_cnt)"
        puts "\$ns_ connect \$udp_($cbr_cnt) \$null_($cbr_cnt)"

        puts "\$ns_ at $stime \"\$cbr_($cbr_cnt) start\""

        incr cbr_cnt
}

proc create-tcp-connection { src dst } {
        global rng cbr_cnt opt

        set stime [$rng uniform 0.0 100.0]

        puts "#\n# $src connecting to $dst at time $stime\n#"

        puts "set tcp_($cbr_cnt) \[\$ns_ create-connection \
                TCP \$node_($src) TCPSink \$node_($dst) 0\]";
        puts "\$tcp_($cbr_cnt) set window_ 32"
        puts "\$tcp_($cbr_cnt) set packetSize_ $opt(pktsize)"

        puts "set ftp_($cbr_cnt) \[\$tcp_($cbr_cnt) attach-source FTP\]"


        puts "\$ns_ at $stime \"\$ftp_($cbr_cnt) start\""
        puts "\$ns_ at [expr ($stime+0.5)] \"\$ftp_($cbr_cnt) stop\""

        incr cbr_cnt
}

# ======================================================================

getopt $argc $argv

if { $opt(type) == "" } {
    usage
    exit
} elseif { $opt(type) == "cbr" } {
    if { $opt(nn) == 0 || $opt(seed) == 0.0 || $opt(mc) == 0 || $opt(rate) == 0 } {
        usage
        exit
    }

    set opt(interval) [expr 1 / $opt(rate)]
    if { $opt(interval) <= 0.0 } {
        puts "\ninvalid sending rate $opt(rate)\n"
        exit
    }
}

puts "#\n# nodes: $opt(nn), max conn: $opt(mc), send rate: $opt(interval), seed: $opt
(seed)\n#"

set rng [new RNG]
$rng seed $opt(seed)

set u [new RandomVariable/Uniform]
$u set min_ 0
$u set max_ 100
$u use-rng $rng

set cbr_cnt 0
set src_cnt 0

for {set i 0} {$i < $opt(nn) } {incr i} {

        set x [$u value]

        if {$x < 50} {continue;}

        incr src_cnt

        set dst [expr ($i+1) % [expr $opt(nn) + 1] ]
        #if { $dst == 0 } {
            #set dst [expr $dst + 1]
            #}
        if { $opt(type) == "cbr" } {
                create-cbr-connection $i $dst
        } else {
                create-tcp-connection $i $dst
        }

        if { $cbr_cnt == $opt(mc) } {
                break
        }

        if {$x < 75} {continue;}

        set dst [expr ($i+2) % [expr $opt(nn) + 1] ]
        #if { $dst == 0 } {
                #set dst [expr $dst + 1]
        #}

        if { $opt(type) == "cbr" } {
                create-cbr-connection $i $dst
        } else {
                create-tcp-connection $i $dst
        }

        if { $cbr_cnt == $opt(mc) } {
                break
        }
}

puts "#\n#Total sources/connections: $src_cnt/$cbr_cnt\n#"
```

```
#
# nodes: 100, max conn: 20, send rate: 0.0, seed: 1
#
#
# 1 connecting to 2 at time 1.4204660437165137
#
set tcp_(0) [$ns_ create-connection  TCP $node_(1) TCPSink $node_(2) 0]
$tcp_(0) set window_ 32
$tcp_(0) set packetSize_ 512
set ftp_(0) [$tcp_(0) attach-source FTP]
$ns_ at 1.4204660437165137 "$ftp_(0) start"
$ns_ at 1.9204660437165137 "$ftp_(0) stop"
#
# 4 connecting to 5 at time 31.296177176430906
#
set tcp_(1) [$ns_ create-connection  TCP $node_(4) TCPSink $node_(5) 0]
$tcp_(1) set window_ 32
$tcp_(1) set packetSize_ 512
set ftp_(1) [$tcp_(1) attach-source FTP]
$ns_ at 31.296177176430906 "$ftp_(1) start"
$ns_ at 31.796177176430906 "$ftp_(1) stop"
#
# 4 connecting to 6 at time 81.64760516101849
#
set tcp_(2) [$ns_ create-connection  TCP $node_(4) TCPSink $node_(6) 0]
$tcp_(2) set window_ 32
$tcp_(2) set packetSize_ 512
set ftp_(2) [$tcp_(2) attach-source FTP]
$ns_ at 81.64760516101849 "$ftp_(2) start"
$ns_ at 82.14760516101849 "$ftp_(2) stop"
#
# 6 connecting to 7 at time 30.90790576809454
#
set tcp_(3) [$ns_ create-connection  TCP $node_(6) TCPSink $node_(7) 0]
$tcp_(3) set window_ 32
$tcp_(3) set packetSize_ 512
set ftp_(3) [$tcp_(3) attach-source FTP]
$ns_ at 30.90790576809454 "$ftp_(3) start"
$ns_ at 31.40790576809454 "$ftp_(3) stop"
#
# 7 connecting to 8 at time 16.414540641202844
#
set tcp_(4) [$ns_ create-connection  TCP $node_(7) TCPSink $node_(8) 0]
$tcp_(4) set window_ 32
$tcp_(4) set packetSize_ 512
set ftp_(4) [$tcp_(4) attach-source FTP]
$ns_ at 16.414540641202844 "$ftp_(4) start"
$ns_ at 16.914540641202844 "$ftp_(4) stop"
#
# 7 connecting to 9 at time 4.2794557308216836
#
set tcp_(5) [$ns_ create-connection  TCP $node_(7) TCPSink $node_(9) 0]
$tcp_(5) set window_ 32
$tcp_(5) set packetSize_ 512
set ftp_(5) [$tcp_(5) attach-source FTP]
$ns_ at 4.2794557308216836 "$ftp_(5) start"
$ns_ at 4.7794557308216836 "$ftp_(5) stop"
#
```

```
# 8 connecting to 9 at time 11.380824824506801
#
set tcp_(6) [$ns_ create-connection  TCP $node_(8) TCPSink $node_(9) 0]
$tcp_(6) set window_ 32
$tcp_(6) set packetSize_ 512
set ftp_(6) [$tcp_(6) attach-source FTP]
$ns_ at 11.380824824506801 "$ftp_(6) start"
$ns_ at 11.880824824506801 "$ftp_(6) stop"
#
# 9 connecting to 10 at time 42.365673623218051
#
set tcp_(7) [$ns_ create-connection  TCP $node_(9) TCPSink $node_(10) 0]
$tcp_(7) set window_ 32
$tcp_(7) set packetSize_ 512
set ftp_(7) [$tcp_(7) attach-source FTP]
$ns_ at 42.365673623218051 "$ftp_(7) start"
$ns_ at 42.865673623218051 "$ftp_(7) stop"
#
# 9 connecting to 11 at time 17.480525382552543
#
set tcp_(8) [$ns_ create-connection  TCP $node_(9) TCPSink $node_(11) 0]
$tcp_(8) set window_ 32
$tcp_(8) set packetSize_ 512
set ftp_(8) [$tcp_(8) attach-source FTP]
$ns_ at 17.480525382552543 "$ftp_(8) start"
$ns_ at 17.980525382552543 "$ftp_(8) stop"
#
# 11 connecting to 12 at time 34.874102536064619
#
set tcp_(9) [$ns_ create-connection  TCP $node_(11) TCPSink $node_(12) 0]
$tcp_(9) set window_ 32
$tcp_(9) set packetSize_ 512
set ftp_(9) [$tcp_(9) attach-source FTP]
$ns_ at 34.874102536064619 "$ftp_(9) start"
$ns_ at 35.374102536064619 "$ftp_(9) stop"
#
# 11 connecting to 13 at time 25.808794855051115
#
set tcp_(10) [$ns_ create-connection  TCP $node_(11) TCPSink $node_(13) 0]
$tcp_(10) set window_ 32
$tcp_(10) set packetSize_ 512
set ftp_(10) [$tcp_(10) attach-source FTP]
$ns_ at 25.808794855051115 "$ftp_(10) start"
$ns_ at 26.308794855051115 "$ftp_(10) stop"
#
# 13 connecting to 14 at time 46.611593638831565
#
set tcp_(11) [$ns_ create-connection  TCP $node_(13) TCPSink $node_(14) 0]
$tcp_(11) set window_ 32
$tcp_(11) set packetSize_ 512
set ftp_(11) [$tcp_(11) attach-source FTP]
$ns_ at 46.611593638831565 "$ftp_(11) start"
$ns_ at 47.111593638831565 "$ftp_(11) stop"
#
# 14 connecting to 15 at time 86.206728120430711
#
set tcp_(12) [$ns_ create-connection  TCP $node_(14) TCPSink $node_(15) 0]
$tcp_(12) set window_ 32
```

```
$tcp_(12) set packetSize_ 512
set ftp_(12) [$tcp_(12) attach-source FTP]
$ns_ at 86.206728120430711 "$ftp_(12) start"
$ns_ at 86.706728120430711 "$ftp_(12) stop"
#
# 15 connecting to 16 at time 21.715945946851718
#
set tcp_(13) [$ns_ create-connection  TCP $node_(15) TCPSink $node_(16) 0]
$tcp_(13) set window_ 32
$tcp_(13) set packetSize_ 512
set ftp_(13) [$tcp_(13) attach-source FTP]
$ns_ at 21.715945946851718 "$ftp_(13) start"
$ns_ at 22.215945946851718 "$ftp_(13) stop"
#
# 15 connecting to 17 at time 24.122562782896011
#
set tcp_(14) [$ns_ create-connection  TCP $node_(15) TCPSink $node_(17) 0]
$tcp_(14) set window_ 32
$tcp_(14) set packetSize_ 512
set ftp_(14) [$tcp_(14) attach-source FTP]
$ns_ at 24.122562782896011 "$ftp_(14) start"
$ns_ at 24.622562782896011 "$ftp_(14) stop"
#
# 16 connecting to 17 at time 67.734894327695898
#
set tcp_(15) [$ns_ create-connection  TCP $node_(16) TCPSink $node_(17) 0]
$tcp_(15) set window_ 32
$tcp_(15) set packetSize_ 512
set ftp_(15) [$tcp_(15) attach-source FTP]
$ns_ at 67.734894327695898 "$ftp_(15) start"
$ns_ at 68.234894327695898 "$ftp_(15) stop"
#
# 16 connecting to 18 at time 76.223189279540989
#
set tcp_(16) [$ns_ create-connection  TCP $node_(16) TCPSink $node_(18) 0]
$tcp_(16) set window_ 32
$tcp_(16) set packetSize_ 512
set ftp_(16) [$tcp_(16) attach-source FTP]
$ns_ at 76.223189279540989 "$ftp_(16) start"
$ns_ at 76.723189279540989 "$ftp_(16) stop"
#
# 17 connecting to 18 at time 40.551907727751839
#
set tcp_(17) [$ns_ create-connection  TCP $node_(17) TCPSink $node_(18) 0]
$tcp_(17) set window_ 32
$tcp_(17) set packetSize_ 512
set ftp_(17) [$tcp_(17) attach-source FTP]
$ns_ at 40.551907727751839 "$ftp_(17) start"
$ns_ at 41.051907727751839 "$ftp_(17) stop"
#
# 17 connecting to 19 at time 10.919847158212145
#
set tcp_(18) [$ns_ create-connection  TCP $node_(17) TCPSink $node_(19) 0]
$tcp_(18) set window_ 32
$tcp_(18) set packetSize_ 512
set ftp_(18) [$tcp_(18) attach-source FTP]
$ns_ at 10.919847158212145 "$ftp_(18) start"
$ns_ at 11.419847158212145 "$ftp_(18) stop"
```

```
#
# 20 connecting to 21 at time 94.62649533274886
#
set tcp_(19) [$ns_ create-connection  TCP $node_(20) TCPSink $node_(21) 0]
$tcp_(19) set window_ 32
$tcp_(19) set packetSize_ 512
set ftp_(19) [$tcp_(19) attach-source FTP]
$ns_ at 94.62649533274886 "$ftp_(19) start"
$ns_ at 95.12649533274886 "$ftp_(19) stop"
#
#Total sources/connections: 13/20
#
```

Appendix F

Publications

1. **Mobile Ad Hoc Networks: New Technology, New Challenges.** Online Article. South West Branch British Computer Society (BCS) Website. Unrefereed.

2. **Attacks against Mobile Ad Hoc Networks Routing Protocols.** In Proceedings of the Fifth Annual Postgraduate Network Symposium (PGNeT 2004). Reviewed.

3. **Intrusion Detection in Mobile Ad Hoc Networks.** Poster Presentation in the International Network Conference (INC 2004).

4. **A Two-tier Intrusion Detection System for Mobile Ad Hoc Networks.** In Proceedings of the 4th European Conference on Information Warfare and Security (ECIW 2005). Reviewed.

5. **A Two-tier Intrusion Detection System for Mobile Ad Hoc Networks – A Friend Approach.** In Proceedings of the Intelligence and Security Informatics (ISI 2006). Reviewed.

Mobile Ad Hoc Networks: New Technology, New Challenges

S.A. Razak, S.M. Furnell, P.J. Brooke

Network Research Group, School of Computing, Communications & Electronics
University of Plymouth, Plymouth, United Kingdom
E-mail: nrg@plymouth.ac.uk

Introduction

Networks are now one of the most popular manifestations of computing technology, and the most significant example is the Internet. Online banking, email applications, online trading, instant messaging and news broadcasting are only a few examples to show the massive use of the Internet. Recent progress and advances in the communication technologies have introduced another type of computer network, Mobile Ad Hoc Networks (MANETs). MANETs represent a combination of peer-to-peer techniques, wireless communications and mobile computing, and have become an important field of research in recent years. This new technology has been widely used to support communications in an environment that might be impossible to deploy infrastructure networks such as in military battlefields and disaster recovery sites. In addition, this technology might be used to replace infrastructure networks where employing the wireless networks is more practical [1]. However, employing MANETs in any environment is not as easy as one might think. Besides the routing protocols, other issues such as security, Quality of Service (QoS), resource management, and auto-configuration must be considered before deployment. In this paper, we will introduce some of the important characteristics that make MANETs unique compared to the other type of computer networks. We will also present some of the research challenges that can be explored in the area.

MANETs and the Wireless Families

Wireless networks assist in the communication activities between two nodes to provide more flexible and easier connections. According to the National Institute of Standards and Technology (NIST) in [2], we can categorize wireless networks into three main categories; Wireless Wide Area Network (WWAN), Wireless Local Area Network (WLAN), and Wireless Personal Area Network (WPAN).

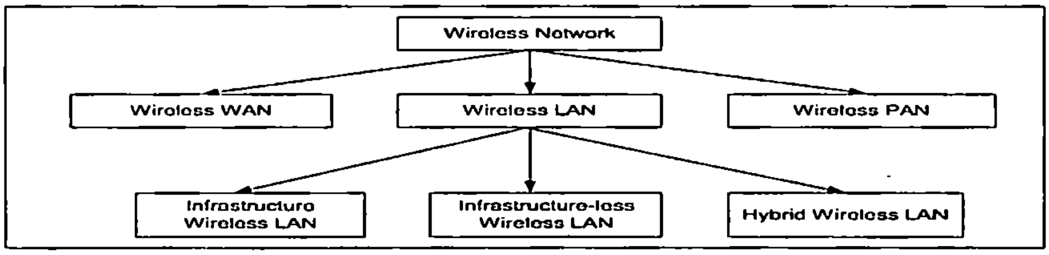


Figure 1: Three types of wireless network

Figure 1 illustrates three types of wireless networks and the descriptions for each technology are as follows:

- *Wireless WAN* is a computer network using wireless networking devices to transfer data in a wide coverage area. Such technology is generally managed by a service provider and usually offers services to a huge number of users. Examples of WAN technology are CDMA, GSM, GPRS, CDPD and satellite networks [3].

- *Wireless PAN* is a collection of personal devices connecting to each other in a limited coverage area. Technologies related to wireless PAN are IrDA, RFID and Bluetooth. Unlike in a wireless WAN, network connectivity in wireless PAN is completely controlled by the user who operates it, not by the service provider. Another significant difference compared to a wireless WAN is that there is no charge for connection, as it uses free unlicensed frequency allocation. More explanation about licensed and unlicensed frequencies can be found in [4].

- *Wireless LAN* is a computer network designed to allow greater flexibility and mobility in a local area network connection. Similar to wireless PAN, this technology also uses an unlicensed frequency allocation to establish wireless connection. Since no service provider exists in the networks, the users must take responsibility to control and manage all the network operations by themselves. In general, wireless LAN can be divided into two main categories, namely infrastructure and infrastructure-less networks as described in [5]. However, recent research in [6] has introduced the third type of wireless LAN technologies; the hybrid wireless LAN. The descriptions of each wireless LAN technology are as follows:

- *Infrastructure wireless network* consists of several mobile devices connected directly to an access point using wireless transmissions. An access point is a station that transmits and receives data from users within the network and can serve as the point of interconnection between the WLAN and a fixed wire network.

- *Infrastructure-less wireless network*, which is also known as MANET, is a network comprised only of mobile wireless devices. Nodes communicate directly with each other without the aid of any access points or wired backbone.

- *Hybrid networks* can be used to ease the deployment of an infrastructure wireless network. The main problem in infrastructure wireless networks is the constraints in placing the access points. By exploiting multi-hop capabilities in the ad hoc networks, all nodes (including those in the outer range) are able to reach the access point to connect to the Internet.

Figure 2 illustrates an example of wireless LAN technologies consisting of infrastructure wireless LAN, MANET and hybrid wireless LAN. White arrows show that mobile devices in MANET can connect to each other without an access point. On the other hand, each mobile node in an infrastructure wireless LAN needs to connect directly to the access point to establish connections amongst them. Black arrows show direct connections between nodes and access point. Nodes in the circle area connected to each other to establish connection between MANET and infrastructure wireless LAN. Wireless networks created in this way are also known as hybrid wireless networks.

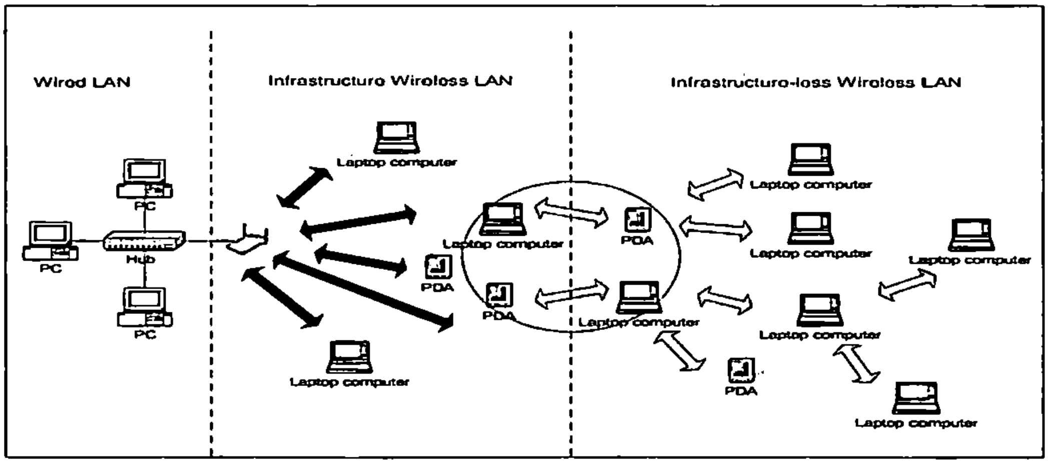


Figure 2: Wireless LAN technologies

MANET Characteristics

The Mobile Ad Hoc Network is one of the emerging technologies nowadays. The emergence of this technology are not restricted to the variety of wireless devices and applications introduced to the market, but also in the number of research works carried out in this field. Generally, there are three main research areas in MANET: enabling technology; networking; and application and middleware [7]. Among these areas, only networking has been given special attention by researchers. However, less focus in the two remaining areas does not mean that they are not important. The focus has been given to the networking area because MANETs have a set of unique characteristics, which add the difficulties in providing effective and efficient communications. In general, MANETs have the following characteristics [5, 8]:

- *Dynamic network topology*: Ad hoc networks are highly dynamic in nature. Nodes in the ad hoc network are mobile and connected to each other via wireless links. Wireless connectivity allows nodes to join the network and dynamically associate to establish routing among themselves. The associations are often created and torn down without prior notice and thus make the ad hoc network topologies unpredictable. The topologies become more complex when nodes in the ad hoc networks established a connection to any public infrastructure network.

- *Distributed operations*: Operations in the ad hoc networks are performed in a distributed manner. Successful routing in ad hoc networks, for example, needs participation among nodes to collaborate in the route discovery process. Besides, since there is no central control for the networks, all the management processes in ad hoc networks must be carried out in a distributed manner.

- *Infrastructure-less*: In ad hoc networks, fixed infrastructure and specialized hardware that help in communication operations are necessarily absent. Besides, nodes participating in the network have not been given any specific roles such as servers, routers or gateways. These situations prevent the deployment of hierarchical node relationships and thus make security mechanisms that depend on these relationships inappropriate.

- *Limited resources*: Generally, most ad hoc network enabled devices are small mobile devices ranging from notebooks to PDAs and cellular phones. Most of these devices can be assumed to rely on batteries for their power supplies. Complex computational tasks must be avoided, as these operations may drain power quickly. Bandwidth is another important resource. Usually MANETs have a lower bandwidth capacity than a fixed network, and for that reason traffic used for connection and maintenance must be kept as minimal as possible. In addition, MANETs also have a limited CPU processing capability and limited data storage.

- *Multi hop routing*: MANETs connectivity can be single hop based or multi-hop based depending on the distance between source and destination nodes. Communications among nodes in MANETs are generally within a short range. Nodes communicate directly using a single hop routing algorithm if they are close to each other. However, because of the geographical constraint and distance between source and destination nodes, data will usually traverse through the network via one or more intermediate nodes before it reaches the destination. In this situation, connectivity between sender and receiver is no longer in single hop mode but it is now in multi-hop mode.

- *Shared transmission media*: The transmission medium used in MANETs is not as stable as transmission medium used in a fixed network. Communication in MANETs is subjected to noise, interference and even constraint to bandwidth limitation. Moreover, security requirements are usually higher in mobile ad hoc network than in the wired network because wireless links used for communication are subjected to external attacks such as spoofing, eavesdropping and link jamming attacks.

Research Challenges

All of the MANET characteristics discussed in the previous section have introduced many research issues in such networks. Some of them are quite similar to what we have in the other networks, but others are specific to the MANET environment. Figure 3 illustrates the main important issues related to the networking area in mobile ad hoc network [7].

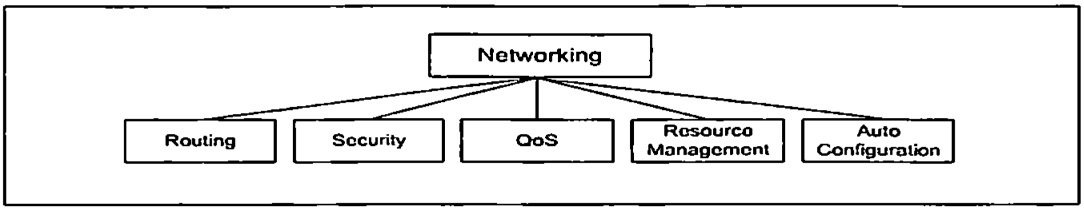


Figure 3: Main research issues in MANETs

- *Routing*: Providing robust and reliable routing mechanisms is the most important issue in ad hoc network, and has thus attracted many researchers to tackle this issue. Proactive and reactive protocols are two approaches in uni-cast routing algorithms considered. Both of these protocols have their own advantages and weaknesses, but the most important thing is that both rely upon the cooperation of all nodes in the networks. Other concerns include multicast routing algorithms and broadcast routing algorithms. Because of the random movement of nodes in MANETs, providing efficient multicast and broadcast routing algorithms becomes more complex and challenging than in a wired network. Various protocols have been proposed to solve all the routing complexities and challenges. However, despite much effort in this area, none of the proposed approaches became a standard protocol for all mobile ad hoc network configurations [9].

- *Auto-Configuration*: MANETs operate in a self-organized manner. Each node in the networks is responsible to configure by itself all the services and applications such as routing protocols, security mechanisms and IP address allocations. However, this configuration is too complex to be done by end users. Providing auto-configuration mechanisms would be very useful and may help in attracting more people to use MANETs.

- *Resource Management*: Resource management is crucial in MANETs. Battery power, bandwidth, CPU processing capability and storage capacity are the most important resources thus proper management towards them is required. All of these resources are limited because of the devices physical constraints. For that reason, communication algorithms as well as services offered in mobile ad hoc network must be optimized to meet the minimum level of bandwidth usage, CPU processing, power utilization and data storage of the ad hoc network enabled devices [8].

- *Quality of Service (QoS)*: QoS is another challenging issue in MANETs [7]. QoS for any network is always related to the characteristics of that particular network. Wireless links used for communication in mobile ad hoc network have a fluctuating link capacity and connectivity, thus making it more difficult to guarantee the QoS in the network. In addition, there are many other unique characteristics in MANETs as described earlier which add the difficulties in providing reliable QoS in MANET.

- *Security*: Providing a robust and reliable security mechanism in MANETs is not an easy task because of the unique characteristics described earlier. Although many security mechanisms (e.g. public key cryptography and firewalls) are found to work well in wired networks, such mechanisms are impractical in MANETs because of the infrastructure constraint [10]. In MANETs, all nodes are expected to operate in a self-organized manner, thus the existence of a central authority to manage the public key infrastructure cannot be assumed. Besides, the nature of instability in the network connections and unpredictable nodes movements add to the difficulty in differentiating between malicious activities and 'natural' network problems.

Providing security for MANETs is the focus of the authors' own research, and in the work to date we have characterized some of the most important attacks that might be launched against such networks. Details of the work can be found in [11].

Conclusions

In this paper, we have introduced mobile ad hoc networks (MANETs), their characteristics, and research challenges. Even though MANETs cannot completely replace infrastructure networks, they could be very useful to enable communications, especially in the case where deploying the infrastructure networks is not practical (e.g. in military operations, or when the infrastructure networks are not available e.g. if destroyed by disaster). Realizing the potentials that MANETs could provide to assist us in our communications, we believe that the research challenges discussed earlier need to be addressed. In our future work, we will address the security issues related to MANETs and later will propose an Intrusion Detection System (IDS) suited appropriately to MANETs characteristics and requirements.

References

[1] Foo Yee Loo, "Ad Hoc Network: Prospects and Challenges," Graduate School Research Paper (Rinkou), Department of Information and Communication Engineering, University of Tokyo, January 2004

[2] Tom Karygiannis and Les Owens, "Wireless Network Security 802.11," Bluetooth and Handheld Devices, NIST Publication (800-48), November 2002

[3] Kevin Chaplin, "Wireless LANs vs. Wireless WANs," Sierra Wireless White Paper, November 2002

[4] http://www.byte.com/art/9405/sec7/art2.htm "PCSes Are Coming," CMP Media LLC, 2003

[5] Jameela Al-Jaroodi, "Security Issues in Wireless Mobile Ad Hoc Networks at the NetworkLayer," Technical Report TR02-10-07, Computer Science and Engineering, University of Nebraska-Lincoln, November 2002

[6] http://www.cs.ucsb.edu/projects/wireless/researchdesc/network/eroyer.shtml, "A Brief Description of the Wireless Project," CISE Wireless Project, 2002

[7] Imrich Chlamtac, Marco Conti, and Jennifer J.-N. Liu, "Mobile ad Hoc Networking: imperatives and challenges," Ad Hoc Networks Journal, Vol. 1, 2003

[8] Khurram Rafique, "A Survey of Mobile Ad Hoc Networks," Columbia University Student Project Report, 2002

[9] Patrick Albers, Olivier Camp, Jean-Marc Percher, Bernard Jouga, Ludovic Mé, and Ricardo Puttini, "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches," In Proceedings of the First International Workshop on Wireless Information Systems (WIS-2002), April 2002

[10] Zygmunt J. Haas, Jing Deng, Ben Liang, Panagiotis Papadimitratos, and S. Sajama, "Wireless Ad Hoc Networks," Draft version of Encyclopedia of Telecommunications, John Wiley, 2002

[11] S. A. Razak, S. M. Furnell, P.J. Brooke, "Attacks against Mobile Ad Hoc Networks Routing Protocols," in Proc. of PGNet Conference, Liverpool, June 28-29, 2004 (*available on request from authors via email*)

Attacks against Mobile Ad Hoc Networks Routing Protocols

S. A. Razak, S. M. Furnell, P. J. Brooke
Network Research Group, University of Plymouth
Plymouth, Devon PL4 8AA
info@network-research-group.org

Abstract-This paper outlines some important issues that relate to security attacks against mobile ad hoc networks from research carried out at Network Research Group, University of Plymouth, on designing intrusion detection system for mobile ad hoc network. In designing security mechanisms for mobile ad hoc networks, one must consider the attacks variations as well as the characteristics of the attacks that could be launched against the ad hoc networks. The discussions of these two aspects are summarized in this paper. This paper also classifies several common attacks against the ad hoc networks routing protocols based upon the techniques that could be used by attackers to exploit routing. messages. Those techniques are modification, interception, fabrication, and interruption.

I. Introduction

Recent advances in computer networking have introduced a new technology for future wireless communication, a mobile ad hoc network (MANET). This technology, which is the combination of peer-to-peer techniques, wireless communications, and mobile computing, provides convenient infrastructure-less communications and could be very useful to provide communications for many applications especially when the infrastructure networks is not feasible. MANET could be used to overcome geographical constraints in a military operation. As it is easy to deploy, it may also very useful to assist in the disaster relief operations where temporary network infrastructure is immediately needed to replace the damaged infrastructure networks.

However, similar to other networks, MANET also vulnerable to many security attacks. MANET not only inherits all the security threats faced in both wired and wireless networks, but it also introduces security attacks unique to itself [1]. As people will be encouraged to use a secured network, it is important to provide MANET with reliable security mechanisms if we want to see this exciting technology become widely used in a next few years. Before the development of any security measure to secure mobile ad hoc networks, it is important to study the variety of attacks that might be related to such networks. With the knowledge of some common attack issues, researchers might have a better understanding of how mobile ad hoc networks could be threatened by the attackers, and thus might lead to the development of more reliable security measures in protecting them.

The purpose of this study is to investigate some of the important issues that might be related to security attacks in mobile ad hoc networks. In Section II, we see how attacks against the ad hoc networks may vary depending upon in which environment the attacks are launched, what communication layer the attacks are targeting, and what level of ad hoc network mechanisms are targeted. After considering these three variations, it is also important to investigate the characteristics of attacks against the ad hoc networks. This issue is discussed in Section III. In this paper, we give a special attention to attacks that could be launched against the routing protocols. We identified that most of the attacks against ad hoc networks routing protocols are actually launched by exploiting the routing messages, and further classify them based upon the techniques that could be used to exploit routing messages in Section IV. Finally, we conclude our study and present our future work in Section V.

II. Attacks Variations

A. Ad hoc networks environments

Ad hoc network can exist in one of three environments; organized, localized, and open environments. Nodes in all of these environments are generally threatened by the same security problems. However, there are some security problems, that are unique to one environment and need more attention in that environment than the others need. Vast numbers of unstructured nodes and the absence of *a priori* relations are some of the main characteristics of the open environment ad hoc networks. Such networks are quite similar to the localized environment networks, but the larger amount of nodes, and the wider coverage area, renders nodes in the open environment to more sophisticated security attacks than the localized networks do. For instance, nodes in both open and localized environments suffer from the absence of a central authority. However, this is not a big issue in a localized environment, because nodes in that environment might have a physical contact with each other to employ any security measures. Security could also be easily enforced in the organized environment because nodes in that environment are usually pre-employed with appropriate security measures before they participate in any specific tasks such as in a military operation.

B. Communication layers

Each layer in the ad hoc networks communication protocols has its own vulnerabilities. In a physical layer, mobile nodes as well as the communication links are vulnerable to both passive and active attacks. Passive eavesdropping, -signal jamming, denial of service (DoS) attacks, and physical hardware tampering are among the most popular attacks in this layer [2]. Such attacks could be made less useful by encrypting the communication signal, employing spread-spectrum communication technology, and using a tamper-resistant hardware.

Similar link jamming and DoS attacks are also threatening the ad hoc networks at the data link layer. At this layer, adversaries might jam the communication links by sending huge data to the networks, or by replaying unnecessary packets to exhaust the networks' resources. Expensive cryptography algorithms and more sophisticated security measures could be very useful at this layer to protect the networks and to distinguish between valid and invalid packets traversed in the networks.

Attackers are also threatening the ad hoc networks at both the transport and the application layers. At the transport layer, messages are exchanged on the end-to-end basis using secured routes established in the network layer. For that reason, ensuring security at the network layer is very important to provide reliable communication at the transport layer. Similar to the other types of networks, attackers can always find a loophole in the ad hoc networks' applications and might use this vulnerability to launch attacks at the application layer. However, since similar attacks also occur in the other types of networks, regular solutions used in wired networks could be reused to defend the ad hoc networks against attacks at the application layer.

Besides providing reliable routes to exchange messages in the transport layer, network layer also provides the most critical service in the ad hoc network, which is the routing protocol. Several routing protocols have been introduced to provide reliable communication among nodes, but less attention to the security aspects when designing such protocols has opened many security holes at this layer [3].

C. Attack level

There are two main levels of attack in the ad hoc network; attacks against the basic mechanisms and attacks against the security mechanisms [4]. Ad hoc networks have their own unique basic mechanisms, such as the use of wireless links for communications, employing their own routing strategies, and operate in a distributed manner. All these basic mechanisms are actually reflecting to their own unique characteristics that differentiate them from other types of networks. Attackers might launch many security attacks against these basic mechanisms. For instance, attackers could launch passive eavesdropping attacks against the wireless links, drain off node's limited resources, and launch active attacks to interrupt the routing mechanisms.

Responding to many security attacks against the ad hoc network basic mechanisms, researchers have introduced a number of security measures to protect the networks. However, all these security measures are also vulnerable to attacks and need to be secure. Examples of attacks against security mechanisms are stealing username and password to get unauthorized access in the networks and modifying public key databases to disrupt authentication, confidentiality, and integrity services.

III. Attack Characteristics

Dynamic topology, distributed operation, and resource constraints are some of the unique characteristics that exist in the ad hoc networks, which inevitably increase the vulnerability of such network. Many characteristics might be used to classify attacks in the ad hoc networks. Examples would include looking at the behaviour of the attacks (passive vs. active), the source of the attacks (external vs. internal), the processing capability of the attackers (mobile vs. wired), and the number of the attackers (single vs. multiple).

A. Passive vs. active attacks

Passive attacks are launched to steal valuable information in the targeted networks. Examples of passive attacks in ad hoc network are eavesdropping attacks and traffic analysis attacks. Detecting this kind of attack is difficult because neither the system resources nor the critical network functions are physically affected to prove the intrusions [5].

While passive attacks do not intend to disrupt the network operations, active attacks on the other hand actively alter the data with the intention to obstruct the operation of the targeted networks. Examples of active attacks comprise actions such as message modifications, message replays, message fabrications and the denial of service attacks.

B. External vs. internal attacks

External attacks are attacks launched by adversaries who are not initially authorized to participate in the network operations. These attacks usually aim to cause network congestion, denying access to specific network function or to disrupt the whole network operations. Bogus packets injection, denial of service, and impersonation are some of the attacks that are usually initiated by the external attackers.

More severe attacks in the ad hoc networks might come from the second source of attacks, which is the internal attack. Internal attacks are initiated by the authorized nodes in the networks, and might come from both compromised and misbehaving nodes. Internal nodes are identified as compromised nodes if the external attackers hijacked the authorized internal nodes and are then using them to launch attacks against the ad hoc networks. Security requirements such as authentication, confidentiality and integrity are severely vulnerable in the ad hoc networks with the compromised internal nodes because communication keys used by these nodes might be stolen and passed to the other colluding attackers. On the other hand, nodes will be classified as misbehaving if they are authorized to access the system resources, but fail to use these resources in a way they should be [6]. Internal nodes might misbehave to save their limited resources, such as the battery powers, the processing capabilities, and the communication bandwidth. Attacks that are caused by the misbehaving internal nodes are difficult to detect because to distinguish between normal network failures and misbehaviour activities in the ad hoc networks is not an easy task.

C. *Mobile vs. wired attackers*

Mobile attackers are attackers that have the same capabilities as the other nodes in the ad hoc networks. Since they have the same resources limitations, their capabilities to harm the networks operations are also limited. For instance, with the limited transmitting capabilities and battery powers, mobile attackers could only jam the wireless links within its vicinity. They are not capable to launch the network jamming attacks to disrupt the whole networks operations.

On the other hand, wired attackers are attackers that are capable of gaining access to the external resources such as the electricity. Since they have more resources, they could launch more severe attacks in the networks, such as jamming the whole networks or breaking expensive cryptography algorithms. Existence of the wired attackers in the ad hoc networks (especially in the open environment networks) is always possible as long as the wired attackers are able to locate themselves in the communication range and have access to the wired infrastructures.

D. *Single vs. multiple attackers*

Attackers might choose to launch attacks against the ad hoc networks independently or by colluding with the other attackers. One man action or single attackers usually generate a moderate traffic load as long as they are not capable to reach any wired facilities. Since they also have similar abilities to the other nodes in the networks, their limited resources become the weak points to them [7]. For instance, complex cryptography algorithms could be used to help in defending the authentication, integrity, and the confidentiality services from a single attacker. As it becomes very expensive for the single attackers to break the encrypted messages, nodes in the networks could share the expensive cryptography workloads with each other by exploiting the distributed operations and the multiple connections they had among them.

However, if several attackers are colluding to launch attacks, defending the ad hoc networks against them will be much harder. Colluding attackers could easily shut down any single node in the network and be capable to degrading the effectiveness of network's distributed operations including the security mechanisms. Adding to the severity, colluding attackers could be widely distributed or reside at the certain area where they presumed high communication rate in the networks exist. If no suitable security measures employed, nodes in that targeted area are susceptible to any kind of denial of service (DoS) attacks that could be launched by the colluding attackers.

IV. ATTACKS AGAINST ROUTING MESSAGES

Routing is one of the most vital mechanisms in the ad hoc networks. Improper and insecure routing mechanisms will not only degrade the performance of the ad hoc networks, but will also render such networks vulnerable to many security attacks. One of the basic elements in the routing mechanism is the routing message, which is used to establish and maintain relationships between nodes in the networks. The importance of the routing message has made it a main target by the attackers to launch attacks against the ad hoc networks [3, 8]. Attacks against the routing messages could be launched in many forms and may include all the characteristics described in Section III. In this work, attacks against routing messages are classified based on the classification suggested by Stallings in [9]. In such classification, information or messages could be deviated from the normal operation flow using modification, interception, interruption or fabrication attacks. In a more severe case, attackers also might use any combination of these attacks to disrupt the normal information flow. As far as our concern, this study is the first to address security attacks against the ad hoc networks routing messages.

A. *Modification*

In a message modification attack, adversaries make some changes to the routing messages, and thus endanger the integrity of the packets in the networks. Since nodes in the ad hoc networks are free to move and self-organize, relationships among nodes at some times might include the malicious nodes. These malicious nodes might exploit the sporadic relationships in the network to participate in the packet forwarding process, and later launch the message modification attacks. Examples of attacks that can be classified under the message modification attacks are packet misrouting and impersonation attacks.

1) *Packet misrouting attacks*: In a packet misrouting attack, malicious nodes reroute traffic from their original path to make them reach the wrong destinations [10]. Attackers might misroute a packet to make it stay in the network longer than its lifetimes, thus render it to be dropped from the network. As a result, the source node needs to retransmit the lost packets and this will consume more bandwidth, as well as increasing the overhead in the networks.

2) *Impersonation attacks*: The impersonation attacks, also called the spoofing attacks, are attacks where malicious node assumes the identity of another node in the networks [11]. By impersonating another node, attackers are able to receive routing messages that are directed to the nodes they faked.

Impersonation attacks are possible in the ad hoc networks because most of the current ad hoc routing protocols do not authenticate the routing packets. As a result, malicious nodes might exploit this loophole to masquerade as another node by modifying the contents of the packets.

B. Interception

Attackers might launch the interception attacks to get an unauthorized access to the routing messages that are not intentionally sent to them. This kind of attack jeopardizes the integrity of the packets because such packets might be modified before being forwarded to the next hop. Besides, the intercepted packets might also be analysed before passed to the destination thus violating the confidentiality. Examples of attacks that can be classified under the interception attacks are wormhole attacks, black hole attacks, and routing packet analysis attacks.

1) *Wormhole attacks*: In the wormhole attacks, a compromised node in the ad hoc networks colludes with external attacker to create a shortcut in the networks. By creating this shortcut, they could trick the source node to win in the route discovery process and later launch the interception attacks. Packets from these two colluding attackers are usually transmitted using wired connection to create the fastest route from source to the destination node. In addition, if the wormhole nodes consistently maintain the bogus routes, they could permanently deny other routes from being established. As a result, the intermediate nodes reside along that denied routes are unable to participate in the network operations.

2) *Black hole attacks*: In this attack, malicious nodes trick all their neighbouring nodes to attract all the routing packets to them. As in the wormhole attacks, malicious nodes could launch the black hole attacks by advertising themselves to the neighbouring nodes as having the most optimal route to the requested destinations. However, unlike in the wormhole attacks where multiple attackers colluded to attack one neighbouring node, in the black hole attacks, only one attacker is involved and it threatens all its neighbouring nodes.

3) *Routing packet analysis attacks*: Since no disruptive action occurs, routing packet analysis could be classified as one of the passive attacks against the ad hoc networks. One way to launch this attack is by exploiting the *promiscuous* mode employed in the ad hoc network. In a *promiscuous* mode, if node A is the neighbour of both nodes B and C at a particular time, node A can always hear the transmissions between node B and node C. By exploiting this nature, node A is able to analyze the overheard packets transmitted between node B and node C. More explanation regarding the *promiscuous* mode in the ad hoc networks can be found in [12]. Besides, malicious nodes could also launch this attack by exploiting the nature in a multi hop routing. In multi hop routing, packets need to be forwarded through several intermediate nodes before reaching the actual destination. Malicious nodes might exploit this opportunity by locating themselves in any location along the route to participate in the message forwarding process and later launch the routing packet analysis attacks.

C. Fabrication

Instead of modifying or interrupting the existing routing packets in the networks, malicious nodes also could fabricate their own packets to cause chaos in the network operations. They could launch the message fabrication attacks by injecting huge packets into the networks such as in the sleep deprivation attacks. However, message fabrication attacks are not only launch by the malicious nodes. Such attacks also might come from the internal misbehaving nodes such as in the route salvaging attacks.

1) *Sleep deprivation attacks*: This kind of attack is actually more specific to the mobile ad hoc networks. The aim is to drain off limited resources in the mobile ad hoc nodes (e.g. the battery powers), by constantly makes them busy processing unnecessary packets. In a routing protocol, sleep deprivation attacks might be launched by flooding the targeted node with unnecessary routing packets. For instance, attackers could flood any node in the networks by sending a huge number of route request (RREQ), route replies (RREP) or route error (RERR) packets to the targeted node. As a result, that particular node is unable to participate in the routing mechanisms and rendered unreachable by the other nodes in the networks.

2) *Route salvaging attacks*: Route salvaging attacks are launched by the greedy internal nodes in the networks. In a mobile ad hoc network, there is no guarantee that each transmitted packet will successfully reach the desired destination node [13]. Packets might not reach the destination node because of the natural network failures or might be under attacks by the adversaries. Therefore, to salvage their packets from such failures, misbehaving internal nodes might duplicate and retransmit their packets although no sending error messages received. The effects of the route salvaging attacks might be more severe if there are many greedy nodes in the networks. Besides draining off more resources in intermediate and destination nodes, this attack might also cause the consumption of unnecessary bandwidth.

D. Interruption

Interruption attacks are launched to deny routing messages from reaching the destination nodes. Adversaries could do this by either attacking the routing messages or attacking the mobile nodes in the networks. Actually, most of the attacks launched in the modification, interception, and fabrication attacks are aimed to interrupt the normal operations of the ad hoc networks. For instance, adversaries aiming to interrupt the availability service in the networks might destroy all paths to a particular victim node by using the message modification attacks. In a message fabrication attack, adversaries could overload the networks by injecting huge unnecessary packets. Examples of attacks that could be classified under the interruption attacks category are packet dropping attacks, flooding attacks, and lack of cooperation attacks.

1) *Packet dropping attacks*: Direct interruption to the routing messages could be done by using the packet dropping attacks. In a standard packet dropping attack, an adversary collaborates as usual in the route discovery process and launches the constant packet dropping attacks if it is included as one of the intermediate nodes. In addition, instead of constantly dropping all the packets, adversaries might vary their techniques using random, selective, or periodic packet dropping attacks to help their interrupting behaviour remain concealed [14].

2) *Flooding attacks*: Adversaries also might interrupt the normal operations in the packet forwarding process by flooding the targeted destination nodes with huge unnecessary packets. Nodes under the flooding attacks are unable to receive or forward any packet thus all the packets directed to them will be discarded from network.

3) *Lack of cooperation attacks*: Lack of cooperation from the internal nodes to participate in the network operations can also be seen as an attempt to launch a refusal of service attack. In such attacks, internal nodes are discouraged to cooperate in the network operations that did not benefit them because participating in such operations will drain off their resources. Misbehaving internal nodes might use different strategies to save their limited resources. They might refuse to forward the other node's packets, not send back the route error report to the sender when failing to forward packets, or might turn off their devices when not sending any packet in the networks.

V. CONCLUSIONS

In this paper, one can see that attacks against the ad hoc networks may vary depend on (1) which environment the attacks are launched, (2) what communication layer the attacks are targeting, and (3) what level of ad hoc network mechanisms are targeted. One can also see that there are several attack characteristics that must be considered in designing any security measure for the ad hoc network. By investigating the characteristics and variations of the attacks, one can make a long list of attacks that could be launch against the ad hoc networks. However, since this study is focusing on the vulnerabilities of the ad hoc networks routing protocols, only some of the common attacks that could be launched against the ad hoc network routing protocols have been investigated. From the investigation, we identified that most of the common attacks against the ad hoc networks routing protocols are actually launched by exploiting the routing messages. From there, we further classify attacks against the routing protocols based upon the techniques that could be used by the attacker to exploit routing messages. In a future work, several security solutions that have been proposed to secure routing protocols will be investigated and classified based on this classification. The investigation will include various techniques that might be employed in protecting, detecting, and responding to the attacks against the routing messages.

VI. REFERENCES

[1] T. Karygiannis and L. Owens, "Wireless Network Security, 802.11, Bluetooth and Handheld Devices," *NIST Publication*, p. 800(48), November 2002.

[2] J. Al-Jaroodi, "Security Issues in Wireless Mobile Ad Hoc Networks at the Network Layer," University of Nebraska-Lincoln, Dept. of Computer Science and Engineering, Technical Report TR02-10-07, November 2002.

[3] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks," in Proc. of 2002 IEEE International Conference on Network Protocols (ICNP), pp. 778-89, Nov. 12-15, 2002.

[4] J. P. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," in *Proc. of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHOC 2001*, pp. 146-155, Oct. 4-5, 2001.

[5] S. Bouam and J. B. Othman, "Data Security in Ad hoc Networks using MultiPath Routing," in *Proc. of the 14th IEEE PIMRC*, pp. 1331-1335, Sept. 7-10, 2003.

[6] S. Ghazizadeh, O. Ilghami, E. Sirin, and F. Yaman, "Security-Aware Adaptive Dynamic Source Routing Protocol," In *Proc. of 27th Conference on Local Computer Networks*, pp. 751-760, Nov. 6-8, 2002.

[7] G. Schäfer, "Research Challenge in Security for Next Generation Mobile Networks," *Position Papers PAMPAS '02 - Workshop on Requirements for Mobile Privacy & Security*, Sept. 16-17, 2002.

[8] H. Li, Z. Chen and X. Qin, "Secure Routing in Wired Networks and Wireless Ad Hoc Networks," Univ. of Kentucky, Department of Computer Science, Term-paper, 2003.

[9] W. Stallings, *Cryptography and network security, Principles and practice, 2nd ed.*, Prentice Hall, Inc, 1999, pp. 6-9.

[10] S. Rajavaram, H. Shah, V. Shanbhag, J. Undercoffer, and A. Joshi, "Neighborhood Watch: An Intrusion Detection and Response Protocol for Mobile Ad Hoc Networks," Student Research Conference, University of Maryland at Baltimore County (UMBC), May 3, 2002.

[11] A. Burg, "Ad hoc networks specific attacks," Technische Universität München, Institut für Informatik, Seminar Paper, Seminar Ad Hoc Networking: concept, applications, and security, Nov., 2003.

[12] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of the 6th annual international conference on Mobile computing and networking*, pp. 255-265, Aug. 6-11, 2000.

[13] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc.*

of the 5th annual ACM/IEEE international conference on Mobile computing and networking, pp. 151-162, Aug. 15-19, 1999.

[14] Y. A. Huang, W. Fan, W. Lee, and P. S. Yu, "Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies," in *Proc. of The 23rd International Conference on Distributed Computing Systems (ICDCS)*, pp. 478-489, May 19-22, 2003.

Intrusion Detection in Mobile Ad Hoc Networks

Shukor Abd. Razak, Steven Furnell, Phil Brooke
Network Research Group, School of Computing, Communications & Electronics, University of Plymouth, UK

For more information, contact:
Shukor Abd Razak
E-mail: srazak@jack.see.plymouth.ac.uk

Motivations & Objectives

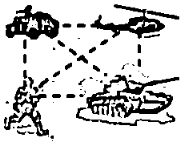
Motivations

- Recent advances in communication technologies introduce mobile ad hoc networks (MANET)
- MANET = autonomous nodes, random network topology, connected by wireless links, self-organized, stand alone or attach to wired network
- Introduces new security threats and inherits all security attacks against wired networks
- Addressing MANET security issues will encourage people to use this technology

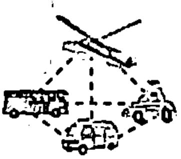
Objectives

- Prevent security attacks
  - Authentication (key management, trust relationships)
  - Secure Routing (incentives, penalties, reputations)
- Detect and respond to security attacks
  - Neighbourhood watch
  - Intrusion Detection System **(our focus)**

Applications



Military operations
- Geographical constraints in battlefield
- Physical connection will disclose location



Rescue operations
- Replace damaged network infrastructure
- Easy to deploy

Conference room
- P2P connections without cable
- Scalable without fixed connections

Issues

Ad Hoc Network Characteristics

- Dynamic network topology
- Distributed operations
- Infrastructure-less
- Limited resources
- Multi-hop routing

Research Challenges

- Routing mechanisms
- Quality of service (QoS)
- Security
- Resource management
- Auto-configuration

Security Issues [1]

Attack Variations

- Attacks against different types of ad hoc networks environments (organized, localized, open environments)
- Targeting different communication layers (physical, data link, network, transport, application layers)
- Launch to disrupt two different important mechanisms in ad hoc networks (basic and security mechanisms)

Attack Characteristics

- Passive vs. Active

Passive	- Difficult to detect
Active	- Intend to interrupt, detectable

- Internal vs. External

Internal	- Compromised, misbehave
External	- Not authorized user

- Single vs. Multiple

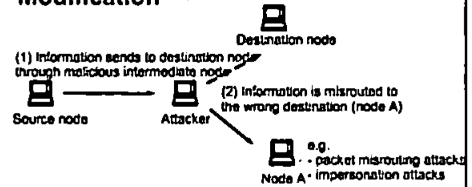
Single	- Moderate attacks
Multiple	- Colluding attackers, severe

- Mobile vs. Wired

Mobile	- Same capability & limitations
Wired	- Gain external resources

Security Attacks [1]

Modification

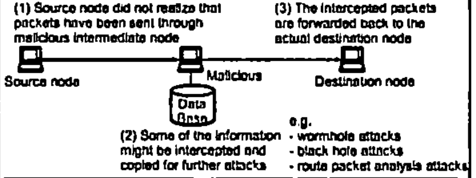


(1) Information sends to destination node through malicious intermediate node

(2) Information is misrouted to the wrong destination (node A)

e.g.
- packet misrouting attacks
- impersonation attacks

Interception



(1) Source node did not realize that packets have been sent through malicious intermediate node

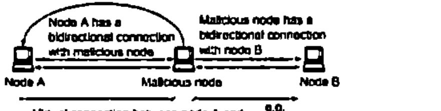
(2) Some of the information might be intercepted and copied for further attacks

(3) The intercepted packets are forwarded back to the actual destination node

e.g.
- wormhole attacks
- black hole attacks
- route packet analysis attacks

Fabrication

Malicious node fabricates fake packets telling node A that it does not have any connection to node B



Node A has a bidirectional connection with malicious node

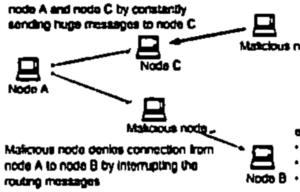
Malicious node has a bidirectional connection with node B

Virtual connection between node A and node B has been denied by the attacker

e.g.
- sleep deprivation attacks
- route salvaging attacks

Interruption

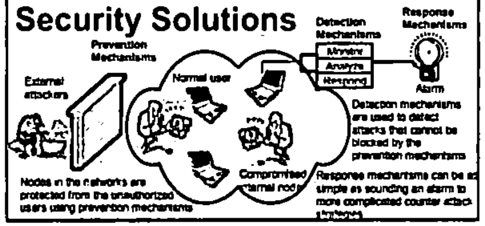
Malicious node denies connection between node A and node C by constantly sending huge messages to node C



Malicious node denies connection from node A to node B by interrupting the routing messages

e.g.
- packet dropping attacks
- flooding attacks
- lack of cooperation

Security Solutions



Nodes in the networks are protected from the unauthorized users using prevention mechanisms

Detection mechanisms are used to detect attacks that cannot be blocked by the prevention mechanisms

Response mechanisms can be as simple as sounding an alarm to more complicated counter attack strategies

Detection Mechanisms

Characteristics of Intrusion Detection System (IDS) [2]

- Audit source location
  - Host-based
  - Network-based
- Detection method
  - Anomaly detection/behaviour-based
  - Misuse detection/knowledge-based
- Usage frequency
  - Real-time detection
  - Periodic detection
- Behaviour on detection/response
  - Passive (alarm)
  - Active (corrective mechanisms, counter attacks)

Conclusions & Future Work

- MANET faces security attacks from both internal and external attackers
- IDS can be a second defensive wall when prevention mechanisms failed to block attacks
- IDS still new in MANET and more research works need to be carried out
- For future work, investigate new methods for effective intrusion detection and response mechanisms in MANET

References

[1] S. A. Razak, S. M. Furnell, P. J. Brooke, "Attacks against Mobile Ad Hoc Networks Routing Protocols," *In Proc. of PGNet Conference*, Liverpool, June 28-29, 2004

[2] H. Debar, M.Dacier, A. Wespi, "Towards Taxonomy of Intrusion Detection Systems," IBM Zurich Research Laboratory, Ruschlikon, Switzerland, 1998

A Two-tier Intrusion Detection System for Mobile Ad Hoc Networks

S. A. Razak, S. M. Furnell, P. J. Brooke
Network Research Group, University of Plymouth, Plymouth, Devon, UK
srazak@jack.see.plymouth.ac.uk
steve@jack.see.plymouth.ac.uk
P.J.Brooke@plymouth.ac.uk

Abstract: Providing a robust and reliable Intrusion Detection System (IDS) for Mobile Ad Hoc Network (MANET) is not as straightforward as in the wired networks because of the characteristics, threats and vulnerabilities, and security requirements related to such network. This paper discusses these issues along with a discussion of the existing research works that have been proposed to secure MANET. After considering these issues, a novel IDS framework (a two-tier IDS for MANET), is proposed to improve the performance of existing IDS in MANET environment.

Keywords: MANET, IDS, Two-tier detection, Trusted friends

1. Introduction

MANET represents a combination of peer-to-peer techniques, wireless communications and mobile computing, and has become an important field of research in recent years. This new technology has been widely used to support communications in an environment where it may be impossible to deploy infrastructure networks, such as in military battlefields and disaster recovery sites. In addition, this technology might be used to replace infrastructure networks where employing the wireless networks is more practical (Foo 2004). Although MANET utilizes a wireless medium for communications as in other wireless networks (e.g. Wireless Wide Area Network (WWAN), Wireless Local Area Network (WLAN), and Wireless Personal Area Network (WPAN)), it has its own unique characteristics that make it differ from the others. With a MANET, nodes communicate to each other in a dynamic network topology, operate in a distributed manner, share the unstable wireless transmission medium, and own limited resources to support their operations. As such, providing robust and reliable communication in MANET can be very challenging. The challenges include routing mechanisms, auto-configuration, resource management, Quality of Service (QoS), and security issues.

Among these challenges, routing issues have been given special attention by previous researchers because of its role as the most basic function in MANET. However, as proven by history, security issues for any new technology need to be addressed at the early stages of its development. For that reason, security issues need to be considered as important as the routing issues, and need to be addressed at the early stages of the MANET emergence. Securing MANET is not as straightforward as securing a wired network. Again, this is because of the unique characteristics possessed by such network. The problem will become more complicated when we try to implement security measures in a different MANET environment. For instance, open MANET will require more complicated security measures to defend it against both internal and external attackers compared to localized and organized MANET, where most of the threats usually came from external attackers. Different environments require different security measures and the requirements depend on several factors such as the type of the user participating in the network, the density of nodes, and the radius of the coverage area. Although MANET requires the same basic security requirements as other networks (i.e. availability, confidentiality, integrity, authentication, non-repudiation, and authorization), they need to be addressed in a specific way that suits the MANET environment.

Several works have been proposed to ensure these security requirements are fulfilled in MANET, and such works can be classified into one of the three steps in a security lifecycle (i.e. prevention, detection, and response) (Mark 2002). These three steps are important and dependent to each other to provide reliable protection against threats in MANET. In this paper, a novel Intrusion Detection System (IDS) that suits the MANET environment is presented to complement existing security measures that have been proposed to secure MANET. This new IDS framework has been designed after considering ideas and suggestions from the previous works as discussed in Section 2. A detailed description of the new IDS framework can be found in Section 3.

2. MANET security: State of the art

In general, security measures that have been proposed to secure MANET can be categorized into two groups: prevention and detection/response mechanisms. The following are examples of previous work that has been proposed to secure MANET.

2.1 Prevention mechanism

Prevention is one of the important phases in the security life cycle and is usually used as a first defensive wall from malicious external attackers (Mark 2002). However, despite being widely used in a wired network, the implementation of such a mechanism in MANET environment faces many problems. For instance, the non-existence of central administrators in MANET to manage the authentication service makes the authentication mechanism one of the challenging issues. Responding to this problem, researchers have devised several mechanisms that suit the MANET environment, and such mechanisms can be broken down into these three categories: authentication schemes, secure routing protocols, and cooperation enforcement mechanisms.

2.1.1 Authentication

Since we cannot assume the existence of a Central Authority (CA) in MANET, most of the researchers suggest that the authentication mechanism in such networks should be carried out in a distributed fashion. (Srdjan 2003) proposed a self-organized authentication scheme for MANET where nodes can independently establish security associations among them in an offline mode. Each node creates its own private and public keys and exchanges the keys with the other adjacent nodes through the secure short-range connectivity channels such as infrared.

Imprinting is another way to establish secure transient associations among MANET nodes in the absence of an online authentication server. In such approach, each node (slave) will be imprinted with a 'soul' that binds it to the other node (master) over a non-wireless channel. Once imprinted with a master soul, slave nodes will only follow instructions that come from the master node throughout its participations in the network operations (Frank 1999). There are also efforts to make the CA virtually exist in MANET environments. For instance, (Lidong 1999) proposed a security mechanism that enable several nodes in the network to hold a partial share of the system private key, which is needed by the new users who wish to join the network. Another approach that might be related to the emulation of CA's role is a cluster-based mechanism (Lakshmi 2000). In such mechanisms, each cluster head will hold a system private key and be responsible to manage, distribute, and authenticate every node in its cluster.

2.1.2 Secure routing

MANET operations require participation from all nodes in the networks. Since MANET nodes are autonomous, there is always a possibility for the authorized nodes (nodes that have been authenticated to use network resources) to misbehave during their participation in network operations. Reasons for nodes to misbehave could be that they want to save some of their limited resources or simply because they are malicious nodes. Most of the researchers agree that this threat can be made less harmful by adding some security features when forwarding packets across the networks. Hash functions (Yih-Chun 2002), end-to-end authentications (Kimaya 2002), and digital signature (Manel 2002) are some of the available security measures that can be used to secure every packet from misbehaving intermediate nodes.

2.1.3 Cooperation enforcement

MANET operations are much dependent on each node's willingness to cooperate in the network. However, some nodes in the network might refuse to cooperate. Two mechanisms have been proposed to overcome this problem. (Sheng 2003) proposed a credit-based system to stimulate nodes' cooperation. Nodes will lose their credit when sending own packets but will gain some credits if forwarding other nodes' packets. After utilizing all their credits, nodes will not be able to send packets to the network. As a result, they will be self-motivated to cooperate in forwarding others' packets. Another way to enforce node's cooperation is by employing a reputation mechanism (Pietro 2002). Each node has its own reputation rate that can be used by other nodes as an indicator of its behaviour. Every node in the network will try as hard as they can to avoid any communication with a node that has a lower reputation rate.

2.2 Detection and response mechanisms

Prevention mechanisms alone are not enough to protect MANET from attacks that might come from external and internal attackers. Detection and response mechanisms (e.g. IDS) could be very useful as a second defensive wall once the prevention mechanism fails to protect the network. Efforts from researchers to provide reliable IDS for MANET can be seen by looking at the methods or strategies they proposed in addressing one of these issues: how to collect the audit data, what is the appropriate method to detect intrusion, how to minimize false alarm, and how to respond to intrusion.

2.2.1 Audit data source

As in a wired network, audit data in MANET can be gathered from two sources: host-based and network-based audit data sources. Since host-based audit data source is not dependent on any network architecture, similar data collection techniques as applied in wired networks can be used in MANET. For instance, we can use a Simple Network Monitoring Protocol (SNMP) to log user activities or by using agent technology to collect available audit data. However, this does not apply to the network-based audit data source. No such concentration point or dedicated node exists in MANET that can be used to collect the whole network information like in the wired networks. However, this does not mean that the network information cannot be collected in MANET environment. One of the most common assumptions made by researchers is that, each node in MANET is capable of hearing the transmission in and out from other nodes in the networks as long as they are within each other's radio range. Researchers claimed that by using this assumption, partial or localized network activities can be collected by each node, which later can be shared among them as a virtual network-based audit data source (Sergio 2000).

2.2.2 Method of detection

As in wired networks, one can use either misuse or anomaly detection techniques to detect intrusion in MANET. Both techniques have their own advantages and disadvantages. However, between these two mechanisms, researchers claimed that the anomaly detection would perform better than the misuse detection in MANET. This is because MANET technology is still new, and the process of compiling the attack signatures for misuse detection technique for such networks is harder than in the matured technology (e.g. wired networks). There is also an issue of updating the attack signatures database. Unlike in an anomaly detection where patterns of normal activities can be trained autonomously, attack signatures used in a misuse detection mechanism need to be managed and updated by a system administrator. However, this does not mean that the misuse detection mechanism is completely inappropriate in MANET. As mentioned earlier, both misuse and anomaly detection mechanisms have their own advantages and disadvantages, and perhaps the combination of these two mechanisms will improve the performance of intrusion detection mechanism in MANET. Examples of anomaly IDS in MANET are (Sowjnya 2002), (Yi-An 2003), and (Yongguang 2003).

2.2.3 False alarm acceptance level

False alarms are very common in intrusion detection systems that employ an anomaly detection technique. They happen when the system misjudges normal activity as being abnormal. It is a very big problem in intrusion detection system because if too many false alarms are triggered in the system, users will start ignoring the alarms, and thus possibly overlook real intrusion attempts. This problem becomes more difficult in MANET because to classify what is normal and what is abnormal activity in such networks is not an easy task. Sometimes, nodes fail to forward packets in MANET because of natural network failure, not because of any malicious activity occurred in the networks. One solution to this problem has been proposed in (Hao 2002) where the authors employ a threshold mechanism to reduce the number of false alarms in their IDS. Each node will not be penalized for a single malicious behaviour. Every malicious activity for each user will be recorded and once the activities reach the threshold value, confirmation can be made that the particular node is malicious and will be discarded from the network. In another approach, (Sergio 2000) proposed a rating mechanism to ease the problem of false accusations from the malicious nodes. Since nodes are able to share the intrusion alerts to speed up the response mechanism, malicious node might take this opportunity to blackmail other nodes in the networks by advertising false accusations. With all nodes having positive or negative reputations according to their behaviours, each node can have an idea whether to accept or discard any accusation received by other nodes in the networks.

2.2.4 Response mechanisms

In MANET, since there is no network administrator to manage the network, each node is responsible for response to any intrusive behaviour. This makes the response issue more challenging than in wired networks. One of the challenges is how to alert other nodes about the detected intrusive activities in the presence of blackmail attackers. One of the solutions to this problem is to trigger a voting mechanism once the intrusion alert is broadcasted to the neighbouring nodes (Sonali 2001).

3. A two-tier MANET IDS framework

3.1 Conceptual framework

A two-tier hybrid IDS for MANET is a novel IDS architecture proposed to improve the efficiency of existing MANET IDS architectures. The main idea of the proposed system is to provide a reliable IDS that can detect any intrusion attempts and at the same time can reduce the number of false alarms raised in the system. With the focus to improve the detection strategies, only a simple response mechanism is deployed in the proposed system as a complement to the detection mechanisms. The conceptual framework of the proposed IDS architecture is as illustrated in Figure 1.

3.2 System Components

The proposed IDS framework has six main modules, covering from the audit data source to the response mechanism. Details of each module and their roles are described as follows.

3.2.1 Real time audit data source (self-experience and friends observation)

This is where the audit data will be gathered for further investigation. In the proposed architecture, two audit data sources have been identified as appropriate to detect intrusive activities in the networks. Any network operation, which has been initiated or having a direct connection with the user itself are classified as *self-experience audit data*. For instance, in a packet forwarding process, the source, destination, and all the intermediate nodes will have a direct experience of such process and are capable of logging the related activities of the process for further investigation if something suspicious is detected. Data are not restricted to be gathered by directly participating nodes (source, destination, and all the intermediate nodes). Neighbours that are close to the participating nodes are also able to capture the overheard network activities using a *promiscuous* mode. This kind of audit data is known as *friends' observation* *audit data* in the proposed framework and can make the detection process faster compared to the intrusion detection system that only relies upon a single audit data source.

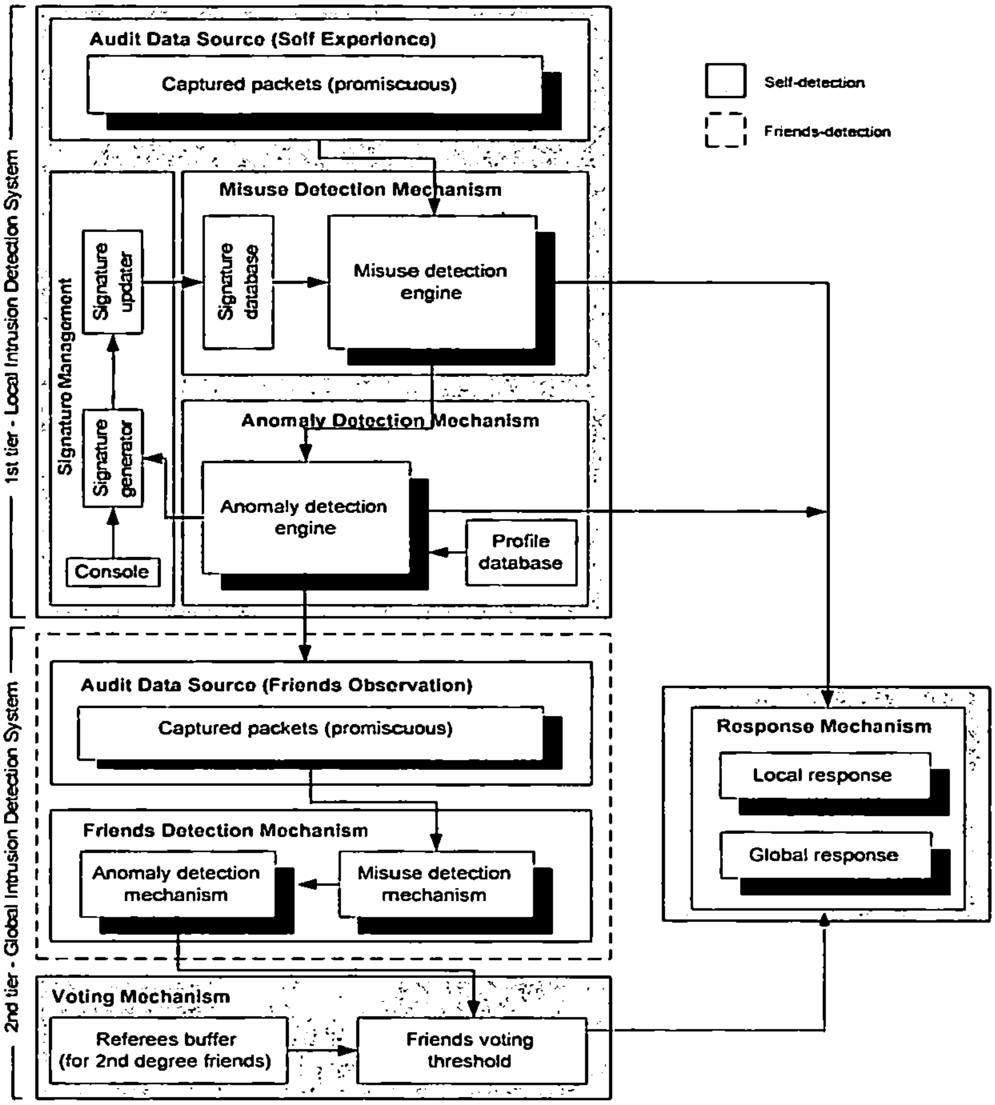

1st tier - Local Intrusion Detection System
Audit Data Source (Self Experience)
Captured packets (promiscuous)
Self-detection
Friends-detection
Misuse Detection Mechanism
Signature Management
Signature updater
Signature database
Misuse detection engine
Signature generator
Anomaly Detection Mechanism
Anomaly detection engine
Profile database
Console
2nd tier - Global Intrusion Detection System
Audit Data Source (Friends Observation)
Captured packets (promiscuous)
Friends Detection Mechanism
Anomaly detection mechanism
Misuse detection mechanism
Response Mechanism
Local response
Global response
Voting Mechanism
Referees buffer (for 2nd degree friends)
Friends voting threshold

Figure 1: Conceptual framework of the two-tier hybrid IDS for MANET

3.2.2 Misuse detection mechanism

The misuse detection mechanism is the primary detection method used in the proposed IDS architecture. At the initial stage, the attack database might only cover a few attack signatures, but as time goes by, with the aid of the anomaly detection mechanism and the signature management module, the attack signature database will reach its maturity level and thus be able to detect more attacks. This module comprises a misuse detection engine to detect activities that match the attack signatures as stored in the signature database.

3.2.3 Anomaly detection mechanism

Attacks that cannot be detected by a misuse detection mechanism will be passed here for further investigation. The failure of detecting the attacks could be because of the attack signature database is still immature or could be because of insufficient evidence. The anomaly detection mechanism applied here is similar to the existing techniques proposed by previous researchers and its main components include an anomaly detection engine and a profile database. Similar to the misuse detection engine, the anomaly detection engine in this module will try to detect malicious activities in the network. However, this time, comparison will be made with the normal user/system behaviour as stored in a profile database.

3.2.4 Signature management

This module will enable a dynamic update to the misuse detection mechanism. The signature generator will automatically generate the attack signature each time the anomaly detection mechanism successfully identifies deviation from normal user/system profiles. The auto-generated attack signatures will then be checked and verified through the management console before being periodically appended to the existing signature database located in a misuse detection mechanism module.

3.2.5 Friends detection mechanism

Any suspicious activity that has not been detected as intrusive by a local detection mechanism will be sent to the global detection mechanism for further investigation. This global detection mechanism requires cooperation from all nodes in the networks to detect intrusions. However, since MANET operates without the aid of a network administrator or third party authentication server, not a single node in the network can be trusted except the node itself. A friend detection mechanism has been proposed to overcome this node trustworthiness problem in MANET. Any suspicious activity that was unsuccessfully detected as intrusive by the misuse and anomaly detection mechanisms will be further investigated with the help of friends. Two types of friends have been identified as appropriate to be applied in the system; 1st and 2nd degree friends. 1st degree friends are nodes in the networks that have a direct connection with the source node (node that initiates the global detection process). Nodes in the network will have a direct connection to each other if they are friend in a real world or they are able to establish a mutual trust between them in a secure channel. On the other hand, 2nd degree friends are nodes in the networks that do not have a direct connection to the source node. 1st degree friends can participate in the global detection process without any problem because their identity can be verified by the source node.

However, source node might only have a few 1st degree friends especially at the early stage of its participation in the networks. As a result, a global detection mechanism might take a longer time to complete or might not be completed because the insufficient number of 1st degree friends' reports received. For that reason, 2nd degree friends' reports can be accepted to speed up the detection process. However, since 2nd degree friends are the indirect friends to the source node and their identity cannot be directly verified, a referee (node that has 1st degree relationships with both source and 2nd degree node) is needed to verify the 2nd degree node's identity. Referees buffer provides temporary storage for all the reports received by the 2nd degree friends before the referee nodes will verify them. It is important to mention here that a referee node must be a 1st degree friend to both the source node (node that triggered the global detection mechanism) and to the 2nd degree friend, which made the report. Reports from both 1st and 2nd degree friends (verified by referees) will be counted by a voting mechanism. Once the reports reached the preset threshold limit, the response mechanism will be triggered.

3.2.6 Response mechanism

This module is responsible to react to any intrusive activity detected by the misuse, anomaly, or friend detection mechanisms. However, since the focus of the proposed architecture is on the detection mechanism, only the basic response strategies have been applied in the system. A local response unit in this module will raise an alarm to alert the local user about the detected intrusive activity. The intrusion alarm then will be broadcasted to the other nodes in the networks by a global response unit. Neighbouring nodes that receive the intrusion alarms will add the intrusive nodes to their own bad node tables to avoid collaborating with the bad nodes in future packet-forwarding processes. However, to avoid false accusations, only alarms received from 1st degree friends can be accepted.

3.3 Related Issues

The proposed IDS framework has been designed after considering several issues, which have been identified from the investigation of previous works. The following are some of the considerations that have been made when designing the proposed IDS framework as summarized in Table 1.

3.3.1 A two-tier detection architecture

The idea of having a two-tier detection architecture is to provide a faster detection mechanism that is capable of detecting intrusive activities at their initial stages. Relying upon a local-based detection method alone is not sufficient to detect intrusion early. Since information gathered in a local-based IDS is limited to the local activities, each node in the network must have enough experiences before any suspicious activities can be judged as intrusive or not. This limitation will slow down the detection process, which can be made faster if the host-based detection method is combined with the network-based detection method. For that reason, a global detection mechanism, which emulates the roles of a network-based detection method, has been proposed in this new IDS architecture. A local-based IDS is located in the first tier, and will be triggered first to investigate any suspicious activity before being passed to the global detection mechanism, which is located at the second tier. This is because the information gathered in a local-based IDS is the first hand information collected from a local audit data source, which can be trusted and can be made available instantly. On the other hand, detection processes in the global detection mechanism will require more time to be completed because the information supplied by the other nodes will require more time to reach the requested node and must be validated to ensure integrity.

3.3.2 Real time audit data source

Since the objective of the proposed architecture is to detect intrusion at an early stage of its appearance, a real time data collection strategy has been proposed in this architecture. A periodic data collection strategy might not be suitable in a high mobility MANET environment as the collected information might be valid for a short period only. For instance, the attackers might already leave the network when the IDS detects their intrusive activities if such activities were gathered using a periodic data collection strategy.

3.3.3 Hybrid detection method

The core element of an intrusion detection system is the detection method. In general, there are two detection methods that can be used and they are misuse detection and anomaly detection methods. Although researchers have made clear the advantages and disadvantages of both detection strategies (Oleg 2002), to choose between these two methods is still not an easy task especially in a MANET environment. Misuse signatures are difficult to build in MANET because of the immaturity and the characteristics (high mobility, transient connection, fluctuate wireless links, etc) of the network. This situation has driven most of the researchers working on the MANET IDS to choose the anomaly detection method for their proposed architectures (Sowjnya 2002), (Yongguang 2003). However, the capability of a misuse detection method cannot be simply ignored. Misuse detection can give results that are more accurate in term of detecting true intrusion attempts and to reduce the number of false alarms compare to the anomaly detection method. Considering both misuse and anomaly detection capabilities, the proposed framework tries to combine these two detection methods in one hybrid system with an aim to study its performance compared to the most frequently used (an anomaly detection strategy), in MANET environment. However, the difficulty of building the attack database is not the only reason that makes researchers choose to employ anomaly detection rather than a misuse detection method. Another reason is the absence of a system administrator in MANET makes the process of updating the attack database more difficult in such networks compared to the infrastructure networks. To ease this problem a signature management mechanism is deployed in the proposed framework. As time goes by, with the ongoing updated attack database, the misuse detection mechanism will reach its maturity level and be able to detect more intrusive activities within the acceptable false alarm rates.

3.3.4 Friend for global detection

The main issue to clarify here is the reason for using friends to assist in the global detection mechanism. As mentioned earlier, MANET operates in a self-organized manner without the existence of any authentication server to authorize each user in the network. Without a reliable identity verification measure, information about any intrusive activities gathered by each node cannot be shared with other nodes as the information might be falsified to blackmail other users. (Yongguang 2003) proposed a voting mechanism to overcome this problem. Voting mechanisms can be very useful to defend against a single blackmail attacker but it is not immune against multiple colluding blackmail attackers. However, this voting mechanism can be improved by filtering the votes. For instance, only votes from friends can be counted to judge any intrusive activity. In the proposed IDS framework, the concept of 1^{st} and 2^{nd} degree friends has been introduced to improve the existing voting mechanism.

Table 1: Problems and vulnerabilities of existing ID systems and how the proposed two-tier IDS can fix the problems

IDS Requirements	**Techniques used in existing IDS**	**Problems/ Vulnerabilities**	**Proposed solutions in two-tier IDS**
Network-based audit data sources	Receive audit data gathered by other trusted nodes	Information might be altered by malicious nodes	**Self-experience**: Capture overheard audit data of adjacent nodes only
	Mobile agents to collect audit data for global detection	Mobile agents are vulnerable to attacks	**Friends-observation**: Audit data for global detection will be captured and analyzed by friends
Detection methods	Misuse detection	Difficult to compile attack signatures	**Signature management:** Generating new attack signatures derived from true anomaly detection alerts
		Unable to detect novel attacks	**Hybrid detection**: Misuse/ anomaly detection
	Anomaly detection	High false alarms	**2^{nd} tier global detection:** Request further investigation and votes from trusted friends before making any decision
Global detections	Receive single report from trusted friends/ neighbours	Vulnerable to blackmail attacks	**Voting mechanism:** Pool reports/votes from 1^{st} and 2^{nd} degrees friends. Votes from 2^{nd} degree friends boost the global detection but require referees for integrity
	Voting threshold for several reports received	Vulnerable to colluding blackmail attackers	

4. Conclusions

MANET is threatened by several threats and vulnerabilities, which require special prevention, detection, and response mechanisms to be deployed in the system. In this paper, we have discussed these MANET security-related issues and have presented a new IDS framework suitable for such networks. The two-tier IDS for MANET has been designed to improve the performance of existing IDS, which have been proposed by other researchers. This new framework is expected to give us a faster and more reliable detection results than what has been offered by previous efforts. In future work, a prototype of this IDS framework will be designed and its performance will be analyzed.

References

Foo, Y. L. (2004) *Ad Hoc Network: Prospects and Challenges*, Graduate School Research Paper (Rinkou), Department of Information and Communication Engineering, University of Tokyo.

Mark K. (2002) "Security Lifecycle- Managing the Threat", GSEC Practical, Vo1 3.

Srdjan;C., Jean-Pierre H., and Levente B. (2003) *Mobility Helps Security in Mobile Ad Hoc Networks*, MobiHoc'03, Annapolis, Maryland, USA.

Frank S. and Ross A. (1999) *The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks*, 1999 AT&T Software Symposium.

Lidong Z. and Zygmunt J. H. (1999) "Securing ad hoc networks", *IEEE Network*, Vol 13, No. 6, pp24-30.

Lakshmi V. and Dharma P. A. (2000) *A Novel Authentication Scheme for Ad Hoc Networks*, IEEE Wireless Communications and Networking Conference (WCNC 2000), pp1268-1273.

Yih-Chun H., David. B. J., and Adrian P. (2002) *SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks*, Fourth IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, New York.

Kimaya S., Bridget D., Brian N. L., Clay S., and Elizabeth M. B-R. (2002) *A Secure Routing Protocol for Ad Hoc Networks*, 2002 IEEE International Conference on Network Protocols (ICNP), pp78-89.

Manel G. Z. and N. Asokan (2002) *Securing ad hoc routing protocols*, ACM workshop on Wireless security, Wise'02, Atlanta, Georgia, USA.

Sheng Z., Yang R. Y., and Jiang C. (2003) *Sprite: A Simple, CheatProof, Credit-Based System for Mobile Ad Hoc Networks*, IEEE INFOCOM'03, San Francisco.

Pietro M. and Refik M. (2002) *Core: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks*, Sixth IFIP Communications and Multimedia Security Conference, Portorosz, Slovenia.

Sergio M., T. J. Giuli, Kevin L., and Mary B. (2000) *Mitigating routing misbehavior in mobile ad hoc networks*, Sixth annual international conference on Mobile computing and networking, pp255-265.

Sowjnya R., Hiren S., Vishal S., Jeffrey U., and Anupam J. (2002) *Neighborhood Watch: An Intrusion Detection and Response Protocol for Mobile Ad Hoc Networks*, Student Research Conference, University of Maryland at Baltimore County (UMBC).

Yi-An H., Wei F., Wenke L., and Philip S. Y. (2003) *Cross-Feature Analysis for Detecting Ad-Hoc Routing Anomalies*, 23rd International Conference on Distributed Computing Systems (ICDCS), pp478-489.

Yongguang Z., Wenke L., and Yi-An H. (2003) "Intrusion Detection Techniques for Mobile Wireless Networks", *Journal of Wireless Networks*, Vol 9, No. 5, pp545-556.

Hao Y., Xiaoqiao M., and Songwu L. (2002) *Self-organized network-layer security in mobile ad hoc networks*, ACM MOBICOM Wireless Security Workshop (WiSe'02), Atlanta.

Sonali B. and Dharma P. A. (2001) *Security Enhancements in AODV protocol for Wireless Ad Hoc Networks*, Vehicular Technology Conference, Atlantic City.

Oleg K. and Ratan G. (2002) *Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks*, IEEE Workshop on Knowledge Media Networking (KMN'02), Kyoto, JAPAN.

A Two-Tier Intrusion Detection System for Mobile Ad Hoc Networks – A Friend Approach

Shukor Abd Razak[1], Steven Furnell[1], Nathan Clarke[1], and Phillip Brooke[2]

[1]Network Research Group, School of Computing, Communications & Electronics, University of Plymouth, Plymouth, United Kingdom
info@network-research-group.org
[2]School of Computing, University of Teesside, Middlesbrough, United Kingdom
P.J.Brooke@tees.ac.uk

Abstract. Existing Intrusion Detection Systems (IDS) in Mobile Ad Hoc Network (MANET) environments suffer from many problems because of the inherent characteristics of the network. Limited audit data, along with the problems faced in achieving global detection and response mechanisms, creates challenges for establishing reliable IDS for MANETs. In this paper, several scenarios are investigated where a 'friend' concept has been applied to solve MANET problems. This same concept is applied to a new IDS framework, and discussion is presented into how it can help in minimizing the problems that are faced in existing IDS. The key advantages of this two-tier IDS framework are its ability to detect intrusion at an early stage of such behaviour in the network, and its capability to minimize the impact of colluding blackmail attackers in the systems.

1 Introduction

MANET is a computer network that combines the capabilities of peer-to-peer, wireless, and mobile network technologies and has been used to support communications in various environments such as in military and disaster relief operations [1]. It has several unique characteristics that make it differ from other types of computer networks. It operates in a fully distributed fashion without the aid of a central authority, has random network topologies, and uses wireless links for communications. Since the conception of MANET several security measures, concepts and architectures have been proposed to counter many of the inherent security concerns the network topology introduces. However, most of these are focused upon prevention mechanisms to protect MANET from external attackers. It is suggested that by employing an IDS as a second line of defense could be very useful whenever prevention mechanisms failed to protect the network. In this paper, two important issues in MANET IDS are discussed: what is the best way to detect intrusions in a collaborative fashion; and how to minimize the impacts of blackmail attacks/false accusations. This paper proposes a new IDS framework for MANET to provide solutions for such problems.

The paper proceeds to provide some background of a friend concept in small world phenomenon and discusses how it can be applied in a MANET environment. Section

3 summarizes some existing work in MANET security, which is related to the concept of friend, and Section 4 outlines some of the important features of a two-tier IDS framework.

2 Friends as Short Cut in Ad Hoc Networks

The small world phenomenon is a concept that suggests any two individuals, selected randomly from almost anywhere on the planet, are connected via a chain of no more than six acquaintances. Milgram [2] conducted an experiment in which he sent 60 letters to various recruits in Wichita, Kansas who were asked to forward the letter to the wife of a divinity student living at Cambridge, Massachusetts. The letters could only be forwarded by hand to personal acquaintances (directly or through friend of a friend) who they thought might be able to reach the recipient. Milgram claimed that he has proved the concept when 3 out of 60 letters that he sent reached the recipients but neglected to say about the low (i.e. 5%) chain completion percentage. However, his experiment has motivated other researchers to investigate more on this concept, such as in the Internet context, as observed by Adamic [3]. In his study, he suggested that the World Wide Web is a 'small world' in a sense that all the sites are highly clustered yet the path length between them is small. Helmy [4] established a relationship between the small world concept and wireless networks. Simulations result from his study proved that by adding a few 'short cut' nodes in the wireless networks, the degree of separation between nodes could be decreased drastically. One question emerging from this study is how to select few 'short cut' nodes in an autonomous, fully distributed, and self-organized ad hoc network. The author proposed the concept of *contacts*, which will act as short cuts to transform the wireless network into a small world. However, the author did not discuss how these *contacts* can be made available in the system, and this problem remains an open issue.

The concept of 'friends' has been introduced in MANET environments to solve many problems, especially those that relate to security issues. One of the common assumptions made by researchers to create friend relationships is that each node must be known to each other in a real world before they can establish a friend relationship in MANET environment. Based upon the concept that a friend in the real world is also a friend in the MANET, along with the concept of six degree separations between friends in real world, we propose a two-tier IDS framework for MANET to investigate how we can benefit from these concepts. Several previous works that make use of 'friendship' concept are discussed in the next section followed by the details of the proposed IDS framework in Section 4.

3 Related Work

Establishing a security association between nodes is very important because without it, secret information such as users' personal information or network information might be passed to unauthorized parties. One way to establish security associations is by deploying an encryption mechanism (e.g. private/public key system). Each message will be encrypted with the recipient's public key so that it can be decrypted using the corresponding private key. While this system could work perfectly well in wired networks, where there exists a central server to manage and to distribute the public keys of each node, the same scenario does not apply in MANET. Since we cannot assume the existence of a central authority to manage and distribute the keys, it seems impossible for each node to know the public keys of others without having a physical contact. However, this problem can be eased with the help of friends as suggested by Capkun et al. [5]. They suggested that each node is capable of establishing a security association with another anonymous node in the system by requesting a recommendation from friends. Friend nodes in their system are nodes that one has physically met in a real world. With a recommendation from a friend, a trustworthiness level for an anonymous node can be determined, thus a security association between two anonymous nodes can be established without the need of a physical contact in a real world.

In a real world, when we apply for a job, usually we are required to supply the employer with names of referees, who know about our background, capabilities and enthusiasm, and may also be used for security purposes. The same concept has been applied in a MANET environment, as proposed by Weimerskirch and Thonet [6]. Their concept is somewhat similar to the work proposed in [5] where recommendation from a friend is needed to authenticate an anonymous node in the system. However, in their system, the anonymous node will supply a few names as its references so that its trustworthiness can be judged.

In another scenario, a friend concept has been used to prevent node selfishness in the routing mechanism. In a MANET, nodes might sometimes refuse to participate in network operations in order to save their own limited resources. As mentioned in [7], nodes can be forced to participate in network operations in two ways; either penalizing them for not cooperating, or rewarding them for their participation. However, this mechanism creates unfairness especially for the nodes that are located outside the 'busy' area. Nodes need credits to send their own packets in the network and the only way to gain credits is by forwarding others' packets. However, for nodes located outside the 'busy' area, the chances for them to be selected in a packet forwarding process are low, and this will make it difficult for them to gain more credits. Miranda and Rodrigues [8] suggest a solution for this problem by using a friend concept. They proposed a concept of selective forwarding, where each node will only participate in a packet forwarding process if the packets come from, or need to be sent to, one of its friends. Each node will advertise to others about its friend list so that it cannot be accused for not forwarding other nodes' packets that are not in its friend list.

4 A Two-Tier IDS for Mobile Ad Hoc Networks

A two-tier hybrid IDS for MANET is a novel IDS architecture proposed to improve the efficiency of existing MANET IDS architectures with the help of friend nodes. The main idea of the proposed system is to provide a reliable IDS that can detect any intrusion attempts and at the same time reduce the number of false alarms raised in the system.

4.1 System Components

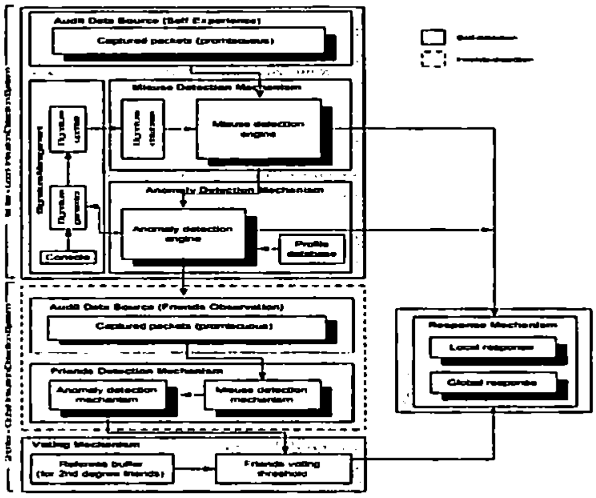

Audit Data Source (Self Experience)
Captured packets (promiscuous)
Misuse Detection Mechanism
Misuse detection engine
Anomaly Detection Mechanism
Anomaly detection engine
Profile database
Console
Audit Data Source (Friends Observation)
Captured packets (promiscuous)
Friends Detection Mechanism
Anomaly detection mechanism
Misuse detection mechanism
Voting Mechanism
Reference buffer (for 2nd degree friends)
Friends voting threshold
Response Mechanism
Local response
Global response

Fig. 1. Conceptual framework of the two-tier hybrid IDS for MANET

Real time audit data source. In the proposed architecture, two audit data sources have been identified as appropriate to detect intrusive activities in the networks. Any network operations initiated by, or having a direct connection with the participating nodes (source, destination, and all the intermediate nodes) are classified as self-experience audit data. Neighbours that are close to the participating nodes are also able to capture the overheard network activities using a promiscuous mode. This kind of audit data is known as friends' observation audit data in the proposed framework.

Misuse detection mechanism. This module comprises a misuse detection engine to detect activities that match the attack signatures as stored in the signature database. At the initial stage, the attack database might only cover a few attack signatures, but as time goes by, with the aid of the anomaly detection mechanism and the signature management module, the attack signature database will reach its maturity level and thus be able to detect more attacks.

Anomaly detection mechanism. Attacks that cannot be detected by a misuse detection mechanism will be passed here for further investigation. The failure of detecting the attacks could be because of the attack signature database is still immature or could be because of insufficient evidence. The anomaly detection mechanism applied here is similar to the existing techniques proposed by previous researchers, and its main components include an anomaly detection engine and a profile database.

Signature management. This module completes the feedback loop by enabling a dynamic update to the misuse detection mechanism. The signature generator automatically generates the attack signature each time the anomaly detection mechanism successfully identifies deviation from normal user/system profiles.

Friends detection mechanism. Any suspicious activity that was unsuccessfully detected as intrusive by the misuse and anomaly detection mechanisms in local detection will be further investigated with the help of friends. First degree friends are nodes in the networks that have a direct connection with the source node (i.e. the node that initiates the global detection process). Nodes in the network will have a direct connection to each other if they are friends in a real world. On the other hand, second degree friends are nodes in the networks that do not have a direct connection to the source node. First degree friends can participate in the global detection process without any problem because their identity can be verified by the source node. However, source node might only have a few first degree friends especially at the early stage of its participation in the networks. As a result, a global detection mechanism might take a longer time to complete or might not be completed because of the insufficient number of first degree friends' reports received. For that reason, second degree friends' reports can be accepted to speed up the detection process. However, since second degree friends are the indirect friends to the source node and their identity cannot be directly verified, a referee (a node that has first degree relationships with both the source and the second degree node) is needed to verify the second degree node's identity. Reports from both first and second degree friends are equal in weight and will be counted by a voting mechanism. Once the reports reached the preset threshold limit, the response mechanism will be triggered.

Response mechanism. A local response unit will raise an alarm to alert the local user about the detected intrusive activity. The intrusion alarm then will be broadcasted to the other nodes in the networks to make them aware about the existence of intrusive nodes. However, to avoid false accusations, only alarms received from first degree friends can be accepted.

4.2 Friends' Role in Two-Tier IDS

Speed up the detection process. Cooperative detection could speed up the detection process but this method is vulnerable to packet modification attacks. Friend detection mechanism in the two-tier IDS can ease this problem as each node in the system will carry out the detection process based on its own local audit data, and will only share the result of the decision whether the suspicious node is malicious or not.

Minimizing the risk of cooperative blackmail attacks. The problem of blackmail attack has been discussed in [9], and the authors suggested that a voting mechanism could ease the problem. However, a voting mechanism could only be used to protect the network from a single blackmail attacker, but not a cooperative blackmail attack. A friend mechanism is capable of minimizing the risk of such problems as only detection results from friends can be accepted in the proposed system. In case of there being a lot of blackmail attackers, the immunity of a friend mechanism can be strengthened by increasing the number of positive detection results that must be gathered from friends before any suspicious activity can be confirmed intrusive.

Reliable global response mechanism. Broadcasting intrusion alerts is a big challenge in a MANET because each node is anonymous to others, and there is always a possibility that some of the alerts are not genuine (i.e. broadcasted by attackers). The reliability of a global response mechanism can be increased with the help of friend nodes. Since each node is only interested in the alerts that came from its friends, all other alerts (including the fake ones) will be dropped. This will solve the false accusation problem caused by the fake alerts in the system.

5 Conclusion

In this paper, a new IDS framework for MANET environments based upon the concept of a friend in a small world phenomenon has been proposed. Current anomaly detection mechanisms as proposed in previous work make the detection process longer, as the system needs to gather sufficient evidence before a decision can be made against any suspicious activity. In another scenario, existing techniques for global detection suffer from the potential for blackmail attackers and false accusations. The proposed two-tier IDS framework has been designed to overcome these issues with the help of friend nodes. For future work, simulations will be carried out to investigate the performance of the proposed IDS framework in various MANET scenarios. It is hypothesized that with the introduction of friend nodes, the impacts of the IDS problems mentioned earlier can be minimized.

References

1. F. Stajano, and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks. In Proc. of the 7th Int. Workshop on Security Protocols, LNCS, vol. 1796, pp172-194, 1999.
2. S. Milgram. The Small World Problem. Psychology Today, pp60-67, May, 1967.
3. L. Adamic. The Small World Web. In Proc. of Eur. Conf. on Digital Libraries (ECDL), pp443-452, September, 1999.
4. A. Helmy. Small Worlds in Wireless Networks. IEEE Communications Letters. Vol. 7, No. 10, October, 2003.
5. S. Capkun, J.-P. Hubaux, and L. Buttyan. Mobility Helps Security in Ad Hoc Networks. In Proc. of MobiHoc'03, Annapolis, Maryland, USA, pp46-56, June, 2003.
6. A. Weimerskirch, and G. Thonet. A Distributed Light-Weight Authentication Model for Ad-Hoc Networks. In Proc. of 4th International Conference on Information Security and Cryptology (ICISC 2001), pp341-354, Seoul, South Korea, December, 2001.
7. B. Raghavan, and A.C. Snoeren. Priority Forwarding in Ad Hoc Networks with Self-Interested Parties. Workshop on Economics of Peer-to-Peer Systems, Berkeley, USA, May, 2003.
8. H. Miranda, and L. Rodrigues. Friends and Foes: Preventing Selfishness in Open Mobile Ad Hoc Networks. In Proc. of the First International Workshop on Mobile Distributed Computing (MDC'03), USA, 2003.
9. Y. Zhang, W. Lee, and Y.-A. Huang. Intrusion Detection Techniques for Mobile Wireless Networks. Journal of Wireless Networks, Vol. 9, No. 5, pp545-556, 2003.