ADDIS ABABA UNIVERISTY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Resource Efficient Key Distribution System for MANETs

Dawit Wubshet

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILMENT FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

December 2014

ADDIS ABABA UNIVERISTY

SCHOOL OF GRADUATE STUDIES

COLLEGE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

Resource Efficient Key Distribution System for MANETs

Dawit Wubshet

Advisor: Dejene Ejigu (PhD)

APPROVED BY EXAMINING BOARD:

1. **Dr. Dejene Ejigu, Advisor** _____

2._____ _____

3. _____ _____

Acknowledgements

First and for most, I like to thank GOD for making this possible. I would like to thank my advisor Dr. Dejene Ejigu for his dedicated support in this work. I also like to thank my family and friends for the invaluable ideas and comments they provided me.

Table of Contents

List of Figures v

List of Tables vi

Acronyms vii

Abstract viii

Chapter One - Introduction 1

1.1 Overview 1

1.2 Statement of the Problem 2

1.3 Motivation 3

1.4 Objectives 4

General Objective 4

Specific Objectives 4

1.5 Methods 4

Literature Review 4

Development of the Simulated Prototype 4

1.6 Scope and Limitations 5

1.7 Application of Results 5

1.8 Organization of the Thesis 7

Chapter Two-Literature Review 8

2.1 Overview 8

2.2 Characteristics of MANETs 8

2.3 Securing MANETs 9

2.3.1 Challenges in Securing MANETs 9

2.3.2 Threats and Attacks 10

2.3.3 Approaches in Securing MANETs 11

2.4 Key Management in MANETs 13
2.5 Clustering and MANETs 15
2.6 Summary 18
Chapter Three-Related Work 19
3.1 Overview 19
3.2 Asymmetric Key Distribution Systems 19
3.3 Symmetric Key Distribution Systems 21
3.4 Group Based Key Distribution Systems 22
3.5 Hybrid Key Distribution Systems 24
3.6 Summary 24
Chapter Four – The Proposed Key Distribution System 28
4.1 Overview 28
4.2 The Proposed key Distribution System Architecture 29
4.3 Key Distribution 30
4.3.1 Key Client 31
4.3.2 Key Server 33
4.3.3 Cache Builder and Key Cache 34
4.4 Clustering Algorithm 37
4.4.1 Selection Process - Phase One 39
4.4.2 Selection Process - Phase Two 42
Chapter Five-Experimentation and Evaluation 56
5.1 Overview 56
5.2 Metrics Used for Evaluation 56
5.3 Experimental Design and Setup 57
5.4 Result and Discussion 59

Chapter Six-Conclusion and Future Work....61
6.1 Conclusion....61
6.2 Future Work....62
References....63
Appendix A. Default Node Configuration for Simulation....67
Appendix B. Snippet of Cluster Head Conflict Resolution for PC....69
Appendix C. Snippet of Modified Cluster Head Conflict Resolution for PC....69
Appendix D. Network Description File Code for the Proposed Key Distribution System....70

List of Figures

Figure 3.1: Example of PIKE Matrix for Key Pre-Distribution 21

Figure 3.2: Cluster Based Group Key Management 23

Figure 4.1: Proposed Key Distribution System Architecture 29

Figure 4.2: Pseudo Code of Initialization Algorithm for the Proposed Key Distribution System 30

Figure 4.3: Pseudo Code of Key Clients Algorithm for the Proposed Key Distribution System 33

Figure 4.4: Pseudo Code of Key Servers Algorithm for the Proposed Key Distribution System 34

Figure 4.5: Snippet of Data Structure for Global Request List 35

Figure 4.6: Pseudo Code of Cache Builder Algorithm for the Proposed Key Distribution System 36

Figure 4.7: Inter Cluster Key Infection 37

Figure 4.8: Ad hoc Host Setup 43

Figure 4.9: Intra Cluster Hop Count Result for PC 46

Figure 4.10: Inter Cluster Hop Count Result for PC 47

Figure 4.11: Intra Cluster Hop Count Result for DDCA 49

Figure 4.12: Inter Cluster Hop Count Result for DDCA 50

Figure 4.13: Intra Cluster Hop Count Result for Lin 51

Figure 4.14: Inter Cluster Hop Count Result for Lin 52

Figure 5.1: Single MANET Host Setup Running the Proposed Key Distribution System 57

Figure 5.2: Snap Shot of a Running Simulation 58

List of Tables

Table 2.1: Summary of Clustering Algorithm Categories 16

Table 2.2: Costs of Clustering 17

Table 3.1: Summary of Related Works 25

Table 4.1: Clustering Algorithm Comparison Result 39

Table 4.2: Clustering Algorithm Comparison Phase One Result Summary 41

Table 4.3: Intra Cluster Communication Efficiency Result Summary 53

Table 4.4: Inter Cluster Communication Efficiency Result Summary 54

Table 5.1: Simulation Parameters 58

Table 5.2: Experimentation Result Summary 59

Acronyms

3hBAC: 3-hop Between Adjacent Cluster heads

AES: Advanced Encryption Standard

AODVUU: Ad hoc On-demand Distance Vector

CA: Certificate Authority

CDS: Connected Dominating Set

CH: Cluster Head

DDCA: Dynamic Distributed Clustering Algorithm

DES: Data Encryption Standard

DOS: Denial of Service

DS: Dominating Set

ECC: Elliptic Curve Cryptography

IDEA: International Data Encryption Algorithm

IDS: Intrusion Detection System

IP: Internet Protocol

KDC: Key Distribution center

LCC: Least Cluster Change

Lin: Lin's clustering Algorithm

MANET: Mobile Ad-hoc Network

MOBIC: Mobility Based Metric for Clustering

MOCA: Mobile Certificate Authority

MR: Mobile Routes

PC: Passive Clustering

PIKE: Pre Intermediaries for Key Establishment

RSA: Rivest, Shamir and Adelman

SEGK: Secure and Efficient Group Key Management

TCP: Transmission Control Protocol

UDP: User Datagram Protocol

v2v: Vehicle-to-vehicle

VoIP: Voice over Internet Protocol

WCA: Weighted Clustering Algorithm

WCDS: Weakly Connected Dominating Set

Wu: Wus Clustering Algorithm

Abstract

Given MANET's lack of static network infrastructure, lack of resource and the use of wireless medium for communication, its key management system is an excellent point of attack for intruders. By compromising on security and using shared keys to avoid the burden of asymmetric key management for each node, vulnerability is created in the network. Different types of applications run on MANET's. There are applications that may require hardened security like onsite emergency response applications and others which may not require hardened security like conference slide sharing applications. To ensure security for those applications that can't afford a security vulnerability due to sharing of keys, a system should use asymmetric keys of which its private key is only known by the owner nodes and its identity confirmed by a certificate authority.

This work proposes a resource efficient asymmetric key distribution system for MANETs. We have used decentralized trust model where each node, can act as a certificate authority and is able to cache certified keys. The system is also supported through clustering. It is able to do its intended task by having a key server, key client, cache builder and key cache modules installed on every node. Each node will perform its task based on the role assigned to it by the clustering algorithm. Since clustering algorithms have their own resource foot prints, we had to search for a clustering algorithm among that will not burden the network with extra cluster formation communication. We were able to experimentally select a clustering algorithm called passive clustering algorithm that will not burden the network with its own clustering information exchange. This clustering algorithm uses existing traffic to maintain clustering information.

We have evaluated the proposed key distribution system by simulating and comparing its efficiency and key delivery rate against zone based key distribution system and MOCA key distribution system. The result showed that the proposed system was able to deliver keys 8.06 ms average end to end delay, 5.22 average hop count and 71.33% key delivery rate, whereas zone based key distribution system scored 14.13 ms average end to end delay, 8.66 average hop count and 75.69% key delivery rate and finally MOCA scored 15.3 ms in average end to end delay, 10.42 hops in average hop count and 67.36% key delivery rate. The proposed key distribution was able to retrieve keys faster using less communication. This is because the key caching capability of the proposed system improved the amount of communication needed to retrieve a key. The errors recorded in the key delivery rate were due to packet losses that occurred during wireless transmission.

Keywords: Key Distribution in MANETs, MANET Trust Model, Cryptography, MANET Security

Chapter One – Introduction

1.1 Overview

Over the past few decades, computing has evolved from large, expensive and standalone computers to small, chip and ubiquitous devices. These devices, unlike their ancestors, are highly mobile and have the ability to communicate with each other without requiring an access point or additional networking infrastructure [1]. This type of communication schema is called Mobile Ad hoc Network [1].

A Mobile Ad hoc Network (MANET) is made up of a set of MANET routers (MRs) [1]. These MRs organize and maintain a routing structure among themselves over dynamic wireless interfaces. As any Internet Protocol (IP) router, a MR may have an attached set of nodes. These nodes access the MANET via the MR to which they are attached. Due, in part, to relative movements of MR and, in part, to environmental effects (especially wireless link characteristics), the network topology and communication links in a MANET may change state more frequently than in fixed wired or fixed wireless networks. These attributes and others influence security protocol design for MANETs. MANETs may in some cases operate as stand-alone networks, or they may be used to extend the wireless mobile range of a more fixed infrastructure network. Even if this communication schema sounds seamless, it comes with a security price [1].

Securing MANETs is a challenging task because, MANETs use trust based routing, are highly dynamic because of mobile nature of nodes and on top of that there is no static infrastructure to enforce security rules [2, 21]. In spite of the mentioned constraints, MANETs have the following strict security requirements [20];

- Authentication: authenticating other nodes.
- Availability: ensuring that services offered are available.
- Integrity: ensuring that message or the entity under consideration is not altered.
- Confidentiality: providing privacy of the wireless communication channels.
- Non-repudiations: preventing malicious nodes from hiding their activities.
- Survivability: ability to provide a minimum level of service in the presence of power loss, failures or attacks.
- Degradation of security services: ability to change security level as resource availability changes.

Therefore, in order to achieve these security requirements, nodes should use a strong cryptography to hide messages over the network. In order for the nodes to use strong cryptography they need an efficient key distribution system.

There are several message relay based key distribution systems and group based key distribution systems [10, 14, 19, 24]. These schemes could use tweaking for better efficiency, since more and more is expected out of MANETs due to expansion of pervasive computing and growing use of MANET applications.

Hence, this work is aimed at proposing a key distribution system using decentralized trust modeled solution with caching capability that allows for maximum security and resource efficiency.

1.2 Statement of the Problem

Several end to end cryptography techniques can be implemented to transmit messages among nodes. In order to achieve end to end cryptography using symmetric keys requires the generation of a key for each communicating node pair, and the generated key has to be transmitted using a secure channel. In case of asymmetric cryptography, in addition to providing security service of non-repudiation inherently [4], a single public key (session key) can be used to achieve multiple end to end secure communications without requiring to setup secure channel to transfer the public key. Hence an efficient public key distribution system is required to manage and share public keys of nodes.

The existing key management systems that are:

- Based on group key sharing [10, 12, 43], require secure cannel to communicate group keys, in addition to the tradeoff of security by sharing group keys that may expose the entire traffic of the group, if an attacker acquires the group key by compromising a single group member.
- Based on symmetric keys [32, 33], require additional secure cannel to share symmetric keys and create burden in managing symmetric keys for each communicating node pair.
- Based on asymmetric keys [19, 24, 20], burden certificate authority nodes and could also use tweaking for better efficiency in resource usage.

This research is aimed at proposing a key distribution system for MANETs that allows for maximum resource efficiency using asymmetric keys generated by the nodes themselves and sharing the public keys using decentralized trust model supported by key caching and clustering; hence easing load from certificate authority nodes and sharing the burden with regular nodes.

1.3 Motivation

MANETs are gaining more and more popularity together with the technological advancements in the field of mobile devices and mobile network technologies. Secure communication in these networks requires the traffic to be encrypted, and there are many security protocols available for this task, but most of them need the key distribution system to work in a secure and efficient manner. It is quite hard to implement secure, efficient and flexible key distribution system with fixed networks where the supporting infrastructures like: name servers and dedicated routers, as well as resources like computational power, memory and bandwidth, are readily available. However, it is even more difficult to solve this key distribution problem with MANETs since they lack these entire supporting infrastructures.

Key distribution is not the only service running in such a resource constrained environment like MANET. Some of the basic network services that a MANET host has to give include:

- Authentication
- Routing
- Quality of Service
- Dynamic Host Configuration

Since MANETs rely on the collaboration principle, a significant amount of communication is needed to provide the mentioned network service. Hence the design of network services given on MANETs need to be resource efficient.

Possible applications of MANETs include: soldiers relaying information for situational awareness on the battlefield, business associates sharing information during a meeting, attendees using laptop computers to participate in an interactive conference, emergency disaster relief personnel that are coordinating efforts at sites of fires, hurricanes, or earthquakes, onsite police response teams and also for the future, Internet of things will utilize MANET technology [6]. Some of these applications cannot afford security holes because of the nature of service they give. Hence they require secure end to end cryptography to hide messages they exchange over wireless medium (which is vulnerable to eavesdroppers within the transmission range).

The motivation for our work comes from the need to optimize the resource usage of cryptographic key distribution systems in MANETs without compromising on security. Hence our work proposes a way of asymmetric key distribution that eases the burden of certificate authority nodes and optimizes resource usage.

1.4 Objectives

General Objective

The general objective of this work is to propose a resource efficient asymmetric key distribution system for MANETs.

Specific Objectives

The following specific objectives will help to accomplish the general objective stated above.

- Do a thorough study of key distribution in MANETs.
- Develop appropriate key distribution architecture.
- Select grouping schema for nodes in the MANET to organize themselves.
- Design and integrate a caching rule and algorithms for the key distribution system that utilizes selected grouping schema.
- Develop a key distribution system prototype under a simulated environment.
- Test and evaluate the efficiency and performance of the prototype against other key distribution systems.

1.5 Methods

The following methodologies will be used in the research process.

- **Literature Review**

Literature review will be done on different areas that are considered to be relevant for our work. Research papers related to key distribution in MANETs, MANET security, and MANET clustering algorithms will be investigated in detail.

- **Development of the Simulated Prototype**
  - **Design**

    In the design phase of the system, the overall system architecture and key distribution algorithms specified in the specific objective will be designed.

  - **Development of Simulated Prototype**

    Due to infeasible coat of MANET devices, the prototype of the system will be implemented on top of a simulated environment. The simulation will account for mobility and number of nodes.

1.6 Scope and Limitations

This research work assumes that the nodes in the MANET are authenticated and thus are trusted. It is not within the scope of this research to make sure that the trusted nodes do not perform malicious activities with the keys they are entrusted with. Our work will focus on the key distribution aspect of the security.

1.7 Application of Results

The proposed key distribution system will enable nodes in MANET to use asymmetric end to end cryptography, by enabling them to share keys in an efficient manner. Up on meeting the objective of the research, the contribution of the work will enable MANET applications to enjoy a resource efficient asymmetric key distribution system.

Typical MANET applications that require secure end to end crypto system and benefit from our work include [2]:

a. Networking on the Battlefield:

There are some major differences between creating a traditional fixed infrastructure network and providing a battlefield network to the tactical edge [28]. The obvious differences are that the networking equipment must be rugged to withstand the harsh battlefield environment and must be size, weight, and power optimized because it will be carried by soldiers or deployed in combat vehicles. On a battlefield, there is no fixed networking infrastructure [5]. Soldiers and assets are mobile. Even if they do not have a connection back to the network at the command post, they still have a need for voice, data, and video communications with each other. Without any fixed networking infrastructure, a battlefield network has to be created "on the fly" using MANETs. The fact that these networks are self-forming and self-healing facilitates deployment and minimizes the need for manual configuration and intervention [28]. They also support multi-hop networking to extend coverage and provide redundant paths for increased resilience [6].

Such networks are key enablers for applications such as:

Vehicle-to-vehicle networking (v2v) [4]: is a MANET automobile technology designed to allow automobiles to "talk" to each other.

Telemetry monitoring [5]: Telemetry monitoring involves watching and analyzing data received at a distance from its source in real time. E.g., monitoring of soldier's health by monitoring his/her cardiac activity and blood pressure in the battle field could play a key role in saving a wounded soldiers life. It also increases situational awareness.

MANET voice over IP [28]: MANET VoIP technology works by converting voice signals to data packets and transferring them on MANET IP network. Every packet has its destination address, so other nodes cannot receive it. By this way different communication groups can avoid interfering with each other. What's more, making use of network security procedures such as encryption improves the safety of the communication further. So it is desirable and necessary to implement VoIP in portable MANET equipment (Tablets, Notebook etc.) and use these equipment's as communication tools.

b. Emerging Pervasive Applications:

In pervasive applications, computing is made to appear everywhere and anywhere [41]. In contrast to desktop computing, pervasive computing can occur using any device, in any location, and in any format. A user interacts with a computer that can exist in many different forms, including laptop computers, tablets and terminals in everyday objects such as a fridge or a pair of glasses [41]. Pervasive applications use MANETs in order to interact with devices in the environment. The mobility of nodes and simplicity of the network makes MANETs an ideal environment for pervasive applications to run on.

Pervasive applications that run on MANETs include:

Smart home [40]: The past decade has seen significant advancement in consumer electronics. Various 'intelligent' appliances such as air conditioners, home security devices, cellular phones, home theatres etc. are set to realize the concept of a smart home. They have given rise to a Personal Area Network in home environment where all these appliances can be monitored using a single controller. Smart home also monitors the activities of the occupant of a home, operates devices in a predefined pattern as the users require and provide solutions to energy saving.

Smart campus [39]: Building smart campus system, aims at the development of services and applications supported by a data gathering platform that integrates real time information systems and intelligent energy management systems that drive a bi-directional learning process such that the user learns how to interact with the building and the building learns how to interact with the user in a more energy efficient way.

c. Emergency Relief Sector:

When stricken by a catastrophic natural disaster, the efficiency of disaster response operation is very critical to life saving. However, communication systems, including cellular networks, usually crash due to various causes making the coordination among a large number of

disorganized disaster response workers extremely difficult [30]. Unfortunately, rapid deployment of an emergency communication system based on existing technologies may not be feasible since most technologies rely on a good transportation system to deliver essential equipment, which is usually not available in a catastrophic natural disaster. Due to their ad hoc nature, MANETs can be used in emergency/rescue operations for disaster relief efforts, e.g., in fire, flood, or earthquake. Emergency rescue operations must take place where non-existing or damaged communication infrastructure [29].

Emergency relief applications that run on MANETs include:

Emergency relief and crowd dispersal using MANET [30]: In a scenario where pedestrians are stranded, confused and they have a common goal to reach a safe destination for example home, they may move together in large groups in proximity to each other. Events in recent history concerning urban locations like London attacks have shown us that hotspots of trouble can erupt everywhere [30]. The idea is to send the confused crowd to their destination safely and without injury, as soon as possible using disaster recovery and evacuation mechanisms with the help of mobile ad hoc networks in absence of cellular infrastructure.

Rescue information system for earthquake disasters based on MANET [31]: Initially a simple MANET is implemented to support emergency information network. Rescue people, voluntary or mission-specific professionals could use their own notebook PCs to construct a multi-hop ad-hoc network to form a basic wireless Intranet. This network is then used to deploy communication applications such as VoIP, Push-to-Talk, Instant Messaging, and mobile social network, etc.

1.8 Organization of the Thesis

The rest of this thesis is organized as follows. State of the art and related work in the domain of MANET security and key distribution in MANETs are presented in Chapters 2 and 3. The proposed resource efficient key distribution system is presented in Chapter 4 followed by evaluation and findings in Chapter 5. Chapter 6 presents conclusion and future works.

Chapter Two-Literature Review

2.1 Overview

A Mobile Ad hoc Network (MANET) is a multi-hop network of wireless mobile hosts that form a temporary network without the aid of any centralized administration or support [2]. In areas where there is little communication infrastructure or the existing infrastructure is inconvenient to use, wireless mobile users may still be able to communicate through the formation of mobile ad hoc networks. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may be multiple hops away from each other [1, 3, 4]. This Chapter discusses issues related with MANET technology and approaches taken to achieve secure communication over MANETs.

2.2 Characteristics of MANETs

MANETs operate in isolation, or may have gateways to an interface with a fixed network. Its nodes are equipped with wireless transmitters/receivers using omni-directional antennas [11]. At a given time, the system can be viewed as a random graph due to the movement of the nodes [3].

According to [1, 3], the following are major characteristics of MANETs:

Autonomous: No centralized administration entity is available to manage the operation of the different mobile nodes.

Dynamic topologies: The network topology may change randomly and rapidly at unpredictable times. Nodes freely roam in the network, join or leave the network at their own will, and fail occasionally.

Resource constraints: The wireless links have significantly lower capacity than wired links. The computation and energy resources of a mobile device are limited.

Infrastructure-less: There is no well-defined infrastructure, access point, or some other central control point available. Moreover, the wireless medium is accessible by both legitimate nodes and attackers. There is no clear boundary to separate the inside network without using distributed authentication services.

Limited physical security: Portable devices are generally small with weak protection. The physical devices could be stolen or compromised.

Multi-hop routing: When delivering data packets from a source to its destination out of the direct wireless transmission range, the packets are forwarded via one or more intermediate nodes.

2.3 Securing MANETs

As various applications of wireless ad hoc networks have been proposed, security has become one of the big research challenges and is receiving increasing attention. Securing communications in resource constrained, infrastructure-less environments such as Mobile Ad Hoc Networks (MANETs) is very challenging because it lacks central authority to handle security services, hence security services must be handled in a distributed manner.

Cryptographic techniques are widely used for secure communications in wired and wireless networks. Most cryptographic mechanisms, such as symmetric and asymmetric cryptography, often involve the use of cryptographic keys. However, all cryptographic techniques will be ineffective if the key management is weak [2, 17, 21].

2.3.1 Challenges in Securing MANETs

The wireless, mobile and ad hoc nature of MANETs brings new security challenges to the network design. According to [2, 21], because wireless medium is vulnerable to eavesdropping and ad hoc network functionality is established through node cooperation, mobile ad hoc networks are intrinsically exposed to numerous security attacks.

During passive attacks, an attacker just listens to the channel in order to discover valuable information. This type of attack is usually impossible to detect, as it does not produce any new traffic in the network. On the other hand, during active attacks an attacker actively participates in disrupting normal operation of the network. This type of attack involves deletion, modification, replication, redirection and fabrication of protocol session control packets or data packets [18].

In addition to the common vulnerabilities of wireless connection, in some ad-hoc network scenarios, the network organization can completely or partially rely on a trust relationship between participating nodes, therefore an ad-hoc network has its particular security problems due to, e.g., nasty neighbor relaying packets [17]. This shows that these distributed network operations requires different schemes of authentication and key distribution. Further, wireless link characteristics also introduce reliability problems, because of the limited wireless transmission range, the broadcast nature of the wireless medium (e.g., hidden terminal problem), mobility induced packet losses, and data transmission errors [25].

Furthermore, the existence of network services such as routing protocols along with several other services like key management, in a dynamic and resource constrained environment of MANETs is a challenge for the mobility management; since each of these services has their own ways of gathering topology information, they cause additional communication overhead.

2.3.2 Threats and Attacks

Many passive and active security attacks could be launched from the outside by malicious hosts or from the inside by compromised hosts. This section gives a brief summary of the different attack classes. It is not our intent to give a complete listing, but to show the threats ad hoc networks are facing, and to show that very few malicious nodes can disable normal network operation.

Any participating node (insider) can spoof or alter routing information. Specific attack behaviors are related to the routing protocol used. For example, for some protocols, the attacker may modify the source route listed in the rout request or rout replay packets by deleting a node from the list, switching the order of nodes in the list, or appending a new node into the list [2, 3] or, when distance-vector routing protocols are used, the attacker may advertise a route with a smaller distance metric than its actual distance to the destination, or advertise routing updates with a large sequence number and invalidate all the routing updates from other nodes [3, 16].

By attacking the routing protocols and key management systems (which is the first line of defense), the attackers can attract traffic toward certain destinations in the nodes under their control, and cause the packets to be forwarded along a route that is not optimal or even nonexistent. The attackers can create routing loops in the network, and introduce severe network congestion and channel contention in certain areas. Multiple collaborating attackers may even prevent a source node from finding any route to the destination, and partition the network in the worst case [16]. In order to successfully launch these attacks, the attacker needs to impersonate a legitimate node participating in regular networking activities; the attacker accomplishes this by attacking the key distribution system and acquiring the group key, the symmetric key or private key of a legitimate user in the network.

In a Sybil attack a single node presents multiple identities to other nodes in the network or the attacker may further subvert existing nodes in the network, or fabricate its identity and impersonate another legitimate node [2, 16].

Sinkhole attacks typically work by making a malicious node looking especially attractive to the surrounding nodes with respect to the routing algorithm. The goal is to absorb as much traffic as possible from a particular area through a compromised node, creating a sinkhole with the adversary at the center. This attack is especially effective in networks with a special communication pattern like sensor networks or surroundings of Internet gateways [2, 16].

In a Wormhole attack, an attacker sends an adversary tunnel messages from one part of the network, usually via a low latency out of bound channel, to another part of the network where

they are replayed. These attacks can, among others, be used to distort routing, create sinkholes and to exploit routing race conditions as well as work even in the presence of authenticated and encrypted routing [16].

HELLO flood attacks are applicable against all protocols that use HELLO messages to form neighborhood relationships among the nodes. A malicious node can send, record or replay HELLO messages into the network with high transmission power and therefore convince every node in the network that it is its neighbor. This attack leaves the network in confusion as most nodes simply send their packets into oblivion [16].

Another type of attack is the denial-of-service (DOS) attack via network-layer packet blasting, in which the attacker injects a large amount of junk packets into the network. These packets waste a significant portion of the network resources, and introduce severe wireless channel contention and network congestion in the MANET [16].

In Man- in- the- middle attack, an attacker sits between the sender and the receiver and sniffs any information being sent between two nodes. In some cases, an attacker may impersonate the sender to communicate with the receiver or impersonate the receiver to reply to the sender [1, 2].

Most attacks can be avoided using strong authentication and using effective cryptographic techniques. In order to implement strong cryptography we need a secure and efficient key distribution system.

2.3.3 Approaches in Securing MANETs

This section will discuss approaches that are taken in order to secure MANETs along with the security services these approaches give.

As described in [2, 4, 5], the main security services can be summarized as follows:

Authentication: The function of the authentication service is to verify a user's identity and to assure the recipient that the message is from the source that it claims to be from.

First, at the time of communication initiation, the service assures that the two parties are authentic; that each participant is the entity it claims to be. Second, the service must assure that a third party does not interfere by impersonating one of the two legitimate parties for the purpose of authorized transmission and reception.

Confidentiality: It ensures that the data/information transmitted over the network is not disclosed to unauthorized users. Confidentiality can be achieved using different encryption techniques such that only legitimate users can analyze and understand the transmission.

Integrity: The function of integrity control is to assure that the data is received exactly as sent by an authorized party. That is, the data received contains no modification, insertion, deletion, or replay.

Access control: This service limits and controls the access of a resource such as a host system or application.

Non-Repudiation: This is related to the fact that if an entity sends a message, the entity cannot deny that it sent that message. If an entity gives a signature to the message, the entity cannot later deny that message. In public key cryptography, a node A signs the message using its private key, and all other nodes can verify the signed message by using A's public key, and A cannot deny the message with its signature.

Availability: This involves making network services or resources available to the legitimate users. It ensures the survivability of the network despite malicious incidences.

In order to provide the mentioned security services the following activities need to be done [17, 25].

a. Node Authentication

In order to achieve authentication, a user trying to gain access to the resource is first identified (authenticated) and then the corresponding access rights are granted. Distributed node authentication schema is another research challenge that is currently getting attention.

b. Encryption of Traffic

Cryptography is an important and powerful tool for secure communications. It transforms readable data (plaintext) into meaningless data (cipher text) [2].

Cryptography has two dominant categories, namely symmetric-key (secret-key) and asymmetric-key (public-key) approaches. In symmetric-key cryptography, the same key is used to encrypt and decrypt the messages, while in the asymmetric-key approach, different keys are used to convert and recover the information. Although the asymmetric cryptography approaches are versatile (can be used for authentication, integrity, and privacy) and are simpler for key distribution than the symmetric approaches, symmetric key algorithms are generally more computation-efficient than the asymmetric cryptographic algorithms. There are varieties of symmetric and asymmetric algorithms available, including DES, AES, IDEA, and RSA [19]. However, most crypto systems rely on the underlying secure, robust, and efficient key management subsystem. In fact, all cryptographic techniques will be ineffective if the key management system is weak. Key distribution is a central part of the security of MANETs.

In MANETs, the computational load and complexity for key management are strongly subject to restriction by the node's available resources and the dynamic nature of network topology. Our work aims to provide a key distribution system considering the mentioned constraints.

c. Intrusion Detection

Many historical events have shown that intrusion prevention techniques alone, such as encryption and authentication, which are usually a first line of defense, are not sufficient [35]. As the system become more complex, there are also more weaknesses, which lead to more security problems. Intrusion detection can be used as a second wall of defense to protect the network from such problems [34]. If the intrusion is detected, a response can be initiated to prevent or minimize damage to the system.

Intrusion detection can be classified based on audit data as either host based or network-based. A network based IDS captures and analyzes packets from network traffic while a host-based IDS uses operating system application logs in its analysis [35].

As in [34, 35], based on detection techniques, IDS can be classified into three categories as follows:

- **Anomaly detection systems**: The normal profiles (or normal behaviors) of users are kept in the system. The system compares the captured data with these profiles, and then treats any activity that deviates from the baseline as a possible intrusion by informing system administrators or initializing a proper response.

- **Misuse detection systems**: The system keeps patterns (or signatures) of known attacks and uses them to compare with the captured data. Any matched pattern is treated as an intrusion. Like a virus detection system, it cannot detect new kinds of attacks.

- **Specification-based detection**: The system defines a set of constraints that describe the correct operation of a program or protocol. Then, it monitors the execution of the program with respect to the defined constraints.

2.4 Key Management in MANETs

Cryptographic algorithms are security primitives that are widely used for the purposes of authentication, confidentiality, integrity, and non-repudiation. Most cryptographic systems require an underlying secure, robust, and efficient key management system. The key management system is responsible for making, distributing and re-keying of cryptographic keys [8, 9]. A key is a piece of input information for cryptographic algorithms. If the key was

released, the encrypted information would be disclosed. The secrecy of the symmetric key and private key must always be assured locally.

Similarly, the public key is protected by the public-key certificate, in which a trusted entity called the certification authority (CA) in a public key infrastructure vouches for the binding of the public key with the owner's identity [18].

Key integrity and ownership should be protected from advanced key attacks. Digital signatures, hash functions, and the hash function based message authentication code [19] are techniques used for data authentication and/or integrity purposes. In systems lacking a TTP, the public-key certificate is vouched for by peer nodes in a distributed manner, such as pretty good privacy (PGP) [8]. In some distributed approaches, the system secret is distributed to a subset or all of the network hosts based on threshold cryptography. Obviously, a certificate cannot prove whether an entity is "good" or "bad". However, it can prove ownership of a key. Certificates are mainly used for key authentication [8].

Key management can be classified based on cryptographic techniques they support and the trust model they use.

As in [7, 8, 9], there are three types of trust models for key distribution in MANETs:

a. Centralized Trust Model

In centralized approaches, a designated entity (e.g., the group leader or a key server) is responsible for distribution of the key to all the participants in a group. In this approach, the MANET is usually assumed to be attached to some static network.

b. Distributed Trust Model

With distributed or contributory key-agreement protocols, the group members cooperate to establish a group key. This improves the reliability of the overall system and reduces the bottlenecks in the network in comparison to the centralized approach.

c. Decentralized Trust Model

In this trust model more than one entity is responsible for making, distributing and re-keying the key. In this category, the group is divided into subgroups which have own subgroup managers. The goal of this structure is to make the protocols to scale better even if the group grows very large, since tasks of the KDC are distributed to multiple instances (the managers of the subgroups).

Works in [7, 8, 9] classify key management based on cryptographic techniques they support as follows:

a. Asymmetric Key Management

Every node in the network has a public/private key pair and it is the responsibility of the dealer to issue the initial certificate for the nodes public key as well as distributing the public key of the certificate authority which is needed to verify the certificates.

Even if public key management provides inherent non repudiation and stricter security than symmetric key, it has high computational and communication overhead for certificate authority nodes.

b. Symmetric Key Management

The key concept here is sharing of a single symmetric key among nodes that exchange messages. Symmetric key management is more difficult because managing keys for every node pair is a challenging task. Managing symmetric keys is usually done by maintaining a database of keys that contains the key for each nod pair in the network. This creates a serious vulnerability by creating a single point of attack which may leave the entire network traffic visible to intruders.

c. Group Key Management

The messages are protected by encryption using the chosen key, which in the context of group communication is called the group key. Only those who know the current group key are able to recover the original message. Group key establishment means that multiple parties want to create a common secret key to be used in the secure exchange of information [12].

However, group key management for large and dynamic groups in MANETs is a difficult problem because of the requirement of scalability and security under the restrictions of nodes' available resources and unpredictable mobility.

d. Hybrid Key Management

According to [23], hybrid key management combines asymmetric and symmetric cryptography to manage keys. Its approach is that, to use asymmetric and symmetric cryptographic keys selectively. Nodes in the same group can use symmetric key, and nodes in different groups may use asymmetric keys.

2.5 Clustering and MANETs

Clustering is an important issue in MANETs, since there is no other means to control the topology of the network [26]. However, a cluster-based MANET has its sideeffects and

drawbacks because constructing and maintaining a cluster structure usually requires additional cost compared with a flat-based MANET.

Many clustering schemes have been proposed for ad hoc networks. A systematic classification of these clustering schemes enables one to better understand and make improvements. Table 2.1 summarizes categories of clustering algorithms in MANETs [22].

Table 2.1: Summary of Clustering Algorithm Categories

Clustering Algorithm Categories	**Objective**
Low-maintenance [22]	Providing a cluster infrastructure for upper layer applications with minimized clustering-related maintenance cost.
Mobility-aware [22]	Utilizing mobile nodes' mobility behavior for cluster construction and maintenance and assigning mobile nodes with low relative speed to the same cluster to tighten the connection in such a cluster.
Combined-metrics-based Clustering [22]	Considering multiple metrics in cluster configuration, including node degree, mobility, battery energy, cluster size, etc., and adjusting their weighting factors for different application scenarios.
Energy-efficient clustering [22]	Avoiding unnecessary energy consumption or balancing energy consumption for mobile nodes in order to prolong the lifetime of mobile terminals and a network.
Dominating set based [22]	Finding a connected dominating set to reduce the number of nodes participating in route search or routing table maintenance.

The cost of clustering is a key issue to validate the effectiveness and scalability of a cluster structure [22]. Table 2.2 summarizes the costs of clustering.

Table 2.2: Costs of Clustering

Cost of Clustering	**Definition and Description**
Ripple effect of re-clustering [22]	Ripple effect indicates that the re-election of a single cluster head may affect the cluster structure of many other clusters and completely alter the cluster topology over the whole network.
Stationary assumption for cluster formation [22]	Mobile nodes must be assumed static in the cluster formation phase so that mobile nodes are able to obtain accurate neighbor information and cluster structure can be promised with specific attributes.
Explicit control message for clustering [22]	Clustering requires explicit clustering-related information exchanged between node pairs. Clusters cannot be formed or maintained by non-clustering-related messages, such as routing information or data packets.
Constant Computation Round [22]	Computation round is the number of rounds that a cluster formation procedure can be completed. The non-constant computation round of a clustering scheme indicates its unbounded time complexity..
Communication (message) complexity [22]	Communication complexity represents the total amount of clustering-related messages exchanged. For clustering schemes with ripple effect, the communication complexity for the re-clustering in the cluster maintenance phase may be the same as that in the cluster formation phase.

According to [13, 22], clustering in MANETs has the following advantages:

- Improving the spatial information reuse, throughput, scalability and power consumption.
- It helps to improve routing at the network layer by reducing the size of the routing tables.
- It decreases transmission overhead by updating the status of the environment after topological changes occur.
- It helps to aggregate topology information as the nodes of a cluster are smaller when compared to the nodes of an entire network. Enabling each node stores only a fraction of the total network information.
- It saves energy and communication bandwidth by minimizing the broadcast and multicast domain.

2.6 Summary

In this Chapter we have tried to review the basic characteristics of MANETs, security challenges faced by MANETs, approaches that taken to secure MANETs and several approaches to manage cryptographic keys in order to achieve secure communication in MANETs. The state of the art of MANET communication security is still evolving. The basic difference of MANETs from other types of networks is the fact that they rely on no fixed infrastructure, no base stations, access points, remote servers, etc. All network functions are performed by the nodes forming the network [5]. Each node performs the functionality of both host and router, relaying data to establish connectivity between source and destination nodes which are not directly within each other's transmission range. What is more, MANETs have limited resources, dynamic topology, distributed control, shared physical medium, and poor physical security. These features are constraints faced by researchers when designing key distribution system and security protocols for MANETs.

Chapter Three-Related Work

3.1 Overview

Recently, research papers have proposed different key management schemes for MANETs. Most of them are based on public-key cryptography. The basic idea is to distribute the CA's functionality to multiple nodes. There are also research papers that are based on the symmetric-key cryptography for securing MANETs. For instance, some symmetric key management schemes are proposed for sensor nodes that are assumed to be incapable of performing costly asymmetric cryptographic computations. Pairwise symmetric keys can be preloaded into these sensor nodes based on random key pre-distribution [16]. Another category of MANET key management research papers is group key based key management. These papers focus on group key generation, group management and distribution of group keys that will be used for secure group communications [8].

Key distribution is a central part of any secure communication, and is the weakest point of system security and protocol design. Hence it is receiving a lot of research attention. In this Chapter we will see several related key distribution protocols categorized by their specific classes as in [15].

3.2 Asymmetric Key Distribution Systems

The work by Yi and Kravets [24], distributes the certificate service to Mobile Certificate Authority (MOCA) nodes. MOCA nodes are chosen based on heterogeneity, if the nodes are physically more secure and computationally more powerful [24]. In cases where nodes are equally equipped, they are selected randomly from the network. These MOCA nodes hold partial key of the private key of the CA. When a node wants to send its public key to another node it acquirers the partial key from the MOCA nodes, reconstructs the private key, singes it's public key and sends it to the requesting node. MOCA is highly dependent on the routing algorithm since it needs the cached paths to the MOCA nodes [9]. The trust model of this scheme is a decentralized trust model since the functionality of CA is distributed to a subset of nodes in the MANET. A client that requires certification services sends a partial key request using cached routs to the MOCA nodes. Any MOCA that receives a partial key request responds with a partial key packet containing its partial signature. The client waits a fixed period of time for k such partial key replies. When the client collects k valid partial keys, the client can reconstruct the full signature and the certification request succeeds. If too few partial keys are received, the client's timer expires and the key request fails. On failure, the client can retry or proceed without the certification service [15, 24].

According to the result of a simulation, with 150 nodes, within 1000m * 1000m area, 30 MOCA nodes and without cached paths to the MOCA nodes (through the use of flooding) MOCA enjoyed 90% successful key delivery rate.

MOCA suffers from low key delivery rate when using unicast (without flooding) scoring 33.7% key delivery rate [24]. Even after MOCAs have been selected and deployed in the system, it is useless if clients cannot contact them and receive services [9]. The communication pattern between a client and k or more MOCA servers is one-to-one-to-many-to-one-to-one, which means that a node with certification need has to contact at least k MOCAs and receive at least k replies [9, 24], which is too much communication overhead for a single certification.

In addition to communication inefficiency, another critical question is how nodes can discover those paths to MOCA nodes securely since most secure routing protocols are based on the establishment of a key service in advance [9].

The work by Sanzgiri and Dahill [36], proposes an authenticated routing for mobile ad hoc networks. This scheme is composed of client nodes, server nodes, combiner nodes and an administrative authority which works as dealer [9, 36]. A dealer is the entity which provides the initial certificate to the mobile nodes. Client nodes are the normal user's mobile nodes that want to come in the MANET. Server nodes have the responsibility of generating the partial certificates and storing the certificates in directory structure, through which mobile nodes can request the certificates of other mobile nodes. Combiner nodes play the important task, which is to combine the partial certificates into the valid certificate [9].

Another work by Yi and Kravets [19], proposes a scheme that is extension of MOCA [24]. It combines the centralized trust and the fully distributed certificate chaining trust models. This scheme takes advantage of the positive aspects of two different trust systems. The basic idea proposed by the authors is to incorporate a TTP into the certificate graph. A single CA that has higher confidence value will represent a group of MOCA nodes. Here, the TTP is a virtual CA node that represents all nodes that comprise of the virtual CA [9]. Some authentication metrics, such as confidence value are introduced in order to "glue" two trusted systems. A node certified by a CA is trusted with a higher confidence level [19]. However, properly assigning confidence values is a challenging task.

According to the result of a simulation with 150 nodes, within 1000m * 1000m area and 30 MOCA nodes, with no certified nodes in the network, only 44% of all possible pairs of nodes can authenticate each other in pure certificate chaining [19]. As the fraction of certified nodes increases, the number of successful authentications for the composite approach increases

significantly. When every node in the network is certified the success ratio increases to 100% successful key delivery. With 10% of the nodes certified 69% successful key delivery rate is recorded. However certifying nodes in the network is one-to-one-to-many(TTP)-to-one-to-one communication [19]. Hence we can see that there is still a communication over head for a resource constrained environment like MANET.

The work by Luo and Lu [20], allows all nodes to act as MOCA nodes. This scheme broadcast signed partial keys to all mobile nodes. Each mobile node in the MANET updates their certificates periodically. The functionality of CA is distributed to all mobile nodes in MANET. The proposed solution takes a ticket-based approach. Each well-behaving node uses a certified ticket to participate in routing and packet forwarding. Nodes without valid tickets are classified as misbehaving. They will be denied from any network access, even if they move to other locations. Thus, misbehaving nodes are "isolated" and their damage to the mobile ad hoc network is confined to their locality.

The simulation experiment was done, with networks of sizes ranging from 50 to 100 nodes [20]. The ticketing success ratio ranged from 80 % to 87 % for ticket renewals. However, periodic certificate renewals are based on broadcasting of tickets in the entire MANET. This causes a communication overhead in the event of certificate renewal.

3.3 Symmetric Key Distribution Systems

The work by Chan and Pergi [32], uses a sensor nodes to establish a shared key. This work is symmetric key agreement scheme using unique secret key in a set of nodes .This model uses the concept of random key pre-distribution. As shown on Figure 3.1, in two dimension case, with each of the O(n) nodes, every mobile node shares a unique secret key in horizontal and vertical dimension. This scheme can be extended to three dimensions or any other dimension. Every pair of mobile node shares a common secret key with at least one or more intermediaries.

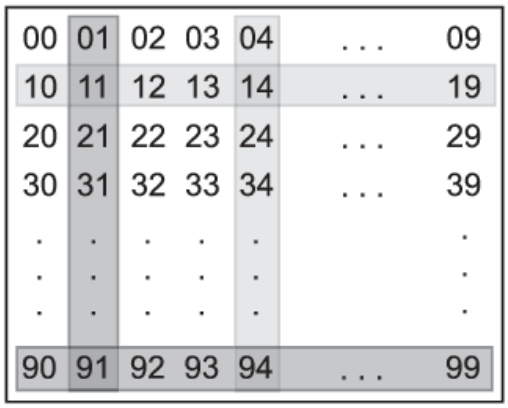


Figure 3.1: Example of PIKE Matrix for Key Pre-Distribution

The basic idea is to use sensor nodes as trusted intermediaries to establish shared keys between nodes. Each node shares (unique) pairwise key with every other node in the network. As can be seen in Figure 3.1, the keys are deployed such that, for any two nodes A and B, it is possible to find some node C in the network that shares a unique pair wise key with both A and B. Since the work pre-distributes unique pairwise keys, the established key is secure if node C (intermediary) is not compromised. Finding the intermediary is done through flooding and requires secure channel to return the key. Features of this model are good security services (with no compromised hosts), and fair scalability.

The simulation result shows that, even if 5% of the nodes are compromised, the pair wise secret keys exposed may increase exponentially.

The work by Anderson, Chan and Pergi [33], proposes a simple scheme where every mobile node participates equally to the key sharing process. In this work there is no need for collaborative effort because node acts as trusted components. These components broadcast their symmetric key to the entire network. Broadcasting symmetric key without establishing secure cannel in advance is highly insecure. The work focus on key distribution in commodity sensor networks where it does not assume a global passive adversary.

In symmetric key distribution systems, the common flaw is that the compromise of single key will expose the whole traffic between nodes sharing a key.

3.4 Group Based Key Distribution Systems

The work by Wu, Wu, and Dong [43], disclosed the SEGK model in 2008. Two multicast trees are constructed in the MANET for improving efficiency, and they are maintained in a parallel fashion to achieve fault tolerance. SEGK model calls one multicast tree as a blue tree and another multicast tree as red tree. The connection of the multicast trees is maintained by a coordinator. Computation and distribution of intermediate keying materials to all members is done by group coordinator through the use of underlying tree links. To make the common group key, each group member, i.e., mobile nodes in the MANET, participate in the making of a final common group key, which is updated periodically. This model presents the reliable double multicast tree formation and maintenance protocol, which ensures that it covers all group members.

The simulation for the work was implemented in Matlab. It was conducted in a 100 × 100 two dimension free-space by randomly allocating a given number of nodes in the range from 50 to 200. The work was evaluated using the computation cost and message cost. The computation cost was evaluated by analyzing the cost of processing group key membership.

When analyzing group key initialization and processing, with group membership rate of p = 10%, 20%, and 30%, computation time increases as the number of nodes increase. For example, when p = 10% and n = 100, computation time is 7.9 seconds, and 19.3 seconds for n = 200. The cost increases quickly with the increase of group membership rate [43]. The same trend was observed when message cost was evaluated. The message cost increased with increased group membership. The results indicate that SEGK has a scalability problem.

The work by Renuka and Shet [51], proposes a group key distribution system based on clustering. In this approach, the entire set of members in the network is divided into a number of subsets called clusters. Each cluster is headed by a cluster head. The layout of the network is as shown in Figure 3.2. The cluster head is similar to other members with same computational capability as other members in the network. Group key is used for encryption of messages exchanged in group communication within the cluster.

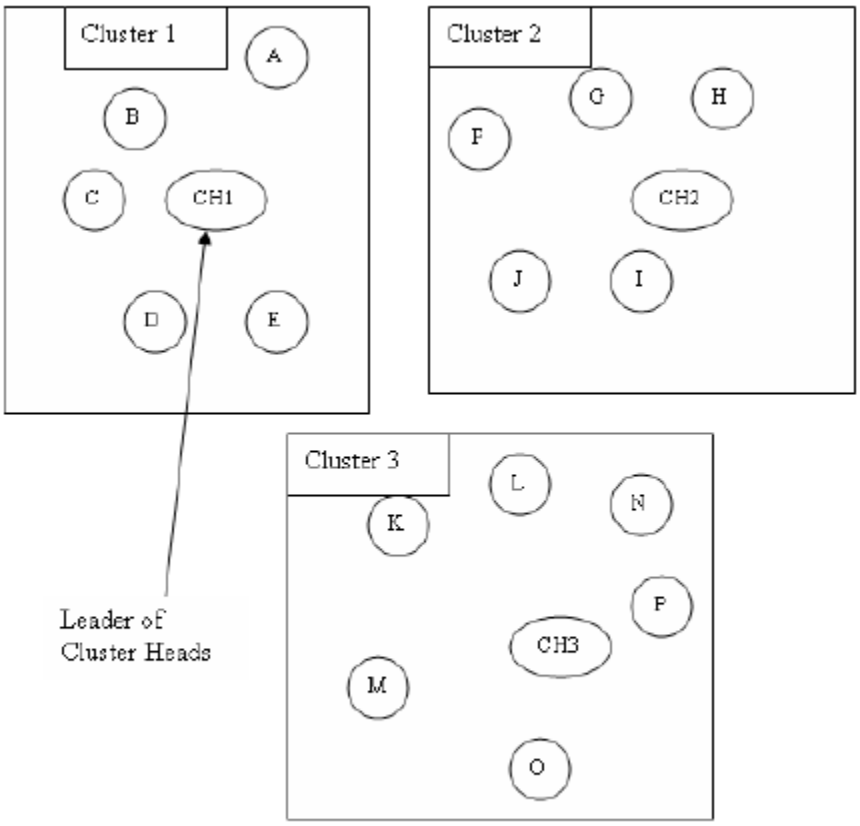


Figure 3.2: Cluster Based Group Key Management

The role of the cluster head is to a generate group key, and manage group key membership. Forward and backward security is maintained by changing the membership key each time the membership changes.

The experimental simulation for the work was conducted in 1000m * 1000m area, with a maximum of 200 nodes divided in 8 clusters. The result showed that the group keys where distributed to members in 480 ms.

The flaw with group key sharing is the compromise of a single node will render the group traffic visible to intruders.

3.5 Hybrid Key Distribution Systems

The work by Khdour and Aref [23] proposes a hybrid key management schema for zone based routing protocol in MANETs. It combines both symmetric and asymmetric cryptosystems where a zone is defined for each node and includes the nodes whose distance (in hops) is at most some predefined number. This distance is referred to as the zone radius. Each node uses symmetric key management inside its zone and asymmetric key management used for inter-zone security, without depending on clustering. The shortcoming of zone based schema is that it shares a symmetric key among nodes in the same hop zone, hence requiring a secure cannel to share keys. Exposure of the symmetric key may render traffic in hop zones visible.

3.6 Summary

Even if the reviewed papers propose key distribution schemas and try to address various issues, there are shortcomings that we aim to address in our work.

Symmetric key distribution suffers from security risk that arises from sharing of a single key for both encryption and decryption. It also suffers from efficiency problem when sharing keys because it tries to deliver pair-wise symmetric key for every node pair in the network [15], whereas most asymmetric key distribution schemas burden the CA nodes, especially if the CA nodes are involved in key generation. It also creates point of attack if CA is in charge of private-public key pair generation. Group based schemas also has security whole that arises from sharing of a single key by a group which may render the entire traffic of a group visible. Table 3.1 summarizes the related works.

Table 3.1: Summary of Related Works

No	Name	Approach		Drawback	Addressed Issues
		Cryptographic key managed	**Trust Model Used**		
1	Mobile Certificate Authority (MOCA) [24]	Asymmetric Key	Decentralized Model	-Establishment of secure path to MOCA nodes is mandatory -If the MOCA quorum to rebuild the PK of the CA is not complete the certification fails. This will result in lower key delivery rate	- By avoiding partial private key exchange with MOCA nodes, remove the necessity of establishing secure path to MOCA nodes -Through key caching and the use of cluster head nodes , reduce one-to-one-to-many(MOCA) one-to-one communication to one-to- one communication
2	Composite Key Management [23]	Asymmetric Key	Hybrid Model	-Building virtual CA (which is virtual TTP) takes group coordination that causes a communication over head	-Through key caching and the use of cluster head nodes , reduce one-to-one-to-many(TTP)-to-one-to-one communication to one-to- one communication

No	Name	Approach		Drawback	Addressed Issues
		Cryptographic key managed	**Trust Model Used**		
3	Secure Routing Protocol [36]	Asymmetric key	Decentralized	-Burdens on the dealer node -Low key delivery rate if cached paths to server and dealer nodes fail	- Reduce burden on key dealer nodes by allowing all nodes to act as CA nodes, given that they have cached the required key
4	Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks [20]	Asymmetric key	Decentralized	-Every node needs a ticket to participate in the network ,this results in communication overhead in dealing certificates	-Avoid broad casting of partial private key of CA nodes in an insecure channel to enhance security
5	Peer Intermediaries for Key Establishment [32]	Symmetric Key	Distributed Model	-The symmetric key pair database at sensor nodes creates a point of attack for intruders	-Avoid using pairwise symmetric key all together
6	Key Infection [33]	Symmetric Key	Decentralized Model	-Highly insecure because it is based on broadcasting of symmetric keys in an insecure channel	-Avoid using pairwise symmetric key all together
7	Simple and Efficient Group Key Management [43]	Group Key	Distributed Model	-Sharing of Group key among a group may render the network traffic visible	-Avoid using group key all together

No	Name	Approach		Drawback	Addressed Issues
		Cryptographic key managed	Trust Model Used		
8	Cluster Based Group Key Management [51]	Group Key	Decentralized Model	-Sharing of Group key among a group may render the network traffic visible	-Avoid using group key all together
9	Zone-Based key Management for MANETS [23]	Hybrid Key	Distributed Model	- sharing of a symmetric key among nodes in the same hop zone requires the establishment of secure channel -Sharing of a group private/public key pair will cause the traffic to be visible if a single node compromises the private key of the group	-Avoid using pairwise symmetric key all together

Our work aims to address the key distribution problem using the following approaches:

- Use of grouping (clustering) for resource efficiency because clustering improves communication and energy efficiency by limiting broadcasting and multi-cast range.
- Use of Asymmetric Cryptography where each node is responsible to generate its own private-public key pair, to enhance security by removing pair-wise/group-wise key sharing.
- To introduce public key caching and use decentralized trust model to ease burden of CA nodes, i.e., each node can act as CA node in the event of public key cache hit.

Chapter Four – The Proposed Key Distribution System

4.1 Overview

This work proposes a key distribution system that uses asymmetric cryptography (where each node is responsible to generate its own private-public key pair) and uses decentralized trust model with key caching capability. This we will show that will optimize resource usage while maximizing security. It does this through distributing the burden of key distribution to ordinary nodes that can act as certificate authorities, and the uses of asymmetric cryptography where the private/public key pairs are generated by hosts (security of the private key is assured locally) will reduce the chance of private keys being exposed to malicious hosts.

The proposed system is also supported through clustering. The idea behind clustering is to divide the network into a number of groups. This decreases the amount of control overhead. Inside a cluster, the one node that coordinates the cluster activities is the cluster head (CH). Ordinary nodes have direct access to this one cluster head and gateways. Gateways are nodes that can hear two or more cluster heads. Ordinary nodes send the key requests to their cluster head that either responds with signed key, or (if the destination is outside the cluster) forwards them to a gateway node to be delivered to the other clusters. In dense networks this significantly reduces the communication overhead, thus solving scalability and resource optimization problems in larger MANETs.

We have accordingly gone through rigorous steps to select and improve a clustering algorithm for our proposed key distribution system architecture. We have selected a clustering algorithm in two phases. In the initial phase of the selection process, we have compared several clustering algorithms proposed by different researchers through the survey done by the work on [22]. The survey on [22] uses costs of clustering summarized on Table 2.2 to compare the clustering algorithms. In the second phase of the selection process, we have experimentally compared the top three algorithms from selection process phase one by considering clustering issues not considered by the work [22].

In this Chapter the proposed key distribution system architecture along with the included sub systems, algorithms executed by the sub systems and selection process of the appropriate clustering algorithm are presented.

4.2 The Proposed key Distribution System Architecture

Figure 4.1 shows the proposed key distribution architecture. The architecture consists of two major components, the key distribution part and the clustering part. The key distribution consists of the key client, key server, cache builder, and the key cache subcomponent. The subcomponents will give their services based on the role of the node in the network. The role of a node in the network may be "ORDINARY","CLUSTER HEAD" or "GATEWAY". The clustering algorithm is part of the system that is responsible for grouping the nodes and assigning these nodes with a role by considering their capabilities. Since there are several clustering algorithms that are proposed by several researchers, we chose to select and customize a clustering algorithm from these works. Section 4.4 is dedicated to reporting the clustering algorithm selection processes and Section 4.3 will present a description each of the subcomponents of the key distribution component.

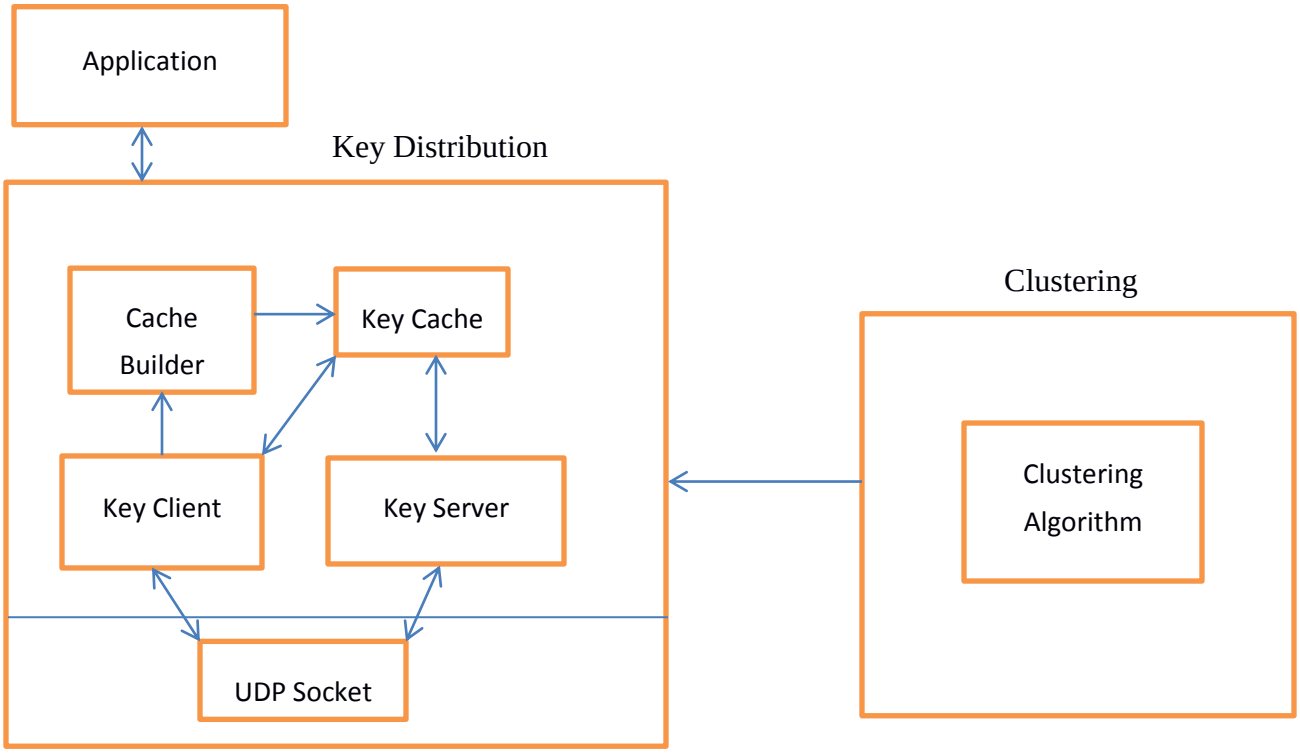


Figure 4.1: Proposed Key Distribution System Architecture

4.3 Key Distribution

The key distribution sub system is composed of five modules that work in collaboration with each other for successful key delivery. It is implemented over UDP socket to reduce resource usage, since UDP sockets use less resource than that of TCP sockets. Reliable packet delivery is handled by having the modules resend requests if responses haven't arrived within a specified timeout period.

The overall responsibilities of the key distribution component are:

- Initially generating private-public key pair of the host
- Introducing its public key to other hosts based on the role of the node
- Receiving key request from applications
- Retrieving and returning signed public key for the key requesting application
- Caching keys based on the caching rule
- Receiving key request from other nodes
- Retrieving, signing and returning key to the requesting node

The main components include key client, key server, cache builder and the cache itself. At initialization each node will generate its own public key, signs it and introduces it to its one hop neighbors and its cluster head, or all nodes under it if the node is a cluster head, through a single multicast message. Figure 4.2 shows the pseudo code for the initialization process.

```
//input: one_hop_list, ordinary_list, gateway_list, CH
//output:returns noting but introduces the public key of a node to other nodes
for each(node_n)
{
    if(node_n_state == "ORDINARY" || node_n_state == "GATEWAY")
    {
        multicast(key_node_n,one_hop_list,CH)//multicast my key to one hop list and cluster head
    }
    Else // the node is cluster head
    {
        multicast(key_node_n,ordinary_list, gateway_list)//multicast my key to ordinary list
    }
}
```

Figure 4.2: Pseudo Code of Initialization Algorithm for the Proposed Key Distribution System

4.3.1 Key Client

The key client is responsible for accepting key requests from applications, and retrieving and returning the retrieved key to the requesting application and the cache builder module. The key client algorithm will initially check whether the requested key is in the cache, and if the key is present in the cache it will retrieve and return it to the requesting application. If the key is not present in the cache it will send key request to its one hop neighbors and initiates a timeout. If two or more one hop neighbors do not return the requested key within the time out period, the key client will send a key request to the cluster head, and initiate another timeout. If the key does not arrive within the timeout period the key client will resend the key request to the cluster head. This process will repeat until the key arrives. When the key receives, it will be passed to the requesting application and the cache builder.

The algorithm executed by the key client module of which its pseudo code is shown in Figure 4.3 performs the following activities:

- Accepting key requests from applications.
- Returning the key for the key requesting application, if key is in the cache (lines 3 through 6 of the algorithm on Figure 4.3).
- Otherwise forward the key request to one hop neighbors, and initiate a timer for a timeout period (lines 9 through 10 of Figure 4.3 shows the multicasting of key request to one hop neighbors and initiating a timeout for replies to arrive).
- If less than two neighbors respond with certified keys with in timeout period, send the key request to cluster head and initiate a timer for timeout (lines 11 through 18 of Figure 4.3).
- If key hasn't been received with in timeout period, resend the key request to the cluster head (lines 11 through 18 of Figure 4.3).
- If the key is returned within the timeout period, return the key to the requesting application and also pass it to the cache builder (lines 22 through 44 of Figure 4.3 shows how the key client handles key replays. If the key replay comes from "CLUSTERHEAD", it will simply return the key to the requesting application and the cache builder otherwise it will wait for more than one replay from one hop neighbors.).

```
1.  //input : requeted_ip, onehop_nbr_list, global_request_list,key,key_cache
2.  //output: returns key to the requesting application and the cache builder module
3.  If (Check_Cache(requeted_ip)) // if key is present in the key cache
4.      {
5.       Return_key_to_app(key)//Return the key to the requesting applications
6.      }
7.  Else If (! Check_Cache(requeted_ip)) // if key is not in the cache
8.      {   insert_into ( global_request_list, requeted_ip)
9.          Multicast_key_request (onehop_nbr_list) // multicast key request to one hop neighbors
10.         Wait (timeout) // wait for a time out period
11.          If (! key_arrived) // if key hasn't arrived with in timeout period
12.             {
13.               While(!key_arrived) /if key hasn't arrived with in time out resend request
14.                  {
15.                   Send_key_req_to_CLUSTERHEAD(requeted_ip)//send key request to cluster head
16.                   Wait(timeout)
17.                   }
18.             }
19.       }
20.  When(key_arives)  // when key arrives
21.      {
22.       If(key_source == "CLUSTERHEAD") if the key came from cluster head node
23.         {
24.           If( chech_key_certeficate(key) )verify digital signature of the arrived key
25.             {
26.                   Return_key_to_app(key) // Return Key to key requesting application
27.                   Hand_key _to_cache _Builder(key)//pass key  to
28.                   Set(key_arived,true)
29.             }
30.         }
```

```
31.          Else if (key_source  != “CLUSTERHEAD”)
32.                {
33.                  If( chech_key_certeficate(key) )verify digital signature of the arrived key
34.                   {
35.                        count++ //more than one node need to sign the key if the key did not come from  CH
36.                              If (count  > 1)
37.                                   {
38.                                    Return_key_to_app(key) // Return Key to key requesting app
39.                                    Hand_key _to_cache _Builder(key)
40.                                    Set(key_arived,true)
41.                                     }
42.                    }
43.                }
44.          }
```

Figure 4.3: Pseudo Code of Key Clients Algorithm for the Proposed Key Distribution System

4.3.2 Key Server

The Key Server is a part of the key distribution system that accepts key requests from external nodes, check if key is present in the key cache and finally returning the key for the key requesting node if the key is in the cache. If the key is not in the cache it will forward the key request to other nodes based on the role of the node in the cluster. The key server also records the request it receives in the global request list to track the requests it receives so that the cache builder module will decide to cache the key (when the key arrives) if the request frequency is greater than one. As can be seen in the pseudo code of key server algorithm in Figure 4.4, the key server is responsible for:

- Accepting key requests from external nodes
- Returning the key for the key requesting node, if the key is in the key cache (lines 3 through 7 of Figure 4.4)
- If key not in the cache (lines 8 through 23 of Figure 4.4) :
    - If the node is cluster head, forward the key request to the gate way nodes.
    - If the node is gate way, forward the key request to neighboring clusters.
    - If node is ordinary do nothing.

```
1.    // Input: global_request_list, gateway_list, cluster_head_List, requeted_ip
2.    // Output: returns key for key request if key is in cache else forward the request accordingly
3.      If (Check_Cache(requeted_ip)) // if key is in the cache
4.            {
5.            Sign_key(key) //Digitally sign the key
6.            Return_key_requesting_node(key) //return the key to the requesting node
7.             }
8.        Else  // if key is not in the cache
9.           {
10.            insert_into ( global_request_list, requeted_ip) // register key request in the global key request list
11.           If (my_node_type == "ORDINARY")
12.                  {
13.                    // do nothing for resource conservation there is a timer on the client side
14.                   }
15.           Else If (my_node_type == "CLUSTERHEAD" )
16.                  {
17.                  Multicast_key_request(nodeip, gateway_list) // multicast request to gateway nodes
18.                   }
19.           Else If (my_node_type == "GATEWAY")
20.                  {
21.                  Multicast_key_request (nodeip, cluster_head_List) //multicast request to cluster head nodes
22.                   }
23.             }
```

Figure 4.4: Pseudo Code of Key Servers Algorithm for the Proposed Key Distribution System

4.3.3 Cache Builder and Key Cache

The cache builder module is responsible for accepting key from the key client and building a key cache based on the role of the node in the cluster. The key cache is simply a temporary storage of keys of which its data sources are keys from other nodes in the MANET. The size of the cache is carefully monitored by the cache builder by referring to the caching rule.

The cache builder will only make the decision to cache the keys it receives based on the nodes role in the MANET, and by referring to the global key request list. The global key request list is a data structure that is used to track key requests. It holds a list of ip addresses and access frequency of nodes of which either external nodes or local applications have previously

requested keys for. Figure 4.5 shows the structure of the global request list the cache builder refers to when deciding to cache received keys based on access frequency of previous key request.

```
struct global_request_list
  {
     int node_id;
     IPvXAddress nodeip;
     int access_frequency;
   };
```

Figure 4.5: Snippet of Data Structure for Global Request List

As can be seen in the pseudo code of the cache builder algorithm on Figure 4.6, whenever the cache builder receives a key from the key client or the node status changes, it operates under the following caching rules:

- If a node is a CLUSTER HEAD (lines 5 through 9 of Figure 4.6), it will cache keys received from any node in its own cluster and keys received from nodes of which their ip address exists in the global request list, and it's access frequency is greater than one (lines 20 through 30 of Figure 4.6).
- If a node is a GATEWAY (lines 10 through 14 of Figure 4.6), it will cache keys received from one hop neighbors, any cluster head it hears from and keys received from nodes of which their ip address exists in the global request list, and it's access frequency is greater than one (lines 20 through 30 of Figure 4.6).
- If a node is ORDINARY (lines 15 through 18 of Figure 4.6), it will cache keys received from one hop neighbors, its own cluster head and keys received from nodes of which their ip exists in the global request list, and it's access frequency is greater than one (lines 20 through 30 of Figure 4.6).

To avoid storage overhead, whenever the status of the node changes, the cache builder will clears the cache (line 4 of Figure 4.6).

Since the key cache holds certified public keys from different nodes, the cache will not create security whole. Even if the node gets compromised and the public keys are exposed to an attacker, and the attacker performs cryptanalysis on the exposed public keys to acquire the private keys, the attacker will not get patterns since the keys are from different sources.

```
1.  // Input: ordinary_List , gateway_list, onehoplist_nbr_list
2.  // Output: returns nothing but populates the key cache
3.      When (change_node _state)//whenever the node state changes
4.          {       clear_cache() //clear the content of the cache
5.              If (my_node_state =”CLUSTERHEAD”)//if node type is cluster head
6.                 {
7.                      Cache_key(ordinary_list) // cache the keys of all ordinary nodes
8.                      Cache_key(gateway_list)  // cache the keys of all gateway nodes
9.                 }
10.                Else If (my_node_type =”GATEWAY”)//if node type is gateway
11.                {
12.                     Cache_key(onehoplist_nbr_list)// cache keys of all one hop list
13.                     Cache_key(clusterhead_lst)//cache keys of all cluster head list
14.                }
15.                Else //if node type is ordinary
16.                {
17.                     Cache_key(onehop_list) cache keys of one hop list
18.                }
19.         }
20.     When (node_key_recived) // key arrives from key client module
21          {
22.           for each  I ∈ global_request_list       // for each element of the global request list
23.                 {
24.                   If (I.ip_addr  == node_key_recived.ip_addr) // if found in the global request list
25.                     {
26.                         If (i.accessfrequency > 1)// if access frequency of the received node is > 1
27.                             Cache_key(node_key_recived)
28.                      }
29.                 }
30.           }
```

Figure 4.6: Pseudo Code of Cache Builder Algorithm for the Proposed Key Distribution System

Why cache one hop neighbors? According to the caching rules, ordinary and gateway nodes cache keys of their one hop neighbors. One hop neighbors of nodes are not necessarily in the same cluster. The idea here is that, to have a key of a node be cached in a node of a neighboring cluster. This increases the probability of a cache hit for a key of a node in a neighboring clusters without causing too much communication overhead. Figure 4.7 shows an example of how one hop neighbors could be found in another cluster. If we increase the ordinary node cache hop radius to two hops, we will increase the communication over head with relatively the same effect of key infection.

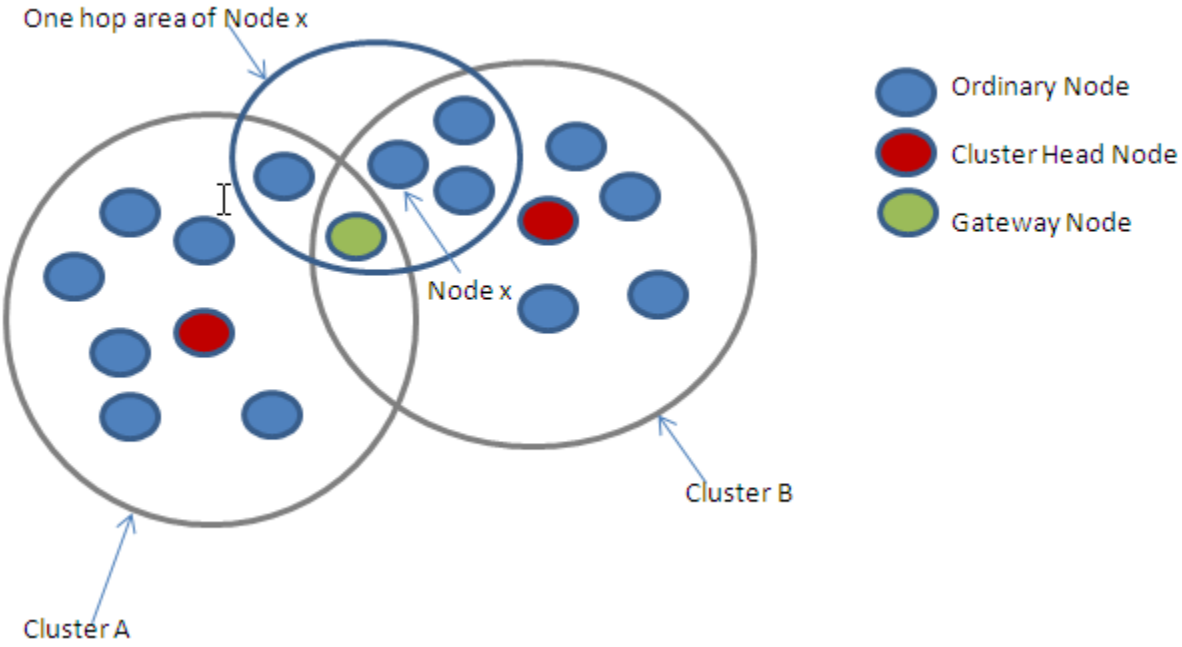


Figure 4.7: Inter Cluster Key Infection

4.4 Clustering Algorithm

The clustering algorithm will generate the cluster information needed by the key distribution in order to do its job. It provides a list of Ordinary, Gateway and Cluster Head nodes in a cluster. Utilizing the topology information gathered by the clustering algorithm, the key distribution system will give its services. The clustering algorithm for the proposed key distribution must be an algorithm with low communication overhead. The key distribution system will utilize the clusters created to deliver keys to nodes independent of the routing protocol used.

Key distribution is not the only service running on MANETs. QOS, Routing, Authentication, Load balancing, etc., are also services that run on MANETs. Each of which may have its own clustering schema. Thus the selected clustering algorithm should use fewer resources as possible in terms of bandwidth, storage and energy.

The following are the overall requirement from a clustering algorithm for the proposed key distribution system.

- Minimal communication in cluster formation
- Minimal communication in cluster maintenance
- Must incorporate cluster head selection
- Should not cause ripple effect when re-clustering
- Should not assume nodes are stationary during cluster formation
- Minimal computation rounds during cluster formation
- The resulting clusters should be stable
- In the resulting clusters, messages from nodes should reach their destination:
  - with minimum amount of hops to nodes of the same cluster
  - with minimum amount of hops to cluster heads of different clusters

The clustering algorithm selection has two phases. In the initial phase, we will compare several clustering algorithms proposed by several researchers through the survey done by the work in [22].

The work [22] considers the following costs of clustering:

- Ripple effect of re-clustering
- Stationary assumption for cluster formation
- Number of computation rounds
- Communication (message) complexity
- Explicit control message for clustering

Not all of our clustering algorithm requirements are addressed by the work in [22]. We aim to address some of the issues not addressed in the second phase of the clustering algorithm selection process. In the second phase of the selection process, the top three contending clustering algorithms from phase one, will be tested and compared experimentally to address clustering issues not addressed by the survey in [22].

Clustering issues tested in the experimental phase of the selection process are:

- Cluster stability
- Inter cluster communication efficiency
- Intra cluster communication efficiency

Selection processes of the appropriate clustering algorithm for the proposed key distribution system are presented in the subsequent Sections.

4.4.1 Selection Process - Phase One

In this phase, clustering algorithms will be measured and compared based on a survey done on the work [22]. According to a survey done in [22], clustering algorithms are classified as summarized in Table 2.1, and to measure and compare the clustering algorithms performance, the work in [22] uses the criteria summarized in Table 2.2.

Based on the criteria summarized in Table 2.2, ten clustering algorithms are measured and their scores are summarized in Table 4.1.

Table 4.1: Clustering Algorithm Comparison Result

Algorithm	**Type by objective**	**Explicit control message**	**Ripple effect of Re-clustering**	**Stationary assumption for cluster formation**	**Constant Computation rounds**	**Communication(message) complexity**
CDS[44]	Ds-based	Yes	Yes	Yes	Two	> $O(2V)$
WCDS[45]	Ds-based	Yes	N/A	Yes	N	> $O(2V)$
LCC[46]	Low maintenance	Yes	Yes	Yes	N	$O(2V)$
3hBAC[47]	Low maintenance	Yes	No	Yes	N	$O(3V)$
Lin[42]	Low maintenance	Yes	No	Yes	N	$O(V)$
PC[38]	Low maintenance	No	Yes	No	Unnecessary to consider	Zero

Algorithm	Type by objective	Explicit control message	Ripple effect of Re-clustering	Stationary assumption for cluster formation	Constant Computation rounds	Communication(message) complexity
MOBIC[48]	Mobility-aware	Yes	Yes	Yes	N	$O(V)$
DDCA[49]	Mobility-aware	Yes	No	No	Unnecessary to consider	N/A
Wu[22]	Energy-efficient	Yes	Yes	Yes	N	$O(2V)$
On-demand WCA[50]	Combined metrics	Yes	Yes	Yes	N	$O(2V)$

* V stands for number of nodes in the MANET

By considering the initial requirements of clustering algorithm for key distribution, we have assigned the following weights to the costs of clustering summarized on Table 2.2. The work in [22] assigns equal significance for costs of clustering summarized on Table 2.2.

Explicit control message for clustering: In the clustering algorithm, if a node explicitly sends cluster formation or maintenance messages, we have assigned it 0 point. On the other hand, if a node does not explicitly send cluster formation or maintenance messages, we have assigned it 1 pint. This is because we are aiming to minimize cluster formation and maintenance communication. Having explicit cluster control messages causes extra communication overhead.

Stationary assumption for cluster formation: If the clustering algorithm assumes that nodes are stationary during cluster formation we have given it 0 point. On the other hand if a node considers mobility of nodes during cluster formation we have given it 1 point.

Has cluster head selection: If a clustering algorithm incorporates cluster head selection we have given it 1 point. If a node does not incorporate cluster head selection we have given it 0 point.

Constant computation round: For our purpose, the clusters should be formed as soon as possible since acquiring keys is the initial stage of any application communication. We have given 1 point if a clustering algorithm finishes clustering the nodes in less or equal to 2 rounds, and 0 if N amount of computation round are needed.

Ripple effect of re-clustering: In a clustering algorithm**,** if reelection of a single cluster head causes re-clustering of the entire MANET we have given it 0 point; otherwise we have given 1 point.

Communication (message) complexity: If the worst case message complexity of the total amount of cluster related communication is zero, we have given 2 points. If it is less or equal to O(2V) we have given 1 point. If it is greater than O(2V) we have given 0 point.

N.B: N/A values are given zero points

According to the weights assigned, the reviewed algorithms scored the results as summarized in Table 4.2.

Table 4.2: Clustering Algorithm Comparison Phase One Result Summary

Algorithm	**Type by objective**	**Result**
CDS [44]	Ds-based	2
WCDS [45]	Ds-based	1
LCC[46]	Low maintenance	2
3hBAC[47]	Low maintenance	2
Lin [42]	Low maintenance	3
PC[38]	Low maintenance	5
MOBIC[48]	Mobility-aware	2

Algorithm	Type by objective	Result
DDCA[49]	Mobility-aware	4
Wu[22]	Energy-efficient	2
WCA[50]	Combined metrics	2

4.4.2 Selection Process - Phase Two

In this phase of the selection process, we tested for the stability, the inter cluster communication efficiency and the intra cluster communication efficiency of the top three contenders from cluster algorithm selection process phase one.

According to clustering algorithm comparison in phase one of the selection processes, the top three algorithms are:

- Passive Clustering Algorithm (PC) [38]
- Lin's Clustering Algorithm (LIN) [42]
- The Distributed Dynamic Clustering Algorithm (DDCA) [49]

In this phase the selected algorithms will be simulated using the Inet framework [37] on top of Omnet++ simulation tool, and compared experimentally by considering factors that haven't been addressed by the work on [22].

4.4.2.1 Clustering Algorithm Selection Experimental Setup

The simulated wireless ad hock host setup supports mobility, IPv4 and IPV6 network protocols, and TCP and UDP as transport protocols. Figure 4.8 shows a simulation of a MANET host used in our experimentation.

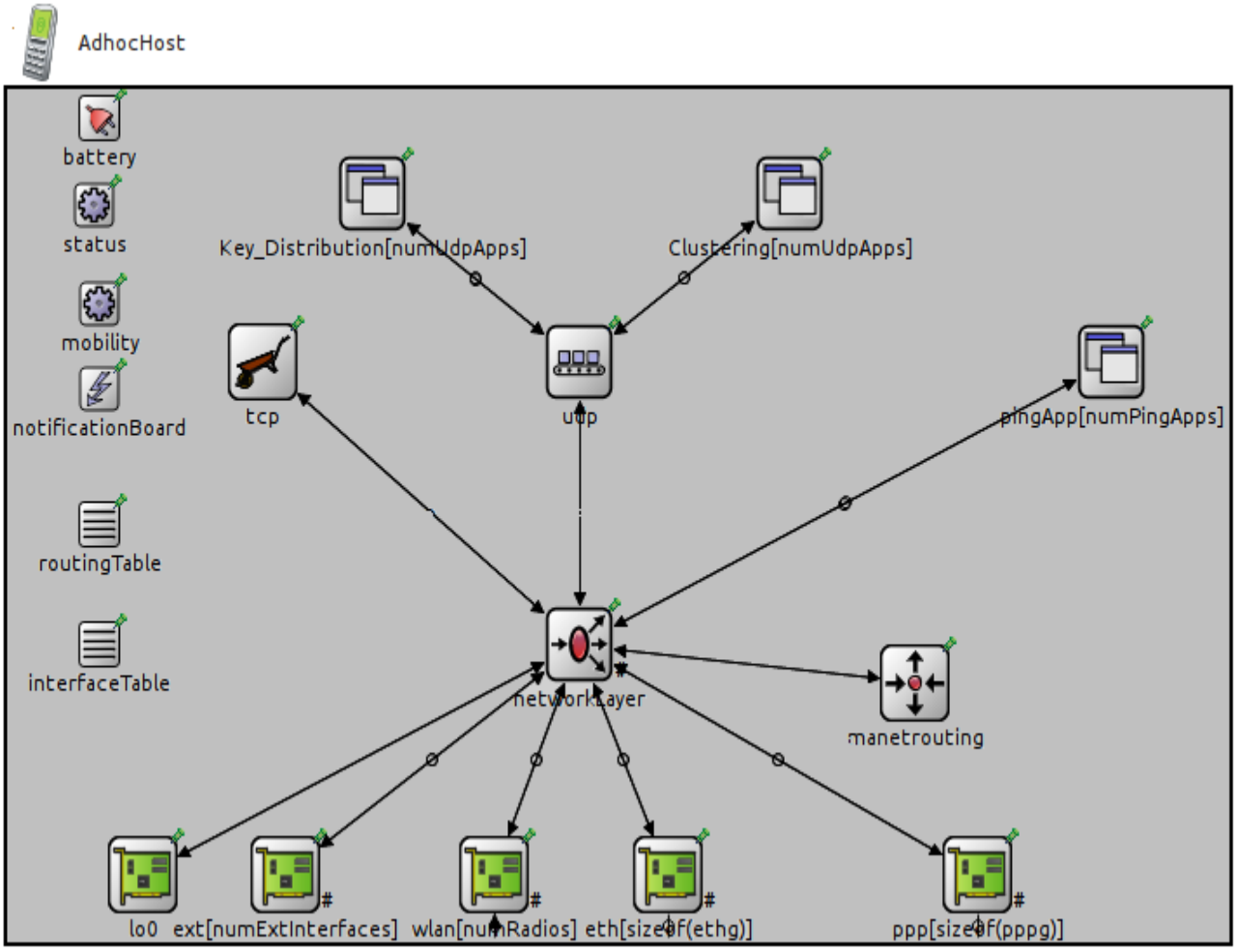


Figure 4.8: Ad hoc Host Setup

The ad hoc host on Figure 4.8 contains a wireless interface module, a loopback interface module, a MANET routing module, ethernet interface module, external interface module, a point to point interface module, a ping application module, a battery module, a network layer module and mobility module.

The wireless interface (wlan0) module simulates a wireless networking interface that receives and sends air frames, whereas the loopback interface (lo0) module responds to the loopback address 127.0.0.1. The battery module simulates a physical battery, and the mobility module simulates movement of the node. Ipv4 and Ipv6 protocols are handled by the network layer module. The ethernet (eth), point to point (ppp) and external (ext) interface modules simulate networking interfaces that can be used optionally for the purpose of simulating different MANET device classes. The ping application module is used for pinging MANET nodes in the network.

The MANET routing module simulates a selection of routing algorithms for MANETs. Since the proposed key distribution system is not dependent on the routing algorithm used, we have chosen to use the default routing algorithm provided by the Inet frame work [37], which is AODVUU.

The routing table, interface table, notification board and status modules are globally accessible data structures that hold packet routing information, list of available interfaces, notifications posted by different modules and status of modules respectively.

The clustering algorithms will be tested with 50 nodes, each of which will participate in cluster formation and sending of test packets to nodes destined within the same cluster and outside of the cluster. The clustering algorithms will be implemented using UDP (connection less) sockets to minimize resource footprint. Packet delivery assurance is handled by the clustering algorithms using timeouts.

All three algorithms will be evaluated based on:

a. Inter Custer Communication Efficiency

Inter cluster communication efficiency refers to how efficiently nodes communicate with nodes outside of their cluster using the clustering information supplied by the clustering algorithm:

- Measured by the number of hops a message takes to reach its destination between different cluster and the time the message takes to reach its destination.
- Based on the information collected by the clustering algorithm, each node will send 50 test packets destined outside of the cluster.

The following metrics will be computed based on how many hops a packet took to reach its destination and the end to end time delay of each packet to reach its destination.

Unit of measurement – Hop count (Among nodes in the different clusters)

- Average Hop Count
- Standard deviation of hop count to test for the stability of the cluster

Unit of measurement –End to End packet delivery delay (Among nodes in the different clusters)

- Average packet end to end packet delivery Delay
- Standard deviation of end to end packet delivery delay to test for the stability of the cluster

b. Intra Cluster Communication Efficiency

Intra cluster communication efficiency refers to how efficiently nodes in the same cluster communicate with each other, and with the cluster head with in the same cluster using the clustering information supplied by the clustering algorithm:

- Measured by the number of hops a message takes to reach its destination with nodes in the same cluster and the time the message takes to reach its destination.
- Based on the information collected by the clustering algorithm, each node will send 50 test packets destined to nodes inside of its cluster.

The following metrics will be computed based on how many hops the packet took to reach its destination and the end to end time delay of each packet to reach its destination.

Unit of measurement – Hop count (Among nodes in the same cluster)

- Average Hop Count
- Standard deviation of the hop count to test for the stability of the cluster

Unit of measurement – packet delivery delay (Among nodes in the same cluster)

- Average packet end to end packet delivery Delay
- Standard deviation of end to end packet delivery delay to test for the stability of the cluster

4.4.2.2 Data Collection

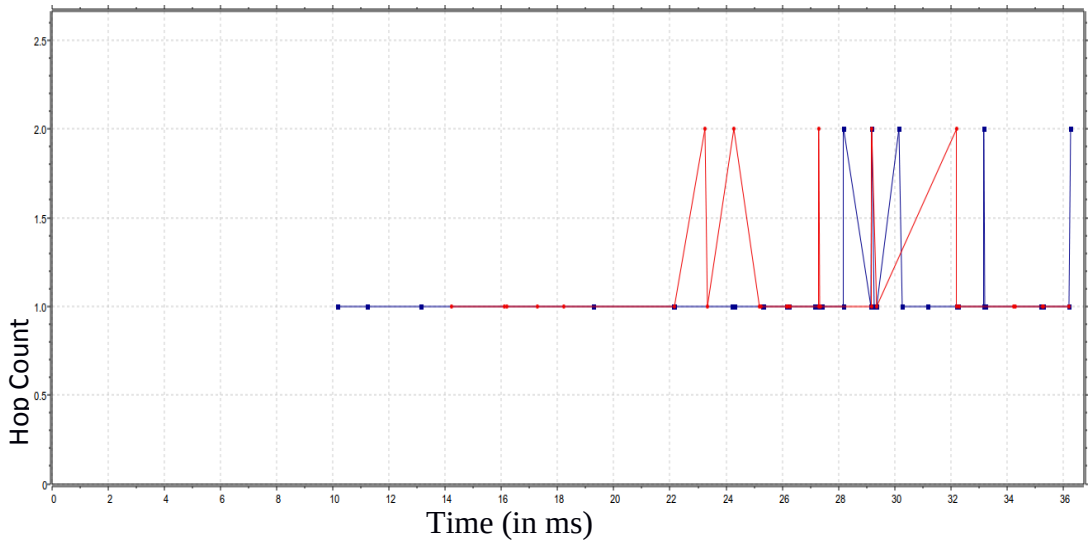
Omnet++ provides data collection and analysis API. Data vectors are recorded as the algorithms are running and are stored in a spread sheet file. The file can later be analyzed by analysis tools provided by Omnet++.

4.4.2.3 Results and Discussion

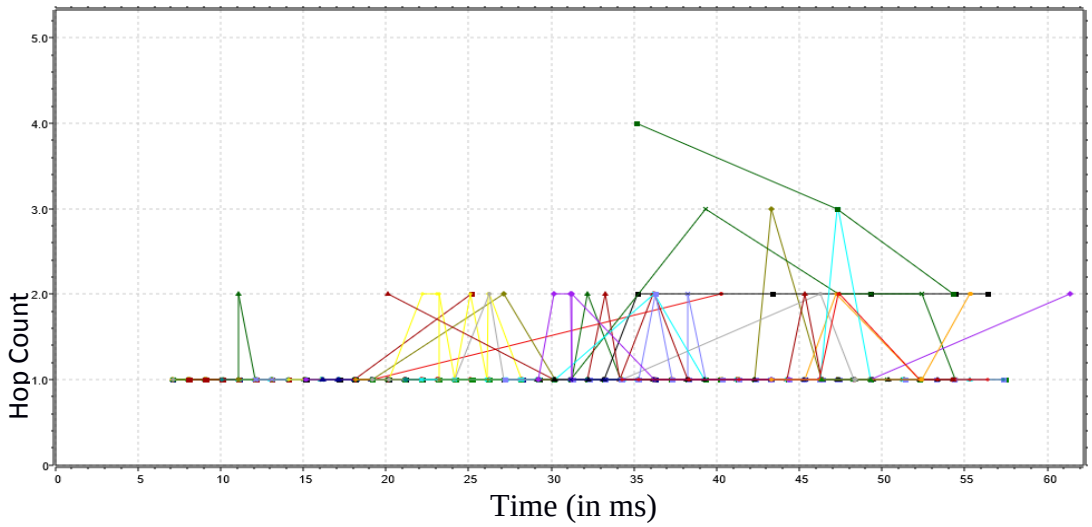
i. Passive Clustering (PC)

a. Intra Cluster Communication

The intra cluster communication efficiency of passive clustering was measured by having each node send test packets to the cluster head and the neighboring nodes in the same cluster. The hop count averaged 1.4 hops; whereas the end to end delay averaged 0.5 ms. The result indicates that most of the nodes in the same cluster are one hop away from each other. Figure 4.9 shows the hop count recorded in real time as test packets reach nodes in the same cluster. The different line colors on the graphs represent different MANET nodes.



a. Result recorded by two nodes when test packets are received



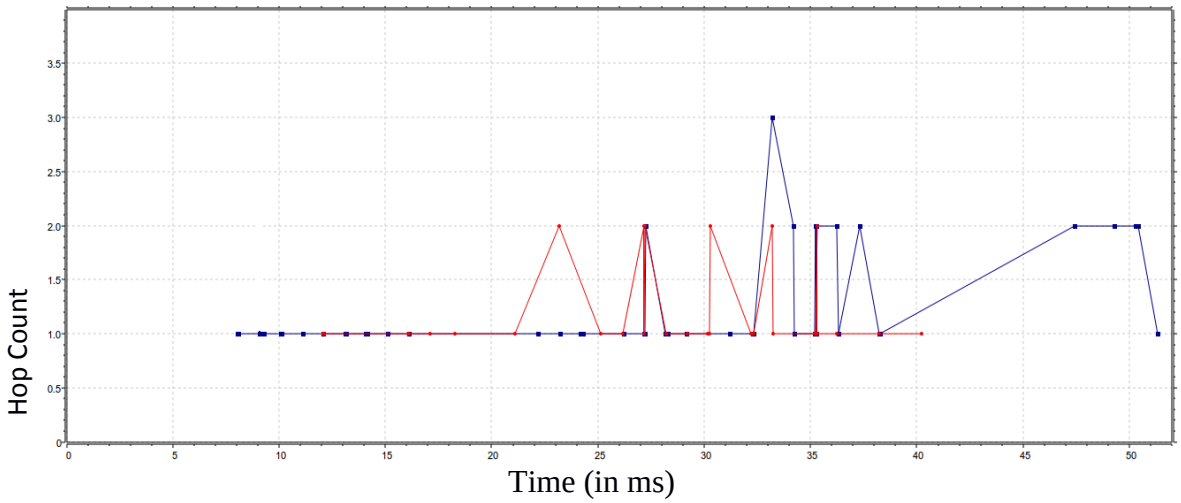
b. Result recorded by all nodes when test packets are received

Figure 4.9: Intra Cluster Hop Count Result for PC

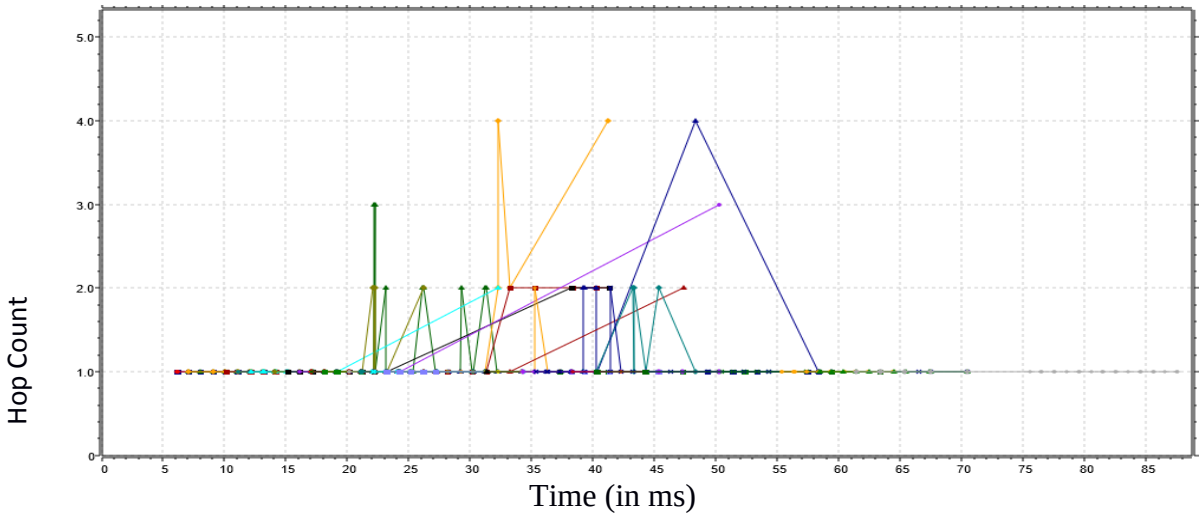
We can observe from Figure 4.9 that, even if most nodes in the same cluster are one hop away from each other, as the time goes on and nodes move around the hop count increased. The maximum result recorded is four hops. We also have noticed that there is a late start to send the test packets, but that is the time it takes for the algorithm to initialize and collect the necessary information.

b. Inter Cluster Communication

The inter cluster communication efficiency of passive clustering was measured by having each gateway node send test packets to cluster head nodes of neighboring clusters. The hop count averaged 1.85 hops; whereas the end to end delay averaged 0.8 ms. The average hop count result indicates that, most cluster head nodes of neighboring clusters are two hops away. Figure 4.10 shows the hop count recorded in real time as packets reach their destinations, which are cluster head nodes in different clusters. The different line colors on the graphs represent different MANET nodes.



a. Result recorded by two nodes when test packets are received



b. Result recorded by all nodes when test packets are received

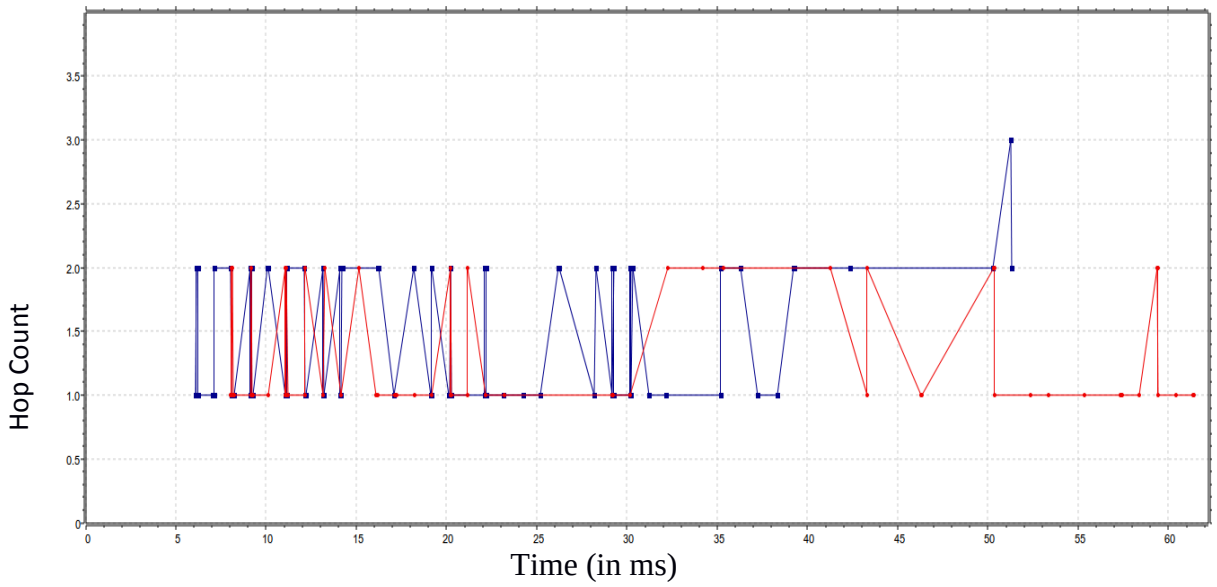
Figure 4.10: Inter Cluster Hop Count Result for PC

When observing the inter cluster communication hop count result in Figure 4.10, we observed that gateway nodes can access neighboring cluster head nodes in one hop, but some gateway nodes might be further apart from neighboring cluster heads, hence needing up to four hops to reach cluster head nodes in neighboring clusters. Again, we've have noticed that there is a late start to send the test packets, but that is the time it takes for the algorithm to initialize and collect the necessary information.

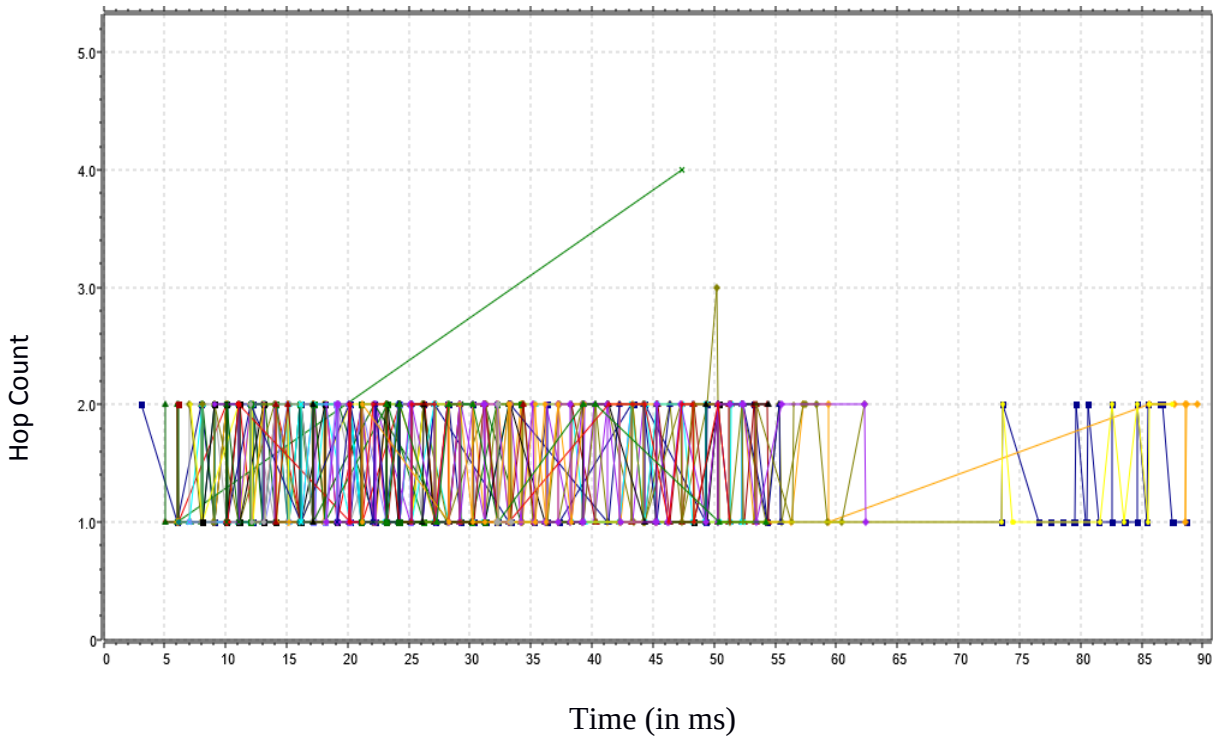
ii. The Distributed Dynamic Clustering Algorithm (DDCA)

a. Intra Cluster Communication

The intra cluster communication efficiency of DDCA was measured by having each node send test packets to the cluster head and the neighboring nodes in the same cluster. The hop count averaged 1.8 hops; whereas the end to end delay averaged 0.9 ms. The average hop count indicates that most of the nodes in the same cluster are two hops away from each other. Figure 4.11 shows the hop count recorded in real time as packets reach nodes in the same cluster. The different line colors on the graphs represent different MANET nodes.



a. Result recorded by two nodes when test packets are received



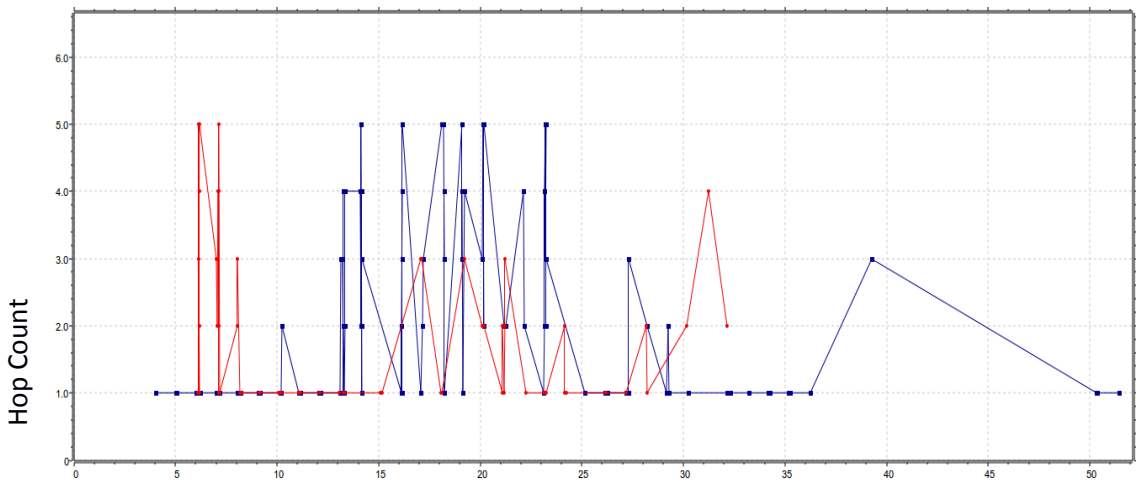
b. Result recorded by all nodes when test packets are received

Figure 4.11: Intra Cluster Hop Count Result for DDCA

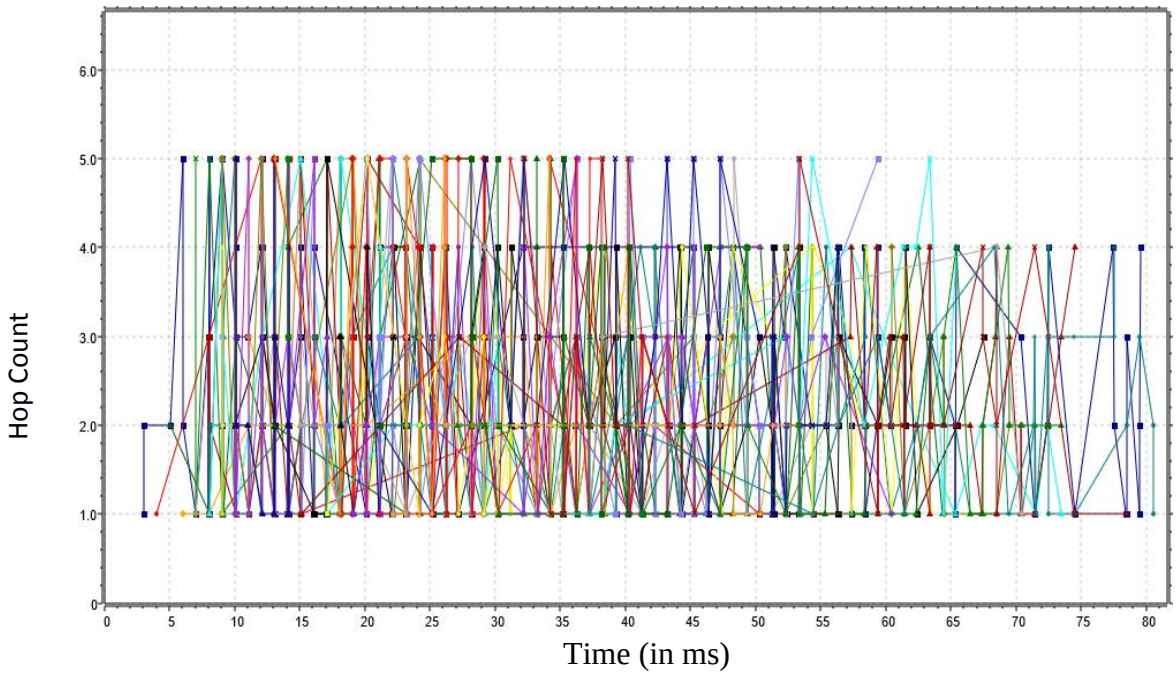
We can see from Figure 4.11 that nodes in the same cluster can reach each other and the cluster head in one or two hops, but we have recorded a single instance that has reached to four hops. We have noticed that there is a late start to send the test packets, but that is the time it takes for the algorithm to initialize and collect the necessary information.

b. Inter Cluster Communication

The inter cluster communication efficiency of DDCA was measured by having each node send test packets to cluster heads of neighboring clusters. The hop count averaged 2.6 hops; whereas the end to end delay averaged 1.2 ms. The average hop count result indicates that cluster head nodes of neighboring clusters are two or three hops away. Figure 4.12 shows the hop count recorded in real time as packets reach cluster head nodes in neighboring clusters. The different line colors on the graphs represent different MANET nodes.



a. Result recorded by two nodes when test packets are received



b. Result recorded by all nodes when test packets are received

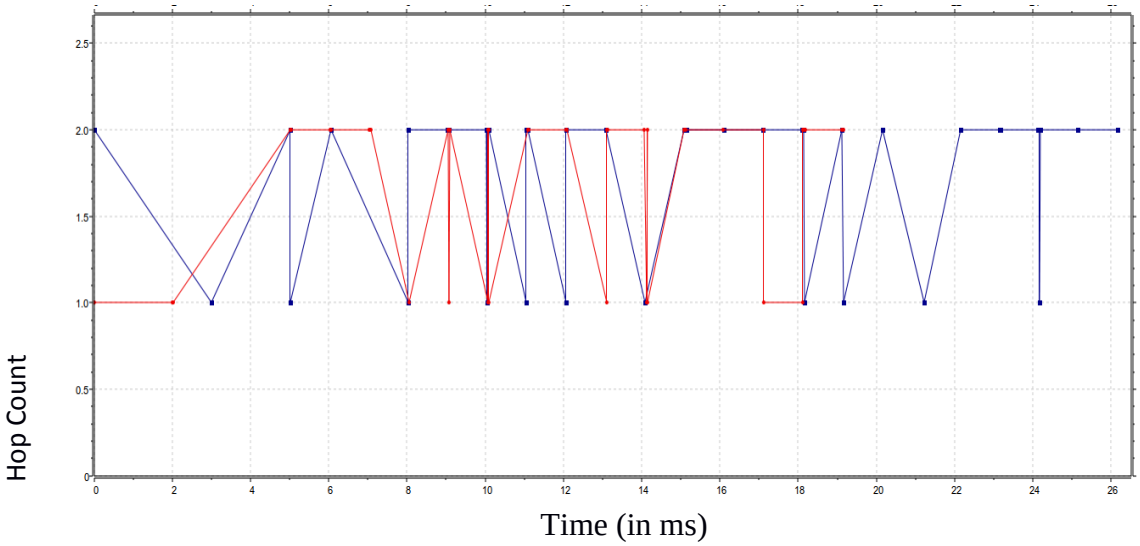
Figure 4.12: Inter Cluster Hop Count Result for DDCA

When looking at the hop count result for nodes in different clusters of DDCA on Figure 4.12, we observe that nodes are from one to five hops away from neighboring cluster head nodes. This is because DDCA is only concerned about the existence of the path regardless of hop count.

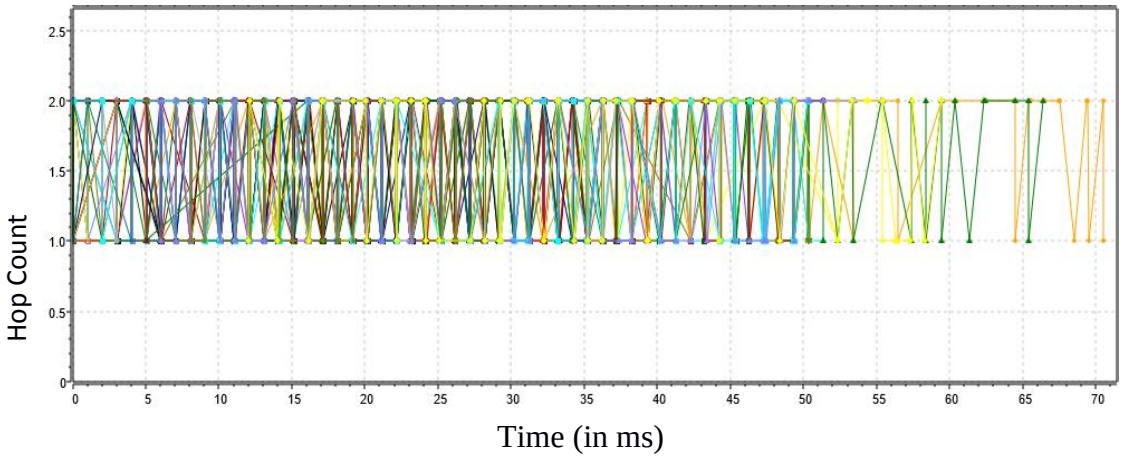
iii. Lin's Algorithm (Lin)

a. Intra Cluster Communication

The intra cluster communication efficiency of Lin was measured by having each node send test packets to the cluster head and the neighboring nodes in the same cluster. The hop count averaged 1.6 hops; whereas the end to end delay averaged 0.7 ms. The average hop count result indicates that most of the nodes in the same cluster are two hops away from each other. Figure 4.13 shows the hop count recorded in real time as packets reach nodes in the same cluster. The different line colors on the graphs represent different MANET nodes.



a. Result recorded by two nodes when test packets are received



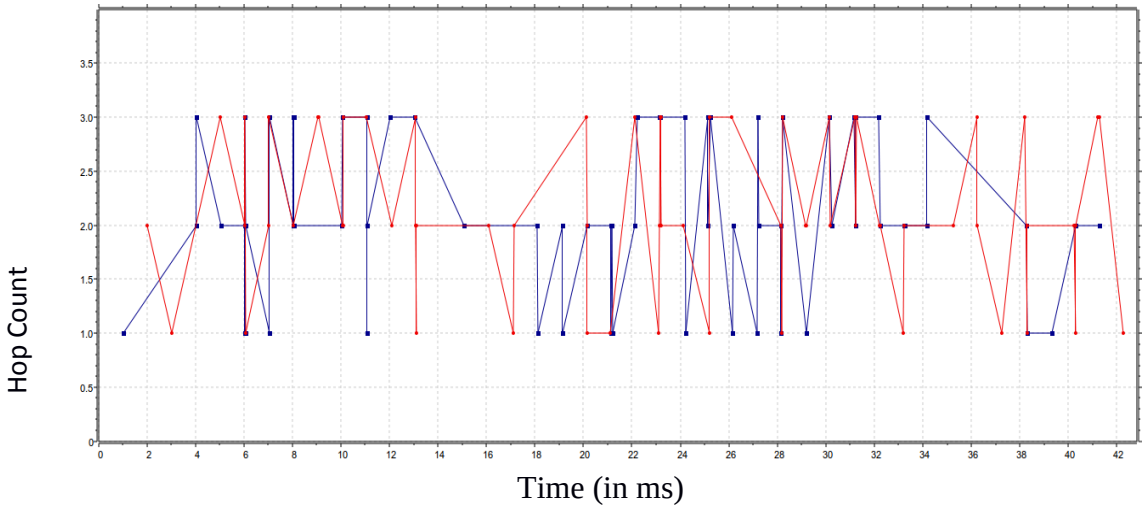
b. Result recorded by all nodes when test packets are received

Figure 4.13: Intra Cluster Hop Count Result for Lin

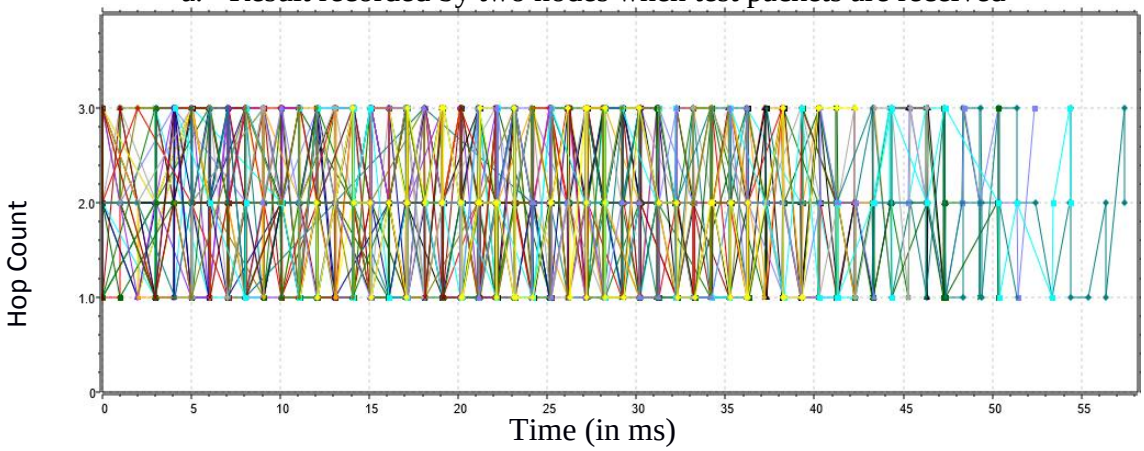
As shown in Figure 4.13, Lin's clustering algorithm has its nodes in the same cluster; one or two hops away from each other. This is because if any node in the same cluster is more than two hops away from the cluster head Lin initiates cluster maintenance. Hence no node can be more than two hops away from the cluster head.

b. Inter Cluster Communication

The inter cluster communication efficiency of Lin was measured by having each node send test packets to cluster heads of neighboring clusters. The hop count averaged 2.1 hops; whereas the end to end delay averaged 0.9 ms. The average hop count result indicates that most of the cluster head nodes in neighboring clusters are two hops away. Figure 4.14 shows the hop count recorded in real time as test packets reach cluster head nodes in neighboring clusters. The different line colors on the graphs represent different MANET nodes.



a. Result recorded by two nodes when test packets are received



b. Result recorded by all hosts when test packets are received

Figure 4.14: Inter Cluster Hop Count Result for Lin

As shown in Figure 4.14, cluster heads of neighboring clusters are reached between one and three hops. This is because Lin has strict cluster maintenance that avoids overlapping clusters.

Scores are given based on the algorithms rank on each category. The highest ranking algorithm will be given the highest point in each category and the points are added to form a total score. The highest point is given to the algorithm with the lowest average hop count and end to end delay and the lowest standard deviation of hop counts and end to end delay. Table 4.3 and Table 4.4 summarize the observed results.

Table 4.3: Intra Cluster Communication Efficiency Result Summary

Name	**Average Hop Count**		**Standard Divination hop count**		**Average packet delivery Delay (in ms)**		**Standard Deviation packet delivery delay (in ms)**		**Total Score**
	Result	**Score**	**Result**	**Score**	**Result**	**Score**	**Result**	**score**	
Passive Clustering (PC)	1.43	3	0.202567	3	0.52	3	0.4171	1	10
Lin's Algorithm	1.624	2	0.910171	2	0.754	2	0.21246	2	8
The Dynamic Clustering Algorithm (DDCA)	1.8	1	1.099284	1	0.978	1	0.19652	3	6

Table 4.4: Inter Cluster Communication Efficiency Result Summary

Name	**Average Hop Count**		**Standard Divination hop count**		**Average packet delivery Delay (in ms)**		**Standard Deviation packet delivery delay (in ms)**		**Total Score**
	Result	**Score**	**Result**	**Score**	**Result**	**Score**	**Result**	**score**	
Passive Clustering (PC)	1.85	3	0.55556	3	0.864	3	0.27839	2	11
Lin's Algorithm	2.123	2	1.03124	2	0.955	2	0.5426	1	7
The Dynamic Clustering Algorithm (DDCA)	2.602	1	1.24652	1	1.225	1	0.2254	3	6

In both inter and intra cluster communication efficiency, passive clustering algorithm scored the highest and hence is selected as the clustering algorithm for the proposed key distribution system.

In passive clustering algorithm, there is no explicit clustering control message exchange which will reduce the communication overhead. All clustering information exchange is by piggy backing other packets. In spite of the advantage of its passive nature, the way it selects its cluster head is not good enough for our purpose. We need nodes with better computing performance

and, longest remaining battery power to be a cluster head, but PC selects nodes with lowest node ID as cluster head.

We have modified the cluster head selection process of PC in such a way that, the algorithm will select nodes with better computing power, and longest remaining battery life as cluster head.

PCs cluster head selection is based on "First Declaration Wins" initially when a node has packets to send, it change its external state to cluster head and stamps cluster information (the nodes ID and state) to the outgoing packets and claims the role of cluster head [38]. After a successful transmission of the claim packet, every node in the sender's radio coverage can learn the presence of the cluster head through overhearing that packet. In the event that there is already a cluster head, conflicts are resolved using the lowest ID Competition. Declarations of multiple cluster head can happen due to the dynamic change of topology and packet delivery delay. In order to resolve those conflicts, a node that has the lowest ID among conflicted cluster head kills other declaration. If a cluster head has received a packet from the cluster head that has the lower ID than its own, then it gives up its role changes its state to ordinary. After one cluster head gives up the cluster head role and has no outgoing packet, then it send extra "CH Give-Up Message" which set "G" as "TRUE" in the IP option field [22, 38]. Appendix B shows the snippet of the cluster head resolution process, as is done by PC.

In our case, instead of using the ID of the node as cluster head conflict resolution metrics, we have used our own conflict resolution metrics obtained by adding the cpu capacity, memory capacity and remaining battery life. Nodes stamp this metrics instead of their ID on cluster information to be used as cluster head conflict resolution metrics. Appendix C shows the snippet of the modified cluster head resolution process. Thus nodes send "CH Give-Up Message" with set "G" as "TRUE", if they have lower computational power than that of the contending cluster head node.

Chapter Five-Experimentation and Evaluation

5.1 Overview

In this Chapter we will report on how we measured and compared the performance and efficiency of the proposed key distribution system against other previously proposed key distribution systems. Two key distribution systems are selected, of which we will simulate and compare their performance and efficiency against ours.

The selected key distributions are:

- Mobile Certificate Authority [24]
- Zone Based Key distribution for MANETs [23]

Efficiency can generally describe the extent to which time, effort or cost is well used for the intended task or purpose. It is often used with the specific purpose of relaying the capability of a specific application of effort to produce a specific outcome effectively with a minimum amount or quantity of waste, expense, or unnecessary effort [27]. For our purpose we measured the efficiency of key distribution systems by measuring the communication and memory effort required, for the distributing and retrieving of keys.

In the next Section we will discuss the metrics used to measure the efficiency and performance of the proposed key distribution system.

5.2 Metrics Used for Evaluation

a. Memory Efficiency

i. Average storage (memory) usage: Measured by the average memory that a node uses, when participating in the key distribution system.

ii. Standard deviation of storage (memory) usage: Measured by the standard deviation memory used by nodes when participating in the key distribution. This metrics is also used to measure the degree of distribution of key storage burden among nodes.

b. Communication Efficiency

To measure the communication efficiency of the key distribution we classified the communication costs to: initialization and maintenance communication cost and key retrieval cost.

i. Initialization communication complexity: is measured by the big O number of messages exchanged for initialization.

ii. Key retrieval communication cost: is measured by the average number of messages exchanged between nodes to retrieve a single key. This is done by measuring how many hops a key request travels until the key is found.

c. Overall Performance

i. Average end to end key delivery delay: The average end to end time delay for a key to be delivered to the requesting node

ii. Key delivery rate [24]**:** calculated by (*Number of successful key request*) *100

(*Number of total key request*)

5.3 Experimental Design and Setup

All three key distribution systems are developed on a simulated MANET environment, using Omnet++ simulation tool. We have used the Inet framework [37] on top of Omnet++ to simulate a MANET host. Detailed simulation parameters are presented in Appendix A. Figure 5.1 shows a simulation setup of a node configured to run the proposed key distribution system. The code for the network description file of the simulation setup, shown in Figure 5.1, is presented in Appendix D.

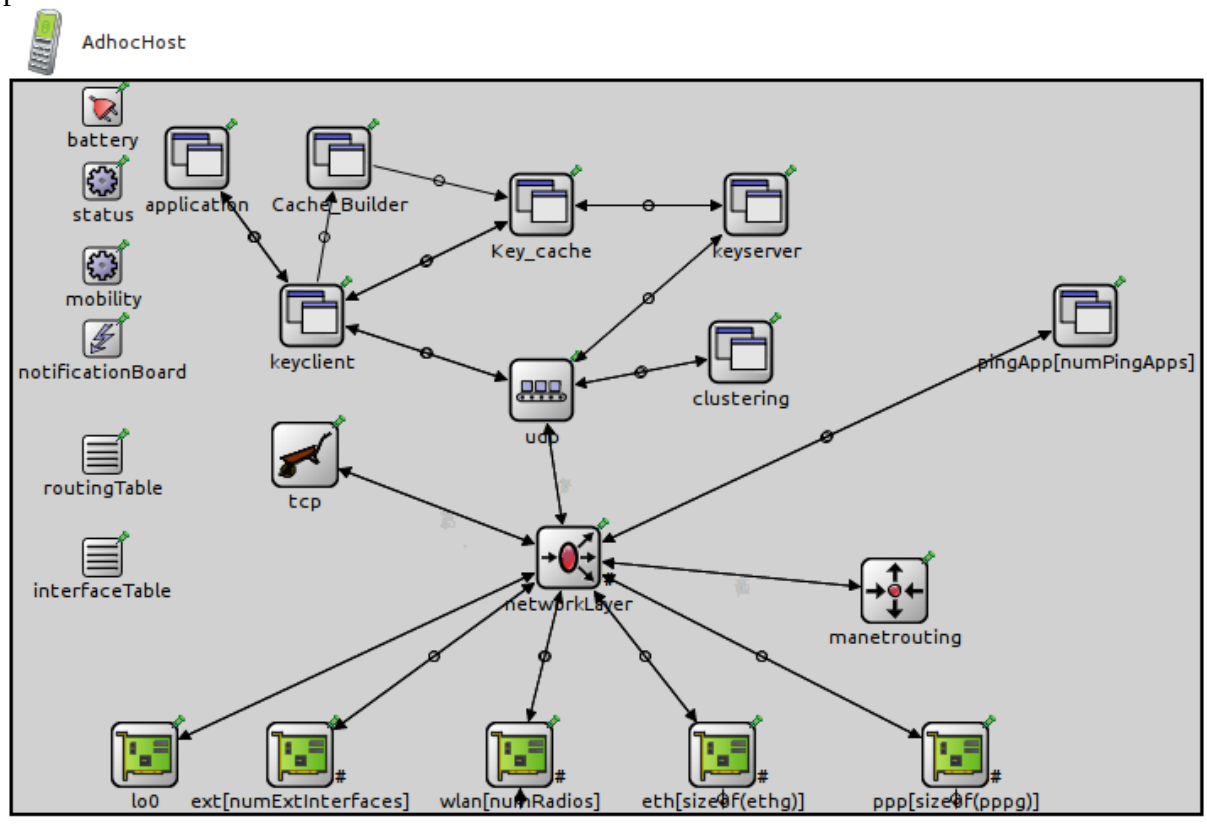


Figure 5.1: Single MANET Host Setup Running the Proposed Key Distribution System

Figure 5.2 shows a snap shot of a running simulation.

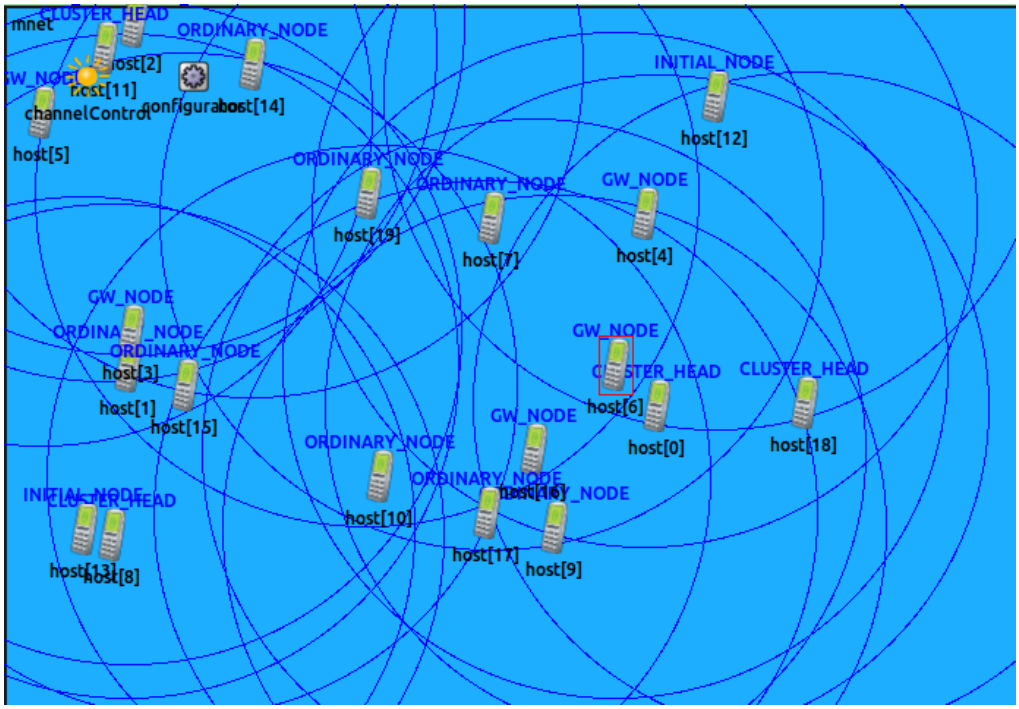


Figure 5.2: Snap Shot of a Running Simulation

Table 5.1 summarizes the parameters used in the simulation and evaluation of the proposed key distribution system.

Table 5.1: Simulation Parameters

Experimental Parameter	Value
Number of MANET Nodes	50
Area	200*m x* 200*m*
Number of key requests (Requests are sent to random nodes in the network, with random intervals)	10 by each node
Total simulation time	1000 Sec
Wireless Link Speed (bitrate)	2 Mbps
Wi-Fi standard	IEEE 802.11n 2009

Data collection was done as the simulation is running using Omnet++ data collection API. Omnet++ also provides analysis tools to analyze the collected data.

5.4 Result and Discussion

Table 5.2 summarizes the observed results based on the metrics explained earlier in Section 5.2.

Table 5.2: Experimentation Result Summary

Key Distribution system	**Memory Efficiency**		**Communication efficiency**		**Over All Performance**	
	Average memory usage (in bytes)	**Standard deviation of memory Usage (in bytes)**	**Initialization Communication Complexity**	**Key Retrieval Communicati-on Cost (in Average hop counts)**	**Average End to End key Delivery Delay (in ms)**	**Key Delivery Rate**
Mobile Certificate Authority[24]	2376	65.36	N/A	10.42	15.3	67.36%
Zone Based Key distribution for MANETs[23]	1816	59.96	O(n)	8.66	14.13	75.69%
Resource Efficient Key distribution for MANETs	2244	46.36	O(n)	5.22	8.06	71.33%

We were able to observe in the memory usage aspect of the experiment that Zone based key distribution performed the best, our proposed key distribution came second and MOCA came third. But when we look at the standard deviation of the memory usage the proposed key distribution performed the best. This indicates that, due to key caching, there is a better burden sharing among hosts, which will reduce bottlenecks that may occur during key requests.

When it comes to communication efficiency, MOCAs initialization phase complexity could not be computed because, MOCAs mobile certificate authority selection and partial key distribution phase is highly dynamic and depends on several variables. Whereas both zone based and the proposed key distribution complete their initialization phases at O(n) message complexity (where n is the number of nodes in the network). In key retrieval communication cost our proposed key distribution system performed the best because, clustering and caching improved the number of hops that a key request travels until the desired key is retrieved.

In the end to end key delivery delay aspect of the experiment our key distribution system performed the best because, signed keys are cached by nodes using our caching rule, hence the keys were found quicker. This resulted lower end to end key delivery delay.

When analyzing the result of the key delivery rate experiment, we have observed that zone based performed the best. This is because zone based uses a shared symmetric key for nodes within three hop zone radius. MOCA performed the worst because there are times in which mobile certificate authority nodes fail to deliver partial keys because they may be out of range or packets get dropped. Our key distribution system trade off on flooding, by restricting key requests to cluster head and gateway nodes to reduce communication burden, therefore came second.

Chapter Six-Conclusion and Future Work

6.1 Conclusion

In this work, we have proposed a key distribution system for MANETs that uses asymmetric cryptography and aims to maximize the efficiency of resource utilization by the use of clustering and key caching. We used asymmetric cryptography because the private key of a node will not travel through the network hence maximizing security. But this creates a burden on CA node because the CA node is responsible for signing and delivering public keys to requesting nodes. This is where the distributed trust model comes in; the distributed trust model will resolve the burden of the CA by allowing ordinary nodes to act as the CAs. This is done through key caching. By allowing nodes to cache keys, they are enabled to help the CA.

We have proposed to achieve efficiency through the use of clustering and key caching. We have managed to select a clustering algorithm with low resource consumption called passive clustering. We have done this by comparing ten clustering algorithms through literature, and measuring intra and inter cluster communication costs and the stability of the top three contending clustering algorithms through experimentation. What makes passive clustering ideal for key distribution in MANETs is that, without generating its own traffic, PC dynamically partitions the network in clusters interconnected by gateway nodes. PC removes control overhead of background cluster formation and maintenance using the concept of "passive" cluster construction. It collects clustering information using the existing traffic.

We have also proposed a key client, key server and cache builder algorithms that will utilize the cluster information generated by PC clustering algorithm to reliably deliver keys. The algorithm has both key server and key client systems. This allows a host to act as the CA. Nodes will cache keys based on what type of node they are, i.e., if a node is a cluster head (the CH node is the node with higher computation power and remaining battery life) it will cache a larger amount of public keys. Ordinary nodes will request a key from the CH node, only when they are unable to find a key with in one hop zone around them.

The simulation results show that the proposed key distribution can deliver certified keys quicker and using less communication. MANETs operate under different scenarios and applications need different level of security based on their nature. Some applications with sensitive content such as military MANET applications may require secure end to end cryptography. Other applications such as conference room slide sharing applications may not be so sensitive. The proposed key distribution can be used where secure end to end crypto systems are needed in spite of bandwidth constraints.

6.2 Future Work

The proposed key distribution system allows MANETs asymmetric end to end cryptography in an efficient manner. This gives applications the option of sharing of other group key distribution schemas in non-security sensitive environments, and to use asymmetric end to end cryptography security sensitive environments. Hence we have the following recommendation for future work.

- For the proposed key distribution system to be implemented as a library with its own API, so that applications with end to end security requirements could use it.
- An authentication system to be integrated to the key distribution system using digital signatures. An authentication system can easily be integrated to the key distribution system since the key distribution system is already dealing with signed digital certificates. These certificates can easily be used to identify hosts for an authentication system.
- Work on node trust rating system, so that the trust on the source node of cached certificate could be rated.

References

[1] J. Macker and S. Corson, "IETF Mobile Ad-Hoc Networking Working Group Charter", http://www.ietf.org/dyn/wg/charter/manet-charter, March 26, 2014

[2] P. Goyal , V. Parmar and R. Rishi , "MANET: Vulnerabilities, Challenges, Attacks, Application , IJCEM International Journal of Computational Engineering & Management", Vol. 11, January 2011

[3] J. Hoebeke, I. Moerman, B. Dhoedt and P. Demeester , "An Overview of Mobile Ad HocNetworks: Applications and Challenges", Global Journal of Computer Science and Technology Vol. 7 Issue 2 (Ver 1.0), April 2009

[4] A. Aarti and S. Tyagi, "Study of MANET: Characteristics, Challenges, Application and Security" Attacks International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013 ISSN: 2277 128X

[5] X. Wang, "Mobile Ad hoc Networks Applications", InTech Publishing, 2011.

[6] J. YI and P. Nantes ," A Survey on the Applications of MANET" , InTech Publishing Feb 2008

[7] R. Dalal ,Y. Singh and M. Khari , "A Review on Key Management Schemes in MANET ", International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.4, July 2012

[8] N. Vimala and R. Balasubramaniam "Distributed Key Management Scheme for Mobile Ad-Hoc Network-A Survey ",Global Journal of Computer Science and Technology Vol. 10 Issue 2 (Ver 1.0), April 2010

[9] B. wu, j. wu and M. cardei , "A survey of key management in mobile ad hoc networks", Handbook of Research on Wireless Security May 2010

[10] M. Salah and M. Bouali , "On the Performance of Group Key Management Protocols in MANETs" , Joint Conference on Security in Network Architectures and Information Systems (SAR-SSI'07), Annecy : France 2007

[11] P. Guptar and P. R. Kumar, "The Capacity of Wireless Net-works," IEEE Trans. Info. Theory, vol.-IT 46.2, Mar. 2000, pp 245-.262

[12] J. Kaavi , "Group Key Distribution in Ad-Hoc Networks Using MIKEY" , Helsinki University of Technology 2009

[13] A. B. MaDonald and T. F. Znati, "A Mobility-based Frame Work for Adaptive Clustering in Wireless Ad Hoc Networks," IEEE JSAC, vol. 17, Aug. 1999., pp. 1466–1487

[14] D.SuganyaDevi and G.Padmavathi , "Secure Multicast Key Distribution for Mobile Adhoc Networks" , (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 2, 2010

[15] A. Chan , "Distributed Symmetric Key Management for Mobile Ad hoc Networks", IEEE INFOCOM'04, vol.4, pp. 2414- 2424 2010

[16] C. Schwingenschlog and S. Eichler "Certificate based Key Management for Secure Communications in Ad Hoc Networks" University of technology in Munich Institute of communication network, 2007

[17] L. Zhou. and Z. Haas. "Securing Ad Hoc Networks", IEEE Network Magazine, Vol. 13,pp. 24-30

[18] L. Raju and R. Akbani. "Mobile Ad HocNetworks Security" (2004). Annual review of communications, Vol 58, pp. 625-628

[19] S. Yi. and R. Kravets "Composite Key Management for Ad Hoc Networks".MobiQuitous'04, pp. 52-61

[20] H. Luo and S. Lu. (2004). "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks", IEEE/ACM Transactions on Networking". Vol. 12, pp. 1049-1063.

[21] P. Rai and S. Singh "A Review of 'MANET's Security Aspects and Challenges", IJCA Special Issue on "Mobile Ad-hoc Networks" MANETs, 2010

[22] Y. Yu and H. J. Chong, "Survey of Clustering Schemes for MANETs" IEEE COMMUNICATIONS the Electronic Magazine of Original Peer-Reviewed Survey Articles 2009

[23] T. Khdour and A. Aref , "A Hybrid Schema Zone Based Key Management for MANETs" Journal of Theoretical and Applied Information Technology 31st January 2012. Vol. 35 No.2

[24] S. Yi and R. Kravets , "MOCA : Mobile Certificate Authority for Wireless Ad Hoc Networks" University of Illinois at Urbana-Champaign 2010

[25] S. Kuldeep, K. Neha and M. Prabhakar "An Overview Of security Problems in MANET "Anna University, Jul 2011

[26] R. Rajaraman, "Topology Control and Routing in Ad Hoc Net-works: A Survey," ACM SIGACT News, vol. 33, no. 2, June 2002, pp. 66–73.

[27] [Online]. Available: "http://dictionary.reference.com/browse/computer_efficiency" [Accessed july15 2014]

[28] H. Bakht , "Applications of mobile ad-hoc networks" School of Computing and Mathematical Sciences Liverpool John Moores University 2007

[29] A. Agarwa, S. Aggarwal, M. Agrawal and N. Batra, "Emergency Relief and Crowd Dispersal using MANET Protocols" , Nov 16 2014

[30] Y. N. Lien, H. C. Jang, and T. C. Tsai, "A MANET Based Emergency Communication System for Catastrophic Natural Disasters " Department of Computer Science, National Chengchi

University Taipei, Taiwan, R.O.C. 2010

[31] H. Chin J. Y. Lien and T. Tsai "Rescue Information System for Earthquake Disasters Based on MANET Emergency Communication Platform" Department of Computer Science, National Chengchi University No. 64 Taipei Taiwan, R.O.C 2012

[32] H. Chan and A. Pergi, "PIKE: Peer Intermediaries for Key Establishment", Carnegie Mellon University, 2005

[33] R. Anderson, H. Chan and A. Pergi, "Key Infection: Smart Trust for Smart Dust" Carnegie Mellon University, 2004

[34] J. Wu and T. Anantvalee, "A Survey on Intrusion Detection in Mobile Ad Hoc Networks", Department of Computer Science and Engineering Florida Atlantic University, Boca Raton, FL 33428, 2008

[35] Y. Zhang, W. Lee, and Y. Huang, "Intrusion Detection Techniques forMobile Wireless Networks," ACM/Kluwer Wireless Networks Journal (ACM WINET), Vol. 9, No. 5, September 2003

[36] K. Sanzgiri and B. Dahill. "A Secure Routing Protocol for Ad Hoc Networks", Dept. of Computer Science, University of California, Santa Barbara, CA 93106

[37] "INET Framework for OMNeT++ Manual" , Generated on June 21, 2012

[38] Y. Yi and M. Gerla, "Pasive Clustering Algorithm in MANETS" Internet Engineering Task Force May 2002

[39] [Online].Available:"http://ec.europa.eu/information_society/apps/projects/factsheet/index.cfm?project_ref=297251" [Accessed 13 November 2014]

[40] [Online]. Available: "home.howstuffworks.com/smart-home.htm" [Accessed 15 November 2014]

[41] [Online].Available:"http://www.computer.org/portal/web/computingnow/pervasivecomputing" [Accessed 15 November 2014]

[42] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wire-less Networks," IEEE JSAC, vol. 15, Sept. 1997, pp. 1265–75

[43] B. Wu, J. Wu and Y. Dong, "An efficient group key management scheme for mobile adhoc network", International Journal and Networks, Vol. 2008

[44] J. Wu and H. L. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," Proc. 3rd Int'l. Wksp. Discrete Algorithms and Methods for Mobile Comp. and Commun, 1999

[45] B. Das and V. Bharghavan, "Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets," in Proc. IEEE ICC'97, June 1997, pp. 376–80

[46] C. Chiang et al., “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” in Proc. IEEE SICON’97, 1997

[47] J. Y. Yu and P. H. J. Chong, “3hBAC (3-hop between Adjacent Cluster heads): a Novel No overlapping Clustering Algorithm for Mobile Ad Hoc Networks,” in Proc. IEEE Pacrim’03, vol.Aug. 2003, pp. 318–21

[48] P. Basu, N. Khan, and T. D. C. Little, “A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks,” in Proc. IEEE ICDC-SW’01, Apr. 2001, pp. 413–18

[49] B. McDonald and F. Znati, “Design and Performance of Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks,” in Proc. 34th Annual Simulation Symp. Apr. 2001, pp. 27–35

[50] M. Chatterjee, S. K. Das, and D. Turgut, “An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks,” in Proc. IEEE Globecom 2000, pp. 1697–701

[51] A. Renuka and K.C.Shet ,”Cluster Based Group Key Management in Mobile Ad hoc Networks”, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.4, April 2009

Appendix A. Default Node Configuration for Simulation

```
network = mnet
#record-eventlog = true
#eventlog-message-detail-pattern = *:(not declaredOn(cMessage) and not declaredOn(cNamedObject) and not declaredOn(cObject))
num-rngs = 3
**.numRadios = 1
**.numUdpApps = 1
**.messageLength = 1B
**.sendInterval = 10ms
**.multicastInterface = "wlan0"
**.mobility.rng-0 = 1
**.wlan[*].mac.rng-0 = 2
#debug-on-errors = true
tkenv-plugin-path = ../../../etc/plugins
**.constraintAreaMinX = 0m
**.constraintAreaMinY = 0m
**.constraintAreaMinZ = 0m
**.constraintAreaMaxX = 200m
**.constraintAreaMaxY = 200m
**.constraintAreaMaxZ = 0m
**.debug = true
**.coreDebug = false
**.host*.**.channelNumber = 0
# channel physical parameters
*.channelControl.carrierFrequency = 2.4GHz
*.channelControl.pMax = 2.0mW
*.channelControl.sat = -110dBm
*.channelControl.alpha = 2
*.channelControl.numChannels = 1
# mobility
```

```
**.host*.mobilityType = "MassMobility"
**.host*.mobility.initFromDisplayString = false
**.host*.mobility.changeInterval = truncnormal(2s, 0.5s)
**.host*.mobility.changeAngleBy = normal(0deg, 30deg)
**.host*.mobility.speed = truncnormal(20mps, 8mps)
**.host*.mobility.updateInterval = 100ms
**.wlan[*].bitrate = 2Mbps
**.wlan[*].mgmt.frameCapacity = 10
**.wlan[*].mac.address = "auto"
**.wlan[*].mac.maxQueueSize = 14
**.wlan[*].mac.rtsThresholdBytes = 3000B
**.wlan[*].mac.retryLimit = 7
**.wlan[*].mac.cwMinData = 7
**.wlan[*].mac.cwMinBroadcast = 31
**.wlan[*].radio.transmitterPower = 2mW
**.wlan[*].radio.thermalNoise = -110dBm
**.wlan[*].radio.sensitivity = -85dBm
**.wlan[*].radio.pathLossAlpha = 2
**.wlan[*].radio.snirThreshold = 4dB
[Config Ping1]
*.numHosts = 50
```

Appendix B. Snippet of Cluster Head Conflict Resolution for PC

```
 if (strcmp(revived-packet->getState(),"CLUSTER_HEAD")==0 && pki->getG() != 1)
   {
    if(getParentModule()->getIndex() > revived-packet ->getNodeid() && strcmp(my-state,"CLUSTER_HEAD") ==
0)
          {
             socket.setOutputGate(gate("udpOut"));
             setSocketOptions();
             IPvXAddress destAddr = chooseDestAddr();
             clusteringmsg *payload = new clusteringmsg(msgName);
             payload->setG(1);
             payload->setNodeid(getParentModule()->getIndex());
             emit(sentPkSignal, payload);
             socket.sendTo(payload, destAddr, destPort);
             EV << "send cluster head give up";
          }
   }
```

Appendix C. Snippet of Modified Cluster Head Conflict Resolution for PC

```
 if (strcmp(revived-packet->getState(),"CLUSTER_HEAD")==0 && pki->getG() != 1)
    {
      if((my-conflict-metrics <  revived-packet ->my-conflict- metrics) && strcmp(mystate,"CLUSTER_HEAD") ==
0)
              {
               socket.setOutputGate(gate("udpOut"));
               setSocketOptions();
               IPvXAddress destAddr = chooseDestAddr();
               clusteringmsg *payload = new clusteringmsg(msgName);
               payload->setG(1);
               payload->setNodeid(getParentModule()->getIndex());
              emit(sentPkSignal, payload);
              socket.sendTo(payload, destAddr, destPort);
              EV << "send cluster head give up";
              }
     }
```

Appendix D. Network Description File Code for the Proposed Key Distribution System

```
module MANET_Host extends NodeBase
{
  parameters:
     @display("i=device/cellphone");
      wlan[*].mgmtType = default("Ieee80211MgmtAdhoc");  // use adhoc management
      string routingProtocol default("AODVUU ");  // used mobile routing protocol. see: inet.networklayer.
      IPForward = default(true);
    submodules:
          keyclient: keyclient {
          parameters:
              @display("p=177,141,row,60");
       }
       keyserver: keyserver {
          parameters:
              @display("p=306,97,row,60");
       }
       Cache_Builder: keycache {
          parameters:
              @display("p=212,58,row,60");
       }
          application: sampleapp {
          @display("p=134,19,row,60");
       }
       Key_cache: keycache {
          parameters:
              @display("p=313,74,row,60");
       }
      manetrouting: <routingProtocol> like IManetRouting if routingProtocol != "" {
          @display("p=522,307");
       }
       tcp: <tcpType> like ITCP if hasTcp {
          parameters:
              @display("p=173,225");
```

```
    }
    clustering: UDPBasicApp {
        parameters:
            @display("p=453,77,row,60");
    }
    udp: <udpType> like IUDP if hasUdp {
        parameters:
            @display("p=313,186");
    }
    pingApp[numPingApps]: <default("PingApp")> like IPingApp {
        parameters:
            @display("p=635,141,row,60");
    }
  connections :
    tcp.ipOut --> networkLayer.tcpIn if hasTcp;
    tcp.ipIn <-- networkLayer.tcpOut if hasTcp;
    for i=0..numUdpApps-1 {
        clustering.udpOut --> udp.appIn++;
        clustering.udpIn <-- udp.appOut++;
        keyclient.udpOut --> udp.appIn++;
        keyclient.udpIn <-- udp.appOut++;
        keyserver.udpOut --> udp.appIn++;
        keyserver.udpIn <-- udp.appOut++;
        keyclient.keycacheout --> Cache_Builder.keyclentin;
        keyclient.keyci <-- Key_cache.keyco;
        Key_cache.keyci <-- keyclient.keyco;
        keyclient.appout --> application.sampleappin;
        application.sampleappout --> keyclient.appin;
        keyserver.keyserverin <-- Key_cache.keyserverout;
        keyserver.keyserverout --> Key_cache.keyserverin;
        Cache_Builder.keyco --> Key_cache.keybi;
            }
    udp.ipOut --> networkLayer.udpIn if hasUdp;
    udp.ipIn <-- networkLayer.udpOut if hasUdp;
    for i=0..numPingApps-1 {
        networkLayer.pingOut++ --> pingApp[i].pingIn;
        networkLayer.pingIn++ <-- pingApp[i].pingOut;
    }
}
```

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials for the thesis have been duly acknowledged.

_____

Dawit Wubshet

This thesis has been submitted for examination with my approval as an advisor.

_____

Dejene Ejigu (PhD)