

**DEVELOPMENT OF A PRIVACY AIDED TRUST ROUTING
ALGORITHM BASED ON SOCIAL SIMILARITY IN OPPORTUNISTIC
NETWORK**

BY

SULEIMAN AHMED

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AHMADU BELLO UNIVERSITY ZARIA

NIGERIA

MARCH, 2019

**DEVELOPMENT OF A PRIVACY AIDED TRUST ROUTING
ALGORITHM BASED ON SOCIAL SIMILARITY IN OPPORTUNISTIC
NETWORK**

By

**SULEIMAN AHMED, PGD (ABU) 2016
P16EGCP8012**

**A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES,
AHMADU BELLO UNIVERSITY, ZARIA**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A
MASTER DEGREE IN COMPUTER ENGINEERING**

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

MARCH, 201

DECLARATION

I hereby declare that this dissertation entitled “Development of Privacy Aided Trust Routing Algorithm Based on Social Similarity in Opportunistic Network” was carried out by me in the Department of Computer Engineering, Ahmadu Bello University as part of the Requirements for the Award of Degree of Master of Science in Computer Engineering. This research to the best of my knowledge has never been submitted anywhere for the award of any degree or diploma. The information derived from literature has been duly acknowledged in the text and a list of references provided.

Suleiman AHMED

Signature

Date

CERTIFICATION

This Dissertation entitled “DEVELOPMENT OF A PRIVACY AIDED TRUST ROUTING ALGORITHM BASED ON SOCIAL SIMILARITY IN OPPORTUNISTIC NETWORK” meets the regulations governing the award of degree of Master of Science (M.Sc.) in Computer Engineering of the Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

Dr. E. A. Adedokun
(Chairman, Supervisory Committee)

Signature

Date

Dr. M. B. Adulrazaq
(Member, Supervisory Committee)

Signature

Date

Prof. M. B. Mu’azu
(Head of Department)

Signature

Date

Prof. S. Z. Abubakar
(Dean, School of Postgraduate Studies)

Signature

Date

DEDICATION

This dissertation is dedicated to my parents, Malam Ahmed Ishaqu and Malama Adisetu Umoru, my beloved wife, Malama Safiyyah Umar Suleiman, and my lovely children: AbdurRahman, Aishah, and Nana Khadijah.

ACKNOWLEDGEMENT

All praises and adoration are due to Allah (S.W.T), the most gracious, the most merciful. May His peace and blessings be upon His prophet, Muhammad (S.A.W), his household, his companions all those who follow his noble path till Armageddon.

I wish to express my utmost gratitude to my supervisor, role model and chairman of my supervisory committee, Dr. E. A. Adedokun for his time, immense contributions and valuable guidance towards the success of this work. Indeed, the completion of this work could not have been possible without your consistent participation and assistance. I am proud to have you as my supervisor. My thanks also go to my co-supervisor Dr. M. B. Abdulrazaq for his valuable input and encouragement throughout the stages of the work. I really appreciate the training I received from you.

I acknowledge and appreciate the contribution of all the lecturers of department of Computer Engineering, Ahmadu Bello University, especially: Prof. M. B. Mu'azu whose mentorship is unquantifiable, Dr. H. T. Sikiru, Dr. I. J. Umoh, Dr. I. A. Bello, Dr. A. T. Salawudeen, Dr. Y. Basira, Dr. B. O. Sadiq, Engr. I. Yusuf, Engr. M. Shehu, and Engr. M. A. Zainab.

My deep appreciation goes to the members of Computer and Control Research Group for their contributions and constructive criticism during all the stages of my presentations, especially, Engr. Zaharudeen Haruna, Engr. Ajayi Ore-Ofe, and Engr. A. Abdulfatah. I also thank all my class mates and colleagues. Wish you all the best.

My gratitude also goes to my siblings, family members and friends whose advice, love, care and prayers have kept me strong and sound throughout my studies. I thank you all.

Ahmed Suleiman

March, 2019.

ABSTRACT

This research work presents the development of a privacy aided trust routing model using social similarity in an Opportunistic Network (OppNet). OppNet is a delay tolerant network where link is highly unpredictable. It is an ideal solution in situations where the deployment of wired and conventional wireless networks is difficult or practically impossible. However, OppNet is faced with several challenges. Most of these challenges are related to routing, buffer management or security. The most promising amongst the several approaches by researchers at addressing these challenges are the ones that explored social behavior of human beings. The mobile nature of OppNets however brought to the fore the need for a more efficient and effective security model. This work, therefore, intends to further strengthen privacy in an OppNet. This is done by developing a community-based privacy preservation scheme using a symmetric cryptographic model, based on a differential equation of a curvi-circular function by applying Galois theory. The key distribution is made possible by partial application of onion routing scheme. The developed privacy scheme is incorporated into trust routing based on social similarity (TRSS) to become Privacy Aided TRSS (PATRSS). This approach improves the security in terms of preservation of privacy of messages and that of the relay nodes without compromising other network performance indicators. Using metrics, such as delivery ratio, delivery cost and average trust performance, PATRSS is simulated on Opportunistic Network Environment (ONE) simulator. PATRSS outperformed TRSS by 7.8%, 30.7% and 9.4% in terms of delivery ratio, delivery cost and average trust value respectively. Finally, the TRSS and PATRSS which are originally routed on spray and wait routing algorithm are implemented on Epidemic, PRoPHET and MaxProp. Using numbers of messages duplicated, those aborted and those delivered as metrics, the results are presented to clearly show the network performance, the improvements, and the costs. These demonstrate the effectiveness and the efficiency of PATRSS.

TABLE OF CONTENTS

DECLARATION	i
CERTIFICATION.....	ii
DEDICATION	iii
ACKNOWLEDGEMENT.....	iv
ABSTRACT.....	v
LIST OF FIGURES	ix
LIST OF TABLES	x
LIST OF ABBREVIATIONS	xi
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background to Study.....	1
1.2 Significance of Research.....	3
1.3 Problem Statement	4
1.4 Aim and Objectives	5
1.5 Motivation	6
1.6 Thesis Organization.....	6
CHAPTER TWO	7
LITERATURE REVIEW	7
2.1 Introduction	7
2.2 Review of Fundamental Concepts.....	7
2.2.1 Opportunistic Network	7
2.2.2 Classification of opportunist network routing protocols	16
2.3 Opportunistic Network Environment Simulator	22
2.4 Network Security and Privacy.....	24
2.4.1 Onion Routing	27
2.5 Fundamentals of Galois Theorem	28
2.5.1 Groups and group theory	28
2.5.2 Symmetric groups.....	29
2.5.3 Rings and polynomials	31
2.5.4 The galois theorem	31

2.6	Review of Similar Works	32
CHAPTER THREE		42
METHODS AND MATERIALS		42
3.1	Introduction	42
3.2	Methodology	42
3.3	Datasets and Trust Model.....	44
3.4	Trust Routing.....	45
3.5	Development of Privacy Preservation.....	46
3.5.1	Development of message privacy.....	46
3.5.2	Development of relay node's privacy.....	50
3.6	Incorporation of Privacy into trust routing.....	51
3.7	Modelling The Datasets	54
3.8	Model Test Case.....	54
3.8.1	Simulations.....	55
3.8.2	Validating of PATRSS in epidemic PROPHET and MaxProp.....	57
CHAPTER FOUR.....		58
RESULTS AND DISCUSSION		58
4.1	Introduction	58
4.2	Dataset Modelling	58
4.3	Performance of PATRSS in Terms of Delivery Ratio	59
4.4	Performance of PATRSS in Terms of Delivery Cost	60
4.5	Performance of PATRSS in Terms of Average Trust Value	62
4.7	Validation of PATRSS with Common Routing Protocols	64
CHAPTER FIVE		68
CONCLUSION AND RECOMMENDATIONS.....		68
5.1	Introduction	68
5.2	Summary	68
5.3	Conclusion.....	68
5.4	Contribution to Knowledge.....	68
5.5	Recommendations for Further Work.....	69
REFERENCE.....		70

Appendix.....	82
----------------------	-----------

LIST OF FIGURES

Figure 1. 1: Message Forwarding in OppNet.....	1
Figure 2. 1: Inter-relationship of some Ad-hoc Networks	15
Figure 2. 2: Classification of OppNets based on existence of infrastructure.....	18
Figure 2. 3: Context based routing.....	21
Figure 2. 4: ONE simulator package.....	24
Figure 2. 5: Privacy and Security.....	26
Figure 2. 6: Structure of Onion Routing	28
Figure 2. 7: Symmetric permutations.....	29
Figure 3. 1: Onion Routing Structure of the Message	49
Figure 3. 2: Stages of Encryption	49
Figure 3. 3: Flow Chart of the Privacy-Aided TRSS.....	53
Figure 3. 4: Message Forwarding in TRSS.....	56
Figure 3. 5: Message forwarding in PATRSS	57
Figure 4. 1: Modelled Datasets	58
Figure 4. 2: Delivery Ratio against percentage of misbehaving nodes.....	60
Figure 4. 3: Delivery cost against percentage of misbehaving nodes.....	61
Figure 4. 4: Variation of Average Trust Value with Time	63
Figure 4. 5: Number of Started messages	65
Figure 4. 6: Number of Aborted messages	66
Figure 4. 7: Number of Delivered Messages	67

LIST OF TABLES

Table 2. 1: Advantages and disadvantages of ad-hoc routing	12
Table 2. 2: Differences between MANETs and OppNets.....	16
Table 2. 3: Permutation group of order three.....	30
Table 3. 1: Composition of the imotes.....	44
Table 4. 1: Simulation Timing.....	59
Table 4. 2: Mean Performance of PATRSS TRSS and PO	64
Table 4. 3: Percentage Improvements.....	64

LIST OF ABBREVIATIONS

Acronyms	Definition
OppNet	Opportunistic Network
DTN	Delay Tolerant Network
MANET	Mobile Ad-hoc Network
VANET	Vehicular Ad-hoc Network
FANET	Flying Ad-hoc Network
IoT	Internet of Things
OMSN	Opportunistic Mobile Social Networks
TTL	Time to Live
TRSS	Trust Routing Based on Social Similarity
PATRSS	Privacy Aided Trust Routing Based on Social Similarity
ONE	Opportunistic Network Environment
PRoPHET	Probabilistic Routing using Past History of Encounter and Transitivity
MaxProP	Maximum Probability with Priority
AP	Access Point
DPA	Digital Personal Assistance
HomeRF,	Home Radio Frequency
IEEE	Institute of Electrical and Electronic Engineering
WLAN	Wireless Local Area Network
IrDA	Infrared Data Association
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
Syn	Synchronization
Ack	Acknowledgment

DSR	Dynamic Source Routing
LAR	Location Aided Routing
DSDV	Destination Sequenced Distance Vector
OLSR	Optimized Link State Routing
WANET	Wireless Ad-Hoc Network
AI	Artificial Intelligence
FANET	Flying-vehicle Aerial Network
GPS	Global Positioning System
UAV	Unmanned Aerial Vehicles
3D	Three Dimensions
DNS	Domain Name System
DHCP	Dynamic Host Configuration Protocol
VoIP	Voice over Internet Protocol
SMTP	Simple Mail Transfer Protocol
NMS	Network Management System
MINT	Mobile Infostation Network Technology
SWIM	Shared Wireless Infostation Model
MULE	Mobile Ubiquitous LAN Extension
LAN	Local Area Network
SMC	Saami Network Connectivity
CAR	Context Aware Routing
MoVe	Motion Vector
NS2	Network Simulator 2
OMNeT++	Objective Modular Network Test bed in C++
OTP	One Time Password

TOR	The Onion Routing
RW	Random Walk
RRW	Restricted Random Walk
IRONMAN	Incentive/Reputation of Opportunistic Network using Social MANET
STC	Secure Two-Party Communication
IBME	Index Based Message Encoding
ROM	Random Oracle Model
CH	Contact Hiding
PPBR	Probabilistic Profile-Based Routing
AOI	Area of Interest
SFDL	Secure Function Definition Language
MPC	Multi-Party Communication
TFT	Tit-For-Tat
PT	Potential Threat
TTP	Trusted Third Party
RLNC	Random Linear Network Coding
CRAWDAD	Community Resource for Archiving Wireless Data at Dartmouth
MSR	Mobile Short Range
FSR	Fixed Short Range
MAC	Media Access Control
ID	Identity
EACK	Encounter-based Acknowledgement
GF	Galois field
XOR	Exclusive Or Operation
ASCII	American System of Code for Information Interchange

ECLA	Eclipse Application platform
IDE	Integrated Development Environment
JVM	Java Virtual Machine
JRE	Java Run-time Environment
SrC	Source Code
GUI	Graphic User Interphase
FN	Fixed Node
MN	Mobile Node

CHAPTER ONE

INTRODUCTION

1.1 Background to the Study

An opportunistic network (OppNet) is a type of Delay Tolerant Network (DTN) and an evolution of Mobile Ad-hoc Network (MANET) in which a message can be transferred from source to destination even if a continuous connection from source to destination does not exist at an instance of time. Therefore, it utilizes and relies upon the opportunistic pair-wise contact between nodes in the network. Thus, messages are routed (or forwarded) from one node to another at any point in time they are close enough within the very limited wireless range of each other. This may happen repeatedly, so that eventually, the message gets to the destination. Thus, it uses store-carry-forward networking paradigm (C'amara *et al.*, 2011; Musolesi & Mascolo, 2008). Figure 1.1 illustrates how a message is forwarded hop by hop by the relay nodes from a source node to the destination node.

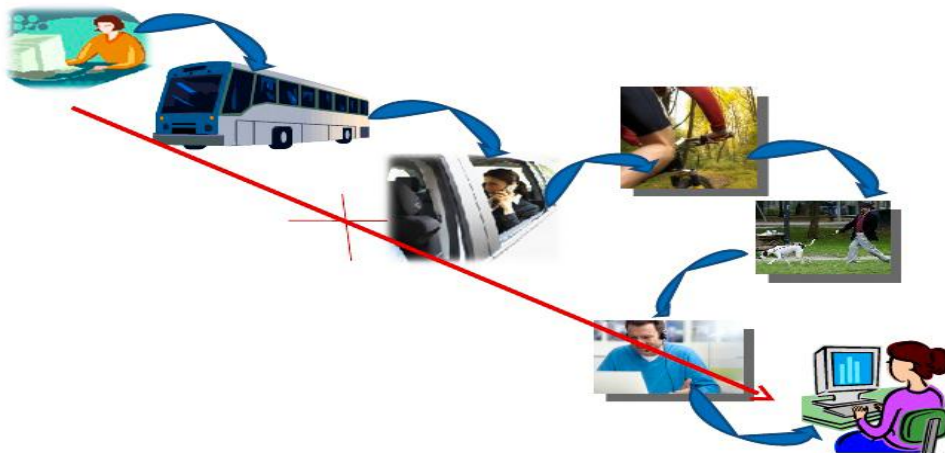


Figure 1. 1: Message Forwarding in OppNet (Noorin, 2009)

Due to the fact that OppNets can be used where there is no network infrastructure, thus having high scalability, it finds applications in many areas such as ubiquitous computing, disaster

management, military surveillance and battle field, wildlife monitoring and Eco-physiology, Vehicular Ad-hoc Networks (VANET's), Internet of Things (IoT), deep space exploration, etc. (Luo *et al.*, 2008; Malladi & Agrawal, 2002; NASA, 2012). Despite OppNets being a potential solution to some of the problems of the traditional networks; the frequent disruption, long delay, dynamic and ad-hoc (self-organising) nature of OppNets present some challenges that need to be overcome in order to optimally reap its usefulness. Researches in OppNets cut across disciplines related to computer networking (computer science and engineering), mathematics, information sciences, and social sciences (sociology and social psychology).

Recently, research interests on OppNets have grown because it is a promising solution to the limits of the traditional infrastructure-based networks. Researchers have done a lot of work on areas related to the routing protocols. Since OppNets use store-carry-forward network paradigm, the buffer management is also a very important aspect of the network. For this, many in-depth works have also been reported on the buffer management strategies (Viscal *et al.*, 2014; Pan *et al.*, 2013). The wireless devices in this network that help route messages may be held by humans, attached to animals, or part of transport systems such as: vehicles, ships, airplanes, amongst others (Balasubramanian *et al.*, 2008; Eagle & Pentland, 2006; McNett & Voelker, 2005). These devices do not just buffer the message but also carry it as they move in the environment until they have pair-wise encounter with the destination or more suitable relay nodes towards the destination. Therefore, the mobility of the carriers of these devices impacts greatly on the performance of OppNets (Camp *et al.*, 2002; Karamshuk *et al.*, 2011; Lin *et al.*, 2004). As such, studies have also been made on this aspect so as to improve the performance of the network. In addition, data from real life contact traces have been gathered in efforts to make an OppNet a reality (Henderson *et al.* 2004; Mtibaa *et al.*, 2008; Piorkowski *et al.*, 2009).

More recently, behavioral pattern of humans from their social interactions in terms of connectivity and mobility has been researched. The social interaction includes: residence, friends, place of work, etc. Others uses behavioral pattern that is related to interest, centrality, community, proximity, amongst others (D'ora & Holezer, 2010; Hui *et al.*, 2011; Fan *et al.*, 2013; Daly *et al.*, 2009; Abdelkader *et al.*, 2013). The behavioral pattern of the social interactions has been identified as the most promising means of routing in many researches (Zhang *et al.*, 2014; Zhou *et al.*, 2013; Usman & Gutierrez, 2018). Thus, because nodes in OppNet are mobile and 'social' sine qua non, (i.e. mobility and social interaction cannot be completely separated in real life), it is also called Opportunistic Mobile Social Networks (OMSN).

However, despite all these works, the usefulness of OppNet's is limited without a serious consideration to the issues of security and privacy (Liang *et al.*, 2013; Kumar *et al.*, 2018; Liu *et al.*, 2018). Because of the nature of OppNets, there is trade-off between routing efficiency and privacy as presented by Costantino *et al.*, (2014). Hence, the need for more thorough and realistic works on this aspect of OppNet.

1.2 Significance of Research

This research work proposes development of a scheme that incorporates privacy preservation in opportunistic network (OppNet) with trust routing. This scheme combines the cryptographic security mechanisms with the key distribution by means of partial onion routing protocol to ensure both the privacy of message and that of the relay nodes are preserved. Without such scheme, the willingness of some potential nodes to participate in the routing/forwarding of messages cannot be realized as a node may not be willing to share its social information which could be very vital in choosing suitable relay nodes. Since message routing/forwarding in OppNet utilizes opportunistic contact of nodes, this causes considerable degradation in performance. Furthermore, if malicious

and selfish nodes are present in the network, they have the potential to harm the cooperative nodes that willingly share their information in the absence of privacy before they are detected by the trust and feedback mechanisms of trust routing.

With privacy preservation the opportunistic contacts of nodes are better utilized. This is because trust ensures communication between nodes that have prior knowledge of each other. Privacy preservation ensures communication between nodes even in the absence of any prior knowledge. Privacy preservation gives a new “unknown” and a “repented” misbehaving (non-cooperating) node better opportunity to join the co-operative nodes in the network and the chance to build a trust profile with time.

In addition, if the sensitivity of a message to be sent requires a node of high trust value, and the number of such nodes is not much in the network, then the cost of getting such message to the destination becomes prohibitively high. This is because such message is kept in the buffer for a very long time and continually duplicated many times so that it is not lost if the message time to live (TTL) expired. Without a scheme such as privacy preservation in trust routing, transmitting sensitive message through the network incurs very high cost.

1.3 Problem Statement

According to Yao *et al.*, (2016), the thrust management model using social similarity performs greatly in securing an opportunistic network against malicious and selfish nodes. This is possible in an ideal scenario where the privacy of the messages been sent does not matter and the relay nodes are willing to share their social information, which helps in the routing, without worrying about their privacy. This ideal case deviates greatly from the reality.

Firstly, this is because such ideal scenarios are only possible in the presence of relay nodes that corporate without worrying about their privacy. Secondly, in a highly challenged, sparse or dense

environment, the nodes may completely be unaware of each other prior to their meeting, yet the need for opportunistic transfer of messages can arise which is the core importance of OppNets. Thirdly, trust routing impacts negatively on the delivery probability. This is due to the fact that the more trust a node demands (for a sensitive message) the less the delivery rate that can be achieved as few numbers of nodes can meet such condition. On the other hand, the lesser the trust level a node demands for routing the easier its security is compromised.

One of the main problems with trust models in OppNets is its inability to effectively utilize the contact opportunities with new nodes for message routing/forwarding. This seriously reduces the delivery rate because OppNet relies mainly on opportunistic contacts between nodes for message delivery.

By incorporating privacy preservation into the trust routing the security and the routing performance can be improved simultaneously. This is because; presence of privacy protection allows the nodes to mutually help one another in the routing/forwarding of messages even before significant trust is established. Furthermore, mutual interaction of a node with an “unknown” node facilitates the establishment of the trust. Consequently, the negative effects of trust routing on message delivery in OppNets is mitigated.

1.4 Aim and Objectives

The aim of this research is to develop a privacy-aided trust routing based on social similarity in opportunistic network.

The objectives are to:

1. Develop a community-dependent privacy scheme.
2. Incorporate the developed privacy into Trust Routing Based on Social Similarity (TRSS) to form privacy aided TRSS (PATRSS)

3. Compare the performance of PATRSS with Privacy in OppNet (PO) and TRSS in terms of delivery ratio, delivery cost, and trust performance.

1.5 Motivation

It is estimated that there are over 3.3 billion mobile phone users around the world. This number is more than a half of the world population. Utilizing the wireless, processing and buffering capabilities of mobile phones and other wireless devices in Opportunistic Network (OppNet) is a promising solution to some of the limits which the traditional infrastructure-based imposed on wireless communications. Although many approaches are proposed, trust based social-aware OppNets have been shown to perform significantly better than others.

However, the basic assumptions upon which trust routings are built limits its usefulness in many applications. This means that, by developing a scheme to mitigate the consequences of these assumptions, the usefulness of OppNets is significantly improved.

1.6 Thesis Organization

Chapter One of this thesis gives background information of this research. Chapter Two follows by providing the review of literatures related to the fundamental concepts on this research and those related to past works on trust routing in OppNets. Thereafter, Chapter Three expatiates on the materials and methods adopted for the success of this work. In addition, Chapter Four presents the results and discusses them. This leads to conclusion and recommendations that are discussed in Chapter Five. And finally, quoted references and appendices are provided at the end of this thesis.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter discusses literatures that are relevant to this research work. It reviews mainly the fundamental concepts, and the recent past works that are closely related to this work.

2.2 Review of Fundamental Concepts

In this section, the fundamental concepts are discussed. They are divided into: opportunistic network, and its classification. This is followed by discussions on Opportunistic Network Environment (ONE) simulator, network security and privacy, onion routing and Galois Theory.

2.2.1 Opportunistic Network

With current explosive advancements in computing technologies and related fields, wireless devices are readily available as handheld pocket sized mobile devices, wearables on parts of the body, as part of transport systems, or used at home, in offices or schools for various purposes and in various sizes, capabilities and functionalities. Despite the presence of wireless infrastructures in so many areas, the need for communication often arises where infrastructure is not available or cannot be used. In many of such cases a contemporaneous end to end link between nodes is not available. Opportunistic computing exploits the computing resources of nodes for communication whenever opportunities for communications is made possible by any factor. Opportunistic network is a communication network that uses opportunistic computing paradigm.

Broadly speaking, a wireless network can be an infrastructure-based or infrastructure-less mode. In an infrastructure-based mode, the nodes access the network through one or more wireless Access Points (APs) usual with a central administration. Where this is not possible due to cost implication, unfavorable weather and environmental conditions, coping with frequent dynamic changes due to nodes presence and disappearance (disruption and delay), scalability, lack of

reliable source of power, etc., the wireless network operates in infrastructure-less mode. In this mode, nodes have to mutually co-operate and self-organize so that a node can forward packets through other nodes. Since the communication links in the infrastructure less mode are formed on-the-fly, this mode is more commonly called an ad-hoc mode (Kurose & Ross, 2017).

An OppNet therefore, is a type of ad-hoc wireless network that is characterized by long delay, frequent disruption, and lack of end-to-end connectivity. These characteristics of OppNets are exceptional cases in ad-hoc networks (Arastouie & Sabaei, 2016; Bamrah *et al.*, 2016). Therefore, OppNets rely on recent pervasive availability and continuous increase in the number and technology of mobile wireless microprocessor-based embedded devices. (Misra, *et al.*, 2016). These devices include, laptops, palmtops, tablets, Digital Personal Assistance (DPA), mobile phones, wristwatches, wearables, digital cameras, vehicles, etc. Although these devices usually have short range radio links, (piconet, HomeRF, IEEE 802.11 WLAN, Bluetooth, IrDA, ZigBee, etc.), they generally have strong computing capability and considerable memory (or buffer) spaces (Wang & Wu, 2015).

2.2.1.1 Delay tolerant ad-hoc and opportunistic networks

The operation of OppNets is about what is to be communicated (message) and how it is to be communicated (relay nodes). From the perspective of the messages, OppNet is related to Delay Tolerant Network (DTN). Whereas, from the perspective of the relay nodes, OppNet is related to ad-hoc network (Misra *et al.*, 2016).

A DTN is explained as follows. In a conventional network, data transmission uses routers that forward data through the links (wire, fiber optics, or wireless). The User Datagram Protocol (UDP) that is used to send packets from source to destination needs a perfect path to succeed. But communication link between routers are not always reliable (i.e. not available all the time). Internet

routers simply discard any packets that cannot be forwarded whenever the link is down. This results in data loss between source and receiver as the end-to-end connection is not perfect. With DTN, routers are replaced with DTN nodes that can store and later forward bundles of data. In other words, DTN nodes must have storage capability. So that if a link is down, the DTN node will hold the bundles until the link comes back up. DTN uses this store and forward mechanism to handle disruptions (resulting to delay) so that consequently all bundles can get through (Kurose & Ross, 2017).

Sometimes, even if a router or a DTN node “thinks” a link is up, a packet or bundle can still be lost en route. For this reason, both the standard internet and DTN protocols have reliability protocols to retransmit missing data. The internet has TCP for reliability. TCP establishes a connection first with control packets (syn, ack). TCP requires end-to-end acknowledgement (ack). If a packet is lost and therefore, unacknowledged, the sender will retransmit the packet (Kurose & Ross, 2017). Thus, the standard internet reliability requires:

1. Complete end-to-end path to establish connection.
2. Retransmission from sender to receiver

In contrast, a DTN node uses end-to-end custody transfer. Thus successive nodes take custody and the bundles progressively migrate towards the destination. In case, if a bundle is lost, and therefore custody is unacknowledged by the next node, the last custodian retransmits. This means that, DTN provides the following:

1. Reliable data transfer without a complete contemporaneous end-to-end path from sender to destination.
2. Retransmission from closest relay node rather than the sender
3. Custody transfer and return receipt operation.

Some researchers have used the terms DTN and opportunistic network (OppNet) interchangeably (Wang & Wu, 2015) and some even claimed that they are synonymous (Conan 2013; Pelusi, *et al.*, 2006). Though they have a lot in common, strictly speaking they are not the same. There is a salient difference between them. From the explanations above the difference between a conventional DTN and an OppNet can be summarized thus:

1. In conventional DTN's, protocols are designed with the assumption that there is no delay, no frequent disruption, and no lack of end-to-end connectivity and to some extents considers exceptions, in cases, if one or more of the assumptions above do not hold, hence data transmission in this network can be said to have ability to tolerate delay (i.e. delay tolerant).
2. In OppNets, however, protocols are designed with the assumption that there will always be delay, frequent disruption, and lack of end-to-end connectivity except in advantageous cases in which one or more of the assumptions above do not hold, hence data transmission in this network relies almost completely on opportunistic contacts of the nodes (Huang *et al.*, 2008).

The similarities between DTN and OppNet can be summarized as follows:

1. In both traditional DTN and OppNet, end-to-end communication path is not a necessary condition for end-to-end communication
2. They both use store-forward network paradigm
3. Delay is not taken seriously. The main concern in DTN and OppNet is delivery despite disconnections, disruptions, and partitions.

4. They both set to address the problem of how to ensure end-to-end communication between nodes in networks that are not connected. Each does this from two different perspectives and priorities.

Based on their priorities, differences between DTNs and OppNets can be put together in two points thus:

1. In DTNs the main concern is to ensure there is interoperability between networks that are internet-like in nature. Because of this, DTNs need a priori knowledge of topology of the network's gateways. On the other hand, in OppNets the main concern is that each and every node should be a gateway. As a result, new networking protocols have to be designed for OppNets.
2. DTNs are designed as overlay networks on the standard internet architecture and uses store-forward paradigm to handle delay and custodian transfer to handle disruption and partition. Whereas, OppNets are designed to utilize every opportunity that allows for communication. Therefore, it exploits nodes mobility (including user behavior and information) in a store-carry-and-forward paradigm to handle delay, and uses buffering and duplication of messages to ensure the message survives disruptions, partition and long delay.

On the other hand, wireless ad-hoc network nodes directly communicate with each other in a self-organize manner. This means that all the wireless devices within the wireless range of each other directly discover and communicate in point to point fashion without involving a central Access Point (AP). For this, an Ad-hoc routing can be single or multi-hop. In contrast, OppNets routings are inherently multi-hop. As pointed out by (Pelusi, *et al.*, 2006), the main evolution of multi-hop wireless ad-hoc networks are OppNets and mesh networks.

In addition, not only do conventional ad-hoc routing protocols not able to take care of network partitions but also do assume that at most of the instance of time there is end-to-end connection between source and destination. These and other characteristics of Ad-hoc networks made reactive (or on-demand) routing protocols such as Dynamic Source Routing (DSR), Location Aided Routing (LAR), etc. and proactive (or table-driven) routing protocols such as Destination Sequenced Distance Vector (DSDV) and Optimized Link State Routing (OLSR), etc., and their hybrid possible. In proactive routing protocols, the source learns the network topology, as nodes exchange routing information, to destination before forwarding. Whereas, in a reactive routing protocol, the source obtains path to destination only when it needs to send some data. The advantages and disadvantages of ad-hoc routing, (Kurose & Ross, 2017), can be summarized in Table 2.1.

Table 2. 1: Advantages and disadvantages of ad-hoc routing

Advantages	Disadvantages
Ease of deployment	Limited wireless transmission range
Speed of deployment	Packet loss due to transmission errors
Decreased dependence on infrastructures	Battery constraints

In Opportunistic networks however, reactive routing protocols of the conventional wireless ad-hoc network cannot be realized. This is because its algorithm cannot successfully find a continuous link at all times. Similarly, the proactive routing protocols cannot be realized as its algorithm cannot converge due to the fact that network dynamics are very high (Abdelkader *et al.*, 2013; Lu *et al.*, 2010).

2.2.1.2 WANETS

Ad-hoc networks (also called wireless Ad-hoc networks, WANETs) are characterized by the facts that they are decentralized, self-organizing, dynamic and scalable. In terms of intended application of WANETs, they are classified as Mobile Ad-hoc Network (MANET), Vehicular Ad-hoc Network (VANET), Fly Ad-hoc Network (FANET), Opportunistic Mobile Social Network (OMSN, mostly called OppNet), Wireless Mesh Network (WMN), Wireless Sensor Network (WSN), etc, (Kurose & Ross, 2017). This sub-section discusses MANET, VANET and FANET due to their close relationship to OppNets.

MANET, as the name implies, is made up of mobile nodes that communicate in ad-hoc mode. Therefore, it is a wireless ad-hoc autonomous network in which the mobility of the mobile nodes, most of which are hand-held mobile phones by humans, are explored in order to achieve connectivity (Wang & Wu, 2015). MANETs generally use protocols that simultaneously use mobility and route discovery. This is because, as the mobile nodes move in the network environment, their routing changes continuously and adaptively so as to ensure end-to-end link does not break. Presence of bidirectional link made it possible for MANETs to use Transmission Control Protocol (TCP) to provide reliability in data transmission (Bjurefors 2012; Conti *et al.*, 2014).

VANET is the application of some of the principles of MANET to road transport vehicular scenarios (Kurose & Ross, 2017). This scenario usually involves either vehicle to vehicle or vehicle to roadside communications. The differences between the protocols of MANET and VANET emanate mainly from their mobility (Bouroumine *et al.*, 2016). This is because vehicular mobility is more organized, and more restricted, in speed, position and trajectory. Popular mobility models include: Random walk, random waypoint, deterministic, random Gauss-Markov, semi-

Markov, fluid flow, correlated diffusion, particle-based, hierarchical influence, behavioral, steady-state generic, graph-based, group-based, autoregressive, swarming-based, virtual game-driven, non-recurrent, social based, community based, orbit based, entropy-based, knowledge driven, etc. and their variants. Some of these are more applicable to VANET domain than MANET because issues such as safety, real-time traffic monitoring and traffic laws are very important VANET scenarios. Application of VANET include electronic break light, platooning, traffic information system, road transport emergency service, etc. (Zenjireh & Larijani, 2015). Recently, Artificial Intelligence (AI) agents are implemented on VANETs to create intelligent transport systems, for autonomous navigation, collision avoidance in vehicles, etc.

FANET is a sub-class of MANET. It usually consists of swarm of flying vehicles that are equipped with Global Positioning System (GPS), sensors and cameras. It is an ad hoc network between some Unmanned Aerial Vehicles (UAVs) such that at least one of them communicates with a ground station or satellite. Recently, Amazon has used UAV for package delivery. UAVs have also been proposed as an effective OppNet node in natural disaster scenarios to facilitate communication (Albuquerque *et al.*, 2016). Protocols in FANETs differ from that of VANET and MANET mainly because FANETs mobility is considerably different from them. Its velocity is higher and it continuously moves in three-dimensional (3D) space. Therefore, network topology changes more frequently and abruptly (Bekmezci *et al.*, 2013).

In summary, the relationship between MANET, VANET, FANET and OppNet; is depicted in Figure 2.1.

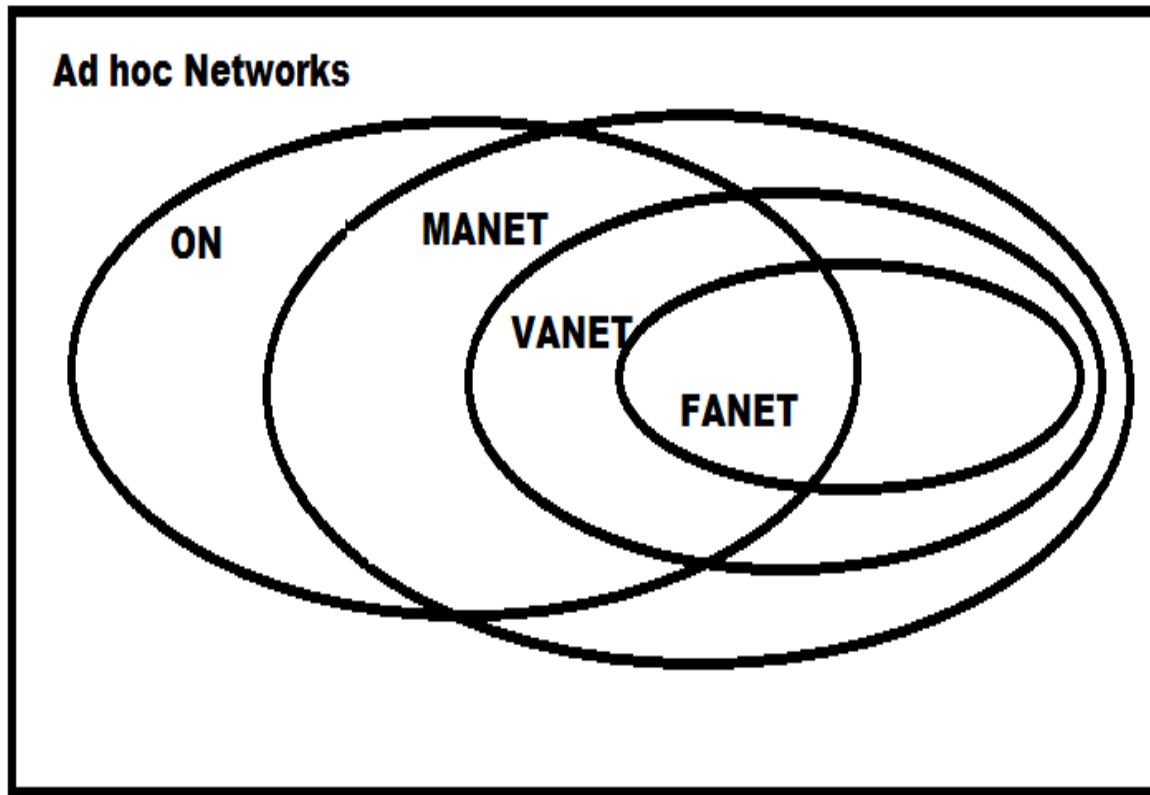


Figure 2. 1: Inter-relationship of some Ad-hoc Networks (Kurose & Ross, 2017)

In contrast, end-to-end path in OppNets does not always exist. Therefore, in OppNets, reliable data transmission by means of feedback loops (in form of acknowledgement) using TCP is not possible. Consequently, some researchers have referred to OppNets simply as an “interesting evolution of MANETs” (Basira 2015; Pelusi *et al.*, 2006). Differences between OppNets and MANETs (Including VANETs and FANETs) are summarized in Table 2.2

Table 2. 2: Differences between MANETs and OppNets (Kurose & Ross, 2017)

	MANETs	OppNets
1	Its protocols are designed for scenarios in which end-to-end path exists	Its protocols are designed for scenarios that end-to-end path may not exist.
2	It uses TCP for acknowledgement	It cannot use TCP for acknowledgement
3	Its protocols resides in network layer	Its protocols resides in application or bundle (between application and transport) layer
4	It transmits packet	It transmits messages
5	It requires prior knowledge of network topology	It does not require prior knowledge of network topology
6	It is infrastructure-less	It utilizes every opportunity (infrastructure based or infrastructure-less)
7	It considers users mobility to be a challenge	It exploits users' mobility and social behavior

2.2.2 Classification of opportunist network routing protocols

OppNet protocols are designed to ensure that messages are delivered despite the challenges in the network. Each of these protocols is designed to cater for some sets of challenges. Therefore, each has its strengths and weaknesses. To understand and use the protocols effectively they are classified. There are different bases for classification which include (Arafath *et al.*, 2017; Wahid *et al.*, 2014):

1. existence of infrastructure
2. heuristic for routing

3. intended application environment

For the purpose of this work, classification based on infrastructure is further discussed.

2.2.2.1 *Classification of protocols based on existence of infrastructure*

In communication and computer networks, an infrastructure is a set of facilities, hardware, software, and services that connects the various devices together and make the network services possible for communication. They include: Ethernet, optical fibers, wireless access points, repeaters, routers, switches, hubs, bridges, gateways, proxies, servers, firewalls, intrusion detection/prevention systems, Identity/access management, key management, certificate authority, Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Voice over Internet Protocol (VoIP), Simple Mail Transfer Protocol (SMTP), Network Management System (NMS), etc.

Based on existence of infrastructure, OppNet protocols are classified into two: those that use infrastructure and those that do not make use of infrastructure. Figure 2.2 shows a typical OppNet classification based on existence of infrastructure. Infrastructure based protocols are further divided in terms of mobility into two: mobile and stationary while infrastructure-less protocols are divided in terms of routing-dependency on context into two: context based and context oblivious (Pelusi, *et al.*, 2006).

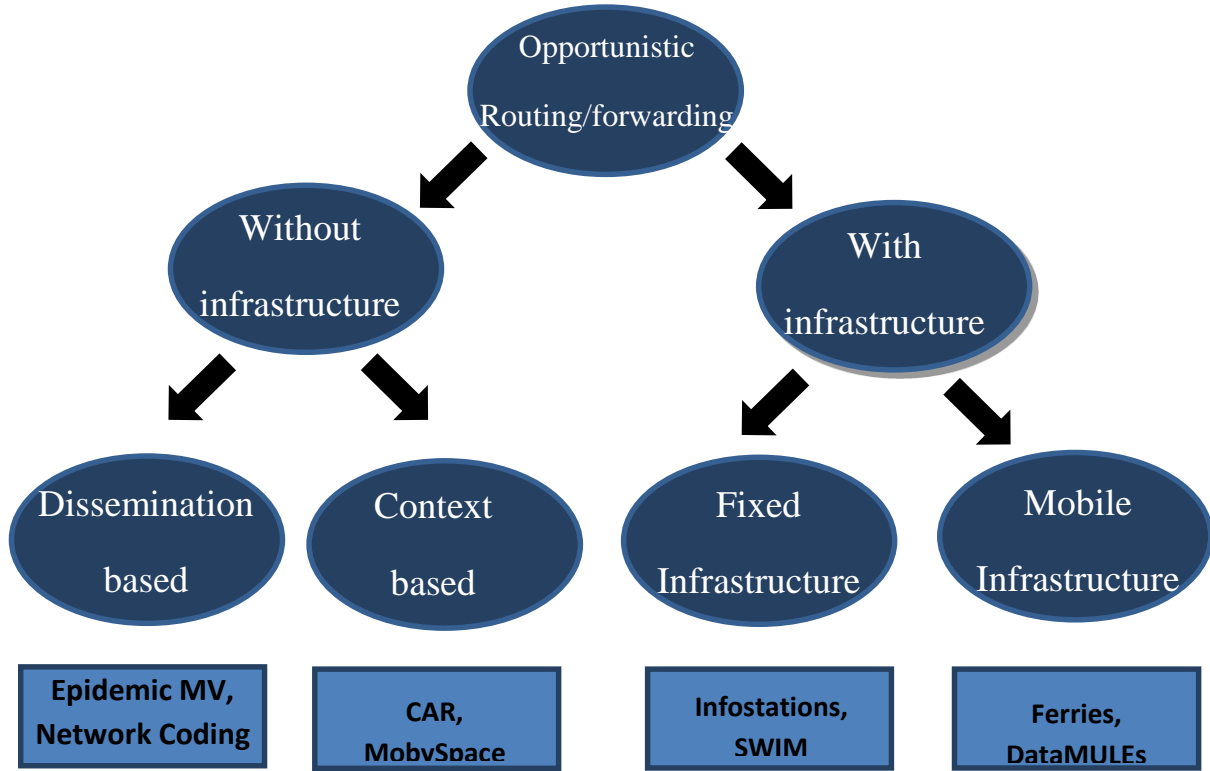


Figure 2. 2: Classification of OppNets based on existence of infrastructure (Pelusi, *et al.*, 2006)

From this, OppNets are classified into four sets. In each sets three protocols are briefly explained:

1. Infrastructure based fixed OppNets:

- a) Infostation: uses protocol that ensures that data are differentiated in terms of location and time and users are differentiated in terms of channel strength and importance. They are then rated and controlled by Mobile Infostation Network Technology (MINT) such that users are prioritized and served. It was first proposed for battle field communication (Rajappan, *et al.*, 2004)
- b) Shared Wireless Infostation Model (SWIM): In this, wireless tags are attached to whales. As the whales move around in a water body, for each pair-wise contact, data are compared and updated by their tags. This continues until data gets to any of the data

- repository buoys on the water body. It was proposed for collecting data in water bodies such as river, lakes, ocean etc. (Small & Haas, 2003)
- c) ZebraNet: Was designed for wildlife monitoring and tracking. Wireless tracking devices are hung around Zebra neck and peer to peer networking principle is used to communicate in a way similar to that of SWIM to some stationary data repository (Liu *et al.*, 2004).
2. Infrastructure based mobile OppNets (Pelusi *et al.*, 2006)
- a) Ferries: Both ferries and mules uses similar concepts. The main difference between them is that, in ferries both the infrastructure and the nodes are mobile while in mules the infrastructure is mobile while the nodes are stationary
 - b) Data mules: data is forwarded in Mobile Ubiquitous LAN Extension (MULE). In this case, a mobile device of very large memory capacity moves in space and communicates data with other nodes. This system is sometimes used in collection of data in some Wireless Sensor Network (WSN).
 - c) Saami Network Connectivity (SMC): It was designed to provide internet connectivity to a group of isolated sparse settlements who are reindeer herders of Saami in Sweden to reduce social and political exclusion.
3. Infrastructure-less context based OppNets (Huang *et al.*, 2008)
- a) MobySpace: uses a virtual coordinate routing which the authors called mobility space (MobySpace) pattern. It uses the concept of closeness to calculate the probability that a relay node will meet and deliver message to the destination.
 - b) Context Aware Routing (CAR): In CAR, if the chance of a message being delivered in time is low, the message is sent to a host node whose chance of delivering the message

is higher based on its context. The context is evaluated using Kalman filter to determine its delivery probability.

- c) MoVe: uses Motion Vector (MoVe) which is calculated from the relative velocities of its neighbors to predict their future locations and determine most suitable forwarders of messages towards the destination.

4. Infrastructure-less context oblivious OppNets

- a) Epidemic: This uses epidemic algorithm which was first developed based on principle of epidemiology by social scientists tracking crime in a population. A node can only infect (send a message) to another node with a disease (the message) if and only if the later has not been infected before. Therefore, when two nodes encounter they exchange summary vector to determine which message they have and thereafter update each other. Thus the message floods the network until it gets to the destination (Vahdat & Becker, 2000).
- b) Network Coding: In network coding, as a relay node receives messages it continuously combines it with the previous message to form one message before sending to the next relay node. Thus message is coded and flooded through the network hop by hop until it eventually reaches the destination. The destination uses an algorithm to detect and remove its message and combine the remaining with the message it has prior and send as one message to other relay nodes.
- c) Spray and Wait: Spray and wait uses flooding mechanism but limit it so as to avoid congestion through spray and wait strategy. A message is duplicated L times and $L/2$ of the message is sent to the first sets of relay nodes. The next gets half of the number

of remaining messages. Thus the process is repeated until destination is reached (Spyropoulos *et al.*, 2005)

2.2.2.2 Context Based OppNet

Context based routing can be further classified into social oblivious and social aware. All the routing protocols discussed on context based infrastructure-less protocols in section 2.2.2.1 are social oblivious context routing. This is because the protocols do not use the social information or behavior of the nodes. Social aware protocols exploit users' social information and behavior to determine the next forwarder of a message towards the destination. Figure 2.3 shows a classification of context based routing (Wei *et al.*, 2014).

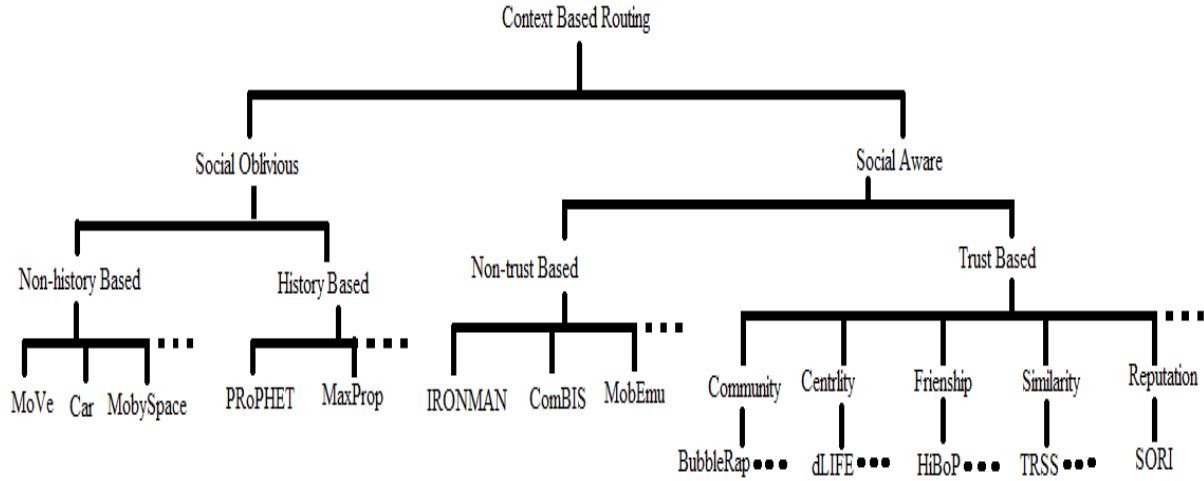


Figure 2. 3: Context based routing (Wei *et al.*, 2014)

Some of the context-based routing protocols that are related to this research are briefly explained as follows.

1. Probabilistic Routing using Past History of Encounter and Transitivity (PRoPHET): In PRoPHET, when a sender node encounters a potential relay node it calculates the probability of successful delivery to the destination node based on past encounter the relay

node had with other nodes. It therefore, uses history of encounter, aging factor and transitive probability to calculate the most suitable next hop (Lindgren *et al.*, 2003).

2. MaxProp: Messages are sent in a peer to peer manner, while the order of transmission and hops are used to sort messages and determine the most suitable node to convey the message to the destination. In addition, it uses a light-weight means of acknowledgement to determine if a message reaches the destination successfully. If so, the copies of the message are deleted from its buffers to conserve space. (Burgess *et al.*, 2006).
3. Trust Routing Based on Social Similarity (TRSS): Uses the similarity in the social profile of a node to compute trust. If the trust value is found to be greater than the trust threshold, it is chosen as a promising node and used as next hop. When two nodes encounter, trust of a node is increased or reduced based on its cooperativeness. This is made possible either from directly evaluating the trust based on acknowledgement feedback received or computed from recommendation of a trusted node from its history table (Yao, *et al.*, 2016).

2.3 Opportunistic Network Environment Simulator

Opportunistic Network Environment (ONE) simulator is a Java-based, agent-based, discrete event simulator designed and optimized to improve the realism in simulation of OppNet scenarios. For this, simulation in ONE is made up of packages for reporting, routing, visualization, and mobility modeling. Visualization can be important during development of an algorithm. For instance, if an algorithm for a mobility model is being tested during development, a researcher would need to intuitively see, and validate, the nodes movement behaviors and responses in the simulation environment as the simulation process progresses. Giving the inputs and collecting the output results from a computer system may not be enough. MANET simulators such as NS2 and OMNeT++, etc., would not properly simulate OppNet scenarios. This is because their routing

protocols cannot properly reflect the opportunistic characteristics of OppNet environment. Similarly, DTN simulators such as dtnsim1, dtnsim2, etc., cannot simulate OppNets. This is due to the fact that these simulators consider only conventional DTN routing protocols and, therefore, do not give consideration for mobility. Furthermore, lack of mobility, make them not readily usable to recent routing protocols in OppNets, especially, those that exploit human social behaviors and traces (Keranen, 2008).

In cases where these simulators consider mobility, they are based on random way point mobility model. This model assumes that a node within the simulation environment in a position randomly and chooses destination and maximum speed and then move from current position to the destination. And once it reaches the destination, it pauses for some time and then repeats the processes all over again. This continues until the end of the simulation time. Though this is a very simple model, it does not reflect a realistic human and vehicular mobility. What it means in reality is that a vehicle can be moving on a highway and suddenly leave the highway and move through a thick forest or river. It also implies that human beings are trekking a distance of over 50km or moving through the walls of buildings. All these deviate greatly from realistic scenarios.

ONE simulator software program is written in Java programming language. Java is a high level object oriented program for developing application software for mobile phones, stand-alone computers systems, servers, etc. It has advantages over its predecessors and some of its successor of being simple, dynamic, multi-threaded, high performance, portable, architecture neutral, secure, robust, interpreted, distributed, and platform independent.

An overview of the working of ONE simulator as explained by (Keranen 2008) is shown in Figure 2.4. It consists of event generator, movement model, routing, visualization and results, simulation engine, etc.

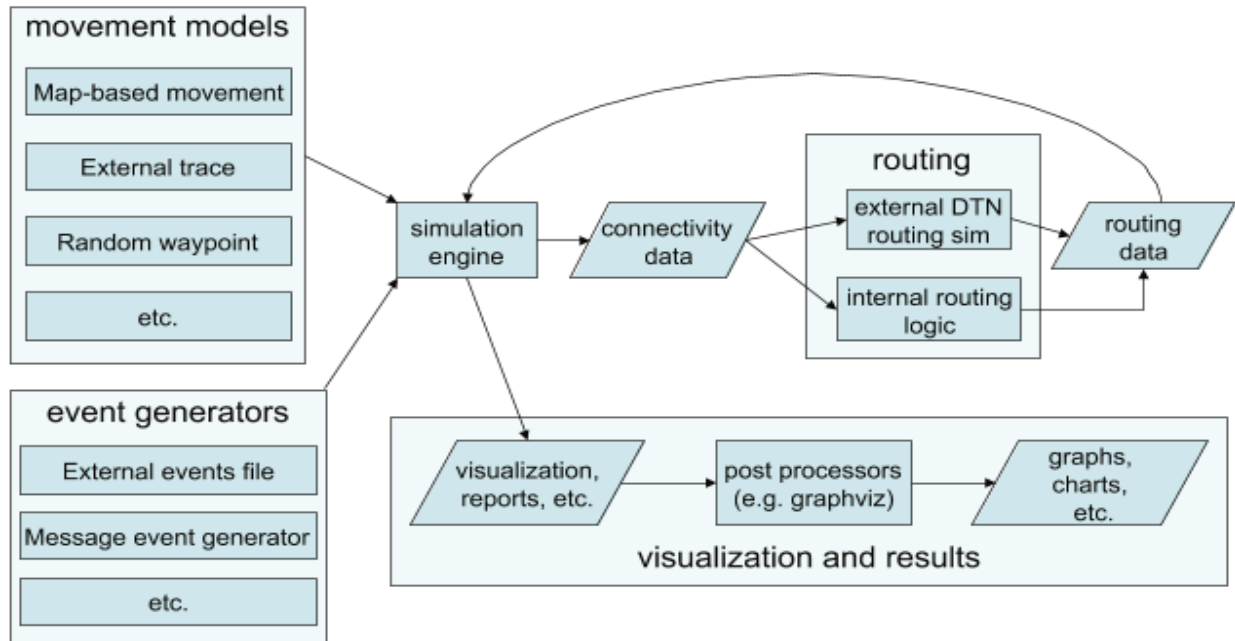


Figure 2. 4: ONE Simulator Package (Keranen 2008)

The core of ONE simulator is the simulation engine. This is the central module that processes all the activities in ONE. It has two input module: movement models and event generators; two output modules: results/visualization and connectivity data; and a feedback loop from the routing data module.

2.4 Network Security and Privacy

A network information system can be said to be secured if it can preserve the confidentiality, integrity and availability of its resources against security threats/attacks. These resources include the hardware and software needed for effective performance. Due to pervasive nature of modern computer network systems, security must be given optimum attentions (Stallings 2011).

Confidentiality ensures information access; disclosure and privacy are restricted to the authorized entities. While integrity ensures that information modification, destruction, non-repudiation and authenticity are protected. And Availability ensures that information access and use is timely and reliably available for the authorized users (Atul 2013).

A security attack can be passive or active. An active attack may involve alteration, modification or destruction of the information while passive attack does not involve any of these. Examples of an active attack include: masquerade, replay, modification, denial of service, etc., Examples of a passive attack include: eavesdropping, traffic analysis, etc. (Stallings 2011).

Confidentiality is a component of security service that inadvertently prevents passive attacks (eavesdropping, traffic analysis, etc.) and may improve protection against active attacks (masquerading, message modification, etc.).

Privacy is a proper subset of confidentiality which is, in turn, a proper subset of security as shown in Figure 2.5. Privacy disclosure is prevented using some security methods which include: steganography and cryptography. Steganography is a method of hiding much smaller information (i.e. plaintext message) within much bigger information. This has an advantage of hiding not just the intended information but also the ability of an unauthorized party to detect the presence of the information (or being communicated) from one entity to the other. Though this method is very effective, it has a disadvantage of requiring very large overhead (Stallings 2011).

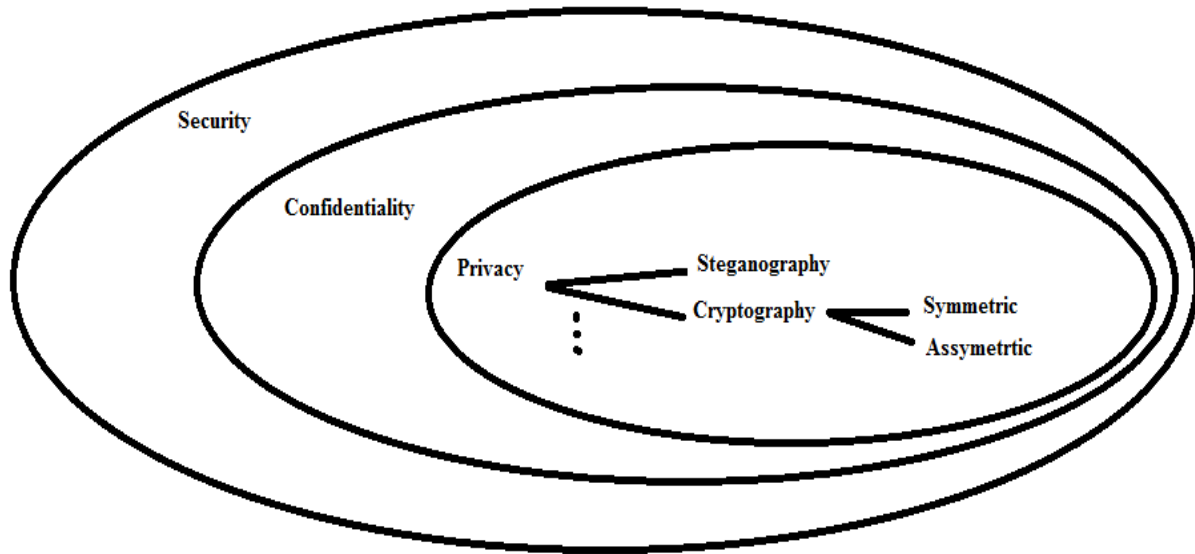


Figure 2. 5: Privacy and Security (Stallings 2011)

Cryptography is the method of making information (plaintext message) not usable by any entity except the intended recipient by transforming it to another form (i.e. the cipher text). The process is called encryption. This method has an advantage of requiring less overhead but a disadvantage of requiring security of the keys to revert the cipher text back to the original intelligible message. This revision process is called decryption. The key is said to be symmetric if same key can be used in the encryption and decryption processes, else it is asymmetric (Lakshmi & Neelima, 2017). Symmetric key cryptography is also called secret key cryptography because the key has to be known to the intended recipient in secret. Asymmetric key cryptography is also called public key cryptography because even if the encryption key is made public, the decryption key cannot be deduced from it (Vennela *et al.*, 2018).

From cryptographic point of view, a secured system may either be unconditionally secured or computationally secured. A system is said to be unconditionally secured if it is completely secured against all possible security threats. This condition is hard to be met in real life. The only security mechanism that is unconditionally secured is One Time Password (OTP) (Stallings, 2011). OTP

has its pros and cons that make it not suitable in an OppNet environment. On the other hand, an information system is said to be secured, if at least, it is secured computationally. A system is computationally secured if the efforts required to break into the system greatly supersede the derivable usefulness of the action.

In OppNets, unlike the conventional network (wired or wireless) privacy preserving approaches, as rightly pointed out by Constantino *et al.*, (2014), can be classified mainly as cryptographic and non-cryptographic based. The former has the merit of providing a stronger privacy protection but more computationally cost while the latter is relatively weaker in terms of privacy protection but uses less computational resources.

2.4.1 Onion Routing

The implementation of the onion routing is the TOR (The Onion Routing) that is used in dark web, to provide anonymity. It was developed in the mid-nineties by the United State (US) naval research. It is made to bounce around connections between different routers so that they are hard to track. Anonymity is different from confidentiality that uses encryption. By confidentiality, a message can be seen but cannot be read. With anonymity, the message (and sometimes together with its source) is not seen *let alone* being read.

Figure 2.6 shows the structure of an onion routing scheme with three layers of encryption. The original message (M) is encrypted with a key to form M1 after which it is encrypted again to have M2. Lastly M2 is encrypted with another key to produce M3. When M3 passes through the network it is progressively decrypted in reverse order (top down) such that the original message is known only by the destination node and each of the intermediate nodes does not know other nodes beyond its immediate neighbors. For example, if a client wishes to “talk” to a server and wants to remain anonymous (its identity unknown). Nested encryption process is performed on the messages such

that the first (uppermost) layer of encryption is decrypted by the node closest to the sender and the last layer (innermost) layer is decrypted by the destination. And the destination replies following the same strategy. Therefore, the communication through the network remains anonymous (Camenisch & Lysyanskaya, 2005). This research, applies Galois theory in the encryption of the messages such that the keys are embedded in nested encryption by partial onion routing.

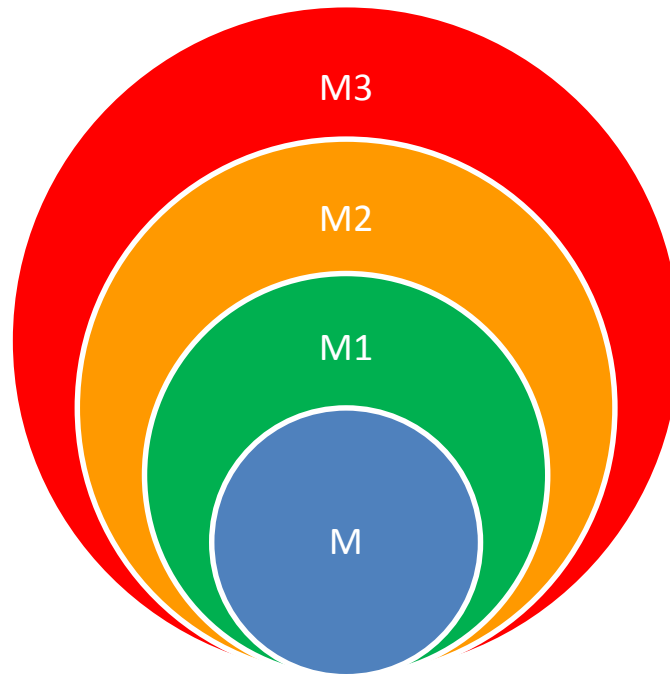


Figure 2. 6: Structure of Onion Routing (Camenisch & Lysyanskaya, 2005).

2.5 Fundamentals of Galois Theorem

In modern (or abstract) algebra, an algebraic structure can be defined as a set that contains the set of finite operations that can be performed on finite set of values and the set of these values on which these operations can be performed. Fundamental algebraic structures include: groups, rings, fields, lattices, vectors spaces, modules, or algebras.

2.5.1 Groups and group theory

A non-empty set A can be said to be a group (A, ϕ) if and only if by performing a binary operation ϕ on any two members of set A , the following axioms are satisfied:

1. Closure: if operation ϕ is performed on any two elements in set A, the resultant is an element in set A, thus:

$$a_1 \phi a_2 = a_3 \quad \forall a_i \in A \quad (2.1)$$

2. Associativity: Set A under operation ϕ is associative if and only if:

$$a_1 \phi (a_2 \phi a_3) = (a_1 \phi a_2) \phi a_3 \quad \forall a_i \in A \quad (2.2)$$

3. Identity: There exist an identity element “a” in set “A” such that

$$a_1 \phi a = a_1 = a \phi a_1 \quad \forall a_i, a \in A \quad (2.3)$$

4. Inverse: There is an inverse element for each element in set A such that

$$a_1 \phi a_1^{-1} = a_1^{-1} \phi a_1 = a \quad \forall a_i, a_i^{-1}, a \in A \quad (2.4)$$

Examples of groups include $(\mathbb{R}, +)$, $(n(p), \text{mod } +)$, $((n(p), \text{mod } x))$ among others.

2.5.2 Symmetric groups

Given n number of distinct objects there are n! permutations that can be done on the objects where:

$$n! = n \times (n-1) \times (n-2) \dots 3 \times 2 \times 1 \quad (2.5)$$

A symmetric group S_n is a set of permutations of the elements of a group and their composition.

By this, a symmetric group is a group in which permutations of the elements of the group does not change the value of the polynomial (usually called symmetric polynomial) that the group produced. Symmetric groups are generally non-commutative.

For instance, for permutations of symmetric group with 3 elements S_3 , there are $3! = 6$ possible compositions. This is illustrated in Figure 2.7.

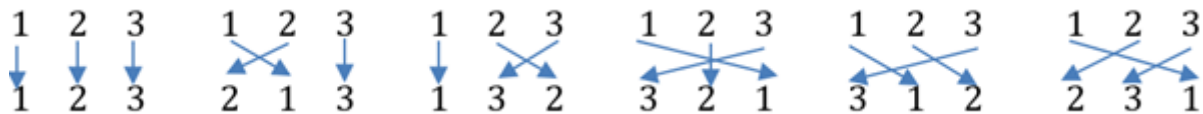


Figure 2. 7: Symmetric Permutations (“Matematisk Institut,” 2009)

This can be written as vectors or alternatively as cyclic permutations as follows: e , $(1, 2)$, $(2, 3)$, $(1, 3)$, $(1, 2, 3)$, and $(1, 3, 2)$ where “ e ” is an identity element. Progressing with this argument, a multiplicative operation can be performed. The row is multiplied by the corresponding column. The result is shown in Table 2.3.

Table 2. 3: Permutation Group of Order Three

X	E	(1, 2)	(2, 3)	(1, 3)	(1, 2, 3)	(1, 3, 2)
E	E	(1, 2)	(2, 3)	(1, 3)	(1, 2, 3)	(1, 3, 2)
(1, 2)	(1, 2)	e	(1, 3, 2)	(1, 2, 3)	(1, 3)	(1, 3)
(2, 3)	(2, 3)	(1, 2, 3)	e	(1, 3, 2)	(1, 2)	(2, 3)
(1, 3)	(1, 3)	(1, 3, 2)	(1, 2, 3)	e	(2, 3)	(1, 2)
(1, 2, 3)	(1, 2, 3)	(2, 3)	(1, 3)	(1, 2)	(1, 3, 2)	E
(1, 3, 2)	(1, 3, 2)	(1, 3)	(1, 2)	(2, 3)	e	(1, 2, 3)

A proper subset of a group that satisfies the closure axioms is referred to as a subgroup of that group. Therefore, S_3 has subgroups such as e , S_3 , $\{e, (1, 2)\}$, $\{e, (1, 3)\}$, $\{e, (2, 3)\}$, $\{e, (1, 2, 3), (1, 3, 2)\}$. An alternating group L is defined as subgroups that involve even permutations.

A group may contain more than one subgroup. If B is a subgroup of A then the order of B divides the order of A , i.e. $(|B| \mid |A|)$. Also, every finite group is isomorphic to symmetric group S_n for some n values.

Study of group theory suggests that groups are intimately connected to symmetries and geometries. Physicists are very interested in the study of group symmetry as it is one of the mathematical foundations upon which quantum mechanics and theory of relativity are built. Computer scientists and engineers are interested in study of groups and other algebraic structures as they are the mathematical foundations of cryptographic algorithms for information security and privacy.

2.5.3 Rings and polynomials

A non-empty set, A can be said to be a ring if and only if, if two binary operations ϕ and ψ are performed on any two elements of the set, the following axioms are valid:

1. An Abelian (or commutative) group in (A, ϕ) is formed
2. There is associativity in (A, ψ)
3. Operation ϕ is distributive over ψ and vice versa.

2.5.4 The galois theorem

The Galois theorem proposed that at most quartic polynomials (i.e. polynomials of order 4 and less) are solvable by radicals. In other words, only polynomials of order 1 through 4 can be solved in terms of their coefficients using arithmetic and radical operations only. The theory arose from finding solutions to polynomial equations. Given an nth order polynomial function $P(x)$ (Jia 2018):

$$P(x) = x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-2}x^2 + a_{n-1}x + a_n \quad (2.6)$$

Where a_1 through a_n are the coefficients, the polynomial can be re-written by replacing each of the coefficients a_i by the corresponding symmetric group S_i (Khudaverdian 2006).

$$P(x) = x^n - S_1x^{n-1} + S_2x^{n-2} - \dots + (-1)^{n-1}S_{n-1}x + (-1)^nS_n \quad (2.7)$$

Any nth order univariate polynomial can be factored into n linear factors in terms of the roots x_1 through x_n (Khudaverdian 2006).

$$P(x) = (x - x_1)(x - x_2)(x - x_3) \dots (x - x_n) \quad (2.8)$$

By expanding and comparing the coefficients of the polynomial $P(x)$, then (Backer, 2012)

$$S_1 = \sum_{i=1}^n x_i \quad (2.9)$$

$$S_2 = \sum_{i < j} x_i x_j \quad (2.10)$$

$$S_3 = \sum_{i < j < k} x_i x_j x_k \quad (2.11)$$

$$\vdots \quad \vdots$$

$$S_n = \prod_{i < j < k < \dots < n} x_i x_j x_k \dots x_n \quad (2.12)$$

Therefore, by applying these properties with natural numbers (especially prime numbers) as a group encryption and decryption algorithms can be developed to suit the needs of a network for secured end-to-end secure communications.

2.6 Review of Similar Works

Review of similar works in this research considers what other researchers have done that is directly related to trust, privacy and security in opportunistic network. This is done through the critique of knowledge gap of what they do, how they do it, result they obtained, and the significance of their work each in a chronological order. This is so as to present the evolution of the research problem. **Baglioni *et al.*, (2010)** developed a cryptography-based scheme for preserving privacy for friend recommendation in a social mobile network. They used their scheme to find out the similarity of the content of contact list of any pair of users. If the similarity was more than certain threshold, then the users were possibly friends else they were not. This was made possible by the use of some hash functions which the users exchanged and then used to obtain Jaccard's index to represent the initial contact list. Thus they proposed a fully decentralized coordination by collecting and processing information that was accessible to any node and then used the information to recommend new friends. They found trade-off between precision and recall by varying the threshold and the number of the hash functions used. They showed that their algorithm was effective in predicting the existence of contact list. Though the information needed to perform recommendation had privacy preservation; the information necessary to implement the application was not privacy preserved which can be explored by malicious nodes.

D'ora and Holezer (2010) developed a non-cryptographic based method using some randomization techniques which they called Hide-and-Lie to protect nodes privacy. They applied

this method to an interest-casting OppNet routing scenario. Thus a node would have an interest profile for various topics and node-to-node routing was based on some probabilistic matching of interests in their message contents. They proposed an obfuscation strategy, where a user could intentionally change his profile to deceive a potential attacker. They showed that with their scheme, the probability of a malicious node tracing a node from the node's profile despite the anonymity in message exchange protocol was considerably lower. Their proposed method involved trade-offs between high privacy level and low data forwarding capability. This method has demerit of being computationally costly as the node had to be continuously filtering message content so as not to have a huge negative effect on the network performance. In addition, their method depended on random walk (RW) and restricted random walk (RRW) mobility models which many researchers had shown deviate greatly from real life scenarios as explained in 2.1.3. Since their method was based on mobility models that did not reflect the reality, in order to minimize the computational complexity that would have been otherwise required, their results were valid only within the simplistic abstract environment they considered.

Bigwood and Henderson (2011) developed an incentive mechanism that utilized existing social network information for detecting and punishing selfish nodes in an opportunistic network. This they called Incentive and Reputation of Opportunistic Network using Social Mobile Ad hoc Network (IRONMAN). The IRONMAN used the information to bootstrap trust relationships without the need for a delivery receipt or an oracle or an infrastructure. They used a trace-driven simulation to demonstrate that their mechanism is better than the earlier mechanisms in terms of accurate and timely detection of selfish nodes and hence improved the delivery performance in the presence of selfish nodes. However, their work advertently assumed that all of the nodes belong to the same social network and used such to build trust. Therefore, incentive mechanism they

proposed failed to deter the selfish behavior of nodes that were not from the same social network and this could be contagious across different social networks.

Chapman *et al.*, (2011) developed a privacy-preserving framework with a secure multi-party computation called “Might Be Evil”. It translated a function $f(x, y)$ into Boolean garbled circuit for Secure Two-Party Communication (STC). A garble circuit was a collection of garbled gates which was a cryptographic communication protocol for two mistrusting parties without the need for any central trusted third party. STC allowed two mobile entities, say n_A and n_B (such that n_A knows only “a” and n_B knows only “b”) to compute a function $f(a, b)$ together without each disclosing what it knew to the other entity. The result of this privacy-preserving method indicated that it can leverage the mobility, location-awareness, data accessibility, etc. Though this method was more flexible than method of De Christofaro *et al.*, (2011), it incurred a huge computational cost. Apart from this, STC had been shown by (Huang *et al.*, 2011) to also have issues of integrity in the presence of malicious and selfish nodes.

De Cristofaro *et al.*, (2011) developed a privacy contact discovery method which was based on Index Based Message Encoding (IBME). They proposed a method that used friend discovery algorithm by using Okamoto’s technique with RSA in their identity-based key agreement session in a Random Oracle Model (ROM). Therefore, they achieved private contact discovery such that two nodes could only learn their common contact list and no more. This protected the network nodes against list manipulation and ensured nodes authenticity. Thus, they allowed two users to find friends whom they both have in common while the privacy remained preserved from their contact list. They further used contact certification to ensure that non-friend users cannot falsely claim friendship. They showed that the overheads in terms of communication and computation compare to the number of contact lists were approximately linear. However, they used Contact

Hiding (CH) security mechanism modeled as a game to secure against revealing any of the non-matching contacts. This model had two major weaknesses. Firstly, the CH-adversary would only find it difficult to differentiate between contact lists whose probability was considerably high. This was due to the nature of the fairness from the balance of gain of knowledge of participants' protocols during the contact discovery process. Secondly, their methods assumed that all the users were in the same social community which may never be the case in real world opportunistic environment.

Aviv *et al.*, (2012) proposed a privacy-aware geographical routing that used Probabilistic Profile-Based Routing (PPBR) to ensure OppNet's geo-routing was secured in terms of privacy without the need for initial friend discovery phase. Geo-routing used multicast protocol where a message was sent to all users in a geographical area, unlike in a unicast where a message was sent to a receiver. The method attempted to do this while preserving certain information of the users such as identity, location, mobility pattern, etc. However, in order to achieve this, the authors had to build mobility profile by splitting the area of interest (AOI) into some cells while continually updating user's coordinates in terms of longitude and latitude and its time duration spent in a cell. They demonstrated the effectiveness of their model by means of a discrete event-driven Humanet Simulator and also showed that their method "leaked" some information and that the delivery efficiency was only 30%. This model would not scale well in a challenged environment, furthermore, if the attacker was able to figure out the first stage of message delivery, the sender's privacy would be compromised.

Goyal and Chaudhary (2013) proposed a privacy preservation algorithm that was titled ensuring privacy in opportunistic (PO) network. It protected the identity (ID) of nodes that subscribed to the privacy preservation from being globally accessible in the network. In their algorithm, nodes

in the network were grouped into clusters. In each of the clusters, there was a fixed node that stored the real ID and generated a virtual ID of the rest of the nodes (which were mobile) in a cluster. If a source node wished to send a message to another node in the network environment, it sent the message through the cluster head (i.e the fixed node). The cluster head encrypted the message, generated a virtual ID of the destination and sent the message to the most suitable relay nodes. This process was repeated until the message got to the destination. Thus nodes could only send messages through the cluster head which generated session key and authenticated the entities. With this, the privacy of the destination nodes and the messages were protected from being disclosed while searching for suitable relay nodes. They showed the through put increased with introduction of the PO. However, their cluster head performed the function of a trusted third party in the network. Existence of trusted third part is not realizable in opportunistic network because nodes presence or otherwise is random.

Constantino *et al.*, (2013) developed an optimized security framework which they called Mobile Fair Play. It used a high-level procedure in a Secure Function Definition Language SFDL to optimize garbled Boolean circuit for Secure Two-Party Communication (STC) protocols. These protocols included millionaire problem, Jaccard's coefficient, set intersection, etc. They demonstrated the usefulness of their approach as a privacy-preserving interest-casting opportunistic network and implemented the framework on Android Smartphones. They claimed that their approach was secured against malicious attackers. Though, this method was able to improve the computational cost in terms of compilation times, it did not scale well in terms of secure function computation times (i.e. the time it takes to have hand-shaking per communication). This was because STC, as variant of Multi-Party Communication (MPC) has complexity that was

quadratic order of the private data involved and exponential order of the number of parties involved.

Chen *et al.*, (2014) designed a trust management protocol for DTNs and applied it to secure routing to demonstrate its utility. The results demonstrated that the trust-based secure routing protocol designed to maximize delivery ratio could effectively trade off message overhead for a significant gain in delivery ratio. They claimed that their trust-based routing protocol outperformed Bayesian trust-based routing and Probabilistic Routing using History of Encounter and Transitivity (PRoPHET) in delivery ratio and message delay as it applied the best trust formation out of social and QoS trust properties. Though their protocol performed well when the percentage of malicious nodes was relatively low, however, the protocol's performance was very low when there was a relatively high percentage of malicious nodes (e.g. 40% of malicious nodes or more) in the network. This was a serious challenge, considering the fact that an OppNet attempts to effectively utilize contact opportunities in an environment and had no control over who joined or left the network.

Distl and Hossmann (2014) investigated factors that culminated the trade-off between privacy and utility. The utility was a metric assigned through ranking to each node in the network based on the social structure of nodes' past encounters with other nodes as stored in their contact graph. They proposed a method of transformation of the contact graph using a greedy search algorithm so as to preserve the privacy of the information therein. However, the performance of their algorithm did not scale well as the network size was increased. This was due to the method they applied for calculation of utility for each node, and applying betweenness centrality in routing decisions in their algorithm. This was because the steps of determining the utility increased drastically with increase in the number of nodes in the network. And also, using betweenness

centrality as a routing metric would require determining the shortest paths which in turn requires global knowledge of the network structure. These set an upper limit to the usefulness of their work as network size increased.

Guo *et al.*, (2015) proposed an incentive based publish/subscribe asynchronous messaging paradigm which they termed ConSub. In ConSub, messages were routed/forwarded depending on its content. They employed a Tit-For-Tat (TFT) mechanism such that any pair of interacting nodes must try to satisfy the interest of each other. The degree of cooperation of node and its contact probabilities were used to determine message content utility, which in turn, determined exchange order. Therefore, nodes acted in such a way as to maximize their content utility. Thus any selfish node whose actions are primarily based on what benefitted itself without caring about how such actions would affect the network were discouraged. In terms of packet delivery and hop counts, they showed that their approach was superior to the previous ones. However, since the contents of messages were stored according to their content utility this required a lot of overhead and also computational complexity. Therefore, as the network size increased the transmission cost increased prohibitively.

Xi *et al.*, (2015) presented a trust management routing scheme that used behavioral feedback. This scheme attempted to build behavior trust in an incomplete self-organized mutual authentication process. According to them, in a harsh sparsely populated environment, a node may not be able to use its encountering opportunities to effect mutual authentications. Therefore, they used certificate chains from social attributes to construct local certificate graphs. These graphs were updated continually until identity trust could be realized from it. At the same time, a successor node generated a verified feedback packet for every cooperative behavior of its predecessor nodes. These packets together with the identity trust formed the behavior trust. They demonstrated that

their trust management scheme can effectively select trust nodes for routing even in the presence of large number of compromised nodes. However, threshold value for their selection algorithm was arbitrarily selected which if optimally determined could improve the performance of their scheme. Another issue with this scheme was that there was the need for adaptive instead of static threshold to reflect the dynamism of behavioral differences in nodes from different environment and this could be computationally expensive.

Ahmad *et al.*, (2016) proposed a technique for detecting malicious paths and malicious nodes in an opportunistic network. The detection of malicious path was modeled base on Merkle tree hashing technique. A node would build trust to destination node using detected path which was in turn used to detect the malicious nodes. By simulation, they demonstrated that their technique had high detection accuracy and could mitigate against selective drop attacks. However, their technique required long period of time to achieve node detection. This was because their technique did not consider relay nodes' co-operation and rely solely on direct trust relationship which took considerably long time to build.

Yao *et al.*, (2016) designed a dynamic trust routing based on social similarity (TRSS). They utilized social similarity to build their trust routing model. Their model established trust either directly from social behavior of a node or indirectly from recommendation of another trustworthy node. Their concept of social similarity was based on assumption that nodes encounter each other due to their common similarity or interest. And a source wishing to send a message would choose a node as the next-hop forwarder towards the destination if it had more number of common social features in its social history record. The untrustworthy nodes were thus removed from cooperative message forwarding. Their simulation results showed that their method was effective in detecting the non-cooperating (selfish and malicious) nodes. However, their method assumed that nodes will

cooperatively and willingly reveal their social profiles without being worried about their privacy. This was not true in reality. In addition, avoiding an “unknown” node as next forwarder because of having low trust value negated the essence of OppNet. These, limited the applicability of their method.

Chhabra *et al.*, (2017) developed a security protocol that used game theory to secure the network against malicious nodes by modeling the behavioral pattern of a potential threat (PT) from among nodes using history of their delivery and reputation. They incorporated their security protocol into the Spray and Wait opportunistic routing protocol and ran the simulation on ONE simulator. They demonstrated the effectiveness of their proposed security protocol in countering black hole attacks in the network using delivery probability, number of packets dropped, network overhead ratio and average transmission delay as metrics. They claimed that the protocol’s defense strategy ensured better utilization of network resources. However, the rigorous mathematical nature of their approach required a lot of processing per node in the network leading to serious trade-offs between efficiency and effectiveness.

Dhurandher *et al.*, (2017) developed a Cryptography-Based Misbehavior Detection and Trust Control Mechanism for Opportunistic Network. Their method aimed at ensuring access control in the network. Their trust mechanism could detect, avoid and isolate malicious nodes. This was achieved as a result of authentication, confidentiality and integrity from their security service. In addition to these, they proved that their method was robust and scalable and their trust could accommodate recent cryptographic techniques. However, their detection and control mechanisms were built as overlay on an infrastructure-based opportunistic network. Therefore, their method could not work in an infrastructure-less OppNet environment.

Khan and Chatzigeorgiou (2018) proposed a method of achieving security in an OppNet environment. They used Random Linear Network Coding (RLNC) techniques to reduce latency and energy consumption while maintaining security. The RLNC ensures that messages were encoded and transmitted securely from source to destination in the presence of eavesdroppers. Furthermore, they developed a framework that detects the probability of an eavesdropper intercepting and breaking the confidentiality of the message being transmitted. With extensive analysis and simulation of their proposed method, they demonstrated the effectiveness of their method. However, RLNC was a non-cryptographic method and therefore less effective compare to cryptographic methods especially in an OppNet environment where nodes could join and leave at any time and eavesdroppers' behavioral pattern may not be known priori.

All of the works considered in this review are those that do not use any trusted third party (TTP). This is because use of TPP is not feasible in OppNet environments. Majority of those reviewed are cryptographic-based. These are generally more secured than those that are non-cryptographic based. However, cryptographic based privacy preservation in OppNet incurs high computational complexities and therefore consumes the resources. This is usually due to the encryption and decryption algorithms used and method of key distribution employed.

CHAPTER THREE

METHODS AND MATERIALS

3.1 Introduction

This chapter discusses the methods and materials used to accomplish this research. It explains the step by step procedures (i.e. methods) together with the data, software and hardware (i.e. materials) used for the success of this research work. It provides vivid explanation of the methodological steps.

3.2 Methodology

The methodology adopted for this research work is as follows:

1. Development of privacy scheme:
 - a) Initializing random nodes with wireless connectivity, mobility and behavioral models, and community.
 - b) Assigning each node some sets of mobility and behavioral models and communities using probabilistic (normal) distribution.
 - c) Dividing time into sliced periods such that trustworthiness and social similarity are ranked based on the time periods.
 - d) Distributing the nodes into hierarchical community structure.
 - e) Revealing Social profiles according to community, and trustworthiness (of a community structure).
 - f) A node wishing to send a message duplicates and divides it into 3 sub-messages and puts the key of a sub-message onto a different one and encrypts using Galois theory according to the equation 3.1 using e, f and g as keys known by the destination priori and x as a function of message content.

$$\frac{d^4}{dx^4}(\sin x + ex^3 + fx^2 + gx) = \sin x \quad (3.1)$$

- g) Applying a partial onion routing scheme so that only when all the three messages arrive at the correct destination that it can be decrypted from equation 3.1 using e, f and g as keys.
 - h) Applying a hash function on social profile of anode base on ranking.
 - i) Ensuring that whenever two nodes encounter, they exchange social profile and compute social trust.
 - j) Ensuring that a message is sent to a node based on sensitivity, social trust, social similarity and rank
2. Incorporation of the developed privacy scheme into TRSS
 - a) Obtaining the data sets of Scott *et al.*, (2006) to build social similarity.
 - b) Replication of TRSS of Yao *et al.*, (2016).
 - c) Adoption of trust routing in TRSS.
 - d) Setting conditions for privacy and TRSS.
 - e) Running of the scenario on Opportunistic Network Environment (ONE) simulator.
 3. Comparing performance of the PATRSS and validating the result
 - a) Replication of PO of Goyal & Chaudhary, (2013).
 - b) Running of various scenarios of PO, TRSS and PATRSS on ONE simulator.
 - c) Comparing the performance of PATRSS with PO and TRSS using delivery ratio, delivery cost as performance metrics.
 - d) Validating the results by implementing PATRSS on Epidemic, PRoPHET and MaxProp routing algorithms using number of duplicated messages, number of aborted messages and number of delivered messages as metrics.

- e) Plotting of the results graphically to show the network performance, the improvements and the costs.

3.3 Datasets and Trust Model

This research used the public data traces of imotes as conducted in Scott *et al.*, (2006) in CRAWDAD at Cambridge. The datasets were generated from the result of experiment done by researchers in Scott *et al.*, (2006). In the experiment, two categories of imotes, stationary and mobile imotes, were used. The stationary imotes were fixed in locations with high chance of people converging there. This location included: pubs, shops, markets, etc. The mobile imotes were held by a group of people made up of researchers, students and professors and each was given an imote. The formation is illustrated in table 3.1. Although, the experiment started with 40 mobile and 19 fixed imotes, at the end only 38 mobile and 18 fixed imotes were reclaimed due to malfunctioning of some imotes before the end of the experiment.

Table 3. 1: Composition of the Imotes (Scot *et al.*, 2006)

Device	Description	No	Battery type
MSR10	Mobile Short Range with interval of 10min	40	CR-2 battery (950 mAh)
FSR10	Fixed Short Range with interval of 10min	15	CR-2 battery (950 mAh)
FSR6	Fixed Short Range with interval of 6min	2	2200 mAh
FLR2	Fixed Long Range with interval of 6min	2	2200mAh

Each of the imotes was programmed to scan its vicinity periodically every 5 seconds. Manual synchronization was used at the beginning with time stamp epoch of Unix. However, dephasing technique was applied to every two consecutive scans to prevent two devices in proximity from

trying to access each other at the same time coincidentally. Thus, an imote searched for presence of other Bluetooth devices around its proximity. If during the scan it detected a device, it prompted the device to send the device's Media Access Control (MAC) address which it stored in the memory using paging function when sent. By doing this, the recorded data had inter-contact time whose duration was more than two consecutive scanning failures.

To create identity (ID) of the MAC address for comprehensive record of social profiles of the people carrying the imotes, each of them was given a questioner form to fill in certain personal data about them before the commencement of the experiment. The data included: name, email, studies, languages, city, country, nationality, affiliation, position, project, airports, Bluetooth, attendance, stay, room-mate, metro, presenter, member, and topics. Thereafter, some sets of arbitrary characters were assigned to each person in order to secure their privacy.

When two devices encountered; the ID, the social profiles, the start and end time of the encounter were thus stored by the discovering device. Although, the contact times stored by each about its neighbor were not mutual due to scan de-phasing, each device held the trace file of devices it had contacted as it moves around in the environment. Thus the trace file of the experiment contains connectivity information of the imotes.

3.4 Trust Routing

In trust routing, forwarding of messages to a relay node is a function of past cooperation of the node in forwarding messages rightly. Trust routing based on social similarity (TRSS) allows the trust to build in the network according to social context of common interest and interaction of the nodes. This is central to improving routing performance in TRSS.

In TRSS, nodes with more social relationship at a particular time have more likelihood of meeting (i.e. encounter) in the future time, and therefore more likely to bring about higher message

delivery. However, the selfishness of the selfish nodes may not be affected by this. A selfish node may choose to drop messages so as to reduce the consumption of its buffer space and energy. This would consequently affect the expected performance of the network negatively. Due to this, the trust routing is adopted as follows:

1. The social similarity is calculated from the social features. It is then combined with behavioral corporation of nodes to form a composite metric. This metric is used to compute the social trust of a node.
2. A node decides its next hop forwarder node based on the social trust of the forwarder. This further reduces the chance of occurrence of selfishness and attacks such as trust-boosting, promise-then-drop and defamation attacks
3. Relay nodes co-cooperativeness in a network is known through direct calculation of the trust by a node or indirectly through obtaining the information from another trustworthy node. This is made possible by the use of encounter-based acknowledgement EACK in the behavioral feedback on each encounter. Once the behavior of a node is made known in the network, the cooperative can be ‘helped’ by other nodes and the misbehaving nodes can be punished by being isolated from corporative forwarding of messages.

3.5 Development of Privacy Preservation

From security point of view, privacy is the ability of an entity to have control over an information on who and what can be accessed of it. This work develops privacy preservation of the message to be sent and that of the relay nodes that cooperates in sending the message hop by hop.

3.5.1 Development of message privacy

The privacy of the message as a confidentiality subset of the security is protected by means of cryptographic mechanism. A source node wishing to send a message, m duplicates the message

into two messages, m and m_i . It divides message m_i into three equal sub-messages $m1$, $m2$, $m3$.

The characters of m_i is sequenced by position numbering p_i according to equation 3.1.

$$m_i = \sum_0^k p_i, \quad k = \text{total number of characters in } m_i \quad (3.1)$$

Division of message content into the sub-messages is done by performing transposition operation on each of the characters of m_i based on their position number p_i into $m1$, $m2$, $m3$. This is illustrated in Equation 3.1.

$$p_i \in \begin{cases} m1 & i = 3n + 1 \\ m2 & i = 3n + 2 \\ m3 & \text{otherwise} \end{cases} \quad n \in \mathbb{N} \quad (3.2)$$

Next each of the sub-messages $m1$, $m2$, $m3$ are then encrypted using Galois theorem and partial application of onion routing. This is done as follows. Consider a mathematical identity in form of fourth order homogeneous differential equation of a curvi-circular function given in equations 3.3 and 3.4.

$$\frac{d^4}{dx^4} y = \sin x \quad (3.3)$$

$$y = \sin x + ex^3 + fx^2 + gx \quad (3.4)$$

What the equation implies is that if a sub-message is changed into a function y and a differentiation operation is performed four times repeatedly on the function y , the differential coefficient which is a sinusoidal function, $\sin x$ is generated. But this resultant sinusoidal function is part of the original function y . In other words, a plaintext sub-message (represented by function, y) is encrypted into a cipher-text sub-message (represented by function, $\sin x$) such that the only way the original plaintext can be recovered during decryption is by knowing the keys (represented by e , f and g) in equation 3.4.

Conversion of a plaintext sub-message into function y is done as follows. Function y is made up of two parts: sinusoidal part which is a circular function and a curve part which is a third order (or

cubic) polynomial function, both in x . This means the most critical part of the conversion is determination of x from the plaintext sub-message.

Determination of x from the plaintext in order to generate function y is however, non-trivial. Instead this research applies Galois Theory to generate the function y and to perform the operations. Let X takes value from 0 to 2π as shown in equation 3.5.

$$X = \{x: 0 \leq x < 2\pi\} \quad (3.5)$$

The entire period (2π) is then sampled at equal interval such that the period is divided by the number of characters (q) in the sub-message according to equation 3.6 (Backer, 2012).

$$x[q] = \frac{2\pi}{q} \quad (3.6)$$

This is then used as the domain of the function y . The codomain is quantized into Galois field $GF(2^7)$. Each element of this field corresponds to an American System of Code for Information Interchange (ASCII) character. The 7 bits of the resultant character's ASCII code is exclusively ored (XOR'ed) with the corresponding plaintext sub-message's character to generate its encrypted representative. Thus repeating this with each character of the plaintext of the sub-message, the sub-message's cipher-text is generated.

Next, the key, (k_1) of a different sub-message (m_1) is inserted into the encrypted sub-message ($E(m_2)$) and further encrypted with key (k_2) as above. However, this time, only the quadrature of the sinusoidal part of the function y is used. The stages are shown in Figure 3.3.

In other words, each of the three sub-messages m_1 , m_2 and m_3 is encrypted with keys k_1 , k_2 and k_3 to generate sub-messages $E(m_1 \leftarrow (k_1, k_2, k_3))$, $E(m_2 \leftarrow (k_1, k_2, k_3))$, and $E(m_3 \leftarrow (k_1, k_2, k_3))$. This encryption is nested into another encryption of which only the quadrature of the initial encryption is used. However, a key is multiplied by a cipher-text sub-message and encrypted with a different key as illustrated by equation 3.7.

$$M_{i+1} = E^1(k_i * E(m_{i+1} \leftarrow (k_i, k_{i+1}, k_{i+2}))) \text{ where } i = \{n: 1, 2, 3 \in \mathbb{N} \bmod 3\} \quad (3.7)$$

The structure of the final message routed is an onion structure (i.e. nested encryption) but partially sent through different opportunistic relay nodes. The structure is shown in Figure 3.2.

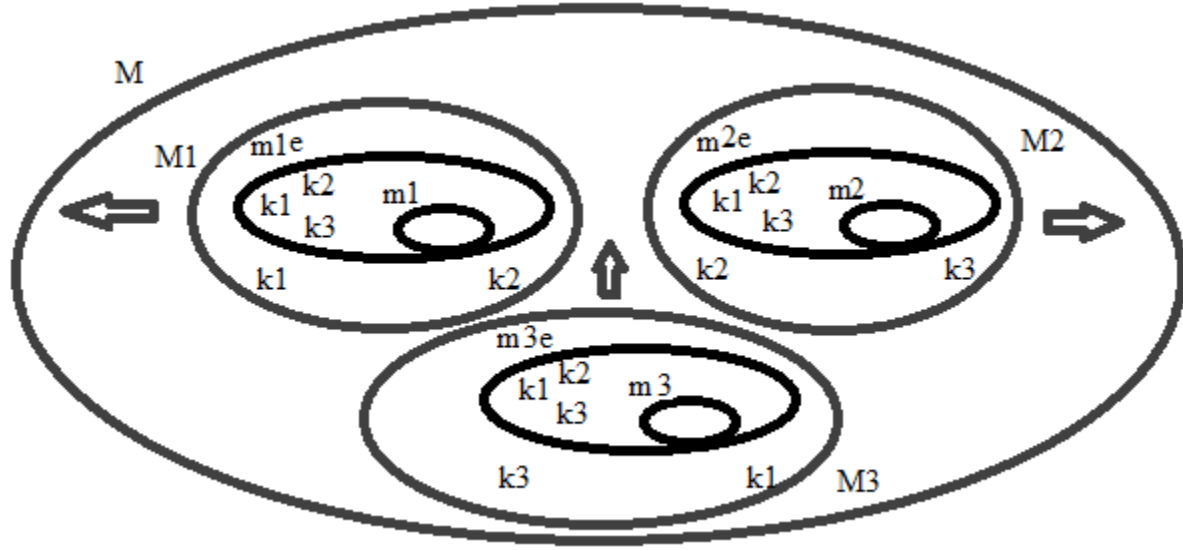


Figure 3. 1: Onion Routing Structure of the Message

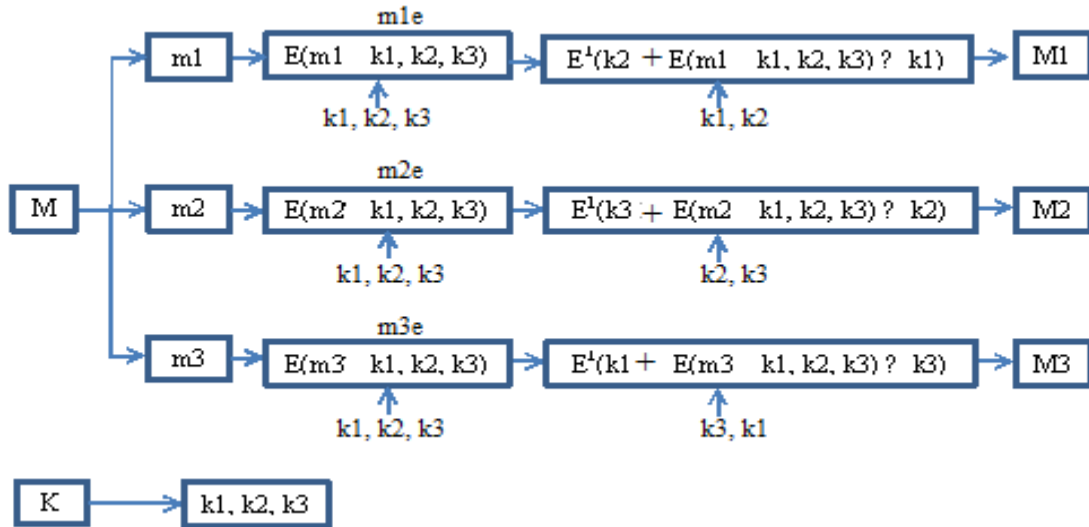


Figure 3. 2: Stages of Encryption

3.5.2 Development of relay node's privacy

All the nodes that helps in forwarding a message hop by hop between a source and the destination node are the relay nodes. The nodes are initialized with wireless connectivity so that messages can be relayed on opportunistic contacts. The connectivity pattern is modelled from the datasets of Scott *et al.*, (2006). These nodes are, pedestrians, cars, buses, and trains. The numbers of pedestrians are relatively more than others, usually between 30% and 70% of the total nodes while that of the trains is least, usually below 10% of the total nodes. Mobility is assigned based on these groups in terms of maximum and minimum allowable speeds and the allowable set of tracks on which each group can move. Nodes are further distributed in community such that the communities have hierarchical characteristics modeled as tree structure. To minimize complexity, three level hierarchy is used. The corporation of the nodes (i.e. behavior), and their mobility, and community use normal probabilistic distribution as in equation 3.8 (Backer, 2012).

$$f(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (3.8)$$

Each of the behavioral corporation in relaying a message, mobility (of a group) and community of a node is a random variable (X), normally distributed with mean (μ) and standard deviation (σ). Therefore, the probability of the random variable X being found within an interval $x=a$ and $x=b$ is shown in equation 3.5 (Backer, 2012). Therefore, x is specific value of X.

$$P(a < X < b) = \int_a^b f(X)dx \quad (3.9)$$

Since the nodes are 'social', their characteristics are not equally distributed at all time. For example, nodes move around their community more in the day time and less otherwise. In other words, the mobility is nomadic in day time (12 hours) and sedentary during night time (12 hours) periods. This means that interaction among nodes is time dependent. Therefore, time is divided

into sliced periods such that trustworthiness and social similarity is ranked based on the time periods.

Social profiles of a node are hashed. Since the size of data and the number of nodes used in this work are not large, table look up is employed. Here, the profile of a node is given arbitrary set of characters in the look up table. An “unknown” potential relay node is sent only the hashed values. Whereas, trustworthy members of a community keep records of both the social profiles and the corresponding hashed values. Depending on the community of a node, their trustworthiness and social interactions among members, the social trust and social similarity are ranked.

A message to be sent by a source node is first ranked as either not sensitive (s1), sensitive (s2) or very sensitive (s3). A node would send a message to next relay node based on the ranking of message sensitivity, social trust of next relay node and social similarity of it with the destination node.

3.6 Incorporation of Privacy into trust routing

Consider a social feature vector F with r numbers of social features and f_i is the i th social feature in the social feature table according to equation 3.10:

$$F = [f_1, f_2, f_3, \dots, f_i, \dots, f_r] \quad (3.10)$$

When two nodes (for example, node “a” and node “b”) that often encounter, meet they exchange social features of the nodes they have encountered priori. This is if trust of both nodes are already established ($T_{b,a} > 0.5$). If not the hashed values of the social features are used. Thereafter, each node updates its history record. From this information, the Social similarity ni, D between node n_i and destination node D is computed thus with equation 3.11 (Yao *et al.*, 2016) where ni, D is the social similarity between relay node “ i ” and destination node D :

$$ni, D = 1 - \sqrt{\frac{\sum_{j=1}^r \alpha_j (1 - \frac{N_{jk}}{\sum_{k=1}^p N_{jk}})^2}{r}} \quad (3.11)$$

This is in accordance with the definition of trust from social sciences. It is interpreted here, as the likelihood that the trustee will behave as the trustor expected rightly. From the social similarity, trust T of node “b” found by node “a” is initialized in accordance with equation 3.12 (Yao *et al.*, 2016).

$$T_{b,a}^{new} = f(f^{-1}(T_{b,a}) \pm T_{a,b}) \quad (3.12)$$

In order to ensure that the value of $T_{b,a}$ is not outside [0,1], a normalizing function $f(x)$ according to in equation 3.13 is used. In this case, β affect the rate of adjustment so that, the $T_{b,a}^{new}$ becomes:

$$T_{b,a}^{new} = \begin{cases} T_{b,a} e^{\beta T_{a,b}}, & y < 0 \\ 1 - (1 - T_{b,a}) e^{\pm \beta T_{a,b}}, & y \geq 0 \end{cases} \quad (3.13)$$

Where: $y = f^{-1}(T_{b,a}) \pm T_{a,b}$

To calculate the indirect (or recommended) trust, a transitive trust is used. This means evaluating the trust that a node (na) has on on another node (nc) due to the recommendation of a trustworthy node (nb). This is indicated in equation 3.14.

$$T_{a,c}^{new} = (T_{b,c} - T_{a,c}) \cdot T_{a,b} + T_{a,c} \quad (3.14)$$

The forwarding of messages is done such that if a known node is encountered its trust is evaluated if it is found to be trustworthy ($T_{ba} > 0.5$), TRSS is used. If it is found not to be trustworthy, and there is a high likelihood of encountering the destination node, the onion routing is employed with hashed values of social features for privacy. The same is done if the node is unknown. This gives PATRSS the ability to utilize more opportunistic encounter than the TRSS. This is summarized in Figure 3.3. From the flow chart the block “T*Rout” is the TRSS of Yao *et al.*, (2016).

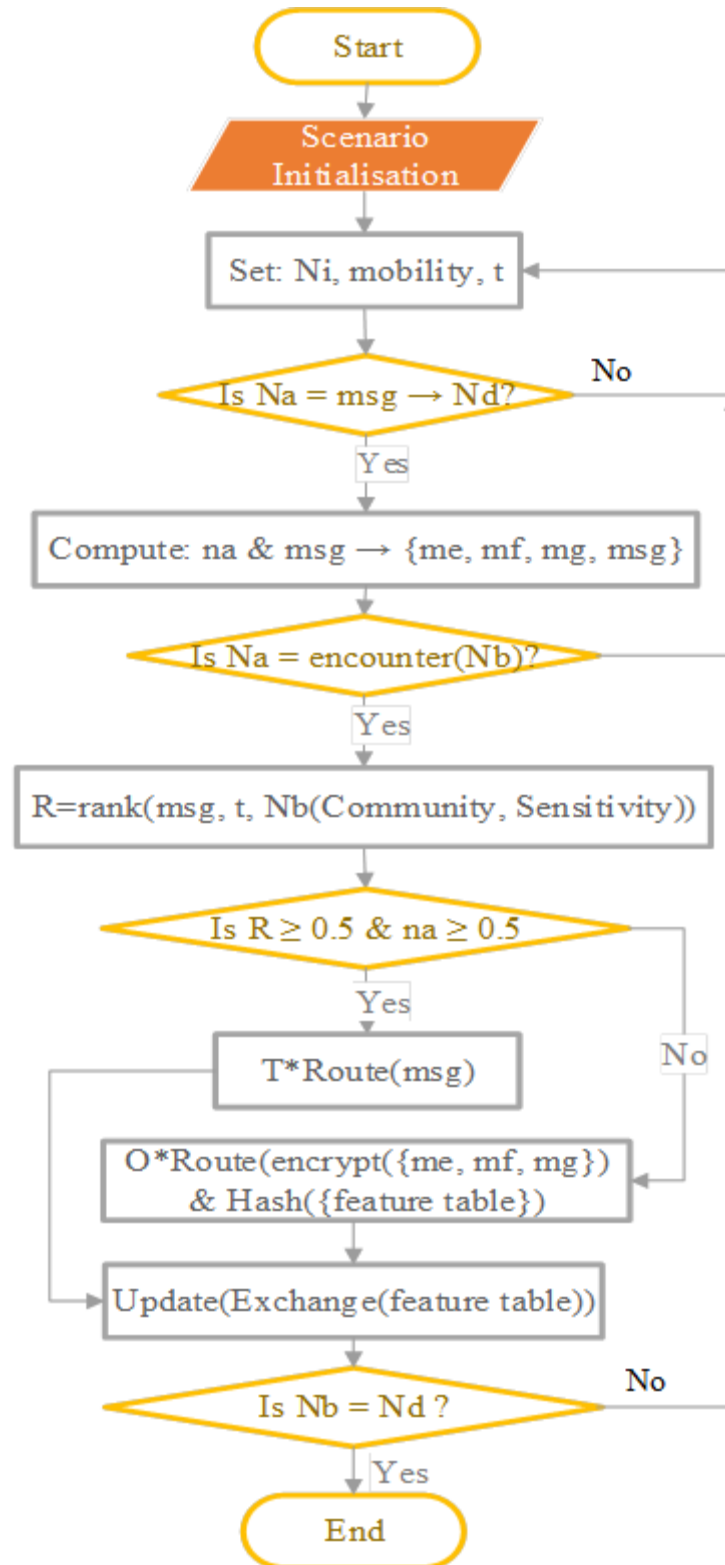


Figure 3. 3: Flow Chart of the Privacy-Aided TRSS

3.7 Modelling the Datasets

The procedure for downloading, setting up and running of the Eclipse Integrated Development Environment (IDE) and Opportunistic Network Environment (ONE) simulator are explained in Appendices A1 and A2. Datasets to generate connectivity are obtained from Scott *et al.*, (2006). It was downloaded from CRAWDAD website after registration and given the password via email to access and download the data with terms and conditions. The datasets consist of data of some researchers and students who held wireless devices (imotes, and mobile phones) such that the imotes records each time it contacts another wireless device. It keeps the records of the identity (ID), starting time and end time of each of each contact. The ID contains 21 sets of data about the person holding the wireless device. However, this work uses only 6 out of these that has the highest entropy (more information). They are nationality, country (of residence), city (of residence), language, job position and affiliation.

These sets of information are randomly distributed to the nodes in the play field of the simulation environment. Next, these nodes were randomly modeled to behave socially in their opportunistic routing pattern using probabilistic distribution. The java code to implement this is shown in Appendix B1.

3.8 Model Test Case

The TRSS and the PATRSS are developed on Spray and Wait opportunistic routing algorithm. The general settings of the scenarios involved are explained thus. 100 nodes are used. The play field used Helsinki map-based environment which covered an area of 450 x 340 square meters. Bluetooth devices are used with transmission of 54Mb/s each having a transmission radius of 10m. Messages are generated randomly with time to live of 400 through 600 seconds. The message size is 0.5 through 1 MB. For each scenario, 20 simulations are run and the average (arithmetic mean)

of the results is taken. The nodes are grouped such that there are: 48 pedestrians (p), 30 cars (c), 20 buses (b) and 2 trains (t). The speed of movement depends on the group and it varies from 0 through 1.5m/s for pedestrians, 15 m/s for trains, 25 m/s for buses and 30m/s for cars. Random way point mobility model is used for all the nodes throughout the simulations.

3.8.1 Simulations

This section gives a brief overview of the simulation of the privacy in OppNet (PO), Trust Routing based on Social Similarity (TRSS) and the developed privacy aided trust routing based on social similarity (PATRSS).

3.8.1.1 Simulation of PO

In this section the work of Goyal & Chaudhary, (2013) which is titled ensuring privacy in opportunistic network (PO) is replicated and run on the ONE simulator. The work is based on division of the network into clusters such that in each cluster there is a fixed node (FN). This node holds the database of the cluster members. This database contains the identification (ID) which consists of username and password of the members of the clusters. Communication takes place in the network using the FN as the intermediary between the message source and destination by message encryption/decryption while the FN holds and send the session key after authentication.

3.8.1.2 Simulation of TRSS

The Trust Routing based on social similarity is replicated and simulated on the ONE simulator. The scenario settings given in Section 3.8 is used. The java source code is shown in Appendix B2. An essential step in message transmission through the intermediate nodes is shown in Figure 3.4.

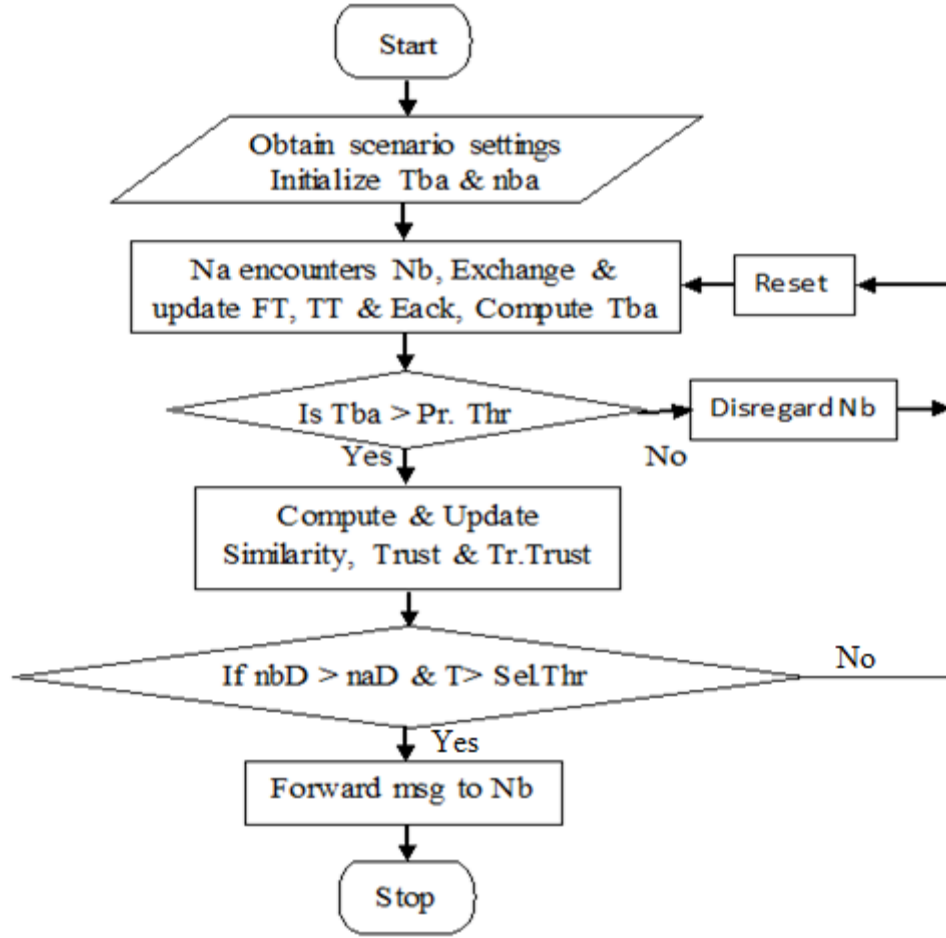


Figure 3. 4: Message Forwarding in TRSS

3.8.1.3 Simulation of PATRSS

Privacy is incorporated into the TRSS to form the PATRSS. This is compiled and run in the ONE simulator using the settings given in section 3.2. The java source code of the PATRSS is given in Appendix B3. Message forwarding steps in PATRSS is summarized in flow chart in Figure 3.5.

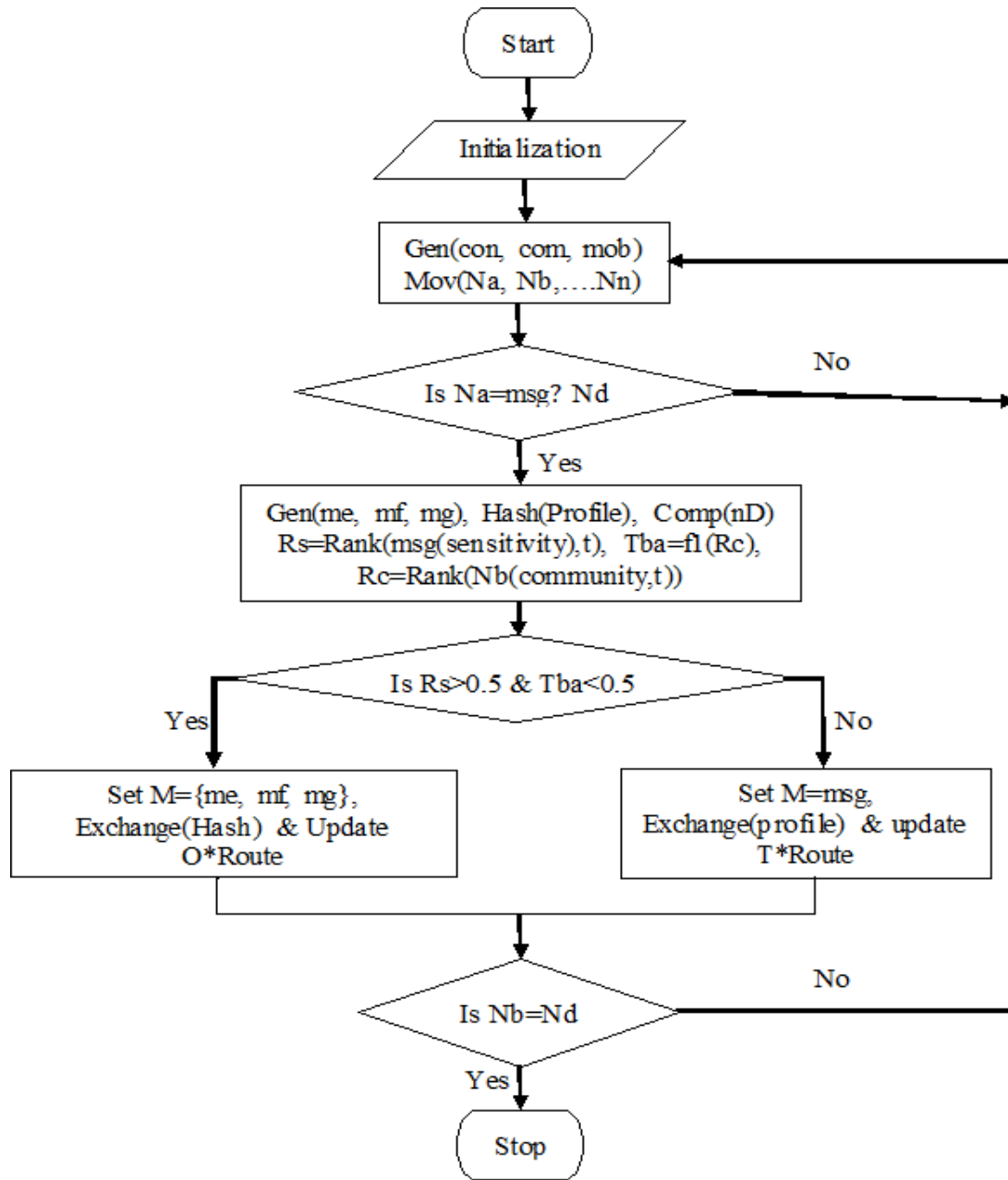


Figure 3. 5: Message forwarding in PATRSS

3.8.2 Validating of PATRSS in epidemic PROPHET and MaxProp

PATRSS is initially routed on Spray and Wait routing protocol. In this section, it is routed on Epidemic, PROPHET and MaxProp routing protocols to validate the results in the Spray and wait. In each of the cases 1200 messages are created.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This chapter discusses the results of this research work. The privacy aided trust routing based on social similarity (PATRSS) is simulated in the ONE simulator environment, the results are compared with privacy in OppNet (PO) which has privacy without trust and trust routing based on social similarity (TRSS) which has trust without privacy. The performance is further compared with that of four common opportunistic routing protocols and the results are graphically presented.

4.2 Dataset Modelling

The hagggle dataset of Scott *et al.*, (2006) is modeled in terms of the contact time as shown in Figure 4.1. It can be observed from the graph that that most of the initial contacts take place at the first one third of the period. This work is then modeled to reflect this using simulation time of 24 hours (one day).

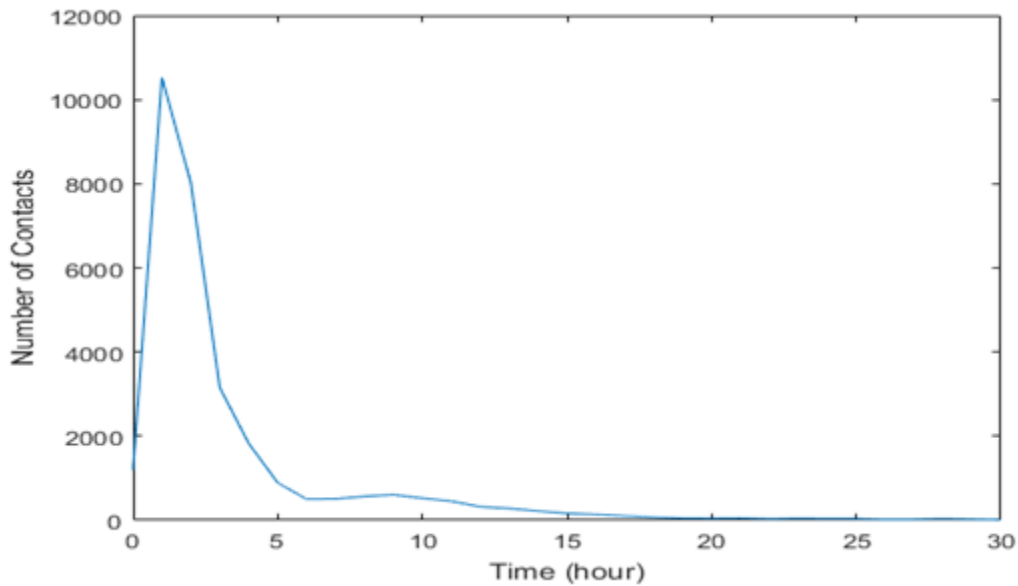


Figure 4. 1: Modeled Datasets

4.3 Performance of PATRSS in Terms of Delivery Ratio

Delivery ratio of a scenario is the total number of messages generated and sent by the source node to the total number of messages successfully delivered to the destination node. It gives a measure of transitivity of messages through the OppNet which in turn “tells” the level of co-operation of the relay nodes. As a security performance measure, it gives the level of safety in relaying of messages in the network against all non-cooperative (selfish and malicious) behavior of nodes that can hamper successful delivery of messages to the destination.

The delivery ratio of PATRSS, in comparison to that of PO and TRSS is measured by taken the average (arithmetic mean) value of it in 20 simulations per scenario. The general settings for simulation timing is shown in Table 4.1.

Table 4. 1: Simulation Timing

Simulation time	Update Interval	Message TTL	Warm up time
24h = 86400sec	0.1sec	5h = 300min	100sec

The commands used in the source code in the generating of the result include the following.

```
Report.nrofReports = 1
```

```
Report.reportDir = reports/Suleiman_result/DelRat
```

```
Report.report1 = MessageStatsReport
```

The delivery ratio of PATRSS, PO and TRSS as the percentage of misbehaving nodes is increased from 0% to 80% of the total number of nodes in the network scenario is shown graphically in Figure 4.2.

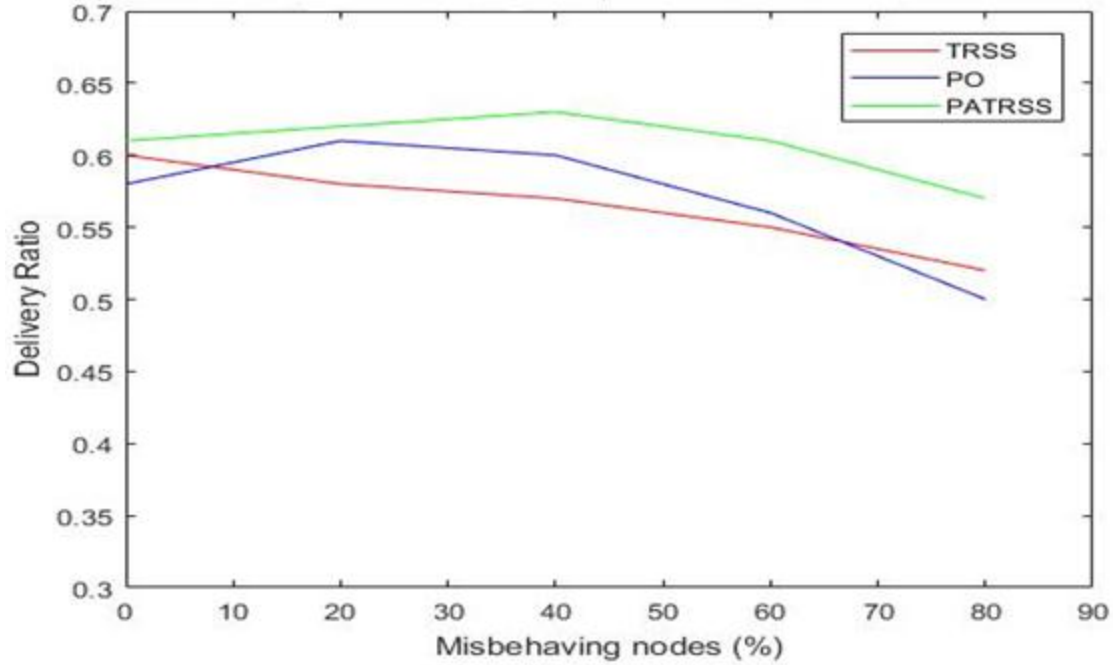


Figure 4. 2: Delivery Ratio against Percentage of Misbehaving Nodes

The delivery ratio of PATRSS is fairly constant and generally higher than the corresponding PO and TRSS as the percentage of misbehaving nodes increases. This is because as the percentage of misbehaving nodes increases in the network, the number of nodes TRSS uses to relay messages successfully is reduced thereby reducing the number of messages that are successfully delivered during the simulation time leading to reduced delivery ratio. Whereas in PO any available node is used, therefore, its delivery ratio is relatively higher compare to TRSS but less than the PATRSS. This is due to the absence of trust mechanism in the PO. In contrast, PATRSS divides its messages into three sub-messages whenever the trust of the relay node is below threshold, this reduces the risk of a complete message being silently dropped or sent to a wrong node by a misbehaving node. Consequently, PATRSS performs better.

4.4 Performance of PATRSS in Terms of Delivery Cost

Delivery cost in relation to this work is the cost of duplication of messages and its delay in the buffer in order to ensure successful delivery. In other words, it is the effort expended through

replication and buffering in taking a message from the source to the destination. Mathematically, it is calculated as the total number of messages created together with the number of messages duplicated divided by the total number of delivered messages multiply by the number of hop count.

Figure 4.3 shows the delivery cost of the PATRSS, PO and TRSS.

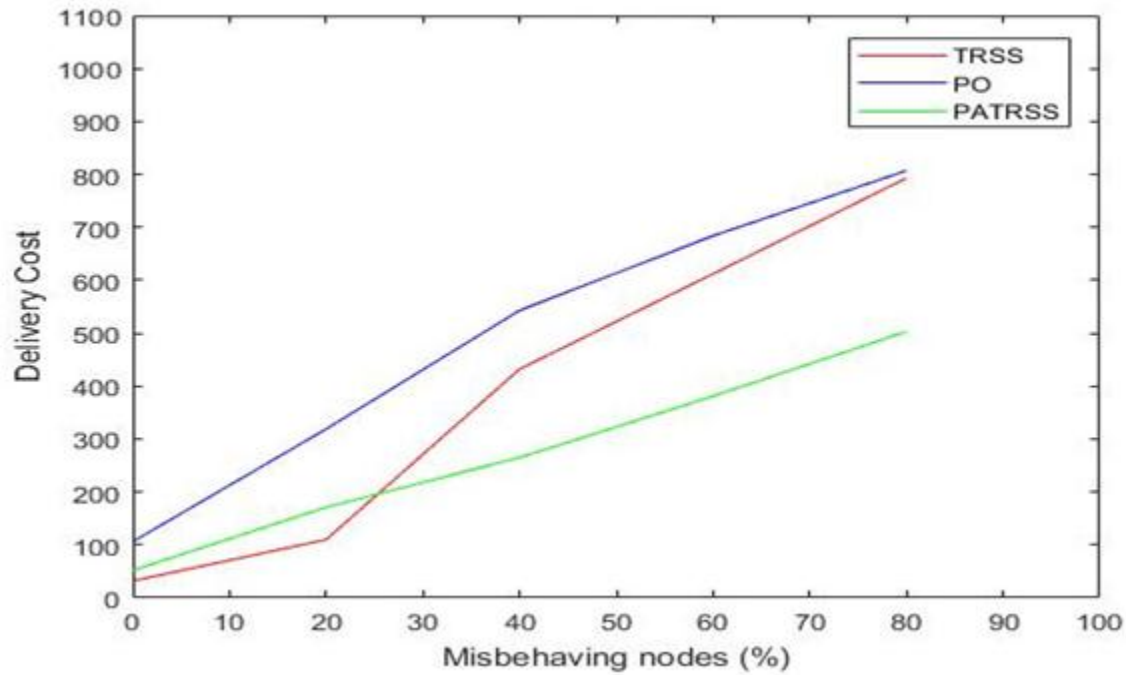


Figure 4. 3: Delivery Cost Against Percentage of Misbehaving Nodes

As the percentage of misbehaving nodes increases in the network environment, the chance of successful transfer of messages from the source to the destination decreases, hence a lot of effort is expended for each successful delivery. Therefore, the delivery cost of the PATRSS, PO and TRSS increase. The delivery cost of PO, if all the nodes are cooperative, is the lowest of the three. This because the PO uses heuristics that works based on clustering of nodes, and for each cluster a stationary node is made available that co-ordinate activities of its members. This results in detection of successful relay of messages which lead to less duplication and consequently results in low delivery cost. However, as the percentage of the misbehaving nodes in the network increases, there is less co-operation of nodes. This results in poor coordination of the cluster

member therefore the delivery cost increases significantly with increase in the percentage of misbehaving nodes. On the other hand, the TRSS uses its direct and indirect trust evaluation through its encounter based acknowledgment feedback mechanism to access the cooperation of a node in message forwarding and therefore avoid any node whose trust is below the trust threshold. Consequently, the delivery cost increases almost linearly as there is more duplication of messages since progressively there is less number of cooperative nodes in the network environment as the percentage of misbehaving nodes increases.

In contrast, PATRSS works with heuristics that combine privacy and trust, the delivery cost is less than PATRSS but higher than PO when all the nodes are cooperative. As the percentage of misbehaving nodes increases, delivery cost in PATRSS does not increase steeply. This is due to the partial onion routing mechanism that is applied. A message is divided into three sub-messages and each encrypted in nested form and sent. If a selfish or malicious node chooses to drop, corrupt or sent the message to a wrong relay or destination node, this can only occur to a fraction of the message ($1/3$) and not the complete message yet such action is detected through the feedback mechanism and such node is isolated. With this, duplication is avoided. This result in less harm from the misbehaving nodes which result to significantly less delivery cost compare to the TRSS as the percentage of misbehaving nodes increases in the network environment.

4.5 Performance of PATRSS in Terms of Average Trust Value

Trust is a measure of confidence that a trustor node has on a trustee node that the trustee node will behave co-operatively in the network environment. It is the probability that a node will forward a message it receives to most suitable relay or destination node as the trustee node receives it. The average trust value is the arithmetic mean of the trust value over a period of simulation time as sampled at regular time interval. The faster trust increases in the network the better the security of

the network. In this research, it is used as a performance index to indicate how privacy preservation lead to a higher trust value. Using an arbitrary trust threshold of 0.5, the average trust value with time is shown in Figure 4.4. PO does not involve trust, as such trust values of TRSS and PATRSS are obtainable.

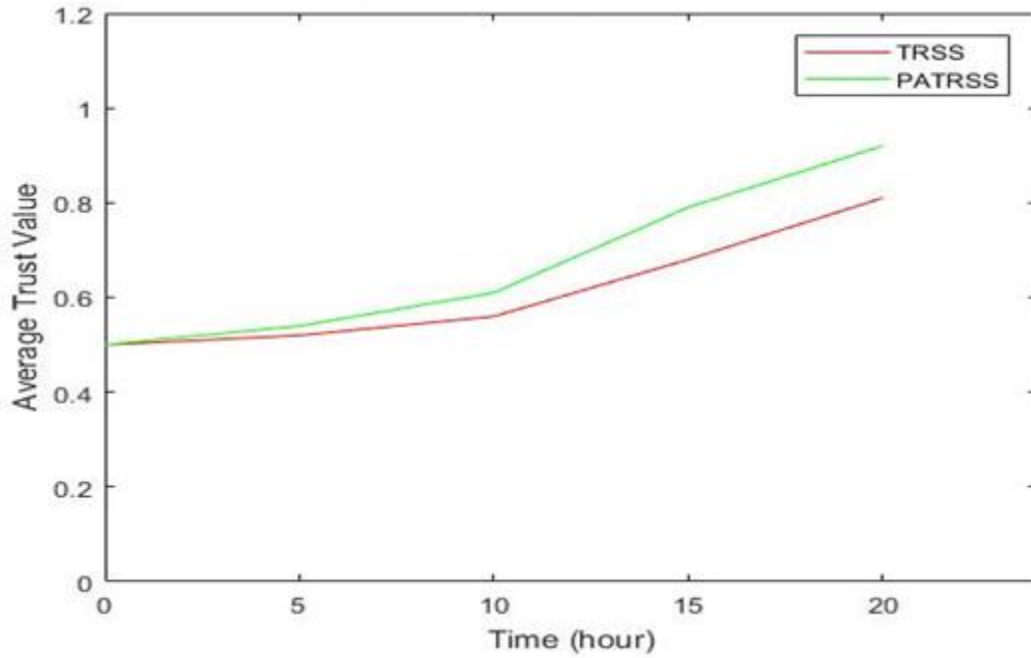


Figure 4. 4: Variation of Average Trust Value with Time

It is observed that the average trust value builds faster in PATRSS than TRSS. This is because while TRSS completely avoid any node whose trust value is below the trust threshold, PATRSS gives unknown nodes the chance to route and if such nodes behave cooperatively their trust value is increased and vice versa. And nodes that are co-operative are rewarded by having other nodes in the network helping them to route their messages while those that are not are isolated. These lead to higher trust value in PATRSS compare to TRSS.

4.6 Mean Performance of PATRSS TRSS and PO

The arithmetic mean of the performance of PATRSS, TRSS and PO in terms of mean delivery ratio, mean delivery cost and mean trust value are calculated based on Figure 4.2, Figure 4.3 and Figure 4.4. The values are presented in Table 4.2.

Table 4. 2: Mean Performance of PATRSS TRSS and PO

	Mean Delivery Ratio	Mean Delivery Cost	Mean Trust Value
PO	0.57	491.8	-----
TRSS	0.564	395.8	0.614
PATRSS	0.608	274.4	0.672

From the table, percentage improvements are found using equation 4.1 and presented in table 4.2. The benchmark in the equation in each case refers to the performance of either TRSS or PO to which that of PATRSS is being compared.

$$\% \text{Improvement} = (\text{PATRSS} - \text{Benchmark}) \times 100\% \div \text{Benchmark} \quad (4.1)$$

Table 4. 3: Percentage Improvements

	Mean Delivery Ratio (%)	Mean Delivery Cost (%)	Mean Trust Value (%)
PO	6.7	44.2	-----
TRSS	7.8	30.7	9.4

4.7 Validation of PATRSS with Common Routing Protocols

The PATRSS which uses Spray and Wait routing protocol initially is routed and reported in this section using common opportunistic routing protocols. The routing protocols are epidemic (flooding based), PROPHET (Encounter and Transitivity based), and MaxProp (probability based).

In each of the cases 1200 messages are created and routed from the source nodes to the destination nodes.

Figure 4.5 shows the number of started messages without using the PATRSS (i.e WPATRSS) and compare to using the PATRSS in a bar chart. Started messages consists of the initial number of messages created together with those replicated by the relay nodes to ensure delivery. In the Spray and Wait, Epidemic, P_{Ro}PHET and MaxProp the number of started messages are respectively 49.9%, 23.1%, 36.6% and 38.3% lower in PATRSS compare to WPATRSS. There is reduction in the number of started messages in the PATRSS in the routing protocols. This implies that the number of replicated messages reduced which means there is improvement in cooperation among the nodes using PATRSS in all the four routing protocols.

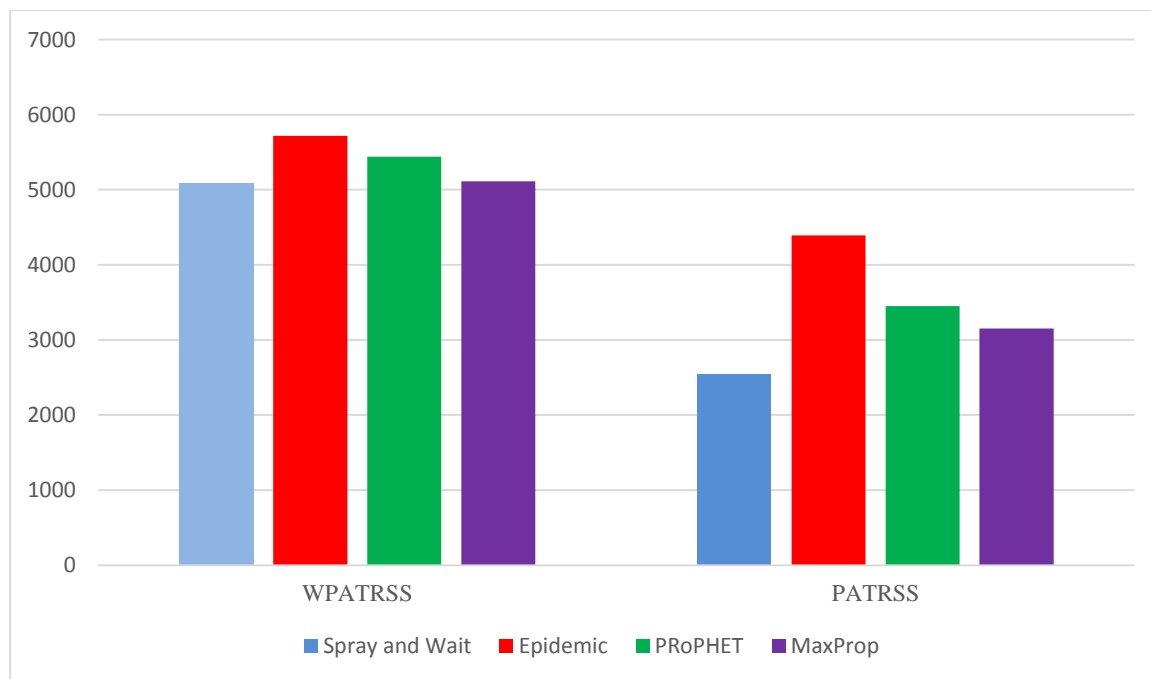


Figure 4. 5: Number of Started Messages where (started = created + duplicated) Messages

Figure 4.6 displays a bar chart showing the number of aborted messages in WPATRSS compare to PATRSS. Aborted messages mean those messages that do not get to the destination. They

consist of messages that are deleted from the buffer when they stay too long in the buffer, that is, more than the message time to live (TTL). This is in addition to those that are deliberately dropped by a relay malicious relay node or selfish relay node in order to conserve energy, buffer space, etc. In the Spray and Wait, Epidemic, P_{Ro}PHET and MaxProp the number of started messages are 21.2%, 3.4%, 7.9% and 11.9% lower in PATRSS compare to WPATRSS. In all the four routing protocols, there is reduction in the number of aborted messages in the PATRSS. This implies improvement in the security, and consequently improvement in efficiency of message routing.

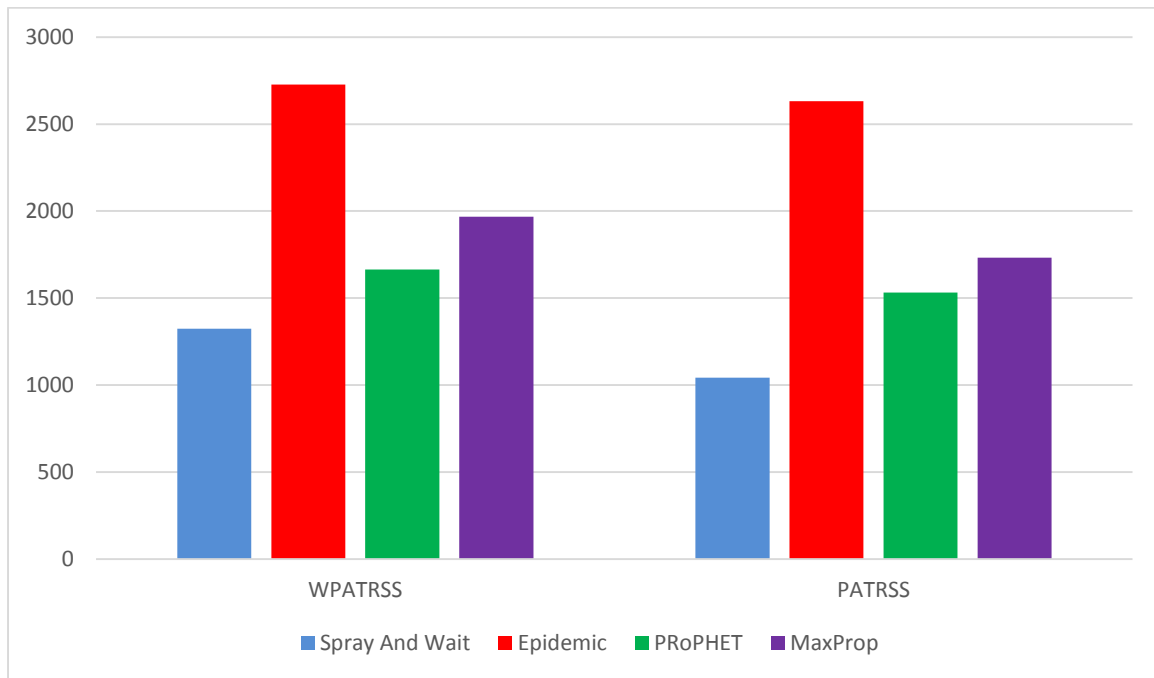


Figure 4. 6: Number of Aborted Messages where (aborted = deleted + dropped) Messages
Figure 4.7 shows the number of delivered messages. There is improvement of 12.6%, 31.1%, 15.5% and 38.5% in Spray and Wait, Epidemic, P_{Ro}PHET and MaxProp respectively with PATRSS over WPATRSS. The improvement in the delivery ratios imply that PATRSS is effective in routing messages.

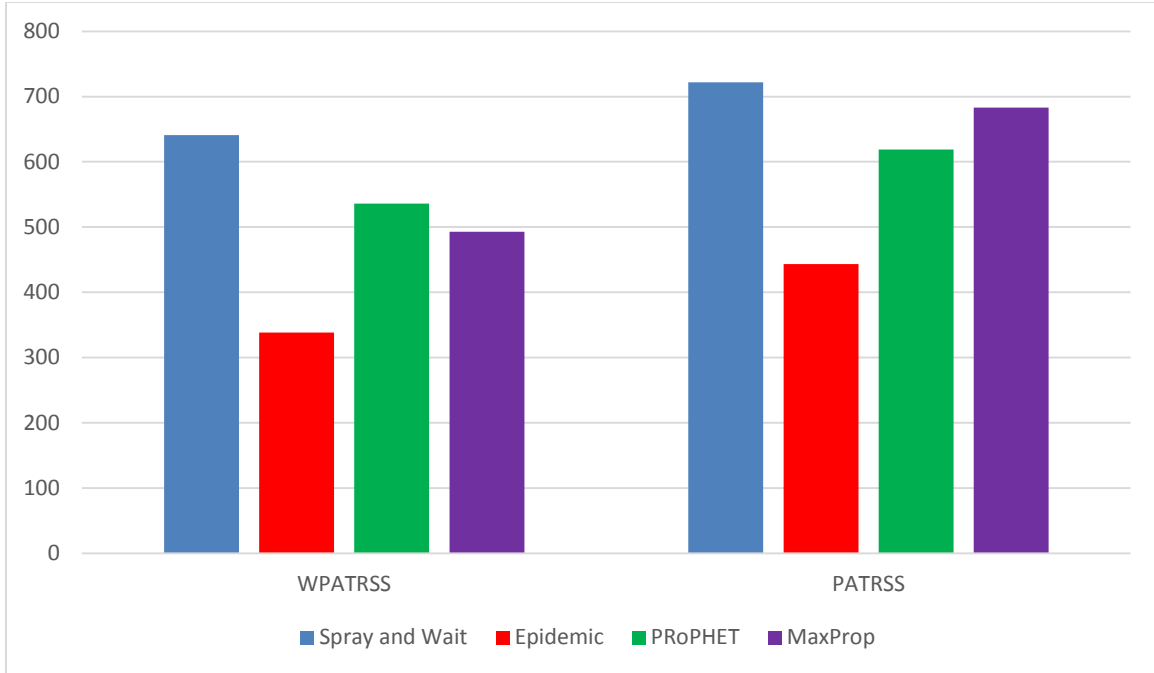


Figure 4. 7: Number of Delivered Messages

These demonstrate that irrespective of the four routing protocol used, OppNet with PATRSS performs better than without PATSS (WPATRSS). Therefore, improvement in security by incorporating privacy into trust routing in OppNet results in more cooperation among the nodes and hence improvement in network performance.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter concludes and discusses the contribution to knowledge, and recommendation for future work on this work.

5.2 Summary

This research work uses an effective cryptographic security with an efficient partial onion routing approach to achieve the development of privacy aided trust routing. The developed privacy is incorporated into the TRSS to form PATRSS. With preservation of privacy, nodes in the network could participate in the forwarding of messages whether or not their trust is established. By simulation on the ONE simulator, the performance of PATRSS is compared with TRSS (without privacy) and PO (without trust). The delivery ratio, delivery cost and trust performance, improve by 7.8%, 30.7% and 9.4% respectively compare to the TRSS. The delivery ratio and the delivery cost also improve by 6.7% and 44.2% respectively compare to PO. The PATRSS is then validated with Epidemic, PRoPHET and MaxProp routing protocols.

5.3 Conclusion

Security and routing are two of the most challenging problems in opportunistic network. This work demonstrates that by improving the security of an opportunistic network which is done by incorporating privacy preservation into trust routing, there is more cooperation among the nodes resulting to better network performance.

5.4 Contribution to Knowledge

The contribution of this work to knowledge is itemized as follows:

1. Development of framework for symmetric cryptography using Galois theory and onion routing for OppNets

2. Incorporation of privacy preservation into trust routing in OppNet

5.5 Recommendations for Further Work

The following recommendations can be considered for improvements on this work

1. Providing a comprehensive analysis of the processing stages and use of buffers to reduce the processing costs and buffer spaces used.
2. This work has assumed a fixed trust threshold of 0.5. A dynamic trust threshold that is network dependent can be developed and used for improvement.
3. Suitable machine learning algorithms can be used to select and fine tune the mobility patterns of the nodes.

REFERENCE

- Abdelkader T., Naik K., Nayak A., Goel N. and Srivastava V. (2013). SGBR: A routing protocol for delay tolerant networks using social grouping, *IEEE Transactions on Parallel and Distributed Systems*, 24(12): 2472–2481.
- Ahmad A., Alajeely M. and Doss R. (2016). Establishing trust relationships in opportunistic networks using Merkle trees. *2016 8th International Conference on Communication Systems and Networks (COMSNETS)*. 978-1-4673-9622-6/16.
- Albuquerque J. C. D., Lucena S. C. D. and Campos C. A. V. (2016). Evaluating data communications in natural disaster scenarios using opportunistic networks with unmanned aerial vehicles. *19th International Conference on Intelligent Transportation Systems (ITSC) 2016*. Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4.
- Arafath M. S., Khan K. U. R. and Sunitha K. V. N. (2017). Opportunistic sensor networks: A survey on privacy and secure routing. Retrieved from: <https://www.researchgate.net/publication/317184962>
- Arastouie N. and Sabaei M. (2016). Data forwarding scheme to minimize end-to-end delay in opportunistic networks. *8th International Symposium on Telecommunications (IST'2016)*. DOI: 978-1-5090-3435-2/16
- Atul K. (2013). *Cryptography and network security*. Third edition. Mc Grawhill (India) Private Limited New Delhi. ISBN 13: 978-1-25-902988-2
- Aviv A. J., Sherr M., Blaze M. and Smith J. M. (2012). Privacy-aware message exchanges for geographically routed human movement networks. *Computer Security—ESORICS*, 7459, 181–198. Springer Berlin Heidelberg.

- Baglioni E., Becchetti L., Bergamini L., Colesanti U. M., Filipponi L., Vitaletti A. and Persiano G. (2010). A lightweight privacy preserving SMS-based recommendation system for mobile users. *In ACM RecSys*, 2010.
- Backer A. (2012). An Introduction to Galois Theory. School of Mathematics & Statistics, University of Glasgow. Retrieved from <http://www.maths.gla.ac.uk>
- Balasubramanian A., Mahajan R., Venkataramani A., Levine B. and Zahorjan J. (2008). Interactive WiFi connectivity for moving vehicles. *In Proceedings of ACM SIGCOMM*, 2008. 427–438.
- Bamrah A., Woungang I., Barolli L. and Takizawa M. (2016). A Centrality-Based History Prediction Routing Protocol for Opportunistic Networks. *10th International Conference on Complex, Intelligent, and Software Intensive Systems*. 2016
- Basira, Y. (2015). Improved integrated routing protocol for opportunistic network with congestion control using pre-emptive data eviction. *A Dissertation Submitted in the Department of Electrical and Computer Engineering, Ahmadu Bello University, Zaria in Partial Fulfillment of the Requirements for the Award of Master of Science (M.Sc.) Degree in Computer Engineering*. (April, 2015)
- Bekmezci L., Sahingoz O. K. and Temel S. (2013). Flying ad-Hoc networks (FANETs): A survey. *Ad Hoc Networks*. doi.org/10.1016/J.ADHOC.2012.12.004.
- Bigwood G. and Henderson T. (2011). IRONMAN: Using social networks to add incentives and reputation to opportunistic networks. *IEEE International Conference on Privacy, Security,*

Risk, and Trust, and IEEE International Conference on Social Computing, DOI: 978-0-7695-4578-3/11

Bjurefors, F. (2012). Measurements in opportunistic networks. *Dissertation for the degree of Licentiate of Philosophy in Computer Science Division of Computer Systems Department of Information Technology Uppsala University, Sweden*. ISSN 1404-5117.

Bouroumine A., Zekraoui M. and Abdelilah M. (2016). The influence of the opportunistic vehicular networks on smart cities management: Study case on Agdal District in Rabat City. DOI: 10.1109/CISIS.2016.143

Burgess J., Gallagher B., Jensen D. and Levine B. N. (2006). MaxProp: Routing for vehicle-based disruption-tolerant networks. *In Proc. IEEE INFOCOM*, April 2006.

C'amara D., Frangiadakis N., Bonnet C. and Filali F. (2011). Vehicular delay tolerant networks. *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts, IGI Global*. ISBN: 9781609600426.

Camenisch J. and Lysyanskaya A. (2005). A formal treatment of onion routing. *International Association for Cryptologic Research 2005*. V. Shoup (Ed.): Crypto 2005, LNCS 3621, pp. 169–187, 2005.

Camp T., Boleng J., and Davies V. (2002). A survey of mobility models for ad-hoc network research. *Wireless Communications and Mobile Computing: Special issue, 2002*.

Chapman P., Huang Y. and Evans D. (2011). Privacy-preserving applications on smartphones. *In Proceedings of the 6th USENIX Conference on Hot Topics in Security, HotSec'11*, Berkeley, CA, USA, 2011. USENIX Association.

- Chen, I.-R., Bao F., Chang, M. J. and Cho, J.-H. (2014). Dynamic trust management for delay tolerant networks and its application to secure routing. *International Journal of Advanced Research in Computer and Communication Engineering*. ISSN (Print): 2319-5940 ISSN (Online): 2278-1021. Vol(3), Issue(1), January 2014
- Chhabra A., Vashishth V. and Sharma D. K. (2017). A game theory based secure model against Black hole attacks in Opportunistic Networks. *IEEE*. DOI: 978-1-5090-4780-2/17
- Conan V. (2013). Mobile opportunistic traffic offloading. *Project Presentation MOTO Consortium* 2013. Grant Agreement No: 317959.
- Conti M., De Salve A., Guidi B., Pitto F. and Ricci L. (2014). Trusted dynamic storage for Dunbar-based P2P online social networks. In: *Meersman R. et al., (Eds) On the Move to Meaningful Internet Systems: OTM 2014 Conferences*. Lecture Notes in Computer Science, vol(8841). Springer, Berlin, Heidelberg. doi.org/10.1007/978-3-662-45563-0
- Costantino G., Martinelli F., Santi P. and Amoruso D. (2013). An implementation of secure two-party computation for smartphones with application to privacy-preserving interest-cast. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, MobiCom '12*, 447–450, New York, USA, 2012. ACM.
- Costantino G., Martinelli F. and Santi P. (2014). Investigating the privacy vs forwarding accuracy trade-off in opportunistic interest-casting, *IEEE Transactions on Mobile Computing*, 13(4): 824–837.
- Daly E. M. and Haahr M. (2009). Social network analysis for information flow in disconnected delay-tolerant MANETs. *IEEE Transactions on Mobile Computing*. 8(5): 606–621.

- De Cristofaro E., Manulis M. and Poettering B. (2011). Private discovery of common social contacts. *In Proceedings of the 9th International Conference on Applied Cryptography and Network Security, ACNS'11*, 147–165, Berlin, Heidelberg, 2011. Springer-Verlag.
- Dhurandher S. K., Kumar A. and Obaidat M. S. (2017). Cryptography-based misbehavior detection and trust control mechanism for opportunistic network systems. *IEEE Systems Journal* 2017. DOI: 10.1109/JSYST.2017.2720757.
- Distl B. and Hossmann T. (2014). Privacy in opportunistic network contact graphs. *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. DOI: 10.1109/wowmom.2014.6919020
- D'ora L. and Holczer T. (2010). Hide-and-lie: Enhancing application-level privacy in opportunistic networks. *In Proceedings of the Second International Workshop on Mobile Opportunistic Networking, MobiOpp '10*, 135–142, New York, NY, USA, ACM.
- Eagle N. and Pentland A. (2006). Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4): 255–268.
- Fan J., Chen J., Du Y., Gao W., Wu J. and Sun Y. (2013). Geo-community-based broad casting for data dissemination in mobile social networks. *IEEE Transactions on Parallel and Distributed Systems*. 24(4): 734–743.
- Guo M-H., Liaw H-T., Chiu M-Y. and Tsai L-P. (2015). Authenticating with privacy protection in opportunistic networks. *11th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QSHINE 2015* 97S-1-63190-063-1. DOI: 10.410S/eai.19-S-2015.2260S62

- Goyal M. and Chaudhary M. (2013). Ensuring Privacy in Opportunistic Network. *International Journal of Computer Applications* 0975 – 8887, 76(4), August 2013. DOI: 10.1.1.403.59 retrieved from www.ijcaonline.org
- Henderson T., Kotz D. and Abyzov I. (2004). The changing usage of amateur campus wide wireless network. In *MobiCom2004: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. 187–201.
- Huang C-M., Lan K-C. and Tsai C-Z. (2008). A Survey of opportunistic networks. *22nd International Conference on Advanced Information Networking and Applications Workshops*. 978-0-7695-3096-3/08. DOI 10.1109/WAINA.2008.292.1672
- Huang Y., Evans D., Katz J. and Malka L. (2011). Faster secure two-party computation using garbled circuits. In *Proceedings of the 20th USENIX Conference on Security, SEC'11*, Berkeley, CA, USA, 2011. USENIX Association.
- Hui P., Crowcroft J. and Yoneki E. (2011). BUBBLE rap: Social-based forwarding in delay-tolerant networks, *IEEE Transactions on Mobile Computing*, 10(11): 1576–1589.
- Jia Y-B. (2018). Roots of polynomials (Com S 477/577 Notes). Retrieved: September 25 2018, from <http://web.cs.iastate.edu/cs577/handouts/polyroots>
- Karamshuk D., Boldrini C., Conti M. and Passarella A. (2011). Human mobility models for opportunistic networks. *Communications Magazine, IEEE*. 49(12):157–165.
- Keranen A. (2008). Opportunistic network environment simulator. *Special Assignment report, Helsinki University of Technology, Department of Communications and Networking*. May 2008.

- Khan A. S. and Chatzigeorgiou I. (2018), Opportunistic relaying and random linear network coding for secure and reliable communication. *IEEE Transactions on Wireless Communications*, vol(17), no(1), January 2018. DOI: 10.1109/TWC.2017.2764891.
- Khudaverdian H. M. (2006). Galois theory a draft of Lecture Notes. Manchester, Autumn 2006 (version 16 XII 2006) Retrieved from: <http://www.maths.manchester.ac.uk/~khudian/Teaching/Galois/gallctrs64>
- Kumar P., Chauhan N. and Chand N. (2018). Security framework for opportunistic networks. Sa P. K. *et al.*, (Eds.), *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, Advances in Intelligent Systems and Computing* 519. DOI: 10.1007/978-981-10-3376-6.
- Kurose J. F. and Ross W. (2017). *Computer networking: top-down approach*. Seventh Edition. Pearson. ISBN: 13-978-0-13-285620-1
- Lakshmi J. V. and Neelima G. (2017). Network security with cryptography. *10th international Conference on Recent Trends in Engineering Science and Management (ICRTESM-17)*. Newton's Institute of Science & Technology, Guntur Dist, Andhra Pradesh, India. Retrieved: 21/02/2018 from www.conferenceworld.com
- Liang X., Li X., Zhang K., Lu R., Lin X. and Shen X. S. (2013). Fully anonymous profile matching in mobile social networks. *IEEE Journal on Selected Areas in Communications*. 31(9): 641–655, 2013.

- Lin G., Noubir G. and Rajaraman R. (2004). Mobility models for ad hoc network simulation. *INFOCOM Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol(1). IEEE.
- Lindgren, A., Doria, A. and Schelen. O. (2003). Probabilistic routing in intermittently connected networks. *SIGMOBILE Mobile Computing and Communication Review*, 7(3), 2003.
- Liu T., Christopher M., Zhang P. and Martunosi M. (2004). Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet. *Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys '04)*, Boston, MA, USA. Pp 256-269. DOI: 10.1145/990064.990095
- Liu X., Xiong N., Zhang N., Liu A., Shen H. and Huang C. (2018). A Trust with abstract information verified routing scheme for cyber-physical network. DOI: 15787-15808
- Lu R., Lin X., Liang X. and Shen X. (2010). Sacrificing the plum tree for the peach tree: A social spot tactic for protecting receiver-location privacy in VANET. *Proceeding of IEEE GLOBECOM*, 1–5.
- Luo P., Huang H., Shu W., Li M. and Wu M. (2008). Performance evaluation of vehicular DTN routing under realistic mobility models. *In Proceeding of the IEEE Wireless Communications and Networking Conference (WCNC)* 2206–2211.
- Malladi R. and Agrawal D. P. (2002). Current and future applications of mobile and wireless networks. *Communications ACM*, 45(10): 144–146, October 2002.
- Matematisk Institut (2009). Rings and polynomials. Retrieved: September 25 2018, from <http://web.math.ku.dk/~olsson/manus/alg3-2009/ek2-2009>

- McNett M. and Voelker G. (2005). Access and mobility of wireless PDA users, *ACM SIGMOBILE CCR*, 9(2):40–55, 2005.
- Misra S., Saha B. K., Pal S. (2016). Opportunistic network advances and application. *Computer Communications and Networks*. DOI 10.1007/978-3-319-29031-7
- Mtibaa A., Chaintreau A., LeBrun J., Oliver E., Pietilainen A-K. and Diot C. (2008). Are you moved by your social network application? *Proceedings of the First Workshop on Online Social Networks (WOSN)*, 67–72, ACM. NewYork, NY, USA.
- Musolesi M. and Mascolo C. (2008). A framework for multi-region delay tolerant networking. *In Proc. of the ACM Workshop on Wireless Networks and Systems for Developing Regions (WiNS-DR)* 37–42
- NASA (2012). Disruption tolerant networking for space operations (DTN). Retrieved from http://www.nasa.gov/mission_pages/station/research/experiments/DTN
- Noorin F. (2009). Opportunistic Networking [Web log post]. Retrieved September 27, 2018, from <https://www.slideshare.net/noorin/opportunisticnetworking>
- Pan D., Ruan Z., Zhou N., Liu X. and Song Z. (2013). A comprehensive Integrated Buffer Management Strategy for Opportunistic Networks. *Journal on Wireless Communication Network 2013*. DOI: 10.1186/1687-1499-2013-103. ISSN: 1687-1449
- Pelusi, L., Passarella, A. and Conti M. (2006). Beyond MANETs: dissertation on opportunistic networking, research report IIT-CNR, August 2006. Retrieved from <http://bruno1.iit.cnr.it/bruno/techreport>.

- Piorkowski M., Sarafijanovoc-Djukic N., and Grossglauser M. (2009). A parsimonious model of mobile partitioned networks with clustering. *In The First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, January 2009.
- Rajappan G., Acharya J., Liu H., Mandayam N. and Seshkar Y. R. (2004). Mobile Infostation Network Technology. *Technical Report* 2004.
- Scott J., Gass R., Crowcroft J., Hui P., Diot C., Chaintreau A. (2006). *CRAWDAD dataset Cambridge/haggle* (v.2006-09-15). Traceset: imote. Downloaded from <https://doi.org/10.15783/C73S3N>.
- Small, T. and Haas Z. J. (2003). The Shared wireless Infostation model: A new ad-hoc networking paradigm (or where there is a whale, there is a way). *Proceedings of the 4th ACM International Symposium on Mobile ad-hoc Networking and Computing*, Annapolis, Maryland, USA. Jun 1-3, ACM New York, pp: 233-244.
- Spyropoulos T., Psounis K. and Raghavendra C. S. (2005). Spray and Wait: Efficient routing in intermittently connected mobile networks. *In Proceedings of ACMSIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.
- Stallings W. (2011). Network security and cryptography principles and practices. *Pearson Education, Inc., publishing as Prentice Hall*. ISBN 13: 978-0-13-609704-4.
- Usman A. B. and Gutierrez J. (2018). DATM: a dynamic attribute trust model for efficient collaborative routing. *S. I.: Queuing Theory and Network Applications*. Retrieved from <https://doi.org/10.1007/s10479-018-2864-5>

- Vahdat A. and Becker D. (2000). Epidemic routing for partially connected ad hoc networks. *Technical Report CS-200006*, Duke University, 2000.
- Vennela G. S., Varun N. V., Neelima N., Priya L. S. and Yeswanth J. (2018). Performance analysis of cryptographic algorithms for cloud security. *Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018)*. ISBN: 978-1-5386-1974.
- Viscal J., Richart M., Saavedra J., Baliosian J. and Grampin E. (2014). Buffer management in opportunistic network. *In Proceedings of the Latin America Networking Conference (LANCI4)*. 14(2). ISBN: 978-1-4503-3280-4/14/09. DOI:10.1145/2684083.2684085
- Wahid A., Kumar G. and Ahmad K. (2014). Opportunistic networks: Opportunity versus challenges survey. *National Conference on Information Security Challenges (NCIC-ISBN: 978-81-7678-220-3*
- Wei K., Liang X. and Xu K. (2014). A survey of social-aware routing protocols in delay tolerant networks: Applications, taxonomy and design-related issues. *IEEE Communications Surveys & Tutorials*, vol(16), No(1), first quarter, 2014.
- Wang Y. and Wu J. (2015). Multicast in opportunistic network. *11th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness QSHINE 2015*.
- Xi C., Liang S., Feng M. A. J. and Zhuo M. A. (2015). A trust management scheme based on behavior feedback for opportunistic networks. *Network Technology and Application*, April 2015

- Yao L., Man Y., Huang Z., Deng J. and Wang X. (2016). Secure routing based on social similarity in opportunistic networks, *IEEE* 1536-1276. DOI: 10.1109/TWC.2015.2476466.
- Zenjireh M. M. and Larijani H. (2015). A survey on centralized and distributed clustering routing algorithms for WSN. *IEEE 81st Vehicular Technology Conference*, Glasgow, Scotland. DOI: 10.1109/VTCSpring.2015.7145650.
- Zhang K., Liang X., Lu R. and Shen X. (2014). Exploiting private profile matching for efficient packet forwarding in mobile social opportunistic network. *Opportunistic Mobile Social Networks*. CRC Press.
- Zhou H., Chen J., Fan J. and Du Y. (2013) ConSub: Incentive-based content subscribing in selfish opportunistic mobile networks. *IEEE Journal on Selected Areas in Communications/Supplement*. Vol(31), NO(9), 0733-8716/13, DOI: 10.1109/JSAC.2013.SUP.051305

Appendix A1

Downloading and Installing Eclipse and ONE Simulator

Eclipse is an Integrated Development Environment (IDE) that is developed and optimized for writing, editing, and developing codes in java language. Eclipse is downloaded from <https://www.eclipse.org/download>. The downloaded eclipse must correspond to the bit (32 or 64 bits) of the installed java which must also in turn correspond to the operating system of the computer system used. If this is not the case, the system continuously malfunctions and indicates error codes.

The Opportunistic Environment (ONE) simulator is downloaded and stored in a directory a computer system. ONE is downloaded from <https://github.com/akeranen/the-one>. The latest version of ONE simulator which is ONEv1.6.0. at the time of this research is used. Although the previous version (ONEv1.5.1-RC2) will equally do for this work. This is because the added features in the latest version is not related to this research work.

To configure the ONE simulator with the eclipse, the following steps are followed after opening the eclipse in the window:

1. Go to file in the eclipse,
2. Click “new” in drop down box
3. Then click java project, a new project box appears titled “new java project”
4. In it, deactivate the “use default location”
5. Click the radio button “browse” and go to the directory where the ONE simulator is stored
6. The absolute address of the directory shows up in “location”
7. On the “project name” type ONE_test

8. Click next on the radio button below
9. Then click order and export
10. Then click finish

With these steps the ONE simulator is configured. If the ONE showed error, it means some of the libraries are missing which can be included manually by the following steps:

1. Right click on ONE in the eclipse environment
2. Click on build path in dropped down menu
3. Click on add external archives
4. Browse to the library folder of the downloaded ONE simulate
5. Highlight DTNConsoleConnection and ECLA
6. Click open
7. In addition, the Junit is located following the above steps and added
8. The DTNsim can now be run in the core package.

Appendix A2

Running the ONE Simulator

In order to run ONE simulator in the Eclipse, the following order is followed: Package explorer → ONE_test → src → core → "right click" DTNsim.java → runas → java application. The ONE simulator GUI is displayed and the simulation processes is started by clicking the play button. If the ONE simulator is to be run from its directory, this can be done as follows. Go to the directory of the ONE_1.6.0, click on compile. The .java source code store on default setting is compiled to generate its .class file (that the JVM can run). The command prompt displays the compiling process. When the source code is successfully compiled, click one.bat. The GUI comes up and simulation process can be started as before.

On the top left corner of the GUI is the name of the scenario that is been run. Below this, is the playfield options to control the display of: underlay image, node name strings, radio coverages, node connections, message buffer, map graphics. Next are the tools to control overlays of the message filter and node filters. Below these, the GUI displays simulation time which can be clicked on to force an update or right clicked to change the time format. This is followed by the simulation second per second. The radio topside radio buttons include: play simulation, step forward (by one interval), the button to enable or disable fast-forward, the button for play until sim_time (which is typed on pop up box). The GUI update can be used to speed up or down the simulation process. Also the simulation environment can be zoomed in out at will and the screenshot of the GUI can be taken. In addition, the GUI displays the number of nodes each on a radio button besides the simulation area and information about a node can be displayed by clicking on it. The event log is shown below the simulation area as the simulation progresses and can be controlled in the event log control.

Appendix B1

Source Code of TRSS Class

```
package routing;

import java.util.ArrayList;
import java.util.List;
import core.Connection;
import core.DTNHost;
import core.Message;
import core.Settings;

public class TRSS extends ActiveRouter {

    public static final String NROF_COPIES = "nrofCopies";
    public static final String BINARY_MODE = "binaryMode";
    public static final String TRSS = "TRSS";
    public static final String MSG_COUNT_PROPERTY = TRSS_NS + "." +
        "copies";

    protected int initialNrofCopies;
    protected boolean isBinary;

    protected List<Message> getMessagesWithCopiesLeft() {

        List<Message> list = new ArrayList<Message>();

        for (Message m : getMessageCollection()) {

            Integer nrofCopies = (Integer)m.getProperty(Similarity);

            assert nrofCopies != null : "SnW message " + m + " didn't have " + "nrof
copies property!";

            if (nrofCopies > 1) {

                list.add(m); }

        }
```

```

        return list;
    }

    public SprayAndWaitRouter(Settings s) {
        super(s);

        Settings snwSettings = new Settings(PATRSS_NS);
        initialNrofCopies = snwSettings.getInt(NROF_COPIES);
        isBinary = snwSettings.getBoolean( BINARY_MODE); }

    protected TRSS(SprayAndWaitRouter r) {
        super(r);

        this.initialNrofCopies = r.initialNrofCopies;
        this.isBinary = r.isBinary; }

    @Override
    public int receiveMessage(Message m, DTNHost from) {
        return super.receiveMessage(m, from); }

    @Override
    public Message messageTransferred(String id, DTNHost from) {
        Message msg = super.messageTransferred(id, from);

        Integer nrofCopies = (Integer)msg.getProperty(MSG_COUNT_PROPERTY);
        assert nrofCopies != null : "Not a SnW message: " + msg;

        if (isBinary)    {
            nrofCopies = (int)Math.ceil(nrofCopies/2.0); }
            else {nrofCopies = 1;}

        msg.updateProperty(MSG_COUNT_PROPERTY, nrofCopies);

        return msg;    }

    @Override
    public boolean createNewMessage(Message msg) {

```

```

        makeRoomForNewMessage(msg.getSize());

        msg.setTtl(this.msgTtl);

        msg.addProperty(MSG_COUNT_PROPERTY,
new Integer(initialNrofCopies));

        addToMessages(msg, true);

        return true;    }

@Override

public void update() {
    super.update();

    if (!canStartTransfer() || isTransferring()) {
        return; }

    if (exchangeDeliverableMessages() != null) {
        return; }

    @SuppressWarnings(value = "unchecked")
    List<Message>
copiesLeft = sortByQueueMode(getMessagesWithCopiesLeft());

    if (copiesLeft.size() > 0) {
        this.tryMessagesToConnections(copiesLeft, getConnections()); } }

protected List<Message>getMessagesWithCopiesLeft() {
    List<Message> list = new ArrayList<Message>();

    for (Message m :getMessageCollection()) {

        Integer nrofCopies = (Integer)m.getProperty(MSG_COUNT_PROPERTY);

        assert nrofCopies != null : "TRSS message " + m + " didn't have " +
            "nrof copies property!";

        if (nrofCopies > 1) {

```

```

        list.add(m);    }    }

    return list;    }

@Override

protected void transferDone(Connection con) {

    Integer nrofCopies;

    String msgId = con.getMessage().getId();

    Message msg = getMessage(msgId);

    if (msg == null) {

        return; }

    nrofCopies = (Integer)msg.getProperty(MSG_COUNT_PROPERTY);

    if (isBinary) {

        nrofCopies /= 2; }

    else {

        nrofCopies--; }

    msg.updateProperty(MSG_COUNT_PROPERTY, nrofCopies);    }

@Override

public TRSS replicate() {

    return new TRSS(this);    }    }

```

Appendix B2

Modelling Datasets for Message Routing

```
package routing.util;

import java.util.ArrayList;

import util.Range;

import util.Tuple;

import core.ArithmeticCondition;

import core.Connection;

import core.DTNHost;

import core.Message;

import core.ModuleCommunicationBus;

import core.Settings;

Scenario.name = infocom06_router-%%Group.router%%
Scenario.simulateConnections = false
Group.movementModel = StationaryMovement
Events.nrof = 2
Events1.class = ExternalEventsQueue
Events1.filePath = suleiman_scenarios/infocom-24hr.txt
public class MessageTransferAcceptPolicy {

    public static final String MTA_POLICY_NS = "mtaPolicy";

    public static final String NROF_MCBACS_S = "nrofMCBACS";

    public static final String MCBACR_S = "MCBRcondition";

    public static final String MCBACS_S = "MCBScondition";

    public static final String MCBCVR_S = "MCBRvalue";

    public static final String MCBCVS_S = "MCBSvalue";

    public static final int TO_ME_VALUE = -1;

    public static final String TO_RPOLICY_S = "toReceivePolicy";
```

```

public static final String FROM_RPOLICY_S = "fromReceivePolicy";

public static final String TO_SPOLICY_S = "toSendPolicy";

public static final String FROM_SPOLICY_S = "fromSendPolicy";

public static final String HOPCOUNT_SPOLICY_S = "hopCountSendPolicy";

private ArrayList<Tuple<String,ArithmeticCondition>>recvConditions = null;

private ArrayList<Tuple<String,ArithmeticCondition>>sendConditions = null;

private Range[] toSendPolicy = null;

private Range[] fromSendPolicy = null;

private Range[] toReceivePolicy = null;

private Range[] fromReceivePolicy = null;

private ArithmeticConditionhopCountSendPolicy = null;

private ArithmeticConditionhopCountReceivePolicy = null;

public MessageTransferAcceptPolicy(Settings nsSettings) {

    Settings s;

    if (! nsSettings.contains(MTA_POLICY_NS)) {

        return; }

    s = new Settings(nsSettings.getSetting(MTA_POLICY_NS));

    addMCBCs(s);

    if (s.contains(TO_SPOLICY_S)) {

        this.toSendPolicy = s.getCsvRanges(TO_SPOLICY_S); }

    if (s.contains(FROM_SPOLICY_S)) {

        this.fromSendPolicy = s.getCsvRanges(FROM_SPOLICY_S); }

    if (s.contains(TO_RPOLICY_S)) {

        this.toReceivePolicy = s.getCsvRanges(TO_RPOLICY_S); }

    if (s.contains(FROM_RPOLICY_S)) {

```

```

        this.fromReceivePolicy = s.getCsvRanges(FROM_RPOLICY_S); }
    if (s.contains(HOPCOUNT_SPOLICY_S)) {
        hopCountSendPolicy = s.getCondition(HOPCOUNT_SPOLICY_S); }
    if (s.contains(HOPCOUNT_RPOLICY_S)) {
        hopCountReceivePolicy = s.getCondition(HOPCOUNT_RPOLICY_S); } }

private void addMCBCs(Settings s) {
    if (!s.contains(NROF_MCBCS_S)) {
        return;    }

    int[] nrof = s.getCsvInts(NROF_MCBCS_S);
    if (nrof[0] > 0) {
        this.recvConditions =
            new ArrayList<Tuple<String,ArithmeticCondition>>(); }
    if (nrof[1] > 0) {
        this.sendConditions =
            new ArrayList<Tuple<String,ArithmeticCondition>>(); }

    addConditions(s, MCBACR_S, MCBCVR_S, this.recvConditions, nrof[0]);
    addConditions(s, MCBACS_S, MCBCVS_S, this.sendConditions, nrof[1]); }

private void addConditions(Settings s, String cPrefix, String vPrefix,
    ArrayList<Tuple<String,ArithmeticCondition>> list,
    int nrof) {
    for (int i=1; i<=nrof; i++) {
        ArithmeticCondition ac = s.getCondition(cPrefix + i);
        String mcbValue = s.getSetting(vPrefix + i);
        list.add(new Tuple<String, ArithmeticCondition>(mcbValue, ac)); } }

```

```

private boolean checkMcbConditions(ModuleCommunicationBus mcb,
    boolean receiving) {
    ArrayList<Tuple<String, ArithmeticCondition>> list =
        (receiving ? this.recvConditions : this.sendConditions);
    if (list == null) {
        return true;    }
    for (Tuple<String, ArithmeticCondition> t : list) {
        if (!mcb.containsProperty(t.getKey())) {
            continue;    }
        if (t.getValue().isTrueFor(mcb.getDouble(t.getKey(), 0))) {
            return false;    }    }
    return true;    }

private boolean checkSimplePolicy(DTNHost host, Range [ ] policy,
    int thisHost) {
    int address;
    if (policy == null) {
        return true;    }
    address = host.getAddress();
    for (Range r : policy) {
        if (r.isInRange(TO_ME_VALUE) && address == thisHost) {
            return true;    }
        else if (r.isInRange(address)) {
            return true;    }    }
    return false;    }

private boolean checkHopCountPolicy(Message m, ArithmeticCondition ac) {

```

```

        if (ac == null) {
            return true;    }

        else    { return ac.isTrueFor(m.getHopCount());    }    }

public boolean acceptSending(DTNHost from, DTNHost to, Connection con,
    Message m) {

    if (!checkMcbConditions(from.getComBus(), false)) {

        return false;    }

    int myAddr = from.getAddress();

    if (! (checkSimplePolicy(m.getTo(), this.toSendPolicy, myAddr) &&
        checkSimplePolicy(m.getFrom(), this.fromSendPolicy,    myAddr)) ) {

        return false;    }

    if (m.getTo() != to && !checkHopCountPolicy(m, this.hopCountSendPolicy)){

        return false;    }

    return true;    }

public boolean acceptReceiving(DTNHost from, DTNHost to, Message m) {

    if (! checkMcbConditions(to.getComBus(), true)) {

        return false;    }

    int myAddr = to.getAddress();

    if (! (checkSimplePolicy(m.getTo(), this.toReceivePolicy, myAddr) &&
        checkSimplePolicy(m.getFrom(), this.fromReceivePolicy, myAddr)) ) {

        return false;    }

    if (m.getTo() != to && ! checkHopCountPolicy(m, this.hopCountReceivePolicy)) {

        return false;    }

    return true;    }    }

```

Appendix B3

Source Code of PATRSS Class

```
package communityRouting;

import core.Settings;

import java.crypto.enc

import routing.DecisionEngineRouter;

import routing.MessageRouter;

import routing.RoutingDecisionEngine;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Map;

import java.util.Set;

public class PATRSS extends TRSS {

String message1=new string(message);

String[] message1= List.toArray(new string message1)

for (int index=0; index<message1.length; index++) {

    if(index=3n+1) M1=arraylist.add(message1);

    elseif(index=3n+2) M2=arraylist.add(message1);

    else M3=arraylist.add(message1);

public byte[] encrypt(obj1 obj2)    {

SecureRandomrandSecRand = SecureRandom.getInstance(M k);

Static Galois.encrypt enc = new M();

Return new(enc.enc1(M K));    }
```

```

public class Comm
implements RoutingDecisionEngine,
CommunityDetectionEngine{

public static final String COMMUNITY_ALG_SETTING = "communityDetectAlg";

public static final String CENTRALITY_ALG_SETTING = "centralityAlg";

protected Map<DTNHost, Double>startTimestamps;

protected Map<DTNHost, List<Duration>>connHistory;

protected CommunityDetection community;

protected Similarity similarity;

private Map<DTNHost, Map<DTNHost, Double>> _weights;

private Map<DTNHost, Double> _importances;

public Comm(Settings s){

this._weights = new HashMap<DTNHost, Map<DTNHost, Double>>();

this._importances = new HashMap<DTNHost, Double>();

if(s.contains(COMMUNITY_ALG_SETTING))

    this.community = (CommunityDetection)

        s.createIntializedObject(s.getSetting(COMMUNITY_ALG_SETTING));

else    this.community = new SimpleCommunityDetection(s);

if(s.contains(SIMILARITY_ALG_SETTING))

    this.similarity = (Similarity)

        s.createIntializedObject(s.getSetting(SIMILARITY _ALG_SETTING));

else    this.centrality = new SWindowCentrality(s);    }

public Comm(DlifeComm proto) {

this._weights = new HashMap<DTNHost, Map<DTNHost, Double>>();

this._importances = new HashMap<DTNHost, Double>();

```

```

this.community = proto.community.replicate();
this.centralitv= proto.centralitv.replicate();
startTimestamps = new HashMap<DTNHost, Double>();
connHistory = new HashMap<DTNHost, List<Duration>>();    }
public void connectionUp(DTNHostthisHost, DTNHost peer){ }
private Map<String, Object> properties;

    private String  appID;

    static          {reset();

    DTNSim.registerForReset(Message.class.getCanonicalName());    }

public Message(DTNHost from, DTNHost to, String id, int size) {

    this.from = from;          this.to = to;

    this.id = id;              this.size = size;

    this.path = new ArrayList<DTNHost>();

    this.uniqueId = nextUniqueId;

    this.timeCreated = SimClock.getTime();

    this.timeReceived = this.timeCreated;

        this.responseSize = 0;

    this.requestMsg = null;      this.properties = null; this.appID = null;

    if(s != 0 || T < 0.5)

        messageTransferred(route.PATRSS)

    else messageTransferred(route.trust)

    Message.nextUniqueId++;

    addNodeOnPath(from);      }

public DTNHostgetFrom() {

```

```

        return this.from;    }

    public void doExchangeForNewConnection(Connection con, DTNHost peer) {
        DTNHost myHost = con.getOtherNode(peer);
        DlifeComm de = this.getOtherDecisionEngine(peer);
        this.startTimestamps.put(peer, SimClock.getTime());
        de.startTimestamps.put(myHost, SimClock.getTime());
        this.community.newConnection(myHost, peer, de.community);    }

    public void connectionDown(DTNHost thisHost, DTNHost peer){
        double time = startTimestamps.get(peer);
        double etime = SimClock.getTime();
        List<Duration> history;
        if(!connHistory.containsKey(peer)){
            history = new LinkedList<Duration>();
            connHistory.put(peer, history);    }
        else    history = connHistory.get(peer);
            if(etime - time > 0)
                history.add(new Duration(time, etime));
        CommunityDetection peerCD = this.getOtherDecisionEngine(peer).community;
        community.connectionLost(thisHost, peer, peerCD, history);
        startTimestamps.remove(peer);    }

    public boolean newMessage(Message m) {
        return true;    }

    public boolean isFinalDest(Message m, DTNHost aHost) {
        return m.getTo() == aHost; }

    public boolean shouldSaveReceivedMessage(Message m, DTNHost thisHost){

```

```

return m.getTo() != thisHost;      }

public boolean shouldSendMessageToHost(Message m, DTNHost otherHost, DTNHost thisHost){
    if(m.getTo() == otherHost) return true;

    DTNHost dest = m.getTo();

    DlifeComm de = getOtherDecisionEngine(otherHost);

    weights = DecisionEngineRouter.weightsCopy;
    importances = DecisionEngineRouter.importCopy;

    boolean peerInCommunity = de.communesWithHost(dest);
    boolean meInCommunity = this.communesWithHost(dest);

    if(checkMessage(m, otherHost)) return false;

    if(peerInCommunity && !meInCommunity)
        return true;

    else if(!peerInCommunity && meInCommunity)
        return false;

    else if(peerInCommunity){
        Map<DTNHost, Double> tempListThis = new HashMap<DTNHost, Double>();
        Map<DTNHost, Double> tempListOther = new HashMap<DTNHost, Double>()

        double ThisWeightToDest = 0.0;
        double OtherWeightToDest = 0.0;

        tempListThis = _weights.get(thisHost);
        tempListOther = _weights.get(otherHost);

        if(_weights.containsKey(thisHost) || _weights.containsKey(otherHost)){
            Set<DTNHost> hostset = tempListThis.keySet();

            Iterator<DTNHost> hostIterator = hostset.iterator();

```

```

        if(tempListThis.size()!=0){
            while(hostIterator.hasNext()){
                DTNHostcurrenthost = hostIterator.next();
                if(currenthost==dest){
                    ThisWeightToDest=tempListThis.get(currenthost); }      }      }

                Set<DTNHost> hostset1= tempListOther.keySet();
                Iterator<DTNHost> hostIterator1=hostset1.iterator();
                if(tempListOther.size()!=0){
                    while(hostIterator1.hasNext()){
                        DTNHostcurrenthost = hostIterator1.next();
                        if(currenthost==dest){
                            OtherWeightToDest=tempListOther.get(currenthost);      }      }      }
                            if(OtherWeightToDest>ThisWeightToDest)
                                return true; }  }

else if(_importances.get(thisHost) != null && _importances.get(otherHost) != null)
    if(_importances.get(otherHost) > _importances.get(thisHost))

return true;

return false; }

public booleancheckWeightToDest(Map<DTNHost,Double>weightList, DTNHostdest){
    if(weightList.get(dest)!=null) {
        Set<DTNHost>hostset= weightList.keySet();
        Iterator<DTNHost>hostIterator=hostset.iterator();
        if(weightList.size()!=0) {
            while(hostIterator.hasNext()) {

```

```

DTNHostcurrenthost = hostIterator.next();

if(currenthost==dest)

return true;    }    }    }

return false;    }

public booleancheckMessage(Message m, DTNHostotherHost){

List<Message> teste = new ArrayList<Message>();

teste.addAll(otherHost.getMessageCollection());

Object [] array = teste.toArray();

String message1 = message1toString();

for (int i = 0; i <array.length; i++) {

if(array[i].toString()==message1) {

return true;}    }

return false;    }

public booleanshouldDeleteSentMessage(Message m, DTNHostotherHost)    {

DlifeComm de = this.getOtherDecisionEngine(otherHost);

return de.communesWithHost(m.getTo()) && !this.communesWithHost(m.getTo());    }

public booleanshouldDeleteOldMessage(Message m, DTNHosthostReportingOld){

DlifeComm de = this.getOtherDecisionEngine(hostReportingOld);

return de.communesWithHost(m.getTo()) &&!this.communesWithHost(m.getTo());    }

public RoutingDecisionEnginereplicate()    {

return new DlifeComm(this);    }

protected booleancommunesWithHost(DTNHost h)    {

return community.isHostInCommunity(h);    }

protected double getLocalSimilarity()    {

return this.centrality.getLocalSimilarity(connHistory, community); }

```

```

protected double getGlobalCentrality()          {
return this.centrality.getGlobalCentrality(connHistory);    }

private DlifeCommgetOtherDecisionEngine(DTNHost h)          {
MessageRouterotherRouter = h.getRouter();
assert otherRouter
instanceofDecisionEngineRouter

        return (Comm) ((DecisionEngineRouter)otherRouter).getDecisionEngine(); }

    public Set<DTNHost>getLocalCommunity()          {
return this.community.getLocalCommunity();                }
}

```