School of Computer Science and Engineering

PhD Thesis

A DYNAMIC TRUST AND MUTUAL AUTHENTICATION SCHEME FOR MANET SECURITY

Mansoor Ihsan

-Supervised by-

Dr MARTIN HOPE

September 2018

CONTENTS

LIST OF ABBREVATIONS **7**

LIST OF FIGURES **9**

LIST OF TABLES **11**

LIST OF EQUATIONS **12**

LIST OF ALGORITHMS **13**

ACKNOWLEDGEMENTS **14**

ABSTRACT **15**

CHAPTER 1 **16**

INTRODUCTION **16**

1.1 STATEMENT OF THE PROBLEM 20
1.2 RESEARCH QUESTIONS 22
1.3 RESEARCH AIMS 23
1.4 RESEARCH OBJECTIVES 23
1.5 RESEARCH PROCESS 24
1.6 THESIS ORGANISATION 30

CHAPTER 2 **32**

BACKGROUND AND LITERATURE REVIEW **32**

2.1 WIRELESS COMMUNICATION IN MANET 33
2.2 MANET INFRASTRUCTURE 35
2.2.1 INFRASTRUCTURE MODE 35
2.2.2 ADHOC MODE.. 35
2.3 MANET ROUTING 36
2.3.1 PROACTIVE ROUTING PROTOCOLS 37
2.3.2 REACTIVE ROUTING PROTOCOLS 39

2.3.3 HYBRID ROUTING PROTOCOLS 40
2.4 MANET SECURITY 41
2.4.1 FORMS OF ATTACKS IN MANET 42
2.4.2 SECURITY THREATS IN MANET 43
2.5 NETWORK SECURITY AND CRYPTOGRAPHY 46
2.5.1 SECRET KEY CRYPTOGRAPHY 48
2.5.2 PUBLIC KEY CRYPTOGRAPHY 50
2.5.3 HASH FUNCTIONS 52
2.5.4 HASHED MESSAGE AUTHENTICATION CODE 54
2.5.5 DIGITAL SIGNATURE 54
2.5.6 PAIRWISE KEYS DISTRIBUTION 56
2.6 MANET SECURITY RELATED WORK 57
2.6.1 SECURE ROUTING PROTOCOL 57
2.6.2 TRUST BASED SCHEMES 59
2.7 SIMULATION AND SOURCE CODE 62
2.8 SUMMARY 63

CHAPTER 3 **64**

AUTHENTICATION, TRUST AND KEY MANAGEMENT **64**

3.1 AUTHENTICATION 65
3.2 AUTHENTICATION MODELS 66
3.2.1 PRE-NETWORK INITIALISATION AUTHENTICATION 67
3.2.2 POST-NETWORK INITIALISATION AUTHENTICATION 68
3.2.3 HYBRID AUTHENTICATION 70
3.3 TRUST AND KEY MANAGEMENT 70
3.4 NODE DEGRESS AND 2-HOP NEIGHBOURS 72
3.5 FALSE POSITIVES AND FALSE NEGATIVES 73
3.5.1 FALSE POSITIVES 73
3.5.2 FALSE NEGATIVES 73

3.5.3 TRUE POSITIVES 73
3.5.4 TRUE NEGATIVES 74
3.6 KEY MANAGEMENT 74
3.7 AVAILABILITY…….. 75
3.8 ENERGY……..... 76
3.9 MOBILITY……. 76
3.10 CONFIDENTIALITY 77
3.11 CONCLUSION…….. 78

CHAPTER 4 **79**

ANALYTICAL MODEL **79**

4.3 PHASE-ONE……..... 80
4.3.1 THRESHOLD CALCULATION 84
4.3.2 ONE-HOP NEIGHBOUR THRESHOLD 84
4.3.3 TWO-HOP NEIGHBOUR THRESHOLD 87
4.3.4 MOBILITY THRESHOLD……..... 91
4.3.5 TRUST THRESHOLD..... 92
4.3.6 AVERAGE TRUST 93
4.3.7 TRUST THRESHOLD..... 93
4.3.8 ENERGY THRESHOLD 95
4.3.9 DYNAMIC THRESHOLD 96
4.4 NUMERICAL MODEL 99
4.5 PHASE-TWO…….. 105
4.5.1 ELLIPTIC CURVE DIFFIE-HELLMAN KEY EXCHANGE 106
4.6 MUTUAL AUTHENTICATION SCHEME 111
4.6.1 AODV AUTHENTICATION PROCESS AT DEST NODE 111
4.6.2 AODV AUTHENTICATION PROCESS AT SOURCE NODE 114
4.7 CONCLUSION……. 116

CHAPTER 5 **117**

IMPLEMENTATION **117**

5.1 SIMULATION ENVIRONMENT 117
5.1.1 MOBILITY MODEL AND DATA RATES 119
5.1.2 MALICIOUS NODES 119
5.2 SCHEME DESIGN 120
5.3 MOBILITY….. ……..... 124
5.4 ENERGY…….……..... 124
5.5 DYNAMIC THRESHOLD SIMULATION……..……..……. 127
5.6 CONCLUSION ….. 129

CHAPTER 6 **130**

PERFORMANCE EVALUATION **130**

6.1 PERFORMANCE METRICS 130
6.2 SCENARIO-ONE (20 NODES) STANDARD VS TRUSTED AODV…… 136
6.2.1 PERFORMANCE SCENARIO-ONE 134
6.3 SCENARIO-TWO (50 NODES) STANDARD VS TRUSTED AODV……145
6.3.1 PERFORMANCE SCENARIO-TWO 147
6.4 STATIC VS DYNAMIC TRUST 152
6.5 PHASE-TWO PERFORMANCE EVALUATION 155
6.6 CONCLUSION……. 156

CHAPTER 7 **158**

CONCLUSION 158

7.1 THESIS SUMMARY 159
7.2 PROPOSED SCHEME EVALUATION 162
7.3 CONTRIBUTIONS 169

7.4 RESEARCH CHALLANGES..... 169
7.5 FUTURE WORK 170

REFERENCES 173

APPENDIX A: AODV 20 NODES TCL FILE 189

APPENDIX B: AODV 20 NODES CBR FILE 193

APPENDIX C: AODV 50 NODES TCL FILE 196

APPENDIX D: AODV 50 NODES CBR FILE 200

APPENDIX E: SOURCE CODE 204

LIST OF ABBREVIATIONS

AES	Advance Encryption Standard
AODV	Ad hoc On Demand Distance Vector
ARAN	Authenticated Routing for Adhoc Network
CA	Certificate Authority
DTTS	Dynamic Trust Threshold Scheme
ECDH	Elliptic Curve Diffie-Hellman
DoS	Denial of Service
DLP	Discrete Logarithm Problem
DSDV	Destination-Sequenced Distance-Vector
DES	Data Encryption Standard
3DES	Triple-DES
DSR	Dynamic Source Routing
DSA	Digital Signature Algorithm
DVR	Distance Vector Routing
ECC	Elliptic Curve Cryptography
IBC	Identity Based Cryptography
IBE	Identity Based Encryption
IETF	Internet Engineering Task Force
IBS	Identity Based Scheme
KDC	Key Distribution Centre
LAN	Local Area Network

MD	Message Digest
MAC	Message Authentication Code
MPR	Multi Point Relay
MK	Master Key
OSLR	Optimized Link State Routing Protocol
PKI	Public Key Infrastructure
PK	Private Key
RSA	Rivest Shamir Adleman
SAR	Security Aware Routing
SAODV	Secure Ad Hoc Distance Vector
SEAD	Secure Efficient Ad Hoc Distance Vector
SPAAR	Security Aware Aided Adhoc Routing
SGC	Secure Group Communication
SHA	Secure Hash Algorithm
TBS	Trust Based Scheme
TTS	Trust Threshold Scheme
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TORA	Temporary Ordered Routing Algorithm
SUARN	Survivable Adaptive Radio Network
MANET	Mobile Ad Hoc Network
WAN	Wide Area Network
WRP	Wireless Routing Protocol
ZRP	Zone Routing Protocol

LIST OF FIGURES

Figure 1.1	Research Process.....	27
Figure 2.1	Types of wireless networks…….....	34
Figure 2.2	Routing protocols in MANET.....	37
Figure 2.3	Network security model…..….....	47
Figure 2.4	Secret key cryptography.....	49
Figure 2.5	Public key cryptography…..….....	50
Figure 2.6	Hash function……....	54
Figure 2.7	Digital signature Algorithm..….....	56
Figure 4.1	DFD phase-one…..….....	83
Figure 4.2	Wireless network graph model.....	85
Figure 4.3	One hop neighbours……....	86
Figure 4.4	Two hop neighbours……....	89
Figure 4.5	NS2 Simulation of Trust Threshold calculation.....	99
Figure 4.6	Static Trust……..	103
Figure 4.7	Dynamic Trust……	104
Figure 4.8	Elliptic Curve over Integer Modulo *p*……..	108
Figure 4.9	Destination node DFD standard vs trusted AODV.....	113
Figure 4.10	Source node DFD standard vs trusted AODV…….....	115
Figure 5.1	AODV operation of route request.....	121
Figure 5.2	Hello packet format of standard AODV…….	123
Figure 5.3	Hello packet format of Trusted AODV……..	124

Figure 5.4	Dynamic trust threshold DFD.....	128
Figure 6.1	Packet statistics.....	138
Figure 6.2	Network throughput.....	139
Figure 6.3	Routing overhead.....	141
Figure 6.4	Average end-to-end delay.....	142
Figure 6.5	Packet delivery ratio.....	144
Figure 6.6	Packet statistics.....	148
Figure 6.7	Routing overhead.....	149
Figure 6.8	Network throughput.....	150
Figure 6.9	Average end-to-end delay.....	151
Figure 6.10	Packet Delivery Ratio……..	152
Figure 6.11	Static verses Dynamic Trust Model Throughput…….	153
Figure 6.12	Static verses Dynamic Trust Model Packet Delivery Ratio…	154
Figure 7.1	New and Existing Node Joining the Network…….	166
Figure 7.2	MANET trust stages…….....	167
Figure 7.3	Acknowledgement Method……...	171

LIST OF TABLES

Table 4.1	Static Trust Algorithm.....	90
Table 4.2	Dynamic Trust Algorithm…….....	91
Table 4.3	Static Trust Matrix.....	93
Table 4.4	Dynamic Trust Matrix …….....	95
Table 4.5	Comparative Analysis RSA and Diffie-Hellman using ECC	100
Table 5.1	Hello Packet Parameters.....	123
Table 5.2	Energy Model.....	126
Table 6.1	Simulation System Environment…....	133
Table 6.2	Node Movement and Network Size…….	134
Table 6.3	Traffic Pattern Parameters 20 Nodes.....	135
Table 6.4	Data parameters for CBR application.....	121
Table 6.5	Node Movement and Network Size……..	145
Table 6.6	Traffic Pattern Parameters 50 Nodes.....	146

LIST OF EQUATIONS

Equation 4.1	Set of Wireless Links…….....	84
Equation 4.2	Node Degree Threshold.....	87
Equation 4.3	Two-hop Neighbours.....	88
Equation 4.4	Two-hop Neighbours Threshold.....	90
Equation 4.5	Mobility Threshold.....	91
Equation 4.6	Average Trust …….....	93
Equation 4.7	Trust Threshold.....	93
Equation 4.8	Trust Threshold Math Equation.....	94
Equation 4.9	Energy Threshold.....	96
Equation 4.10	Dynamic Trust..…..…….	97
Equation 4.11	Trapdoor Function.....	107
Equation 4.12	Elliptic Curve…….....	107
Equation 4.13	Elliptic Curve Condition.....	108
Equation 4.14	Discrete Logarithm Problem…….	109
Equation 6.1	Throughput……	131
Equation 6.2	Routing Overhead…….	131
Equation 6.3	End-to-end Delay……..	131
Equation 6.4	Packet Delivery Ratio……	132

List of Algorithms

Algorithm 4.1	Static Trust Algorithm…….…….....	100
Algorithm 4.2	Dynamic Trust Algorithm…….....	100
Algorithm 5.1	Malicious nodes TCL Script……..	119

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Allah for his mercy and enabling me to successfully complete my research.

I would like to express my special thanks of gratitude to my supervisor Dr Martin Hope for all his valuable support, motivation and technical expertise he offered throughout the research. Every step of my research, I had his support that enabled me to successfully complete my PhD.

I would also thank my friends who offered their support throughout and were always available for help. I had valuable advice from them that guided me in the most appropriate way. I also take this opportunity to thank all the academic staff from my university, whom I met during research, for all their support.

Most of all, I would thank my wife, my parents and all my family for their love, encouragement and kindness. Lastly, I would like to extend my gratitude to everyone who helped me in different aspects of getting this thesis completed.

Mansoor Ihsan

ABSTRACT

MANETs are attractive technology in providing communication in the absence of a fixed infrastructure for applications such as, first responders at a disaster site or soldiers in a battlefield (Kumar, and Mishra, 2012). The rapid growth MANET has experienced in recent years is due to its Ad Hoc capabilities that have also made it prime target of cybercrimes (Jhaveri, 2012). This has raised the question of how could we embrace the benefits of MANET without the increased security risks. MANETs have several vulnerabilities such as lack of a central point, mobility, wireless links, limited energy resources, a lack of clear line of defence, cooperative nature and non-secure communication to mention a few.

This research proposes a two-phase scheme. In phase-one a novel approach is suggested by using concept of exiting trust schemes and adopting the use of Dynamic Trust Threshold Scheme (DTTS) for the selection of trusted nodes in the network and using mutual trust acknowledgement scheme of neighbour nodes to authenticate two communicating nodes. The notion of trust is used for authenticating peer nodes. The trust scheme algorithm is based on real time network dynamics, relevant to MANET conditions, as opposed to pre-determined static values. The phase-one is implemented in AODV and tested in a simulated environment using NS2. The reason for using AODV is that it's reactive and has comparatively low routing overhead, low energy consumption and relatively better performance (Morshed, et al 2010). In order to ensure data confidentiality and end-to-end security, in phase-two, the source and destination generates a shared secret key to communicate with each other using a highly efficient Diffie Hellman Elliptic Curve scheme (Wang, Ramamurthy and Zou, 2006). The shared key is used to encrypt data between the peer nodes.

CHAPTER 1

INTRODUCTION

As the number of mobile devices and wireless network users are continuously growing and the capacity of mobile computer increases, the need for unlimited networking is expected to rise. Easy and quick deployment of wireless networks highlights the significance and importance of MANET networks in future of wireless networks, which is not possible with the existing structure of current centralised wireless systems (Kumar, and Mishra, 2012). A MANET is a collection of autonomous nodes or terminals that communicate with each other by forming a multi-hop, dynamic and purpose specific radio network that maintains connectivity in a decentralised manner (Sun, 2001). The decentralised architecture and dynamic topology of these networks allow the nodes to join and leave network at any point of time, as the nodes move or adjust their transmission and reception parameters. A MANET is deployed in a situation where the normal infrastructure network connections are not available in a given geographical area. MANETs can be used in military to communicate between soldiers, vehicles and military headquarters. It is also used in emergency and rescue services such as flood, fire and earthquake. It is widely used in location aware and educational services (Raja and Baboo 2014).

A MANET is more vulnerable to security attacks than conventional wired and wireless networks due to its distinct characteristics such as dynamic topology, open medium, no centralised access points, distributed cooperation and lack of association. Both authorised and malicious nodes can access the network. As a result, the network is prone to any form of security attacks from both inside and outside. Some mechanism is needed that prevents a node from learning the identity and credentials of other

nodes. Current standard MANET routing protocols do not focus much on the security and privacy issues such as Confidentiality, Integrity, Authentication and Availability (Sharma and Chauhan, 2015). Routing in MANET is based on mutual trust between the nodes but the lack of centralised mechanism prevents the use of conventional security techniques. To complicate matter further, various limitations of nodes such as power and bandwidth constraints, frequent disconnection of links and short battery life poses an important challenge in implementing cryptographic algorithms for providing the required essential security. Nodes having low energy can partially or drop all packets to conserve energy. If a node spends very large part of its battery power then it may not be available (Gupta, and Sexena, 2010).

Security has always been a concern in wireless networks but due to the nature of MANETs, it presents new challenges to security design. Ad-hoc networks are dynamically formed amongst a group of wireless users and do not require a fixed network infrastructure (no central administrator) or pre-configuration (Liang, Poor and Ying, 2011). Nodes function as a router and rely on neighbours to communicate and relay messages if not within the same communication range (Perkins, Belding-Royer and Das, 2003). This fast changing topology and other vulnerabilities make it essential to provide security in such networks at each individual node. As every node functions as a router and do not reside in physically protected places and wireless channel is accessible to all nodes, both legitimate users and malicious attacks therefore, can easily fall under attack (Annarasi and Sevanesh, 2014). Also, the lack of well-defined place, traffic monitoring and access control mechanism cannot be deployed. Open peer-to-peer network architecture, shared wireless medium, limited resources and highly dynamic topology poses a number of additional unconventional challenges to security design (Sharma and Chauhan, 2015).

The level of trust will be represented as the belief probability varying form 0 (not trusted) to 1 (fully trusted) as described in (Cho, Swami and Chen, 2011). According to (Cho, Swami and Chen, 2011), trust is dynamic and not static because the nodes are mobile and the network dynamics change rapidly and as elaborated further by the same author, that due to dynamicity of trust, trust should be expressed as a continuous variable. This led us to our motivation for attempting to devise a method that calculates trust in a dynamic fashion rather than a relying on a static value.

In order to address some of these security challenges; a two phased solution is proposed that is based on taking pure MANET features into account (Nikander, Kempf and Nordmark, 2004). In phase-one a novel approach is suggested by using the following two steps:

1. Developing a Dynamic Trust Threshold Scheme (DTTS) for the selection of trusted nodes in the network by adopting existing trusted scheme (Khan et al, 2014).

2. Using dynamic trust to authenticate two communicating nodes known as mutual authentication scheme

In this phase (phase-one), a trust-based model framework is proposed that provides the foundation for authenticating nodes without having any prior trust relationship. The trust model is highly adaptive and responds to network changes in real time. It will allow all nodes to calculate the trust of neighbour nodes by taking their specific parameters into account. Therefore, the proposed trust model is dynamic as the parameters used are one-hop neighbours, two-hop neighbours, mobility, energy and trust which

are used to calculate the dynamic trust values and this has a direct relationship with MANET operations.

In an attempt to provide end-to-end security solution, a novel method referred to a mutual authentication scheme is introduced in this thesis. Authentication is one of the main principal for security in any system and in order to implement it in a pure MANET environment, it can present a great challenge. The scheme can be used as supplement to existing authentication schemes. Various cryptographic techniques could be applied to this scheme in variety of ways. The mutual authentication can be a topic for further research and explored to enhance authentication process in MANET.

The proposed Dynamic Trust model will be applied in the following way to achieve mutual authentication and its performance will be tested in a simulated environment. Once, the dynamic trust is established in all nodes, then the source and destination nodes request each other's trust values from its corresponding one-hop neighbours. The source node S prior to sending any data will request the trust value of the destination node (D) from D's one-hop neighbour to authenticate D and verify its trust. Destination will repeat the same authentication process for the source S to get the updated trust values from its neighbours.

Phase-two is mainly concerned with applying cryptographic techniques (Wang, Ramamurthy, and Zou, 2006; Koblitz, 1987), to ensure confidentiality through encrypting data between peer nodes. This is the last step of the scheme in which an efficient key exchange solution to encrypt data is demonstrated. This is in-line with the main goal to provide end-to-end security between communicating pair nodes.

The cryptographic schemes that are implemented at this stage are state-of-the-art and one that can provide maximum efficiency given the specific features of MANET.

The algorithm will be simulated and analysed using Network Simulator NS2.33. Linux Ubuntu version 16.04 is used, as NS2 is open source and its Linux based. The scheme will be tested and analysed by introducing malicious nodes in network using AODV protocol to detect and isolate such nodes. Various tests will be conducted to validate that the scheme can mitigate against Denial-of-Service (DoS) attacks such as Blackhole (Jhaveri, 2012) and Wormhole (Anju and Sminesh, 2014). The simulation will be generated under varying network conditions such as mobility, network size area, simulation time, data rate and node count using standard AODV as a reference to compare it with the proposed trusted scheme. At the end, the results will be analysed to evaluate standard AODV performance against trusted AODV in terms of security. The performance metrics tested will be throughput, routing overhead, packet statics, packet delivery ratio and end-to-end delay.

1.1 Statement of the problem

There are inherent security weaknesses and vulnerabilities in MANETs. The cooperative nature and self-organising capability of MANET, makes it considerably challenging to build trust between nodes. Trust of a particular node can be perceived from a perspective of peer node on reliability and accuracy of information received from nodes while traversing the network (Cho, Swami and Chen, 2011). Trust measurement in MANET can pose a considerable challenge due to the nodes power constrains, dynamic and highly cooperative nature between nodes. Hence, trust computation can become highly

complicated given the limitation of the MANET environment and if achieved then the node and network will be deemed highly trusted and secure. The trust computation in static network is relatively simple as they are more behaviour oriented and could be predicted when they are closely observed (Cho, Swami and Chen, 2011).

A Trust based or reputation based system could be referred to one that discovers , records and utilises reputation to form trust threshold and uses trust to influence the foundation for the security scheme to be implemented and provide a multilayer complete security system for the whole network (Cho, Swami and Chen, 2011). The results obtained from the Dynamic Trust Threshold scheme are used to make guided and reputation based decisions about the network, its topology, dynamics and identifying most eligible nodes. The Dynamic Trust Threshold scheme can be seen in isolation or a seamless self-configured security system that provides security solution for routing and data communication. The scheme not only provides the capability to make well informed decisions but reinforces the security against any internal attacks in case of any breach. This thesis proposes a trust based algorithm to support secure communication between nodes and protect against various threats. The requirement and aim for the thesis is to achieve the security goal of Availability, Integrity, Authenticity, and Non-repudiation (Mohandas, Silas and Sam, 2013). This can only be achieved by forming a secure channel between the communicating nodes. The algorithm used, is a combination of a dynamic trust threshold and available efficient cryptographic techniques to achieve the above security goals. The scheme is divided into two phases. The first phase of the algorithm is to build the trust factors which provide a secure platform for the later phase known as phase-two of the proposed scheme, which is the implementation of the secure key exchange scheme to secure communication between nodes in the network.

The dynamic trust algorithm integrates the trust protocol and security attributes of nodes functioning as routers to provide an integrated security metric for the whole network. In MANET the nodes are fully cooperative and rely on each other for routing and message forwarding. The formation of trust between the nodes is a primary goal as it plays a vital role in securing communication (Cho, Swami and Chen, 2011). The continuous evaluation, of a node performance to calculate trust, is needed to reflect the changes in trust level of nodes. The cryptographic operations are performed, once the trust level is achieved. To securely process data, secret key generation techniques and the cryptographic operations are performed to provide authentication, message integrity and non-repudiation, once the desired trust threshold is achieved by a specific node (Zhao, et al 2012)..

1.2 Research Questions

This research endeavours to find a solution to the following main questions:

- How to provide pre-authentication between nodes in dynamic and unsecure wireless Adhoc environment when nodes communicate for the first time.

- How to provide confidentiality to the data being exchange between nodes.

- How can these solutions are provided without using unconventional security methods.

- How the above solutions are compared in terms of computational cost to standard protocol.

Authentication requires how to validate peer nodes and confidentiality requires cryptographic keys exchange (Sharma and Chauhan, 2015). This research is based on finding answers to the above research questions by integrating mutual trust mechanism using AODV routing protocol, which will provide foundation for a trusted framework and devise scheme for a generating a share secret key mechanism that is tailored specifically to MANET environment (Zhao et al, 2012)..

1.3 Research Aims

- The aim of this research is to find a solution to the research questions outlined in section 1.2 and to put in place appropriate countermeasures to mitigate threats against the above mentioned vulnerabilities.

- To provide solutions to the security challenges faced by MANET using tailored and efficient cryptographic techniques.

1.4 Research Objectives

Following are the steps needed to achieve the research aim:

1. Full literature review of MANET, trust based and cryptographic techniques.

2. Define and measure network specific metrics. Design a mathematical model that could calculate trust in nodes based on dynamic trust threshold values.

3. Using various cryptographic techniques and the above metrics as basis to implement security in the network.

4. Final step is testing, analyzing and evaluation of the trusted scheme.

1.5 Research Process

To achieve the above aims and objectives, the following steps are used to conduct a detailed research and design and implement the model.

Step 1: Literature Review

To conduct a detailed study of wireless and mobile Adhoc networks, given the specific infrastructure of MANET and its mobile and highly dynamic architecture. The aim is to thoroughly study its full details, design a workable solution that suits this type of network. Part of this study is to learn about different types of protocol in use for Adhoc networks.

Step 2: Security Challenges

The aim is to research the conventional methods of how security is applied in infrastructure networks and how those techniques could be tailored and applied in MANET. A full analysis of the vulnerabilities and security challenges faced by MANET is then considered. A detailed study will then consider the other features of MANETs, such as the type of protocols, types of wireless communications and forms of security attacks.

Step 3: Encryption Scheme

Cryptography is essential part of security in any network hence; a detailed study of the cryptographic methods considers a security scheme based on Confidentiality, Integrity and Authentication normally referred to as CIA (Zhao et al, 2012). To ensure data integrity and confidentiality various encryption techniques will be tested and implemented. Encryption involves the use to different types of symmetric and asymmetric key techniques that could be implemented in MANET environment. A study and test is then carried out to use the most efficient encryption algorithm to ensure security but not at the expense of performance.

Step 4: Dynamic Trust Threshold

Various parameters that play a vital role in how MANET operates are considered in this work. These parameters include nodes neighbour connectivity, nodes energy and mobility. These parameters form

the basic building block of dynamic trust scheme (Khan et al, 2015). This will provide the foundation for the basic layer of security in the form of trusted and untrusted nodes.

Step 5: Design analytical model

There is a need to design a new algorithm as existing protocols designed for static LAN and WAN have no such schemes implemented. The aim is to design a scheme using AODV. The AODV protocol is modified to implement the proposed scheme and the standard AODV is used as reference to compare and validate the proposed scheme. Details such as the formulas and various equations used in the scheme will also be explained.

Step 6: Trust Based Scheme Design

Trust based algorithm will be researched and will provide a platform for the Dynamic Trust Threshold scheme. This is the first step towards a multi-layered approach to provide security in MANET. Part of our literature review is to thoroughly review the trust based schemes designs so far by different researchers. The aim is to design a scheme by taking into account and adapt methods that best suit the trust scheme.

Step 7: Implementation and testing

In this stage NS2 is used to test and examine the performance of algorithm. NS2 is an open source simulation tool and is Linux based. NS2 supports C++ and TCL programming language (Henderson 2011). Therefore, Object Oriented C++ is essential to modify the existing AODV protocol in order to

implement the proposed scheme. An advanced level understanding of a Linux operating system such as Ubuntu is also needed to fully utilize the simulation environment. This step also includes the testing of the proposed scheme, which involves debugging and troubleshooting, programming and design issues. A few additional steps have been included to AODV code to implement the trusted scheme. Also, some modifications to packet header of hello-packets have been made to exchange the trust and parameters information (Bhanot and Chaudhary 2017).

Step 8: Results analysis

Once, the trust scheme is implemented using AODV, the next step is to test the scheme by comparing its performance with standard AODV under varying conditions. This is an important step, as it will prove how the proposed trusted AODV is performing against the standard AODV and this will be used as reference. The performance will be tested using various parameters, packets statistics and network conditions such as the number of nodes, mobility, scalability, simulation time, area and number of malicious nodes. The performance will be tested using the following metrics:

- **Throughput:** It is the amount of data (bit or packets) per period of time (seconds).

- **Routing Overhead:** Routing overhead represents any control packets required by protocol to perform a specific task.

- **Average End-to-end Delay:** It is the average time taken by packet to reach from source to the destination.

- **Packet Delivery Ratio:** The ratio at which packets are delivered in the network.

Step 9: Modifications

This stage involves testing and optimization for further improvement in algorithm. This is about modification needed after a rigorous process of testing and analysing the results. This is an on-going process until the protocol is successful in achieving the objectives.

Step 10: Thesis Write Up

The final step is to write up the final thesis as shown in figure 1.1 below:

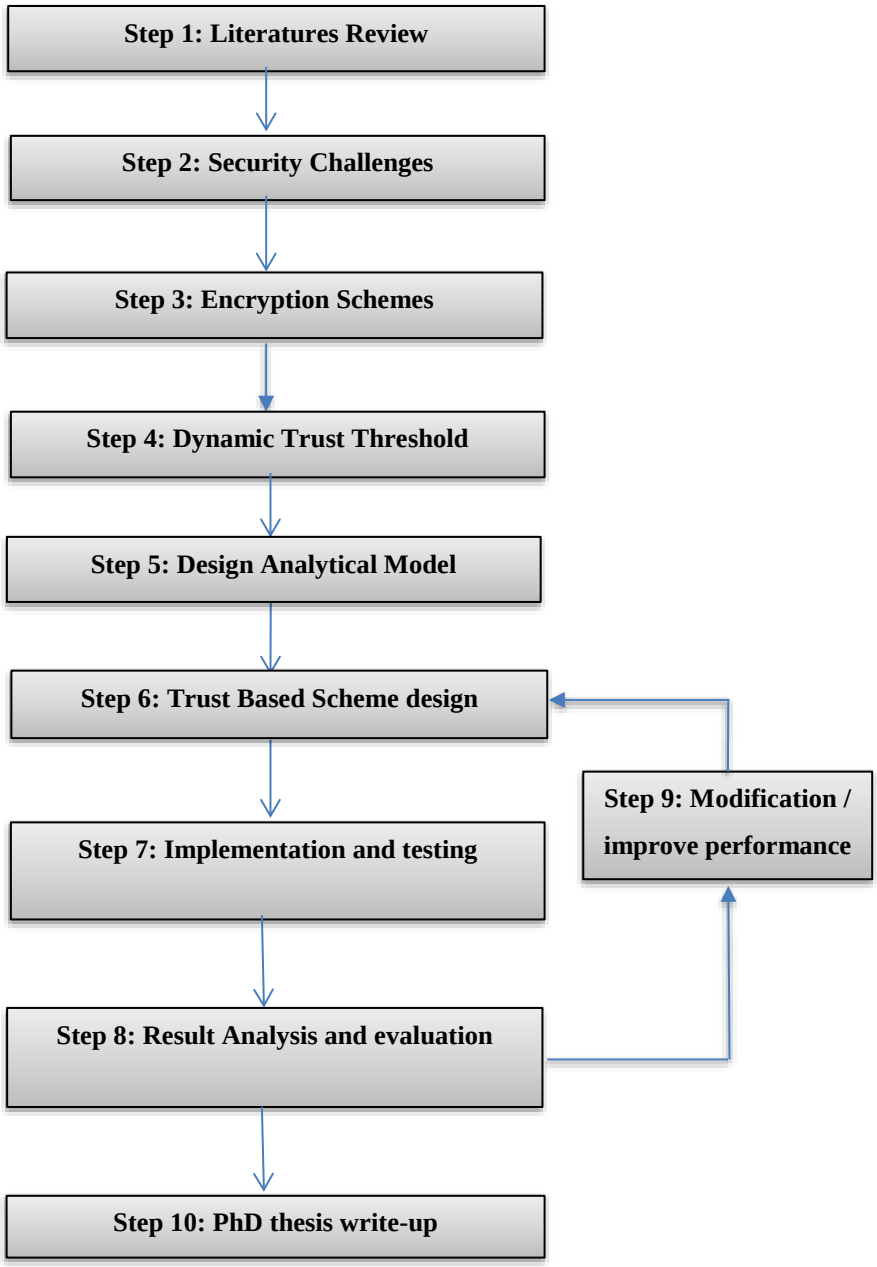


Figure 1.1 Research Process

1.6 Thesis Organisation

Chapter 2 covers the background information, the history and previous research that have been conducted on MANET. A detailed overview of how MANET works; its features and its importance in the modern day research is presented. The security aspect of MANET is then discussed in detail. The conventional security techniques and schemes those can be applicable to MANET. Cryptographic schemes that can be applied in MANET are also discussed. Lastly, the trust based schemes and other security schemes that have already been implemented in MANET are explored.

Chapter 3 is about the importance of authentication and different ways it can be implemented in especially in MANET. Different methods are discussed that are adapted to authenticate nodes using trust in pre and post network initialisation and its advantages and disadvantages. Also, a brief deliberation over various ways of the trust establishment is presented. Once, a certain level of trust is obtained and nodes are authenticated, how the key management protocols can be implemented is also discussed in this chapter. Lastly, the reason for the selection for choosing the parameters needed to dynamically calculate trust threshold are discussed.

In chapter 4, the mathematical model of the scheme has been discussed including the details of the protocol design stages and specifications involved in designing the algorithm are presented. The equations used to implement the scheme in NS2 are also discussed in detail. The parameters used to calculate the trust threshold and how the parameters are derived, have also been presented in detail.

Chapter 5 is about how the trust based scheme is implemented. The simulation tool NS2, which is used to implement the scheme is discussed. The benefits and reasons for its selection are also discussed. How simulation environment is setup, mobility and energy models is selected and implemented, malicious nodes and the modified hello packets used to carry the trust parameters are discussed.

Chapter 6 is about the performance evaluation of the trusted scheme. The performance parameters and various scenarios used to test the scheme are presented. The results are shown and analysed in detail to show the performance of AODV using the dynamic trust scheme as compared to the standard AODV. The numerical model is shown to prove, how the proposed scheme can reduce false positives.

Chapter 7 concludes the thesis by presenting the final conclusion. The proposed scheme is critically analysed and future work are discussed in this chapter. The thesis summary, contribution and the challenges encountered during the research are also presented in this chapter.

CHAPTER 2

Background and Literature Review

In this section, the history, background information and previous work been done on MANETs is analysed. MANET is a wireless network of mobile nodes which is self-organized. Every node can function as a router and communicate with another node thus; it is also called a multi-hop network. The early MANET projects were driven by the military for communication in places with no infrastructure available and there was a need to setup fast and easy networks (Sun, 2001). In those days flooding approach was adopted instead of multi-hop routing. As the basic routing and efficiency in terms of communication were achieved, there arose the need for security. As no provisions for security were taken into account at the time of the development of routing protocol, therefore the protocols were not designed with security of the network embedded (Jhaveri, 2012). Hence, the main focus of the research was diverted towards implementing security solutions.

The research on MANET was initialised by Defence Advance Research Project Agency (DARPA) in the late 70s, and was called PRNET packet radio network, (Kahn et al, 1978). This research was initiated to provide communication between computers and urbanized PRNET. The PRNET then evolved into Survivable Adaptive Radio Network (SUARN), (Kumar and Mishra, 2012). Since the increase in Adhoc network in commercial and domestic areas in the form of PDAs, laptops and pocket PCs. This surge resulted in the increased need for information exchange as well. The importance of Adhoc networking became even greater with the emergence of wireless technologies, such as IEEE 802.11, Bluetooth and HIPERLANE (Xu, Hischke and Walke, 2003). The functioning group of MANET was born in Internet Engineering Task Force (IETF).

2.1 Wireless communication in MANET

There are various wireless communication technologies used in MANET. Most common are IEEE 802.11, ZigBee 802.15.4 and Bluetooth. These are discussed in detail, in the following section;

- **IEEE 802.11:**

In 1997, the Institute of Electrical and Electronics Engineering (IEEE) created the first wireless local area network (WLAN) standard. It is a set of physical layer (PHY) and media access control (MAC) specification for implementing WLAN computer communication in 900 MHz, (IEEE 802.11, no date). They are created and maintained by IEEE (IEEE 802.11, no date). Due to the low speed of 2Mb in 802.11, the IEEE expanded the original standard creating 802.11b, 802.11g, 802.11a, 802.11n and on. This is the one of the common wireless technology used in MANET communication.

- **ZigBee 802.15.4:**

It is an IEEE 802.15.4 based specification that is packet-based radio protocol intended for low-cost, battery operated devices (IEEE 802.15, no date). Some of the features include support for multiple network topologies, low latency and low duty cycles to provide long duty cycles.

- **Bluetooth IEEE 802.15.1**:

It uses low-powered radio communication to link phones, PDA's and computers wirelessly. It is another common wireless standard used for MANET communication. Unlike 802.11 family, Bluetooth

was developed as an alternative wireless network technology with a relatively short range of approximately 10 meters and bandwidth of 1-3 Mbps (Rashid and Yusoff, 2006).

As Radio Frequency RF is used as medium of communication in MANET therefore, it inherits all the properties and characteristics of wireless networks such as the wireless channel security threats and wireless medium unreliability compared to wired network. The routing protocols developed for MANET then replaced the broadcast approaches, which were more efficient and robust. But yet another great challenge faced was securing the network from both passive and active attacks (Jhaveri, 2012; Rifquddin and Sukiswo, 2015). The routing protocols developed for MANET did not take security aspect of the network into consideration. The protocols were geared towards efficiency and speed. Therefore, the researcher drew their attention towards developing an efficient as well as secure protocol, one that require minimum resources and provide maximum security. There has been various protocol developed to address the security issue in MANET. Various security algorithms have been proposed by implementing various cryptographic techniques, considering the conditions of MANET, by different researchers. Some of them include SAODV (Juwad and Al-Raweshidy, 2008), SEAD (Hu, John and Perrig, 2002; Yu and Su 2009), TESLA (Perrig, Canetti and Song, 2002; Yu and Su, 2009), ARIADNE (Hu, Perrig and Johnson, 2005), SAR (Yi, Naldrug and Kravets, 2005), Security Aided Adhoc Routing (Carter and Yasinsac, 2002) and ARAN (ARAN, no date; Yu and Su, 2009). All the above protocols used various types of cryptographic techniques such as secret key, private key and hash functions (Zhao et al, 2012).

In the next section, we will elaborate on the different aspects of MANET functionality, routing protocols and security.

2.2 MANET Infrastructure

MANET is self-organising wireless mobile network; therefore, their protocols are designed taking into account the dynamic connection, with no centralized structure. All nodes behave as a router and take part in discovery and maintenance of routes to other nodes in the network (Kumar, M. and Mishra, S. 2012). Wireless networks are classified into two types:

2.2.1 Infrastructure Mode

Infrastructure mode uses a central device called a wireless access point. The access point is used to connect wireless nodes to an Ethernet network.

2.2.2 Ad Hoc Mode

Adhoc network is the aggregation of mobile nodes, communicating without any centralized mechanism and are also referred to as infrastructure-less network. The Adhoc capability comes at the cost of memory, computation power and limited battery power. The larger the network becomes with more nodes adding, it requires greater processing power, larger memory and bandwidth to maintain accurate routing information (Dodke, Mane and Vanjale, 2016). Figure 2.1 shows the types of wireless networks.

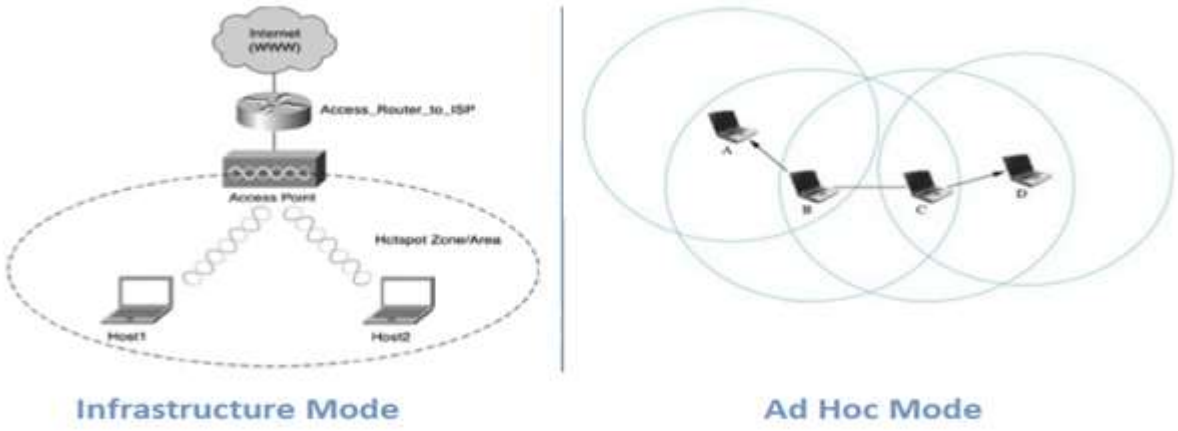


Figure 2.1 Types of Wireless Networks

2.3 MANET Routing

A wide variety of MANET protocols have been developed by researchers to meet the routing and mobility challenges. The protocols for MANET can be classified into three categories i.e. proactive, reactive and hybrid (Kumar and Mishra, 2012). Figure 2.2 shows the classification and types of routing protocols in MANET.

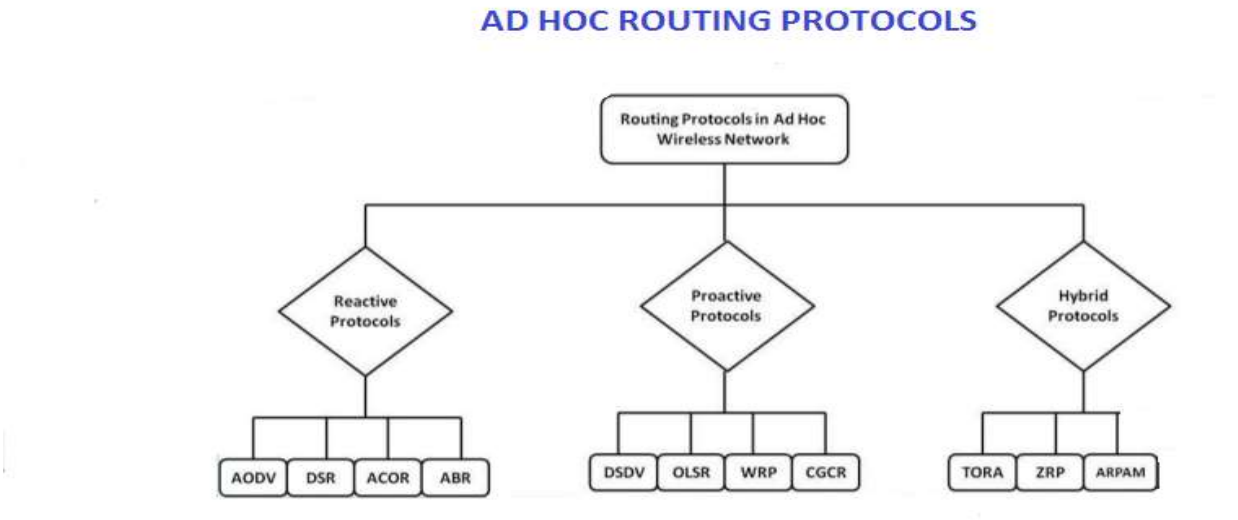


Figure 2.2 Routing Protocols in MANET

2.3.1 Proactive Routing Protocol

These types of protocols are table driven where each node maintains one or more table containing routing information to every other node. The table is updated periodically to maintain the up to date view of the network. The nodes are updated when there is a topology change. Proactive protocols can be distance-vector or link state (Mohandas, G., Silas, S. and Sam, S. 2013).

In distance-vector protocols the routing information is only exchanged between directly connected neighbours. The nodes cannot see beyond its neighbours and therefore, hold no knowledge of the entire topology.

In contrast, the Link-State that is the second type of the proactive protocol where all nodes know about the path reachable by all other nodes. Each node has information about its neighbours and obtains other topology information about the network from broadcast messages it receives from other nodes. Optimized Link State Routing OLSR (Singh and Verma, 2015), Destination-Sequenced Distance-Vector routing protocol DSDV (Gupta and Kumar, 2015; Yu and Su, 2009) and Wireless Routing Protocol WRP (Murthy and Garcia-Luna-Aceves, 1996) are some of the example of proactive routing protocols. A brief overview of these protocols is presented in the following section:

DSDV Destination-Sequenced Distance-Vector routing protocol (DSDV) (Gupta and Kumar, 2015) require all the nodes to send full routing updates. Therefore, every node maintains a routing table that contains the details of neighbours and also details of the routes to all other destinations. It uses Bellman-Ford routing algorithm to calculate routes (Gupta and Kumar, 2015). This protocol is a modified version of Destination Vector Routing DVR and was originally discovered by C. Perkins and P. Bhagwat in 1994 (Singh and Verma, 2015).

WRP Wireless Routing Protocol (WRP) (Murthy and Garcia-Luna-Aceves, 1996) is another table-driven protocol. Every node in WRP protocol maintains a routing table, a link-cost table and message retransmission list. Nodes get updated periodically as well as when there is a topology change. Nodes exchange routing table with their neighbour when a new path is found and always update its routing table in case of a fresh route.

OLSR Optimized Link State Routing (OLSR) is proactive protocol that has routes available when ever required. The protocol declares a dedicated node as Multi-Point Relay MPR and only these nodes are able to broadcast data. This forms a key part of the protocol which also has the advantage of reducing the traffic to entire network (Sing and Verma, 2015).

2.3.2 Reactive Routing Protocol

These types of protocols take a reactive approach as opposed to table drive. The routes are generated as and when required by the communicating nodes. These routes remain valid until the destination is reachable or the route is no longer needed (Mohandas, G., Silas, S. and Sam, S. 2013). Both types of protocols come with their own advantages which will be discussed in later section. Types of reactive protocols and their brief explanation are discussed in the later section. Several protocols are going through IETF standardization, some of them will be discussed in this section.

Perkins, C., Belding-Royer, E. and Das, S. (2003) Adhoc On Demand Distance Vector (AODV) is one of reactive protocol that initiates the process of route discovery when the packet needs to be sent. As there is no route known prior to request to the destination therefore, the protocol initiates the process of route discovery. The routing messages do not contain information about the whole path but only the source and destination information is held. Whenever, a node needs to send a packet to another node, it broadcasts a Route Request RREQ message. Each node that receives a broadcast, sets up a reverse route towards the originator of the message i.e. source node. Once the destination is reached, the destination node sends a Route Reply RREP message. It uses destination sequence number to identify the fresh route (Perkins, Belding-Royer and Das, 2003).

DSR Dynamic Source Routing DSR is another type of reactive protocol that saves bandwidth by restricting the use of control packets. The two main phases of this protocol is route discovery and route maintenance. The major difference between the two phases is that the later does not require periodic hello transmission i.e. its beacon-less (Gupta and Kumar, 2015; Chai-Keong, 1996).

2.3.3 Hybrid Routing Protocol

Hybrid combines features from both reactive and proactive protocols. The aim is to make it more efficient. Some types of Hybrid protocols are Temporary Ordered Routing Algorithm TORA (Murthy and Garcia-Luna-Aceves 1996) and Zone Routing Protocol ZRP (Hass, Pealman and Samar, 2003). Some explanations of theses protocols is given in the next section. TORA Temporary Ordered Routing Algorithm aims to be more efficient by reducing the control messages to minimize the communication overhead. The nodes only have information about neighbour i.e. one-hop. It maintains the state, on the destination basis, like distance vector protocols and its destination oriented nature represents its reactive nature thus, it is known as hybrid (Murthy and Garcia-Luna-Aceves 1996).

ZRP Zone Routing Protocol (Hass, Pealman and Samar, 2003) combines the best of both reactive and proactive routing protocols. As the names suggests, it is based on the concept of zones. It reduces initial route discovery delays by employing reactive routing and control traffic by using proactive routing techniques. Every node is defined into its specific zone. It uses proactive approach in a limited

zone where the maintenance of routing information is easier. For furthest nodes it uses a reactive approach.

2.4 MANET Security

Security is protecting systems, networks, programs and other assets from any digital attacks. As the attacks constantly evolve therefore, they need to be identified and mitigated. Common network is exposed to different types of security attacks. These security attacks can be aimed accessing, tempering or destroying sensitive information. The most common forms of security threats that are faced today are as follows:

- **Malware**

  It is piece of software that runs like any other software and the key difference is in the behavior. A piece of software is considered as malicious, if it shows activities like replicating, disabling certain security feature, data stealing etc., then it can be considered as malware (Shaid and Maarof (2014).

- **Phishing**

  It is a type of social engineering attack in which various methods are used to fool people into disclosing their sensitive information. The common types of phishing attack are spoofed emails, fake social networking accounts, Trojan horse and hacking (Gupta, Singhal and Kapoor, 2016).

- **Denial-of-Service (DOS)**

  This is the type of attack, where a legitimate user is denied access to the information. Some of the types are Blackhole (Jhaveri, 2012), Wormhole (Anju and Sminesh, 2014) are classed as Denial-of-Service (DoS) attacks. Other types of attacks are Byzantine (Yu and Su, 2009), Rushing (Sukiswo and Rifquddin, 2015).

- **Man-in-the-Middle (MITM)**

  It is when an attacker place themselves inline between two devices or people that are communicating. The intention is to manipulate the data as it traverse between the communicating parties (Xu and Liu, 2017).

- **Brute Force Password attack**

  This is one of the most widespread types of attacks in computer networks. In this type of the attack, the attacker continues trying different passwords on the target machine in an attempt ot reveal the loin password (Najafabadi et al 2014).

2.4.2 Forms of Attacks in MANET

The attacks in MANET are of two types; internal or external attacks. These attacks can be further classified into two categories known as active and passive attacks.

- **Internal vs. External Attacks**

Internal attack also called insider attack comes from the compromised node that belongs to the network and might feed other nodes with incorrect information. External attack can be in the form of an adversary injecting harmful information into the network. This type of attack is normally launched by the node that is not part of the network (Meddeb, et al 2017).

- **Active vs. Passive Attacks**

Active type of attack comes with the aim of a node damaging other nodes by performing harmful operations such as network outage, information interruption, modification or fabrication and disrupting the whole functionality of the network. Examples of active attacks are DoS, spoofing, replying, jamming and modification. Passive attacks on the other hand, obtain data exchanged in the network without disrupting the network operations. Examples of passive attacks are traffic monitoring, eavesdropping and traffic analysis (Liang, Y., Poor, H. V. and Ying, L. 2011).

2.4.3 Security Threats in Manet

Routing protocol in MANETs such as AODV, were designed without taking security considerations into account therefore, it is prone to number of security threats as mentioned earlier. There are number of attacks that has been identified and studied in MANET. The types of attacks also depend on which and what type of network has been targeted. We will discuss more advanced attacks that could affect MANET. Some of the types are Blackhole (Jhaveri, 2012), Wormhole (Anju and Sminesh, 2014) are classed as Denial-of-Service (DoS) attacks. Other types of attacks are Byzantine (Yu and Su, 2009),

Rushing (Sukiswo and Rifquddin, 2015), Grayhole (Jhaveri, 2012), Resource consumption (Yu and Su, 2009), and Flooding (Sukiswo and Rifquddin, 2015).

- **Blackhole Attack:**

A malicious node absorbs the network traffic and drops all packets in Blackhole attack. The attacker node drops packets destined for other nodes. Thus, denying services to legitimate nodes in the network. When a malicious node receives a RREQ packet from another node, it sends a false route reply by spoofing its neighbour that, it has best route to the destination. The Blackhole node drops any packets it receives rather than forwarding them to destination node (Sharma, K. S. and Sharma, V. 2016)

- **Greyhole Attack:**

Grey hole is also a type denial-of-service attack where malicious nodes initially act as normal node but starts dropping all or some of the packets it receives (Jhaveri, 2012).

- **Byzantine Attack:**

This can be defined as an attack against the routing protocol in which two or more colluding routers attempt to disrupt routing operation by modifying, fabricating or dropping packets (Yu, Zhou and Su, 2001).

- **Wormhole Attack:**

In this type of attack the adversary node captures the packet at point of the network and tunnels is through to another point (Anuj and Sminesh, 2014). Two colluding nodes normally initiate the attack, where one node is near the source and another near the destination creates a tunnel to direct the flow of packet through the tunnel.

- **Rushing Attack:**

Rushing attack is the type of attack that results in the Denial-of-Service against Adhoc network routing protocols. The attacking node when receive a RREQ, it exploits duplicate suppression mechanism by quickly forwarding RREP to gain access to the data being forwarded by the forwarding group. The receiver accepts the rushed packet and discards other legitimate RREP packets (Sukiswo and Rifquddin, 2015).

- **Flooding Attack:**

Flooding attack is where a malicious node floods the network with fake RREQs or data packets to block the network and hamper any real data transmission by legitimate nodes (Sukiswo and Rifquddin, 2015). The attacker broadcast many RREQ to communicate with node that might or might not exist in the network. All this result in network congestion and the bandwidth is severely is compromised.

- **Jellyfish Attack:**

In this type of attack a delay is added to the packet by holding the packet for some time before they are propagated. The attacking node makes network believe control protocol that there is congestion in the network therefore, the network is incapable of meeting the processing requirement of the user. As a consequence, this leads to the disruption of the network communication as, the network control protocols apply congestion control mechanisms (Kaur, M., Rani, M. and Nayyar, A. 2014).

2.5 Network Security and Cryptography

Cryptography is the art or practice of securing information, in the presence of third party, by converting it into unreadable information. The information can only be read by those in possession of the secret key to decrypt or decipher the message into plain text. In short, cryptography is the science of writing in secret codes. The use of cryptography could be dated back to 1900 B.C (Wang, Ramamurthy and Zou, 2006). Cryptography concerns itself with aiming to achieve four objectives (Davis, 1978; Nie and Zhang, 2009 and Zhao et al 2012);

1. **Confidentiality:** The information can only be read by the receiver for whom it is intended.

2. **Availability:** A service is available to legitimate users when required.

3. **Integrity:** Ensuring the message exchange between sender and receiver is not altered.

4. **Non-Repudiation:** A process to prove that the sender really send the message.

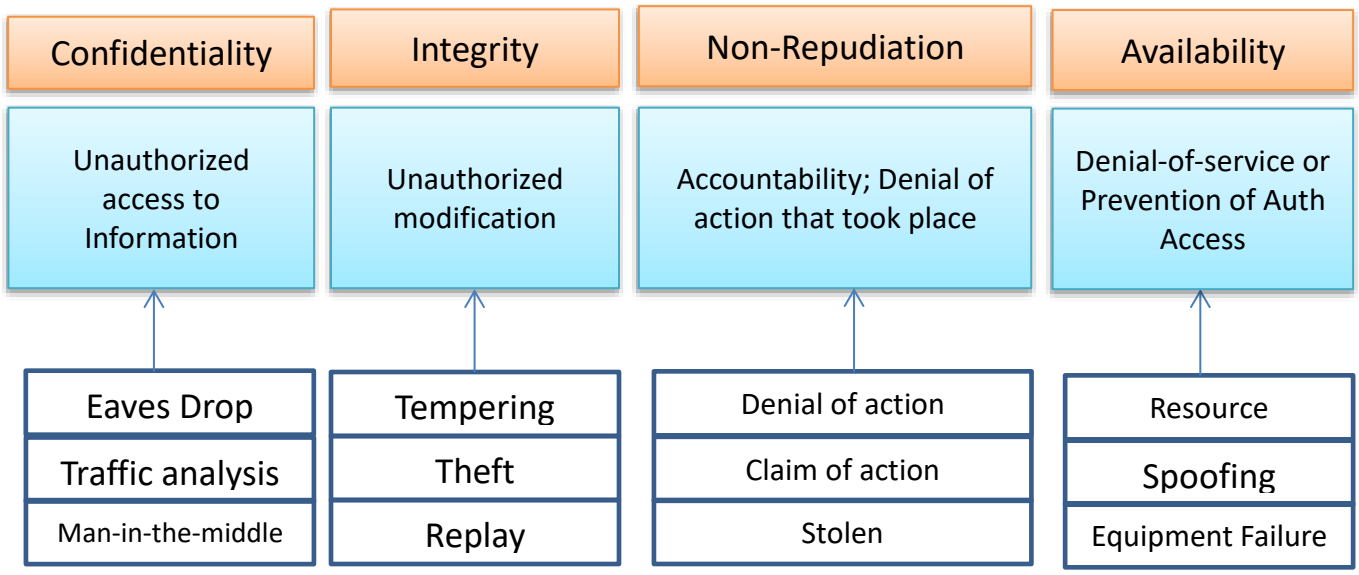


Figure 2.3 Network Security Model

If the above four attributes have been achieved in any systems that has implemented cryptographic techniques then that system would be classed as having some level of security achieved. The security model in figure 2.3 presents the threat landscape; it shows the consequences if any of the attribute fails, it can result in various vulnerabilities. For example, lack of confidentiality can result in an unauthorised access to information.

The common forms of attacks to compromise confidentiality are man-in-the-middle, eaves dropping and traffic analysis as shown in figure 2.3. There are several ways of classifying cryptographic techniques, as given below:

1. Secret Key Cryptography (Symmetric key)

2. Public Key Cryptography (Asymmetric Key)

3. Hash Functions

4. Hashed Message Authentication Code (HMAC)

5. Digital Signatures

6. Pairwise Keys

2.5.1 Secret Key Cryptography

Secret key only uses single key for encryption and decryption of message as shown in figure 2.4. It is also referred to as symmetric key due to the use of single key. The sender and receiver both possess one key to encrypt and decrypt. Some of the known secret key cryptography in use today includes the following:

- Data Encryption Standard (DES) (Davis, 1978; Nie and Zhang, 2009; Rizvi, Hussain and Wadhwa, 2011)

- Triple-DES (3DES) (Nie and Zhang, 2009; Rizvi, Hussain and Wadhwa, 2011)

- Advance Encryption Standard (AES) (Sathiamoorthy, Ramakrishnan and Usha, 2015; Nie and Zhang, 2009; Rizvi, Hussain and Wadhwa, 2011)

- Camellia (Lu et al, 2012)

- TwoFish (Rizvi, Hussain and Wadhwa, 2011)

- Blowfish (Nie and Zhang, 2009; Rizvi, Hussain and Wadhwa, 2011)

Figure 2.4 Secret Key Cryptography

2.5.2 Public Key Cryptography

Modern PKC first proposed by Martin Hellman and Whitfield Diffie in 1976 (Wang, Ramamurthy and Zou, 2006). The paper basically proposed a cryptosystem where by two parties could communicate over a non-secure channel and without having to share a key. The algorithm uses two separate keys to encrypt and decrypt message as shown in figure 2.5. It is also referred to as asymmetric key exchange protocol due to separate key used for encryption and decryption. One key is used to encrypt the plain text known as public key and another key known as private key is used to decrypt the cipher text (Zhao, et al 2012).

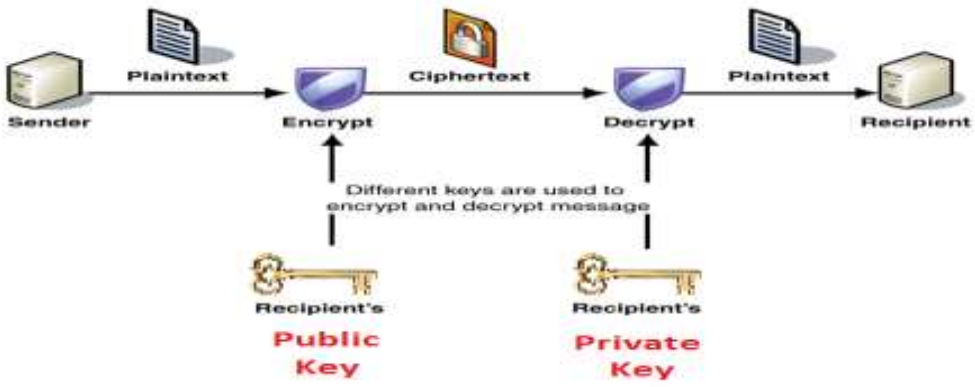


Figure 2.5 Public Key Cryptography

The designated public key can be advertised and therefore, is known to public. The private key however, is kept secret. Although, both the keys are mathematically related but the knowledge of public key does not give away any information or hints about private key. The strength of algorithm

comes from multiplication vs. factorization problem that are easy to compute and relatively difficult to calculate the inverse function (Zhao, et al 2012). This implies that it is easy to multiply two prime numbers and get a product but determining the prime factors of that product is long and hard to calculate, especially when the number is 400 digits long (Wang, Ramamurthy and Zou, 2006; Koblitz, 1987). Exponentiation vs. logarithms is the same one-way function problem where it is easy to calculate the power of a number but finding the inverse is a hard problem. The use of public key cryptography in key exchange and digital signature includes the following:

- RSA – named after Rivest, Shamir and Adleman (Sarkar, Kisku, Misra and Obaidat, 2009)
- Diffie - Hellman DH (Wang, Ramamurthy, and Zou, 2006)
- ElGamal – Based on DH exchange
- Digital Signature Algorithm DSA – Created by NSA (Sathiamoorthy, Ramakrishnan, and Usha, 2015).
- Elliptic Curve Cryptography ECC (Wang, Ramamurthy, and Zou, 2006; Koblitz, 1987).

This thesis uses the Diffie-Hellman ECC. Elliptic Curve Cryptography is a public key crypto system based on Elliptic Curve which was discovered in 1985 Victor Miller and Neil Koblitz (Koblitz, 1987;

Miller, 1986). It creates a mechanism for sharing keys among participants that is based on Discrete Logarithm Problem DLP that is much more difficult to challenge at equivalent length then other algorithm such as RSA. The vast majority of Secure Group Communication SGC (Zou, Ramamurthy and Magliveras, 2005) uses DLP-based Diffie Hellman as a basic key agreement. SGC refers to a scenario where messages are exchanged between groups of participants in such a way that any third party or eavesdropper is unable to glean any information even if they are able to intercept the message (Wang, Ramamurthy, and Zou, 2006; Koblitz, 1987).

2.5.3 Hash Functions

The hash function is an efficient way of mapping a binary string of arbitrary length to binary string of fixed length called hash-value or digest as shown in figure 2.6 (Ragab, Ismail and Farag-Allah, 2001). They are also called Message Digest MD or One-Way encryption. Encryption provides confidentiality but not necessarily integrity. Hash functions therefore are commonly used to provide integrity to messages. It is also used for digital finger printing of file content. There are number of widely used hash function the most common are MD and Secure Hashing Algorithm (SHA) (Juwad and Al-Raweshidy, 2008). The common types of MD hash algorithm are as follows:

- Message Digest2 (MD2): (Ragab, Ismail and Farag-Allah, 2001; Thulasimani and Madheswaran, 2009).

- Message Digest4 (MD4): (Ragab, Ismail and Farag-Allah, 2001).

- Message Digest5 (MD5): This creates 128-bit key (Ragab, Ismail and Farag-Allah, 2001; Thulasimani and Madheswaran, 2009). The SHA have few versions and the difference in the version is based on their efficiency and strength. The following are the most common types of Secure Hashing Algorithm.

- Secure Hash Algorithm1 (SHA-1):

  This creates 160-bit key (Ragab, Ismail and Farag-Allah, 2001)

- Secure Hash Algorithm2 (SHA-2):

  Option includes a digest between 224 and 512 bits

- Secure Hash Algorithm3 (SHA-3):

  Option includes a digest between 224 and 512 bits

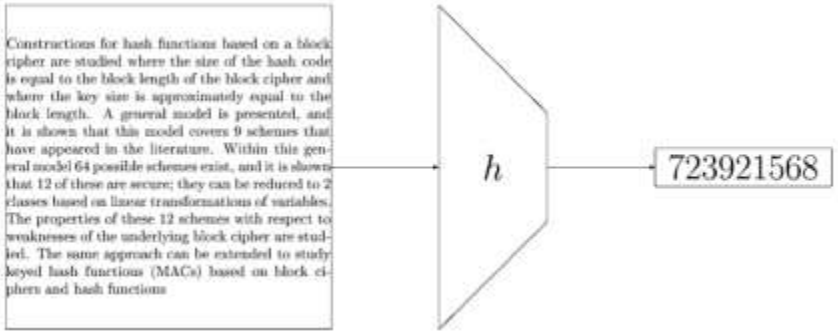


Figure 2.6 Hash Function

2.5.4 Hashed Message Authentication Code (HAMC)

This is a type of hash that uses a secret key of some type in its calculation. The receiving party who knows the secret key can only calculate the resulting hash and can verify it. The attacker or eavesdropper cannot remove or inject data because it doesn't have the key used for calculation (Zhao, et al 2012).

2.5.5 Digital Signature

This is a cryptographic technique that uses the Public Key Infrastructure PKI to generate public and private key issued by Certificate Authority (CA) (Hinds et al, 2012). The sender digitally signs a document using his private key that forms a hash and encrypts the data at the same time. The resulting

digitally signed document can only be decrypted by the sender public key (Zhao et al 2012). In cryptography Digital Signatures provides the following three main benefits:

1. Authentication

2. Data integrity

3. Non-Repudiation

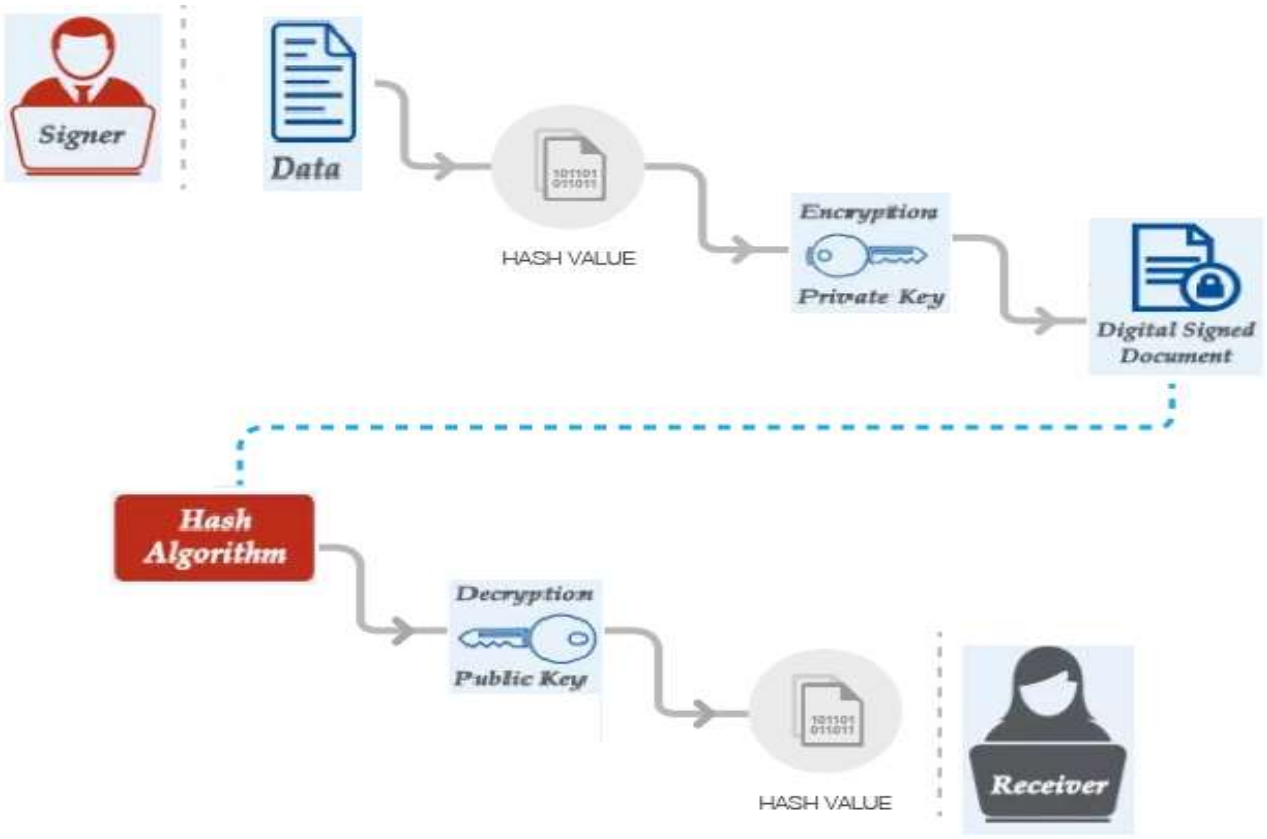


Figure 2.7 Digital Signature Algorithm

2.5.6 Pairwise Keys Distribution

According to this scheme, nodes are pre-loaded with keys and after deployment each node exchanges key information with its neighbours in order to establish a secure link between them (Chang and Agarwal, 2008). The idea is that every node that wants to communicate with other nodes share keys between them.

2.6 MANET Security Related Work

Extensive research has been done and various security protocols have been proposed by the researchers in an attempt to secure different aspects of MANET. The same principles that governs the security of MANET such as authentication, confidentiality, integrity and non-repudiation, are also used in traditional wired network except an important principle i.e. availability (Hinds et al, 2012). The mobility of nodes and constantly changing topology makes availability challenging in MANET. It is essential to the network operations. MANETs are vulnerable to attack on any level of the open system interconnection OSI model including physical attacks such as Denial of Service DOS or wireless jamming techniques as well as attacks on higher-level services such Key Management services (Hinds et al, 2012). Some of secure routing protocols developed for MANETs will be briefly discussed and analysed such as SAODV (Juwad and Al-Raweshidy, 2008), SEAD (Hu, John and Perrig, 2002), TESLA Perrig et al, 2002), Ariadne (Hu, Perrig and Johnson, 2005), SAR (Yi, Naldurg and Kravets, (2005), Security Aided Adhoc Routing (Carter and Yasinsac, 2002) and ARAN (Aran, no date).

2.6.1 Secure Routing Protocol

SAODV: Secure Adhoc On-demand Distance Vector (Juwad and Al-Raweshidy, 2008) routing protocol is used to secure the routing messages for the original AODV. Basically the SAODV uses digital signature to authenticate non-mutable fields and hash chain to authenticate the mutable field i.e. hop count for both route request RREQ and route reply RREP message (Zapata and Asokan, 2002).

ARAN: Authenticated Routing for Adhoc Network (Aran, no date) is another type of MANET security protocol that uses digital signatures to protect the non-mutable fields of the routing messages and uses Open SSL library for certification. This is thought to be time consuming and generate a lot of overhead.

SAR: Security Aware Routing protocol (Yi, Naldurg and Kravets, 2005) is a trust based reactive protocol. It uses trust values and relationships with the nodes which form the basis of its routing decisions. Only trusted nodes can participate in the routing. The protocol does not provide high-end security.

SPAAR: Another protocol proposed called Security Aware Aided Adhoc Routing (Carter and Yasinsac, 2002). It is a location aware protocol which uses geographical information to secure routing information and uses asymmetric cryptography i.e. the use of public key infrastructure for routing.

SEAD: Hu et al (2015) proposed Secure Efficient Ad Hoc Distance Vector and used a protocol, which is based on the design of DSDV. SEAD is designed to prevent attacks such as DoS and resource consumption attacks. Also uses One-Way Hash Chains to secure routing.

DSR: Ariadne also developed by Hu et al (2005) which is based on the operation of (Johnson, Hu, and Maltz, 2008). Ariadne (Hu et al 2005) uses message authentication code (MAC) and secret key shared between two parties to ensures point-to-point authentication of a routing message. Ariadne is a secure on-demand routing protocol and uses symmetric cryptographic operations. The protocol provides

security against one compromised node and prevents many types of denial-of-service attacks. However, it relies on the Timed Efficient Stream Loss-tolerant Authentication TESLA (Perrig et al, 2002). This is not suitable for MANET as it requires clock synchronisation.

2.6.2 Trust based schemes

Trust based routing protocol works by adding trust parameters to the nodes. Nodes operate in promiscuous mode and hear the conversations between other nodes in transmission range. Trust can be computed by taking into account different factors such as packets sent, received, acknowledged and forwarded by various nodes in the network. Therefore, nodes representing high trust can be selected as best path for communication. Trust schemes are used to mitigate security attacks and identify malicious nodes in the network as an alternative to cryptographic methods due to special characteristics of MANET. Extensive research has been carried out on the use to trust threshold schemes in MANET. The next section will discuss some of trust based schemes proposed.

Several techniques have been proposed to detect and eliminate malicious nodes in the network such as (Elhadi, et al, 2013; Marti et al 2000; Nasser and Yunfeng, 2007; Al-Roubaiey et al 2010; Botkar and Chaudry, 2011; Balakrishnan, Deng, Varshney, 2005; Buttyan and Hubaux, 2000; Zhong et al 2003; Buchegger and Le-Boudec, 2002; Jhaveri, 2013). One of the earliest techniques proposed was Watchdog and Pathrater. The Watchdog technique identifies misbehaving nodes while Pathrater technique calculates path avoiding misbehaving nodes (Marti, Giuli, Lai and Baker, 2000). The Pathrater rates every path in its cache and select a path that best avoids misbehaving nodes.

According to Buttyan and Hubaux (2000), the concept of incentives called beans to forward packets is introduced. Each node in return for participating in packet forwarding earns beans. The packet is automatically dropped when the packet run out of beans.

A credit-based scheme known as Sprite was proposed by (Zhong, Chen and Yang, 2003) in which the receipts of all packets sent and received are kept and reported to Credit Clearance Services CCS when there is an internet connection. The CCS can make decision based on its report about the individual nodes.

Scheme called Ex-watchdog proposed by (Nasser and Yunfeng, 2007) was proposed to address the weaknesses of watchdog scheme by discovering malicious nodes which can partition the network by generating false reports.

Another Intrusion Detection System proposed by (Balakrishnan, Deng, Varshney, 2005) relies on watchdog technique to overcome deficiencies in the original watchdog scheme by introducing end-to-end acknowledge called TWOACK.

Another trust based scheme called Adaptive Acknowledge scheme (AACK) (Botkar and Chaudry, 2011) is an attempt to reduce detection overhead while increasing detection efficiency through detecting misbehaving node rather than link proposed in TWOACK Balakrishnan, Deng and Varshney, 2005).

Muhammad et al (2015) proposed Adaptive Trust Threshold Strategy for detecting and isolating misbehaving node. The main difference is that it adapts to changes in topology and therefore, its threshold against which the trust is measured and compared is a dynamic value.

Confident scheme was proposed by Balakrishnan, Deng and Varshney (2005), which is also a reputation based scheme. It has four major components Monitor, Reputation System, Path and Trust Manager. Monitor performs watchdog function, reputation deals with node rating, path is about path rating and trust deals with alert messages.

There have been several detailed surveys conducted to analyse the role of trusted scheme in order to secure the network, some of which are discussed in the above section. The aim for the trusted schemes is to secure routing by detecting misbehaving node that includes both selfish and misbehaving nodes. One of the detailed survey conducted by Cho (2011), gives us an overview into metrics used for MANET trust management. The study is formulated in five tables and conclusion of the survey is made by making various recommendations. One of the recommendation and suggestion is that the trust metric must reflect the unique properties of trust in MANET and trust management design must support collaboration in such a way that every node in the network gets adapted to network conditions and MANET environment including node density, traffic and mobility patterns. This research has taken the similar approach to implement the trust along the same line. The metrics used to calculate dynamic trust reflects this approach and represent pure MANET environment (Nikander, Ed. P., Kempf, J. and Nordmark, E. 2004).

2.7 Simulation and Source Code

NS-2 is used as simulation software to implement the proposed secure AODV algorithm. It is an open source and one of the most frequently used simulation tool. There are many other simulation tools available such as

- QualNet
- Opnet
- GloMoSim
- NS-3
- OMNet++
- Jist

NS-2 is programmed in C++, however there is a separate scripting language used called OTcl, which is an object oriented version of Tool Command Language. The TCL is used to generate scenarios. It is primarily Linux based but can be run from other OS platforms using additional software. The simulator supports a variety of functions, application and network models. The simulator is mainly designed for

general purpose network simulation. This offers a great benefit as this implies that it is not designed or optimised for any specific type of network (NS2 documentation, no date). One of the main advantages apart from NS-2 being open source, it is very popular among research community and there is an online support available.

2.8 Summary

In this chapter the background information of MANET and its history is covered, including the research that has been done so far as well. Different aspects of MANET are thoroughly analysed such as how it works, its various features and how it is important and can contribute to the modern day technology. Security issues and limitation were also discussed and how these issues have been addressed in other research. How trust based schemes have been used to address and implement security in MANET were also critically analysed. A detailed analysis of the proposed trusted scheme was conducted to overcome various security issues in MANET.

CHAPTER 3

Authentication, Trust and Key Management

In this chapter, the association between authentication and trust will be discussed. We argue the importance and different forms of authentication from pre and post network initialisation stand point. How trust has been used as a framework for authentication and the reason for our selection of various parameters to dynamically calculate trust will also be presented. Lastly, how the proposed scheme is used to manage secure secret key protocol is also discussed.

The use of public key cryptography, which involves public and private key and the use of certificate authority, is not feasible for number of reasons (Zhao et al 2012). As explained in the literature review section, it involves the use of a CA to do all types of key management. The deployment of CA in MANET poses many challenges such as physical security, reachability, availability and centralized mechanism to mention a few (Johnson, Hu and Maltz, 2008). On the other hand, the secret key cryptography comes with its own challenges when it comes to implementing them in MANET. One of the approaches is that all nodes share the same secret key for authentication and encryption but due to the physical security of nodes, even this turns out to be a challenge therefore, the common secret key technique is also not fully secure (Davis, 1978; Nie and Zhang, 2009 and Zhao et al 2012). In both the approaches, inspite of their security limitations in MANET, one major challenge is that, the keys have to be pre-deployed in nodes or the nodes already have some form of trust built in amongst them.

It was assumed in this thesis that the network is initialised with no predetermined trust; therefore, all nodes are treated as equally trusted or having no trust at all. The trust value is normally between 0 and 1 (Annarasi and Sevanesh, 2014; Johnson et al 2011; Govindan and Mohapatra, 2010; Eladi et al 2013). When a node joins the network for the first time, it is issued with its reputation that is initialised with value 0 i.e. no trust. The proposed trust based scheme is used to build the initial trust required to identify any malicious activities and nodes with high trust threshold. The trust information is also used to authenticate communicating peer nodes. The scheme proposes a technique that further utilizes the trust information gathered during network formation for authenticating any node in the network. The authentication scheme works by requesting trust information from the neighbours of the target node. In the first instance the neighbour information is requested and then the trust values are obtained from those nodes. Once the trust values are received by the requesting node, it authenticates the node by checking its trust values. This validates and authenticates the target node.

In this section, various initialisation models and how authentication can be achieved using those model has been discussed. We offer a general view of these models and justification of the parameters used and how we have implemented it in the proposed scheme to draw a comparison and show its effectiveness.

3.1 Authentication

It is already established that pre-authentication is vital pre-requisite for any network using authentication and key exchange protocols (Johnson, Hu and Maltz, 2008; Hu, Perrig and Johnson, 2005; Yi, Naldurg and Kravets, 2005). This implies that if the authenticity of the communicating nodes

cannot be ensured then the encryption schemes wouldn't provide full security solutions. Authentication still remains one of the areas that need attention in MANET, especially when new nodes join the network and nodes that are not pre-authenticated. Conventionally, authentication can be provided at two stages i.e. at network initialisation stage or when the network is running. In the first stage, the credentials are exchanges through trusted system and authentication is achieved. In the second stage, the security protocols are distributed when the network is initialised. The second type of authentication is well-suited for new nodes joining the network and in the scenario when no pre-authentication is established at network initialisation stage (Chang and Agarwal, 2008). This leads us to the major and important question of how the nodes are able to prove their identity. The difficulty is in providing a protected channel for secure credential exchange without sharing any credentials (Zhao et al 2012).

In MANET, nodes can react in a variety of ways to different network scenarios. Before discussing different authentication scenario, let us take into consideration some factors that can vary in MANET depending upon the application. The following section discusses some of the scenarios and assumptions made in the authentication model.

3.2 Authentication models

Node authentication in MANET can be made before or after the network initialisation. The proposed authentication model provides a mean of validating peer nodes. The proposed scheme therefore, authentication means validating peer nodes through trust calculated by its neighbour. The trust values requested from peer's neighbour nodes are used as source for authentication. Different forms of authentication models at various stages in MANET life are discussed in the following section.

3.2.1 Pre-network initialisation authentication

- **Pre-Existing Trust**

According to this trust model, MANET network are able to form pre-existing trust based on prior trust among nodes and can be referred to as pre-initialisation authentication. The trust relationship has to exist before the network initialises. Pre-authentication models are suitable for certain kind of networks but not for others. For instance, the pre-existing trust based network is applicable where no new nodes are joining the network. But there is also an issue of mobility as nodes can leave and join the network at any time (Zhong et al 2003; Buchegger and Le-Boudec, 2002; Jhaveri, 2013). Although, in such types of models the trusted node could extend their trust and authenticate new joining nodes but there is an issue of key exchange, performance overhead and new node identity (Hinds et al, 2012).

- **Key pre-distribution protocols**

Pre-initialisation authentication uses key pre-distribution protocols to deal with key distribution. This is achieved by distributing secret keys among nodes, prior to the network initialisation, via a trusted third party node, key distribution centre (KDC) or through bootstrapping (Chang and Agarwal, 2008). Once the network initialises, then the nodes can authenticate each other and securely communicate using secret keys distributed by trusted third parties. These trusted third parties are external nodes and are available at the time of network initialisation. The trusted third parties can be used for individual node keys exchange or it could be available to initialise the network as a whole. But once the keys are exchanged and network initialised then the secret keys have to be refreshed from time to time and there needs to be a mechanism to distribute keys to new joining nodes as well (Carter and Yasinsac, 2002).

The keys have to be distributed through a secure channel and the nodes identity has to be confirmed and verified. The verification of new nodes joining the network is another kind of challenge faced when the network is running (Hu et al 2015), i.e., post-initialisation stage in this thesis.

- **Internet Gateways**

Given a secure communication channel, network operating in infrastructure mode could provide authentication to nodes before the MANET initialises. A trusted third party node connected over the internet gateways could periodically provide authentication as proposed by Merin et al (2007). According to Gupta et al (2014), who proposed similar authentication scheme using internet gateway to facilitate pre-network initialisation authentication. The concept of pre-authentication can be applied to few applications but it's not viable for all MANET applications. For this reason the authentication scheme for MANET has to adapt to basic and unique characteristic of MANET discussed in previous section.

3.2.2 Post-network initialisation authentication

Internal Trusted Third Party

In this section, various options and ways of post initialisation authentication have been explored. In MANET the nodes have no prior trust and are faced with the challenges of authentication, before network initialises as highlighted in the above section. MANET nodes can be authenticated using internal, trusted third party nodes that are distributed in the network. These schemes are the most

challenging of all the authentication schemes that will be discussed. Although, they can address various challenges faced by pre-authentication schemes but are hard to implement, as they are meant to be completely self-organized. This design represents pure MANET (Nikander, Ed. P., Kempf, J. and Nordmark, E. 2004) as there is no centralized mechanism and infrastructure required. Nodes are chosen on the basis of certain characteristics and parameters that can be responsible for distributing and revoking keys. The keys can be symmetric or asymmetric (Hinds et al, 2012).

Symmetric keys or private keys are generated by trusted nodes and distributed to all nodes in the network. Securing MANET using these types of schemes comes with different types of challenges as well. For instance having no centralized network the key distribution, node identity, wireless channel link breakages, mobility and availability of mobile are the main challenges faced. Ariadne proposed by Hu, Y. C., Perrig, A. and Johnson, D. B. (2005) uses message authentication code (MAC) and secret key shared between two parties to ensures point-to-point authentication of a routing message.

Public key exchange protocols have so far turned out to be the most suitable for MANET. These protocols could be applied in variety of ways and combinations, to be able to address the authentication issue. Hu et al (2002) uses one way hash chain for authentication and secure routing protocols. The Secure Adhoc On-demand Distance Vector SAODV uses digital signature to authenticate non-mutable fields and hash chain to authenticate the mutable field i.e. hop count for both route request RREQ and route reply RREP message (Zapata and Asokan, 2002). ARAN (Aran no date) is another type of MANET security protocol that uses digital signatures which is again a Public Key Exchange protocol. Security Aware Aided Adhoc Routing SPAAR (Carter and Yasinsac, 2002), is a location aware

protocol which uses geographical information to secure routing information and uses asymmetric cryptography i.e. the use of public key infrastructure for routing.

3.2.3 Hybrid Authentication

This is a hybrid scheme where external Trusted Third Parties are used to authenticate nodes in pre-initialised state and at the same time the network is equipped with internal Third Party Trusted nodes distributed in the network to take care of the post network initialisation authentication (Liu, Zhang and Zhao, 2013).

3.3 Trust and Key Management

Trust and key management makes are vital sections of the proposed security scheme. Therefore, how the trust scheme has been implemented to authenticate nodes and ECDH algorithm is used to form a shared secret key will be discussed.

All the nodes in the network undergo a process of trust establishment. This process is very challenging in MANET and hard to achieve, as the topology is changing and nodes are mobile. When the trust is fully established in all nodes and any misbehaving and malicious node is identified and isolated, then the trusted nodes are used for key exchange.

Trust based schemes are normally formed using pre-determined threshold values. These values are static which are based on some pre-defined protocol static behaviour and remain same for all the nodes

throughout the life of a network. Before the network is formed the threshold value is set, therefore, the decision to accurately calculate trust, is most challenging. The reason being the dynamic nature of MANET and nodes are not in a single state. Parameters determining the trust state may change due to constantly changing topology. Therefore, making a selection based on pre-determined and static value can result in completely undesirable selection of trusted nodes. This further implies that the static threshold may not always work and could lead to an undesirable outcome. The nodes are mobile and the topology is constantly changing, in those circumstances the static threshold information cannot be fully relied upon and could potentially generate false positives. Therefore, a dynamically calculated threshold value is proposed, which will take the parameters into account such as node neighbours, 2-hop neighbours, mobility and energy to work out the trust. The analytical model section explains in detail how the parameter are calculated and implemented in the proposed trust and threshold calculation.

The trust schemes discussed in the above section relies on static trust value and does not take into account MANET specific conditions into account. For instance, the Watchdog technique identifies misbehaving nodes while Pathrater technique would calculate path avoiding misbehaving nodes (Marti et al 2000) using static trust values. Another example of a scheme using static trust is Adaptive Acknowledge scheme (AACK), (Botkar and Chaudry, 2011) is an attempt to reduce detection overhead while increasing detection efficiency through detecting misbehaving node rather than link proposed in TWOACK (Balakrishnan, Deng and Varshney, 2005). Adaptive scheme is used by Muhammad et al (2015), to determine trust of a node but it doesn't offer any authentication, confidentiality or encryption. They have not considered node energy in their adaptive scheme, which is an important factor in determining node malicious behaviour. The energy factor could result in a node dropping

packet to conserve energy and thus, generate false positive (Gupta, and Sexena, 2010). Also their individual trust calculation is the summation of trust from all nodes but does not clearly state the detail of how the trust is calculated. Lastly, their trust scheme mainly aims at isolating malicious nodes which can partition the network and reducing false positives.

The proposed scheme uses a trust table where the trust values between all nodes are exchanged first via hello packets and added in the table. Hence the table is populated with all the trust values of neighbour nodes. The trust values are used to derive the final trust value called Average Trust Threshold. The trust table holds all the average trust values and Average Trust threshold values which is an important aspect of the proposed protocol. It is important that the Trust table is up to date and it is always updated when the hello packets are exchanged (Bhanot and Chaudhary 2017). Another important factor of the Trust table is to authenticate two communicating peers by sending the Average Trust values when requested.

3.4 Node Degree and Two-Hop Neighbours

Node degree is defined as the number of nodes in node's one-hop neighbourhood. Two nodes are considered as Neighbours when they are within communication range and at a specific time interval. Although there might have been no communication or message exchange between the nodes during the time interval the nodes have been in transmission range but they are still regarded as Neighbours (Khan et al 2015). The process of maintaining neighbour list is via Hello message exchange. Hello message exchange and neighbour list is discussed in detail in implementation chapter.

3.5 False Positives and False Negatives

The trust model mainly depends on the information it receives from neighbour and two-hop neighbour nodes for its trust calculation and as a result identifying trusted and malicious nodes in network. One of the reasons for the selection of Dynamic Threshold calculation for trust calculation is to reduce false positives and false negatives. This is to ensure that the scheme is tuned to give effective results and offer tangible security.

3.5.1 False Positives: When a node classes another node as untrusted where it is a trusted node, it has a significant negative effect on network performance and the trust model (Gupta, and Sexena, 2010). False positives are cumbersome and they burn operation cycle. The node could potentially be regarded as malicious and therefore excluded from routing process.

3.5.2 False Negatives: This is when a node is classed as trusted node but it's a malicious node. False positives are even worst as there is no indication that the node is compromised. It is equally important for the trust model to identify such nodes and accurately calculate trust. If the scheme fails to recognize malicious node it could compromise the whole network (Gupta, and Sexena, 2010).

3.5.3 True Positives: This is when the scheme is doing its job properly and accurately identifying nodes that are trusted. All node are working as they should that is to calculate trust values correctly and passing them to on other neighbour nodes.

3.5.4 True Negative: This is again trust values shared by nodes regarding other nodes trust that have been correctly calculated. True negative is when a node is correctly identified as malicious. This again a positive thing and indicates that the efficacy of the proposed scheme is working.

3.6 Key Management

The Key management issue in MANET could be handled in various ways and there exists different classifications schemes that have been proposed in the past (Hinds et al, 2012). The fact that MANET have no centralized mechanism and infrastructure and nodes are mobile makes the use of cryptographic key management one of the daunting task for securing network using such schemes. These schemes come with issues like performance, resource overhead, wireless links issues and node's physical security. According to Zhao et al (2012), key management deals with Generating Keys, Exchanging or keys distribution, Verifying keys, Storing Keys and Revoking keys at the end of their lifetime (Liu, Zhang and Zhao, 2013).

The keys exchange process is crucial part of the Public Key Infrastructure (PKI). Scheme using PKI for authenticating nodes in MANET, there are various issues that needs addressing to generate, distribute, refresh and revoke expired keys (Hinds et al, 2012).

There are certain factors that need to be taken into account for the keys to be exchanged securely especially when they are applied in MANET. Therefore, various reasons are discussed as why these parameter are important and should be taken into account and hence the reason for the selection.

3.7 Availability

The network is a critical resource as it allows the communication between various devices. In Manet environment the nodes acting as router are acting as communication device at the same time therefore, it is the life blood as it carries the vital traffic from source to the destination node (Liang, Poor and Ying, 2011). The presence of malicious node or an attack on a node could mean a potential failure of the network but at the same time excluding a legitimate node due to false positives could be devastating for the network as well. Any failure could be a Denial-of-Service against the particular service the Manet is used for. If the node is not available to authorized users, the impact could be significant as they rely on one another to form a network and communicate (Mohandas, Silas and Sam, 2013).

In wireless networks availability as whole can be a challenge as wireless networks are highly susceptible to Denial-of-Service attack due to no physical connection needed to launch an attack unlike in wired networks, where a physical connection is required (Mohandas, Silas and Sam, 2013). All nodes before initiating any data communication would need to request the trust values from all their corresponding neighbours in order to authenticate each other. The trust information requested from neighbour nodes needs to be exchanged in specific interval of time, otherwise it may result in causing delay or in worst case scenario the information may fail to reach the target nodes. These failures will result in authentication failure and hence communication will never initiate as the trust cannot be calculated. Therefore, the nodes needs to be available and ready to send the information requested. The simulation have shown that it can be achieved and the overhead caused are presented in the results section.

3.8 Energy

Energy of the node is concerned with power resource. Energy model maintain total energy at each node in wireless network. Energy model in NS-2 is used to set initial energy and monitor it for each node during simulation. As nodes in MANET are mobile and there is no continuous power supply therefore, the energy resource is scarce. This is an important factor and needs consideration in MANET application. The nodes having low power could drop all or some packets to save energy (Gupta, and Sexena, 2010). The power saving act of nodes could easily be mistaken with Grey Hole attack (Jhaveri 2012), where malicious nodes initially act as normal node but starts dropping all or some of the packets it receives. This could result in false positives as the node is not malicious but dropping packets due energy conservation (Gupta, and Sexena, 2010). Hence, if the node's energy level is low and it starts to drop some of the packets, a possibility cannot be ruled out that the node is acting in such a way, only to save energy, rather than jump to a conclusion that it is a malicious node and is excluded from the network.

3.9 Mobility

The basic neighbour discovery requires node to be stable in the regions. Nodes that are constantly changing regions won't have any neighbours. According to Rajesh and Gnanasekar, (2016), the constant motion of nodes across boundaries limits the usefulness of basic neighbour discovery. According to Khan et al (2015), the mobility has significant and direct effect on how a malicious node is detected in the network. The higher the node speed, the lesser is its detection rate, as compared to the node at lower speed. This is due to more frequent changes in neighbourhood composition Khan et al

(2015). Therefore, mobility is considered as an important parameter in determining Trust Threshold of on node in a MANET. A mobile node Dynamic Threshold being lower due to its high mobility has to be taken into account to decide whether it is trusted or not trusted. On the other side, in static trust all nodes are compared against a static value which essentially does not take any mobility into account and considers all nodes the same.

From the above, a conclusion can be drawn that node with relatively low mobility carry more weight than the highly mobile node. Since the outcome for highly mobile node according to the above analysis, is a lower Dynamic Threshold therefore, it will also be excluded from being selected as trust worthy. The Dynamic Threshold value will prevent this as being a lower value obtained due to mobility when compared with static trust value.

3.10 Confidentiality

In key exchange and data exchange one of the most important conditions is confidentiality (Mohandas, Silas and Sam, 2013). This implies that the key information is not leaked and kept secret at all times and only the intended recipient can read it. The data is therefore, encrypted during the transit to ensure the data is not disclosed. But authentication is equally important when it comes to ensure confidentiality (Meddeb et al 2017). Authenticity means the message that claims to be from a given source is in fact from that source. Hence, having confidence in the source of a message is critical. If an unauthorized user obtains the keys for another authorize user could easily authenticates its self. The process of authorizing nodes in a static network could be performed via a centralized authority but in dynamic network such as MANET, this approach cannot be adopted due to MANET environment,

therefore, a different approach is needed (Mohandas, Silas, and Sam, 2013). A novel approach is adopted in the proposed scheme, which is mutual trust authentication scheme for authenticating peer nodes and confidentiality is achieved by using ECDH key exchange protocol.

3.11 Conclusion

The proposed scheme uses the concept of trust to secure the network from malicious nodes. The association between trust and authentication is explained in this chapter and how trust is important in MANET where, no other form of security can be implemented as nodes communicate for the first time. Various forms of trust implementation before and after network initialisation and their benefits will also be discussed. Lastly, the justifications for using the parameters to calculate dynamic trust are presented.

CHAPTER 4

Analytical Model

Routing protocol discussed in previous sections doesn't take and into account any security aspects. In this section, the analytical model of the proposed scheme, details of the protocol design stages and any further specifications involved in designing the algorithm are presented.

The aim of the protocol is to provide a platform and serve as a building block for various other advance network security protocols in MANET. The thesis aim is to keep the specification general as possible to demonstrate that out proposed solution can be adopted and run on majority of underline routing protocol or other classes of MANET application. As no centralized mechanism and highly dynamic environment this protocol can serve as an initiating point to generate and distribute security keys without relying on conventional network setup. According to RFC 3756 (Nikander, Ed. P., Kempf, J. and Nordmark, E. 2004) which discusses Trust Models for various networks, including Adhoc network, it is assumed that a truly Adhoc network is where all the nodes meet and form a network for the first time and there is no prior trust relationship among nodes. The proposed solution is designed by taking into account this particular feature of Adhoc network that there is no prior security relation between the nodes.

A proposed two phased solution is outlined below. While the two-phased approach has a higher start-up cost, there are several reasons that it is beneficial, which will be described later in the sections below.

- Phase One: Dynamic Trust Threshold scheme and Authentication

- Phase Two : How to provide

    1. Confidentiality

    2. Data Encryption

4.3 Phase-One

In the first phase the aim is to work out the trust of each node to identify the trusted and malicious nodes. Any malicious nodes identified are excluded from the network so they cannot take part in any routing process or data communication. Each node performs a local calculation of trust value and dynamic trust threshold value of all its neighbours. The final stage of phase-one is to authenticate two communicating node before any data exchange by requesting trust value from communicating partner's node Neighbours. This step adds an additional layer of security to validate and verify trust. The common way to validate the routing protocol or security protocol in network is through various simulation tools (Sing and Verma, 2015: Elhadi et al 2013 and Al-Roubaiey, 2010). In the following section, an analytical model of the proposed algorithm is presented as an additional way to evaluate and validate the results. The proposed phase-one of Trust threshold scheme can be calculated in the following steps.

Step-1: The trust value is calculated for each node in the network by nodes listening to neighbours in promiscuous mode and recording the number of packets sent and received.

Step-2: Various metrics are used to calculate the dynamic threshold value (**k**) for each node in the network. Out of the five parameters, the trust value is calculated using local values as described in step one by each node, while the other parameters such as one-hop, two-hop, speed and energy are obtained from the neighbour node for which the trust is calculated. These values are passed to neighbour nodes using hello-packets (Bhanot and Chaudhary 2017).

.

Step-3: The trust value calculated using step-one is compared with the static arbitrary trust value of 0.6 (Khan et al 2015). If the trust value is less than the static trust value, the trust value is compared against the dynamic trust threshold value calculated using step-2, in order to dynamically work out the trust of each node. A static arbitrary value is a fixed value and can be changed depending on MANET condition and applications requirements. The static trust value remains fixed for the whole network life and explained in section 4.4 in detail. As a result, nodes having less than static trust but greater than or equal to dynamic trust threshold value will be regarded as trusted or otherwise they will be classed as untrusted as show in figure 4.1 and algorithm 4.2.

Step-4: The trust values are requested by the communicating nodes to authenticate each other. This step involves peer nodes requesting trust values from their corresponding neighbour nodes. As, all nodes recording the trust of its neighbour nodes in the network therefore, these values can be requested by

peer nodes to authenticate its peer. This is another layer to add additional security to an already trusted frame work.

In majority of the trust based schemes proposed (Elhadi, et al, 2013; Marti et al 2000; Nasser and Yunfeng, 2007; Al-Roubaiey et al 2010; Botkar and Chaudry, 2011), the average trust value of nodes obtained, is compared against a predetermined trust threshold value. This predetermined threshold value is a static value and remain static throughout the life of the network. The threshold value is static because it does not take any topology or MANET changes into account. One of the characteristic of a MANET is that, it is highly dynamic and mobile and therefore, these factors have to be taken into account (Cho, Swami and Chen, 2011). By comparing the static trust value against the dynamic trust threshold, two important goals are achieved, firstly, the trusted nodes are identified and secondly, any malicious nodes identified as a result are excluded dynamically. As discussed before the trusted framework will form the first layer called phase-one of the security model and phase-two will be built on top. The data flow diagram in figure 4.1 below represents various step in calculating parameters and threshold values are derived and compared to identify nodes trust dynamically. The trust for all nodes is calculated after the network is initialised.

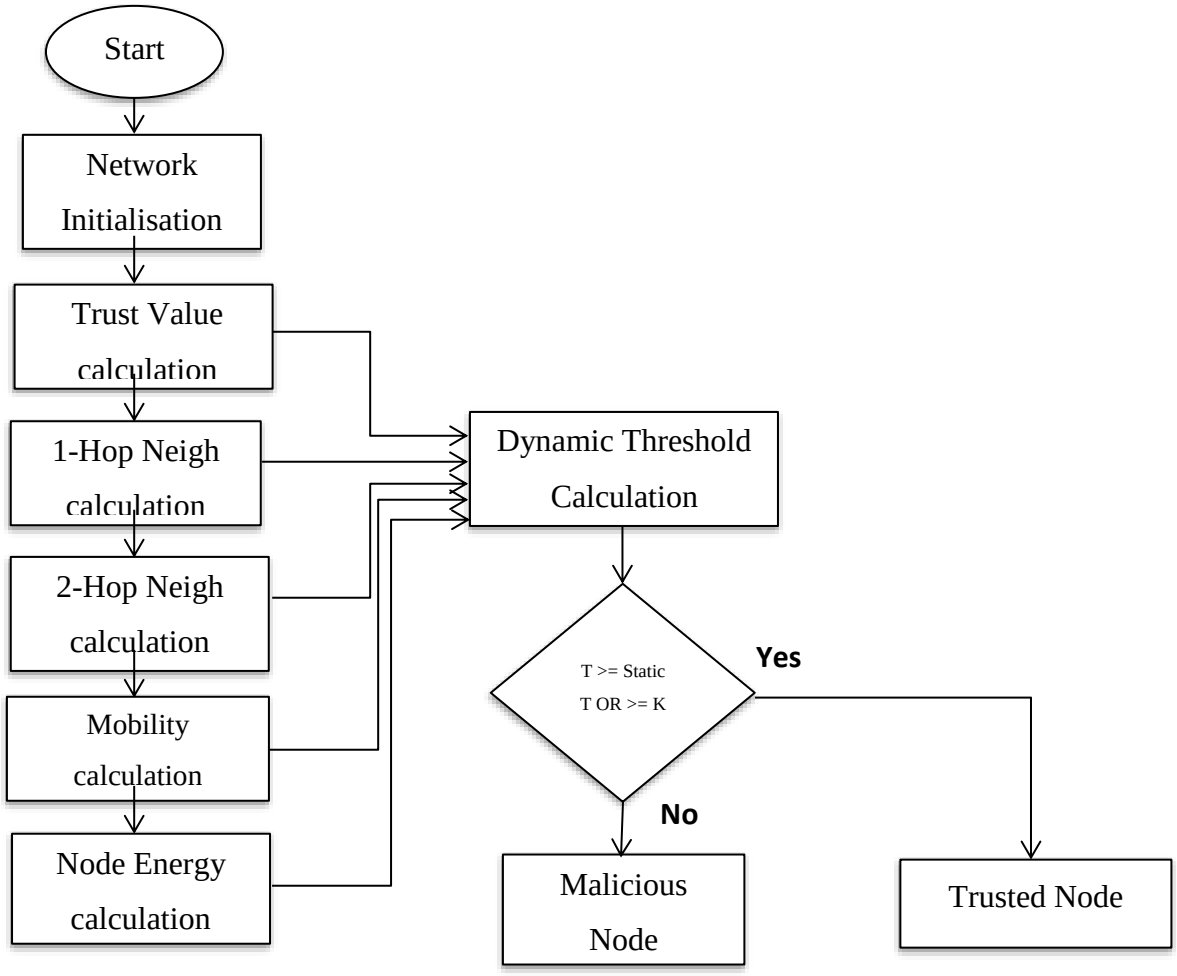


Figure 4.1 DFD showing Phase-one Dynamic Trust Threshold Scheme

In the phase-two, ECDH algorithm is implemented to exchange keys between nodes to secure communication by encrypting any data sent between nodes. The ECDH will be discussed in more details in next the section 4.5 and the analytical model will be presented to show how the proposed scheme works and a complete dissection of the all the layers involved.

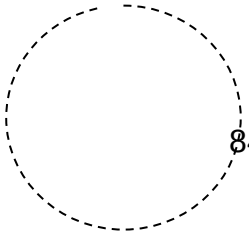
4.3.1 Threshold Calculation **k**

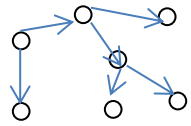
The aim of phase-one is to dynamically calculate and identify the trusted and malicious nodes in the network. The Dynamic Trust value is compared with the trust value calculated by individual neighbour nodes from their personal observation. This Trust value is then compared against a static pre-determined trust value of 0.6 (Khan et al 2015) and Dynamic Trust value to obtain the final trust of all the nodes in the network as described in figure 4.1 in chapter 4. This whole process will provide a foundation for first layer of security in the form trusted nodes in the network and identifying and excluding any malicious nodes. As mentioned earlier, the proposed scheme provides a common platform, which can be adapted to most of the MANET standard routing protocols. A threshold represented as '**k**' is calculated for all the metrics used to calculate dynamic trust. The following section explains in detail, how the threshold of each metric and the dynamic trust is calculated.

4.3.2 One-Hop Neighbour Threshold

We consider a MANET, comprising of a number of mobile nodes. It is a multi-hop wireless network modelled as undirected graph ***G*** represented as ***G (V, E)***, whereas ***V*** is the set of nodes and ***E*** is the set of wireless links $E = V \times V$ (Sukiswo and Rifquddin, 2015) as shown in figure 4.2.

$$E = V \times V \qquad \text{…….. Equation 4.1}$$





Represents Graph *G*

Represents set of wireless nodes *V*

Represents vertices *E*

Figure 4.2 Wireless Network Graph Model

Where:

E: set of wireless link also known as edges

V: set of nodes

Uniform transmission range *r0* is assumed among all nodes *V*. A wireless link between two nodes '*a*' and '*c*' is represented as (*a, c*) and link existence between the two nodes is represented as (*a, c*) ∨ *E*. The above representation is defined as the wireless link or 1-hop Neighbours exists (*a, c*) ∨ *E* if and only if the Euclidean distance between nodes *'a'* and *'b'* is smaller than the transmission range *r0*. Figure 4.2 shows one-hop neighbours screen-shot from NS2 simulation representing 20 nodes.

Figure 4.3 One-Hop Neighbours of Node

Two nodes have a common link and classed as neighbours, if they are within each other's transmission range *r0*. Node degree is thus the number of nodes in any given node's one-hop neighbourhood (Rajesh and Gnanasekar, 2016). The number of One-hop Neighbourhood is directly proportional to overall threshold of any node for the proposed scheme thus, higher the number of neighbour nodes the higher the threshold value.

The node degree of node ***a*** at time *t* can be represented as $\partial a(t)$, with transmission range *r0* for isolated node is 0, which is the minimum value. The node with maximum number of nodes in the

neighbourhood has a higher trust. The range threshold (k) value, ranges from *Min = 0* to *Max* = 1. The threshold for one-hop neighbour is calculated as

$$\boldsymbol{K}\delta a = \frac{\delta a}{|\,V\,|} \qquad \text{.....Equation 4.2}$$

Hence if,

δa = 5 and $|V|$ = 20, then $\mathbf{k}$ = 5/20 = 0.25

Where:

k: is threshold

δa: Neighbours of node *a*

$|V|$: Total number Neighbours of node *a*

4.3.3 Two-Hop Neighbours Threshold

This is defined as a sub-graph of *G* denoted as *Ga* which consists of one-hop and two-hop neighbours of *a*. According to Khan et al (2015), if there is a wireless link between nodes *c* and *e*, then ***e*** is a two-hop neighbour of *a*, and is represented as (*a*, *c*) ν *E*. Figure 4.4 below shows two-hop neighbours of a node. Two-hop neighbours can be represented by the following equation:

$$2\text{-Hop}(a) = \{ c \in V, e \in V : (c, e) \in E \wedge (e, a) \in E \}$$ ……..Equation 4.3

Where:

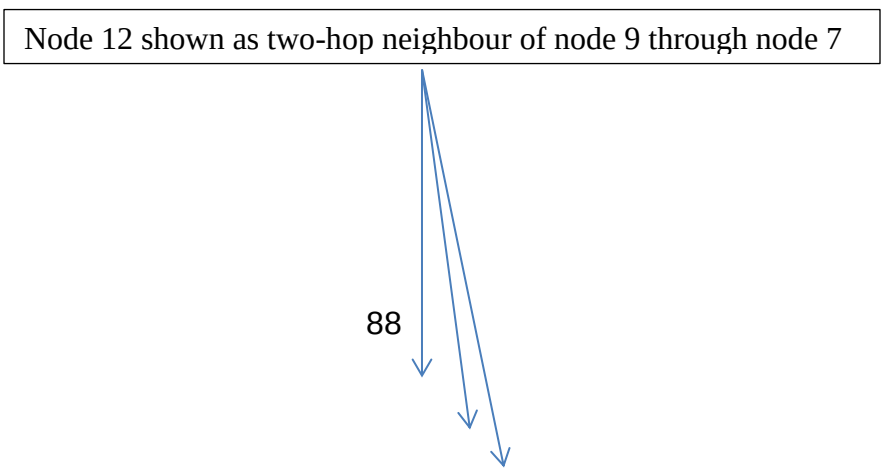
2-Hop(*a*): Two-hop neighbours of *a*

c: Neighbour node of *a*

e: 2-hop neighbour of *a*

E: Represents edges

V: Nodes



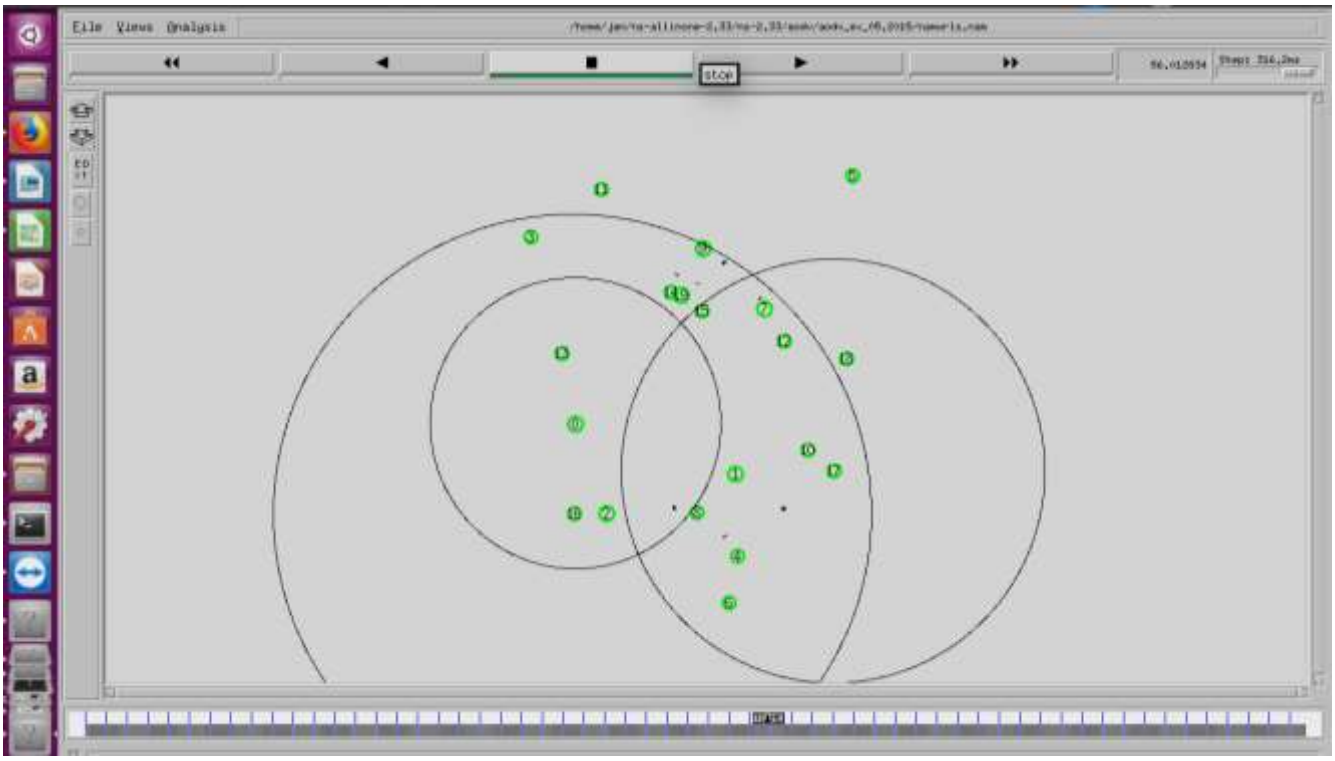


Figure 4.4 Two-Hop Neighbours

The threshold k of nodes is a value between (0, 1), 0 means minimum and 1 means maximum value. The node with highest 1 and 2-hop neighbours has the highest k threshold and a node with no neighbour has lowest threshold (k).

The two-hop of any node can be obtained from the routing table and is defined as b*(a, u)* for node "*a*", as any indirect two-hop nodes "*w*" that could be reached through direct neighbour node "*u*". Having a set *A* of vertices, the neighbourhood of *A* is the union of the neighbourhood of the vertices, and so it is the set of all vertices adjacent to at least one member of *A* (Gupta et al 2014).

$$\mathcal{K}\beta_a = \frac{\beta(a,u)}{|\,2\text{-hop}(a)\,|} \qquad \text{.....Equation 4.4}$$

Hence if,

$b(a, u)$ = 5 and |2-hop(a)| = 20, then $\mathbf{k}$ = 5/20 = 0.25

Where:

k: is threshold

ba : Represents two-hop neighbours of *a*

$b(a, u)$: Two-hop neighbours of node a via node *b*

|2-hop(a)|: Total number of two-hop neighbours of node *a* via all neighbours.

According to the above equation, the threshold for two-hop node can be defined as, the number of 2-hop nodes accessible via node (*u*) divided, the total number of two-hop nodes of node "*a*", through all neighbours. The threshold has maximum value (1) if the $b(a, u)$ is (1) i.e. they are directly proportional. And the threshold will be 0, if $b(a, u)$ is 0, which indicates the number of 2-hop connectivity for this particular node.

4.3.4 Mobility Threshold

Node mobility is defined as speed at which a particular node is traversing the network. According to Khan et al (2015), the node mobility can be calculated as the rate of link change. The mobility of a node is used to determine its mobility threshold. A node is considered to be stable and reliable if it has average mobility relative to other nodes (Anuj and Sminesh, 2014).

Using the above notation, the mobility threshold of a node is calculated. The equation is as follows

$$\kappa_{\mu a} = 1 - \frac{\mu a}{|\mu|}$$ ……Equation 4.5

Hence if,

μa = 5 m/sec and $|\mu|$ = 20 m/sec, then κ = 1-5/20 = 0.75 m/sec

Where:

κ : Threshold

μa : Speed of node *a* m/s

$|\mu|$: Maximum speed of *a* m/sec

According to the equation 4.5, if the mobility is high, than the mobility threshold is 0 and if the mobility is low then the threshold is 1 i.e. high. The detail explanation is presented in chapter 3.

4.3.5 Trust threshold (t)

The trust value is calculated through recording activities of neighbour nodes, as all the nodes operate in promiscuous mode and can hear the transmissions such as packets sent, received, dropped and acknowledgements etc.

Trust calculation is an important factor and hold highest weightage in terms of the metrics. A novel method of calculating trust is proposed that not only select trusted nodes but also identify malicious and

misbehaving nodes in the network. MANET is highly dynamic therefore, the trust is calculated along with the metrics outlined and explained above. This ensures that the network conditions and dynamic nature is taken into account. In this way, the threshold gets adapted to the changing environment and the network has the capability to identify any malicious and misbehaving nodes dynamically. A trust data base is maintained by all nodes and is exchanged periodically through hello packets, therefore, an extension is used and the hello packet is modified to transport the trust values for its Neighbours (Bhanot and Chaudhary 2017). Every node maintains a trust table that records the trust values received from every Neighbours. Nodes listening to Neighbours, records their number of packets received and forwarded in a trust table by using the equation 4.6. There are two types of trust value calculated namely, Average Trust value and Trust Threshold.

4.3.6 Average Trust

This value is calculated by each node for its neighbour nodes. Each node listening in promiscuous mode calculates the average trust value using equation 4.6 and forwards it via hello packet to all neighbours (Bhanot and Chaudhary 2017). Once a node has received average trust value calculated for particular neighbour node from all its corresponding neighbour nodes, then the trust threshold can be derived as.

$$\text{Ave Trust} = \frac{\text{Total Num. of Packets sent/recei ved}}{\text{Total Num. of Packets}} \qquad \text{……Equation 4.6}$$

4.3.7 Trust Threshold

Every node in the network needs to calculate this value in order to compare it with dynamic trust threshold value, to determine the trust level of every node. Once all the average trust values are received from neighbour for a particular node then the trust threshold can be derived using equation 4.7 below:

$$\text{Trust Threshold} = \frac{\text{Sum Ave Trust}}{\text{Total Num. of Neighbour Nodes}} \qquad \text{…….Equation 4.7}$$

The trust threshold value represents the trust of each node. This value also gives us an input for equation 4.10, where the trust threshold is represented as '$\mathcal{K}\tau$' to calculate the dynamic trust threshold. Following is the mathematical representation for calculating trust threshold.

$$\mathcal{K}\tau a = \frac{1}{N}\sum_{i=0}^{N} \tau_i \qquad \text{…….Equation 4.8}$$

Where:

t: Trust threshold value calculated by each neighbour node

N: Total number of neighbours

k: The Final average trust threshold of node *a*.

i: Node index

a: Node

The proposed scheme presents a new concept of each node maintaining trust table where the trust values are exchanged periodically via hello packets (Bhanot and Chaudhary 2017). Once the trust table is fully populated and the trust threshold values are generated for each neighbour then the next step is to calculate the final dynamic trust threshold value.

4.3.8 Energy Threshold

A critical constraint in MANET is that, all nodes employ batteries, so it is difficult to change or recharge batteries on the go. Therefore, all systems, processes and communication protocols or schemes designed for MANET must take into consideration how to minimize power consumption. The aim is to simulate an energy source and keep track of energy consumption of nodes in the network. Energy consumption is an important metric for evaluating the trust threshold (Gupta, and Sexena, 2010). Energy model built in NS2 is implemented to access the energy of node during simulation. Energy model represents level of energy in mobile node.

Nodes in MANET are limited in their energy resources as there is no constant power supply available. Given the energy resource constraint, all the nodes must have sufficient power resource to process information and take part in any data exchange and does not go offline due to no power. The power resource is directly linked with availability, the importance of which was discussed in the section above. Also, nodes in MANET can drop packet either intentionally (malicious) or unintentionally to save energy. Nodes having low power resource can start misbehaving by dropping packets only to conserve energy. This is an important factor in identifying false positive. The energy parameter will take the energy level of nodes into account to find out the reason for node's packet dropping.

The NS2 built-in energy model is used for energy calculation. The initial energy of node is set to 100 and the other energy parameters are presented in detail in the performance evaluation chapter. The initial energy is used as a basis to calculate the overall energy of each node from network initialisation to when the simulation ends.

$$\kappa\varepsilon_a = \frac{\varepsilon\alpha}{|\varepsilon|} \qquad \text{.....Equation 4.9}$$

$\mathbf{e}a$ = 65 and $|\mathbf{e}|$ = 100, then $\mathbf{k}$ = 5/20 = 0.65

Where:

k : Threshold

e: Energy

a: Node

$|e|$: Maximum energy

4.3.9 Dynamic Threshold

This is the final value we need, to complete phase-one of the proposed scheme. The dynamic value is derived from the parameters used in the above sections. Therefore, once all the parameters are calculated including the Trust Threshold value, the next step is to calculate Dynamic Threshold value. The metrics obtained for each node represents its corresponding parameters in the network.

According to Khan et al (2015), the Dynamic threshold of each node in the network can be calculated by combining all the threshold values obtained from equations 4.2, 4.4, 4.5, 4.8 and 4.9 for a particular node that are used to calculate all the thresholds parameters (k_d, k_b, k_m, k_t, k_e).

The Dynamic Threshold equation is derived as follows

$$Ka = \frac{\delta\_w.K\delta a + \beta\_w.K\beta a + \mu\_w.K\mu a + \tau\_w.K\tau a + \varepsilon\_w.K\varepsilon a}{\delta\_w + \beta\_w + \mu\_w + \tau\_w + \varepsilon\_w} \quad \text{.....Equation 4.10}$$

Where:

k_a : Dynamic threshold of node "*a*"

k_{da}: represents the 1-hop neighbour threshold

k_{ba}: is the 2-hop neighbour threshold

k_{ma}: is the mobility threshold

k_{ta} : is the trust threshold

k_{ea}: is the energy threshold

δ_w: Weight representing the weightage of 1-hop neighbours

β_w: Weight representing the weightage of 2-hop neighbours

μ_w: Weight representing the weightage for mobility

τ_w: Weight representing the weightage for trust value

ε_w: Weight representing the weightage for energy

The weight value can be increased or decreased according to need. They are initially set to (1). For instance, the information gathered by neighbours carry more weight than the two-hop neighbour, as neighbour node has first-hand information of node trust. Any node having large number of neighbours

can have greater observation recorded and trust validated. The figure 4.5 shows a screen-shot of when the trusted scheme is run in NS2.

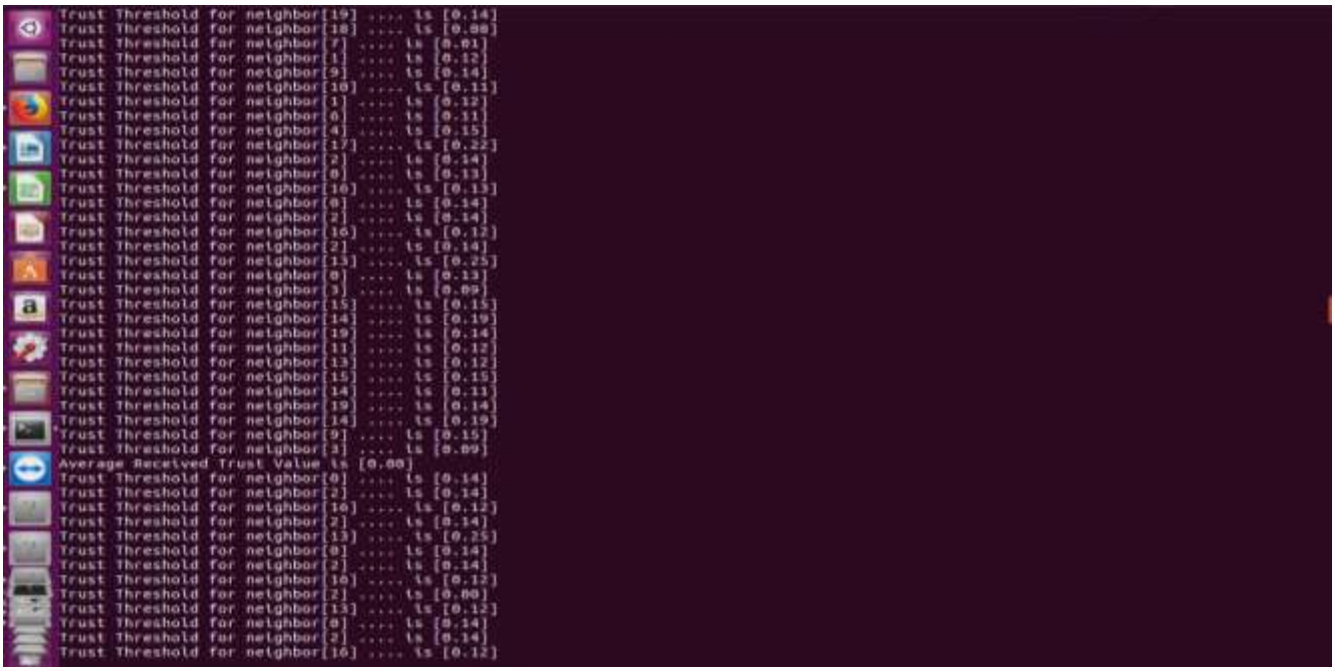


Figure 4.5 NS2 simulation of Trust Threshold calculation

4.4 Numerical Model

This section presents a numerical model by applying sample values shown in table 4.3 and 4.4, to calculate static and dynamic trust respectively. The dynamic trust is calculated by applying value in table 4.4, as an input to the equation 4.10. The results obtained are presented and analysed in this section. The aim is to reduce false positive generated as a result of static pre-determined trust. A numerical model is presented with sample values to demonstrate how static trust model can result in false positive compared to applying the same set of values will result in true positive when dynamic trust model is used along with static model. When a node calculates the trust threshold using equation

4.7 and the dynamic trust threshold value using equation 4.10, the trust of node is computed as shown in algorithm 4.2. While the steps in algorithm 4.1, represent how the node trust is calculated using static trust model and the node is declared as trust or malicious as a result of the computation.

```
Begin
Compute Node Trust
Compute Static Trust
        If Node Trust >= 0.6 then
                Trusted
        Else
                Not Trusted
End
```

Algorithm 4.1 Static Trust Algorithm

The algorithm 4.2 presents the algorithm for calculating dynamic trust. The dynamic trust model takes the static and dynamic trust values into account in calculating node's trust as shown below.

```
Begin
Compute Node Trust
Compute Dynamic Threshold
        If Node Trust >= 0.6 || >=Dynamic Threshold then
                Trusted
        Else
                Not Trusted
End
```

Algorithm 4.2 Dynamic Trust Algorithm

In this thesis an arbitrary value of 0.6 (Khan et al 2015) is used as static trust and can be changed if a stricter trust needs to be applied due to specific network requirements or applications. This is the conventional way of calculating trust referred to as static pre-determined trust model, where the static value remains constant and it never changes. It remains the same for the entire network life (Khan et al 2015).

The algorithm 4.1 presents the steps involved in determining the static trust of a node. If a node has trust value greater than 0.6, then the node is classified as trustworthy. But if the trust value is less than 0.6, then it is classified as untrusted.

But in algorithm 4.2, when the node trust value turns out to be less than 0.6, then it is compared against the dynamic threshold. The dynamic trust would ensure the node trust value is scrutinized dynamically according to node specific conditions in MANET.

The dynamic nature of node in MANET could result in lower trust calculation, as discussed before, and can result in false positives. The trust value could be lower due to MANET specific conditions affecting the trust value but using static trust, this cannot be detected and the node is classed as not trusted.

The numerical model shows the comparison between the static and dynamic trust resulting in true positive, true negative and false positive results. The table 4.4 shows the trust threshold calculated of a particular node trust using equation 4.7 and then compared against static trust. By using static trust, the

result is a false positive for this particular node where the trust value is 0.4, shown in the last column of table 4.3. As the node trust value is 0.4, which is less than 0.6, indicating that the node is untrusted and has resulted in a false positive. But if analysed deeply, the node trust is low because the MANET conditions for this particular node, such as lesser node density, high mobility and low energy has resulted in lower trust. As discussed already, the trust of a node is dependent upon the four metrics and plays a vital role when trust is calculated, which obviously were not taken into consideration when static trust is calculated.

Metrics	True Negative	True Positive	False Positive
Hope-One	0.7	0.3	0.3
Hope-Two	0.8	0.5	0.5
Trust Threshold	**0.3**	**0.8**	**0.4**
Energy	0.7	0.6	0.4
Mobility	0.3	0.3	0.3
Static Trust	0.6	0.6	**0.6**

Table 4.3 Static Trust Matrix

The graph in figure 4.6 presents the static trust model. The graph shows the outcome resulting in a true negative, true positive and a false positive when a static trust is applied to a particular node, given the values in table 4.3 as an input.

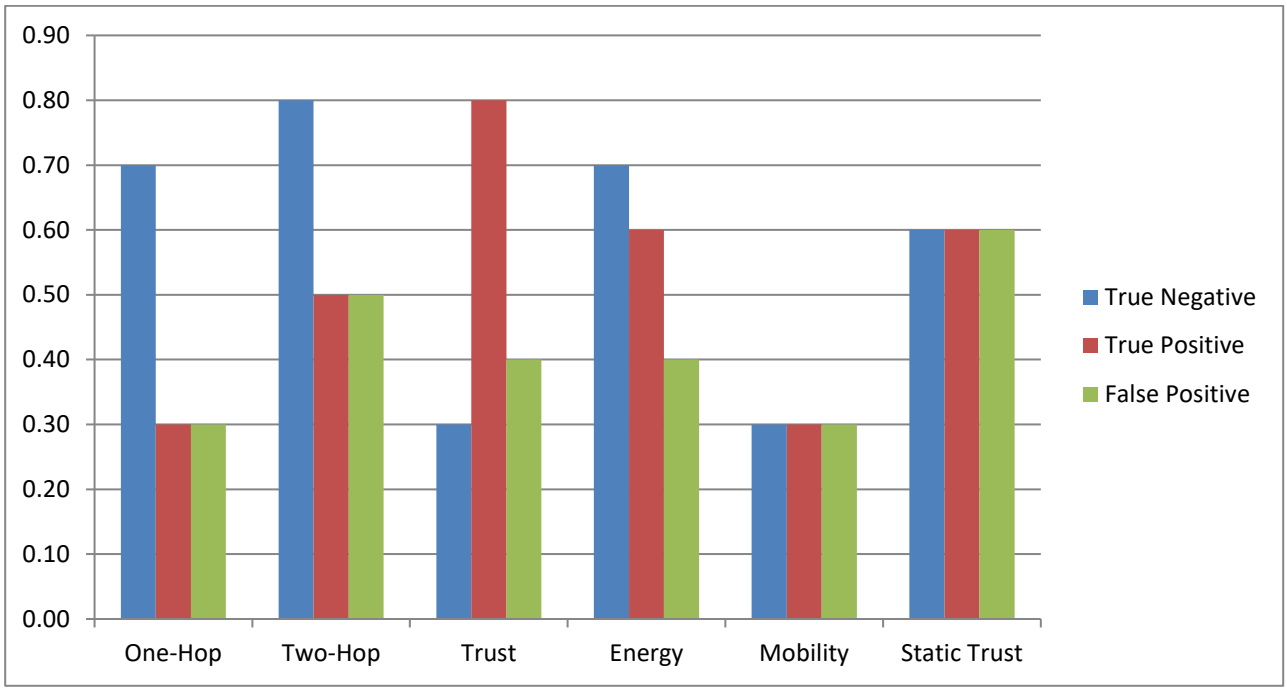


Figure 4.6 Static Trust

But on the other hand, the same node will be classed as trusted, previously declared as untrusted by static trust model using the same data set shown in table 4.4, due to its trust value (0.4). The trust value 0.4 is higher than the dynamic trust value (0.38), as shown in the table 4.4, therefore, using the proposed dynamic trust scheme for these set of variable would result in true positive, which means the node is not malicious but trustworthy.

Metrics	True Negative	True Positive	True Positive
Hope-One	0.7	0.3	0.3
Hope-Two	0.8	0.5	0.5
Trust Threshold	0.3	0.8	0.4
Energy	0.7	0.6	0.4
Mobility	0.3	0.3	0.3
Dynamic Trust	**0.56**	**0.5**	**0.38**
Static Trust	0.6	0.6	**0.6**

Table 4.4 Dynamic Trust

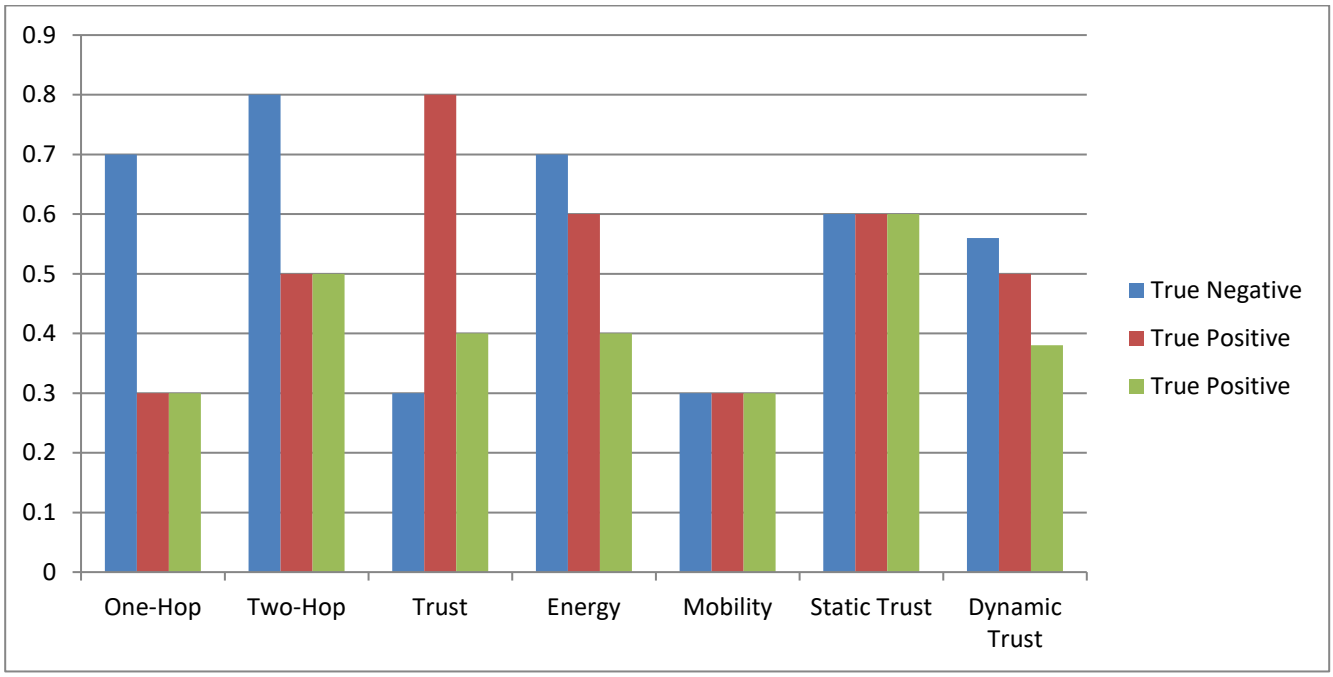


Figure 4.7 Dynamic Trust

In figure 4.6, the graph shows that if only static value is used to calculate trust then it will result in false positive but using static trust combined with dynamic will result in true positive, as shown in last column (true positive) of table 4.4. When the graph is analysed in figure 4.7, all the metric values shown reflect the values given in the table 4.4. This proves that applying dynamic and static trust models together can result in an efficient detection of malicious node.

4.5 Phase-Two

The second security phase which is confidentiality and data encryption, the model of AODV protocol process is presented. It involves implementing the cryptographic protocol before any data exchange between communication nodes**.** When both phase-one and phase-two are combined they provide a foundation to secure key exchange between any communicating nodes. The key exchange process is explained in detail in the next section. The secret key could be used for the following:

1. Authentication and authorization

2. Encrypting data exchange between nodes

3. Provide protection against

    - Blackhole attack
    - Rushing attack

4.5.1 Elliptic Curve Diffie-Hellman Key Exchange

DH scheme allows us to exchange secure information between sender and the receiver over insecure channel. This is an example of asymmetric algorithm. According to this algorithm, two nodes exchange public keys and then each performs a calculation on their individual private key and the public key of the other. The result of this whole process gives us an identical shared key (Nikama and Raut, 2015). The shared key obtained is used for encrypting and decrypting data between two nodes. The scheme provides a framework about how to perform key generation and exchange between parties or devices that do not yet have secure connection to establish shared keying material (key that can be used with symmetrical keying algorithm such as AES, DES, HMAC) therefore, it is more a key-agreement protocol than an encryption algorithm (Elhadi, et al 2013 and Misic, 2008). Elleptic Curve Diffie Hellman is more efficient variant of Diffie-Hellman key exchange protocol which will be used in the proposed scheme. They are used in public key cryptography for conceiving efficient factorization algorithm (Wong, Ramamurthy and Zou, 2006).

According to Gajbhiya, Karmakar and Sharma (2015), Public key protocols are designed on the principle of hardness of solving the following two problems:

1. Factorization of large integers

2. Discrete Logarithm Problem DLP

The main idea behind the above concept is the trapdoor one way function (Gajbhiya, Karmakar and Sharma, 2015).

A one way Trapdoor function, as shown in equation 4.11 is such that it satisfies the following conditions:

- Given x, Y= $f(x)$ is easy to compute
- Given Y, it is computationally infeasible to calculate x = f^{-1}(Y) ….Equation 4.11

Elliptic curves are set of points defined by the solution to the equation 4.12 below:

$$E = \left\{ (x, y) \,\middle|\, y^2 = x^3 + ax + b \right\}$$

$$a, b \in K$$

…….Equation 4.12

Where:

a: is an element of field

b: is an element of field

K: is a field.

Some of the fields K that elliptic curves are defined over are

- *R*: Real numbers
- *Q*: Rational Numbers

- *C*: Complex numbers
- *Z*: Integers modulo *p* represented as *Z/pZ*

Following is the example of a graph of elliptic curve over real numbers *R*.

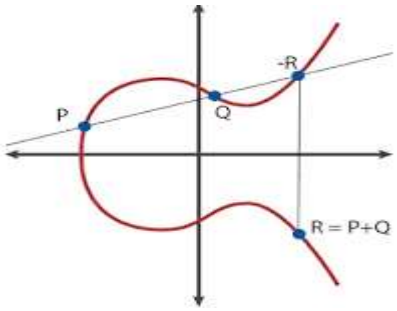


Figure 4.8 Elliptic Curve over Integer Modulo *p* (Gajbhiya, Karmakar and Sharma, 2015)

Also there is a point at infinity represented as *O* and a condition that:

$$4a^3 + 27b^2 \neq 0 \qquad \text{……Equation 4.13}$$

According to Wong, Ramamurthy and Zou (2006), Discrete Logarithm Problem DLP is a type of one-way function as explained above, in which exponentiation is easy but logarithm is difficult to compute. The types of cyclic groups used in public key cryptosystem are

Example of DLP in Zp^*

- Given the finite cyclic group Zp^* of order p-1 and a primitive element $a \in Zp^*$ and another element $b \subset Zp^*$

- The DLP is the difficult computation of determining the integer $1 \leq x \leq p \quad 1$ such that

$$a^x \equiv b \bmod p \quad or \quad x = log_a b \qquad \text{……..……Equation 4.14}$$

Elliptic curves uses shorter encryption keys hence consume fewer memory and CPU resources. It offers more security per bit in increase in size and is more computationally efficient then the first generation RSA and Diffie-Hellman public key systems (Tottanesce 2012). The figure 4.5 below shows the comparison of Diffie-Hellman and RSA key exchange protocols using elliptic curve (Misic 2008).

Symmetric Encryption (Key Size in Bits)	**RSA and Diffie-Hellman (Modulus Size in Bits)**	**ECC Key Size (in Bits)**
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Table 4.5 Comparative Analysis between RSA and Diffie-Hellman and DH using ECC

The above comparison shows, that the elliptic curve keys are much smaller (Misic 2008). The ratio of the key lengths utilizing the protocol from multiplicative group using modulus mod p as shown in table 4.1, to the key length of elliptic curve protocol is increased from 6:1 for 80 bits, 12:1 for 128 bits and 30:1 for 256 bits (Gajbhiya, Karmakar and Sharma, 2015). This implies that the more security is required, the more efficient ECC becomes.

The following section describes various steps needed to configure ECDH protocol.

Let E be an elliptic curve over a finite field k.

Let P, Q be points on E such that $P = nQ$ for some integer n.

Let $|P|$ denote the number of bits needed to describe the point P.

If one wishes to find an algorithm which determines n and has runtime polynomial in $|P| + |Q|$, so, this problem seems hard. This is also referred to as discrete logarithm problem (DLP) where "adding is easy on elliptic curve but undoing is hard", (Tottanesce 2012).

Using a multiplicative group of points on an elliptic curve the ECDH protocol works as follows

1. Node A and node B agree on an elliptic curve E over a Field Fq and a base-point $P \in E/Fq$.

2. A generates a random secret kA and computes $PA = kAP$.

3. B generates a random secret kB and computes $PB = kBP$.

4. *A* and *B* exchange *PA* and *PB.*

5. *A* and *B* compute *PAB = kaPB= kbPA*

The secret *kA* and *kB* is a random value ∈ {1, ..., n−1}, where n is the order of the group generated by *P* (Gajbhiya, Karmakar and Sharma, 2015) and are exchanged over non secure channel without revealing identity of the secret (Wong, Ramamurthy and Zou, 2006).

4.6 Mutual Authentication Scheme

As mentioned above, the AODV is used as reference to compare the proposed scheme by modifying AODV. The standard AODV is compared with trusted AODV that has the proposed scheme embedded, to draw a comparison and validate the findings. There are three types of messages RREQ, RREP, and RERR defined by AODV protocol. To implement the proposed trusted scheme, there has been a modification made to the RREQ message at destination node and RREP at the source node to request corresponding values for authentication before any data exchange.

4.6.1 AODV Authentication process at Destination Node *D*

This section describes how AODV can be used to implement the proposed mutual authentication scheme. When a source node *S* wishes to communicate with destination Node *D*, and doesn't have a route to destination node *D*, it sends a RREQ. In the normal AODV operation the destination node sends a reply to the source node with the valid root when the RREQ reaches the destination node *D* and

the last action performed is a |Send Reply| message sent. After the AODV operation is complete and before any data communication is performed by both nodes, the authentication and authorization stage begins which concludes the first phase of the proposed scheme.

According to this stage, the destination node requests trust values from source *S* and all its neighbour nodes. Once the trust values are received from all the corresponding neighbour nodes of *S* then the trust values are evaluated to calculate final trust value. The node is authenticated if the trust value is equal to and higher than the values received from all neighbours, and authentication fails if the trust value is low. The same process is repeated by the source node *S* to authenticate destination node by requesting source and its neighbours trust values recorded for the source node.

The AODV process at destination node is shown in figure 4.6. The figure 4.6 presents the difference between standard and trusted AODV process. The trusted AODV requests the trust values from source neighbour node and if authentication is successful, a reply is sent in the form of RREP message. Before any data is exchanged the ECDH algorithm is implemented. The additional steps are shown at the end of trusted AODV in figure 4.9.

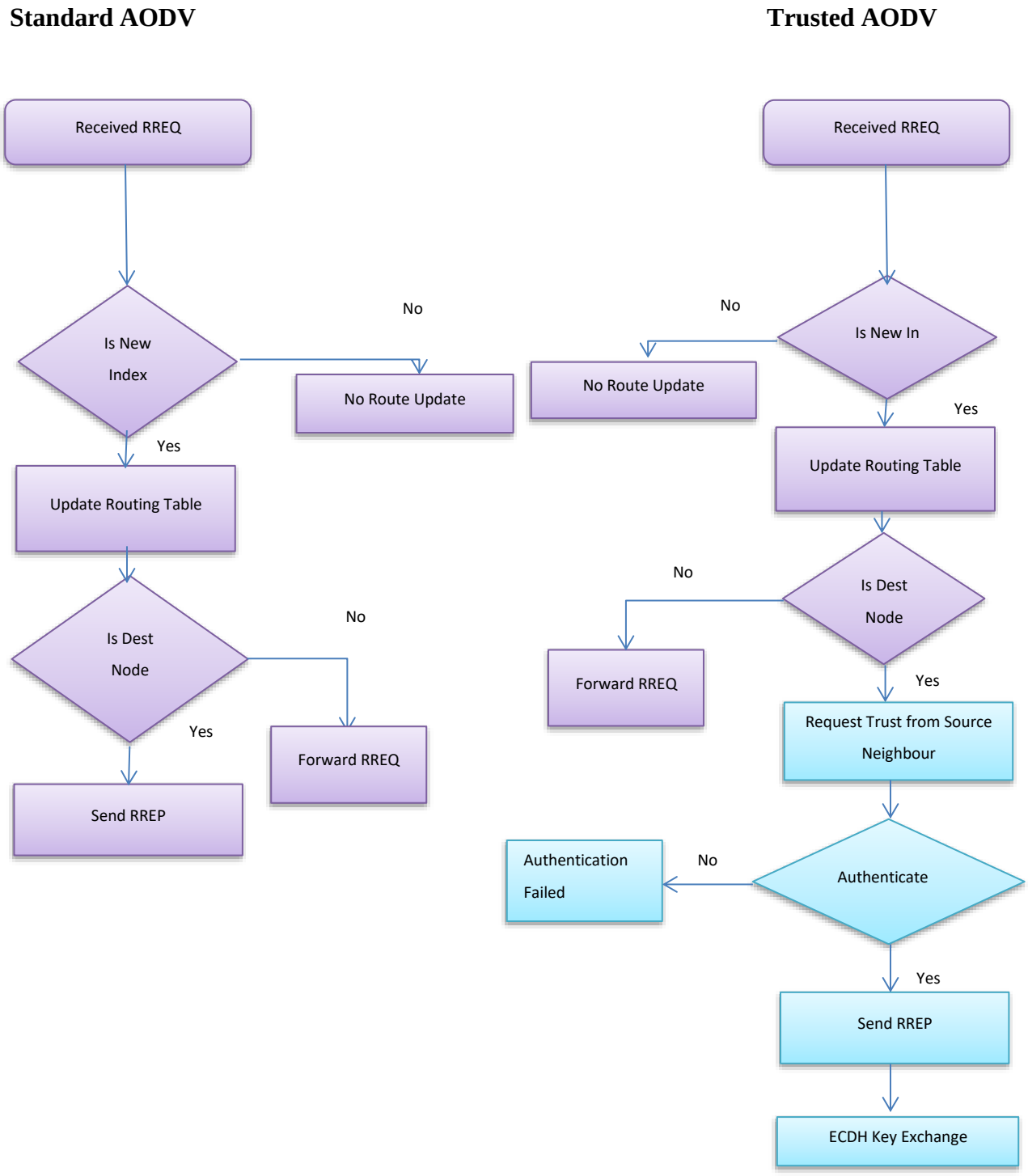


Figure 4.9 Destination Node DFD Standard versus Trust AODV

4.6.2 AODV Authentication process at Source Node *S*

The source node *S* waits for a route reply RREP after sending a RREQ in order to communicate with the destination node *D*. When it received a RREP from destination node *D*, the source node *S* then repeats the same process performed by the destination node. Source node also requests the trust values from all the neighbours of the destination node. Upon receiving the trust values of destination neighbours, the source compares the trust values and authenticates the destination node to establish communication. As both nodes *S* and *D* have no security association with one another to exchange data, hence the proposed scheme provides that layer of security by using trust to authenticate destination node. The figure 4.10 shows the steps in AODV, when the Dynamic Trust Based Scheme is implemented. The steps highlighted in the end, where the source receives the RREP, it requests the trust from destination's neighbours followed by ECDH, which constitutes the last step.

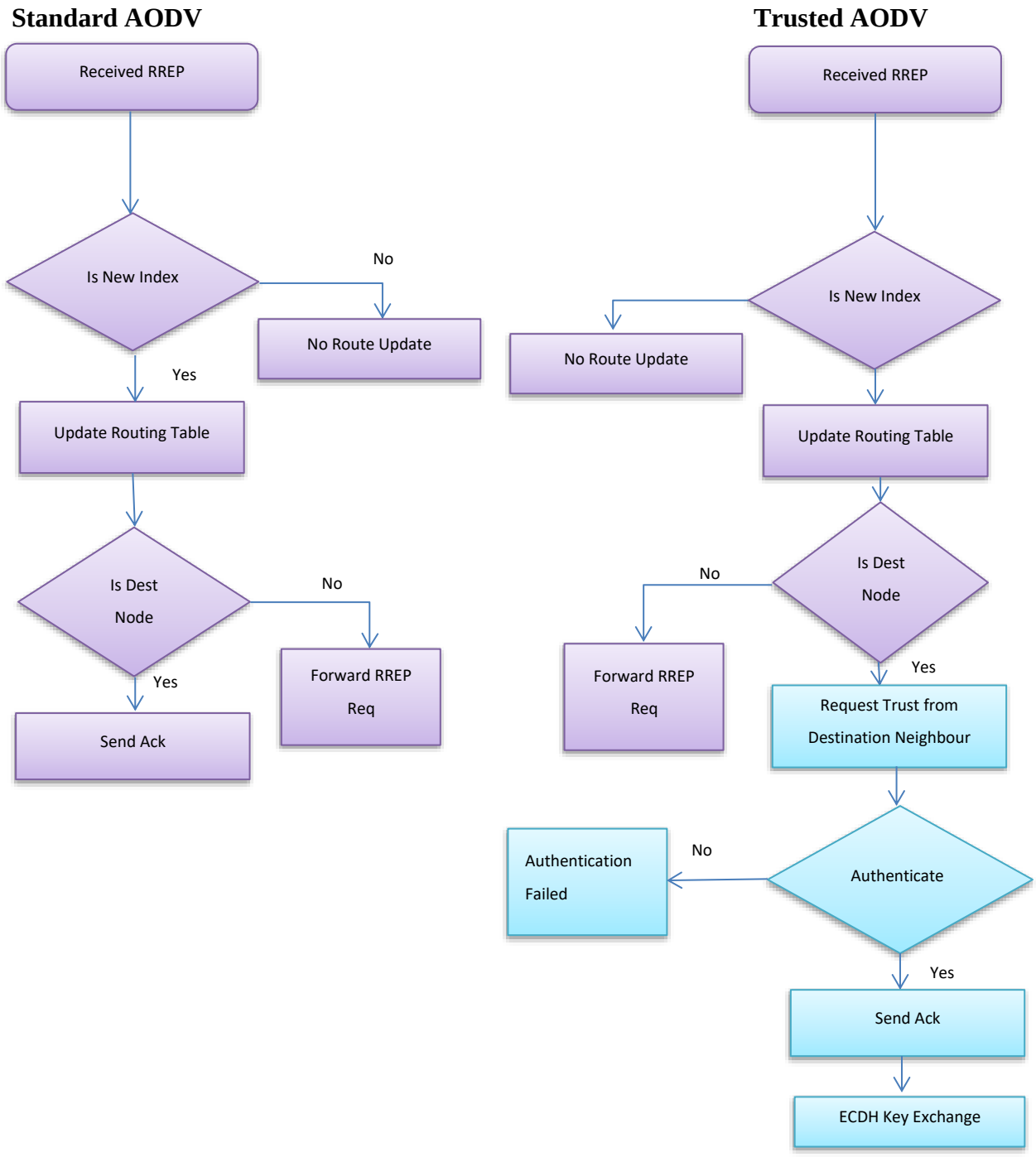


Figure 4.10 Source Node DFD Standard versus Trust AODV

The last step of the proposed scheme is the key exchange mechanism to encrypt messages using secret keys. The keys are exchanged using Diffie-Hellman key exchange. This would ensure the data is encrypted and could not be intercepted or tempered with by eaves dropper between source *S* and destination *D*.

4.7 Conclusion

Different stages of the algorithm and details of its functionalities are discussed in this chapter. As the proposed scheme provides a foundation for MANET routing protocol to implement a layer of security that enables a distributed, trusted and secure key exchange algorithm when the network initialises, therefore, this scheme can be implemented on various other MANET routing protocol as well. The aim was to provide a scheme that does not rely on a specific routing protocol apart from few modifications presented in this chapter. As mentioned above that the proposed scheme is not dependent on any specific routing protocol, therefore, it could be extended to any Adhoc routing protocol by making few modifications to packet header.

Chapter 5

Implementation

In this chapter, the details of the implementation of the proposed scheme are presented. This includes the methods adopted and implemented using AODV routing protocol and the evaluation through simulation with Network Simulator NS2.33 using Ubuntu 16.04 (NS2 Documentation, No date). The simulation results are evaluated extensively under varying network conditions such as mobility, network size area and node count. The simulation environment and different parameters used to generate results are also discussed. The full TCL, CBR and scenarios file is attached in appendices D.

NS2 is used to test the proposed scheme because of its support for various protocols. It is open source and available on Windows, Linux and MAC platforms. However, it uses Cygwin as a platform for implementation in Windows. There are active forums; and help and support is available online (NS2 documentation, no date). NS2 is a simulation tool that provides better support and documentation for its users to help understand how protocols work and interact with different topologies. The underlying language, it is built on is, Object Oriented C++ however, an additional language TCL is used for scripting, which makes it relatively less efficient and difficult to understand (Henderson 2011).

5.1 Simulation Environment

There were two main parts when implementing the trust algorithm in NS2. Firstly, modifying the existing C++ code to implement the trust based scheme. Secondly, generating variety of scenario and

implementing them in TCL to run the simulation. The source code mainly involved using Hello packet and transmitting and receiving of RREQ and RREP packets. The Hello packets were used to carry the parameters values needed to work-out the dynamic trust of neighbour node, while RREQ and RREP packets were used to carry the trust values for mutual authentication. The RREQ packets were used to request and RREP were used to send the trust value between communicating pair nodes. The code is designed as efficiently as possible. This is achieved by making the use of online libraries and code re-use in form of modifying AODV member functions. As highlighted previously, one of the reasons for selecting NS-2 is the online resources and help. By code re-use we mean that the original code designed for RREQ and RREP designed for AODV was modified to carry the trust information. The snippet of the code requesting and sending trust information in the form of TREQ and TREP is given in the appendix E.

To test the proposed security scheme it is implemented in NS2 using Tool Command Language (TCL) script, to build the network scenario and using CBR as traffic generator. Varying the number of nodes, speed, cover area and simulation time are some of parameters used to test the proposed scheme.

Node in MANET could be laptops, PDA's, cell phone and any other device using wireless technology. The simulation environment is NS2 and the detail parameters are listed in the section 6 in chapter 5. A wireless channel using 802.11 as MAC protocol is used to run the proposed scheme in the simulation. The type of applications used is CBR. AODV is used as routing protocol.

5.1.1 Mobility Model and Date Rate

The mobility model used is Random Waypoint Mobility and the speed details are shown in the table 6.1 in the next chapter. There are different numbers of nodes used with varying parameters to analyse and test the outcome. The details are listed in the table 6.1 and 6.3. The date rate is set to 11Mb. The simulation covers the area between minimum x=400 and y=400.

5.1.2 Malicious Nodes

Nodes that intentionally drop data packets instead of forwarding them are known as malicious nodes. They are introduced to the network to test and analyse how standard AODV reacts to these nodes as compared to AODV running trusted scheme. The TCL script below from Appendix-C, shows which nodes are acting as malicious in the network having 20 nodes.

```
# Adding malicious nodes
$ns_ at 0.0 "[$node_(15) set ragent_] malicious" // Node 15 is set as malicious
$ns_ at 0.0 "[$node_(25) set ragent_] malicious" // Node 25 is set as malicious
$ns_ at 0.0 "[$node_(35) set ragent_] malicious" // Node 35 is set as malicious
```

Algorithm 5.1 Malicious nodes TCL script

Also, to test the proposed scheme against Denial of Service attacks such as Blackhole and Greyhole attacks, malicious nodes were introduced in the network. Malicious node drops any data packet it

receives. The number of malicious node and their index is given in the table 6.2. When the network initialises and nodes start to communicate, the proposed scheme is expected to workout trust for each node in phase-one and identifies malicious nodes in the network. The proposed scheme successfully workout the trust threshold and dynamic trust threshold of all nodes as demonstrated and shown in results section. If the trust threshold of a node is less than the static trust and dynamic trust threshold, then those nodes are classed as malicious and excluded from routing.

5.2 Scheme Design

The routing protocol as mentioned above is AODV. Simulation scenario is designed to test the proposed scheme in the presence of a malicious node. When the TCL script is executed the network animator called NAM displays the layout of nodes in the simulation window and console generating output and displaying results.

The protocols is initiated when a source node send a RREQ. The data flow diagram in figure 5.1 shows how the proposed scheme is initiated using AODV as routing protocol. It also shows the very first step of ADOV operation, to find route from source to destination. The AODV is modified to implement the mutual trust authentication step as presented in figure 4.9 and 4.10.

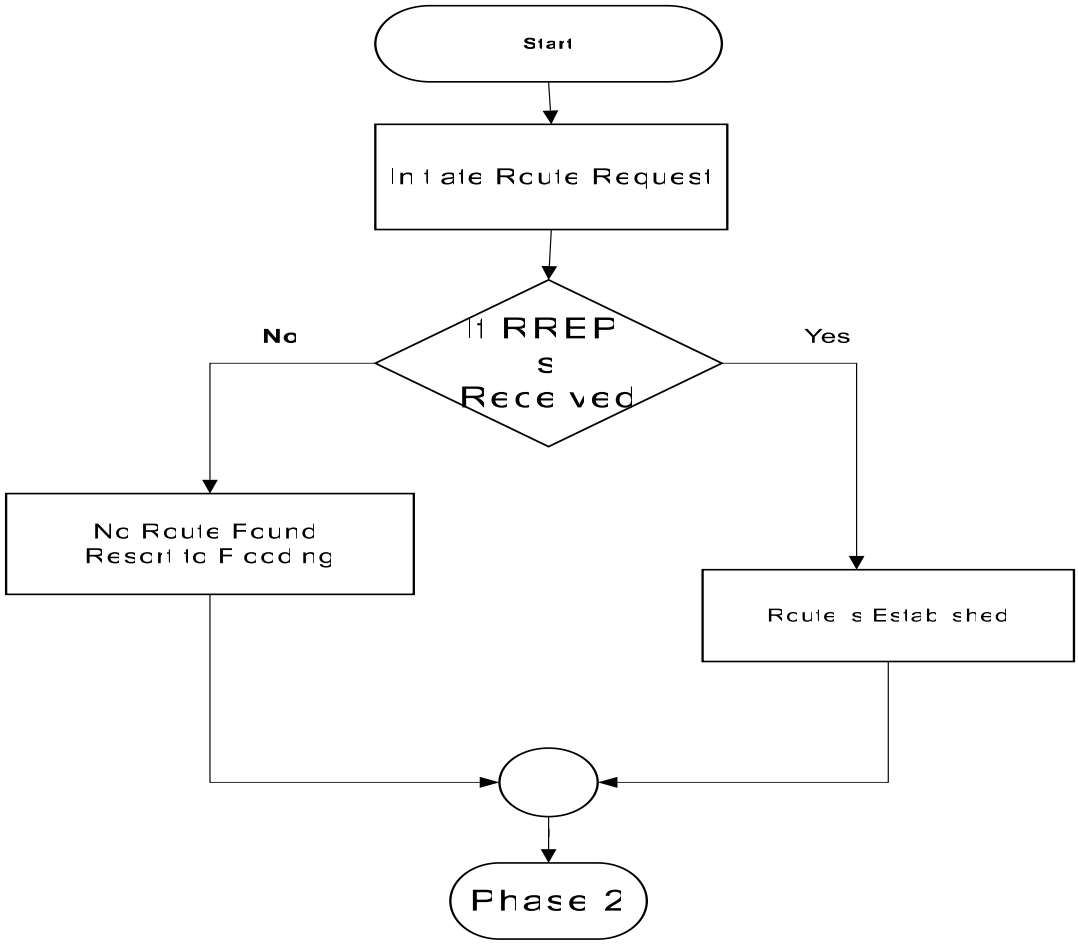


Figure 5.1 AODV Operation of Route Request

5.2.1 Trust metrics exchange

All nodes operating in promiscuous mode are listening to packets sent and received by all neighbour nodes. When the simulation begins and nodes start communicating, source node checks, if it has a route to destination, and if it doesn't have a route it broadcasts a RREQ. Destination nodes also known as sink node in this thesis as it has been referred to by same name in NS2.

All nodes maintain a trust table and calculate the trust values of its neighbour and along with the trust value, calculated and recorded by a node, it also sends the value to its neighbours every time a hello message is sent.

The communication between neighbour nodes is carried using hello packets. The metrics used to calculate dynamic trust are also exchanged through hello packets in order to avoid increasing network traffic then absolutely necessary. The use of hello packet to carry the metric information also enabled us to update such values each time a hello packet is sent; hence the trust values are automatically updated. This endorses the initial statement that the scheme is self-configuring and gets itself rectified automatically. The code in Appendix-E shows how the hello packet is modified to carry various metrics.

Once the trust table is populated with its trust values and trust values received from all neighbours regarding every neighbour, then the node is able to calculate its own and trust threshold of all neighbours using equation 4.8 in section 4. The average trust is used to calculate the trust of node. The node is considered trust worthy if the trust threshold is greater than the dynamic threshold or static trust.

Modifications have been made to Hello Packets structure to exchange Trust Threshold value and other parameters such as nodes density, two-hop neighbours, energy and mobility as mentioned in section 4.3. Hello packet in AODV uses the route reply (RREP) packet format to send hello messages to neighbours (Bhanot and Chaudhary, 2017). Hello messages are broadcasted using the RREP packet with the following parameters shown in table 5.1 below:

Parameter	Value
Destination Address	Node's IP address
Destination Sequence number	Latest sequence number of sending node
Hop Count	0
Life Time	ALLOWED_HELLO_LOSS X HELLO_INTERVAL

Table 5.1 Hello packet parameters

The figure 5.2 below shows the standard AODV hello packet structure whereas the figure 5.3 shows the modified hello packet of the AODV.

Type	**Flags**	**Prefix Size**	**Hop count**
Dest_addr			
Dest_sequence_#			
Source_addr			
Lifetime			

Figure 5.2 Hello packet format of standard AODV

Modified Hello packet format is appended below with added fields highlighted:

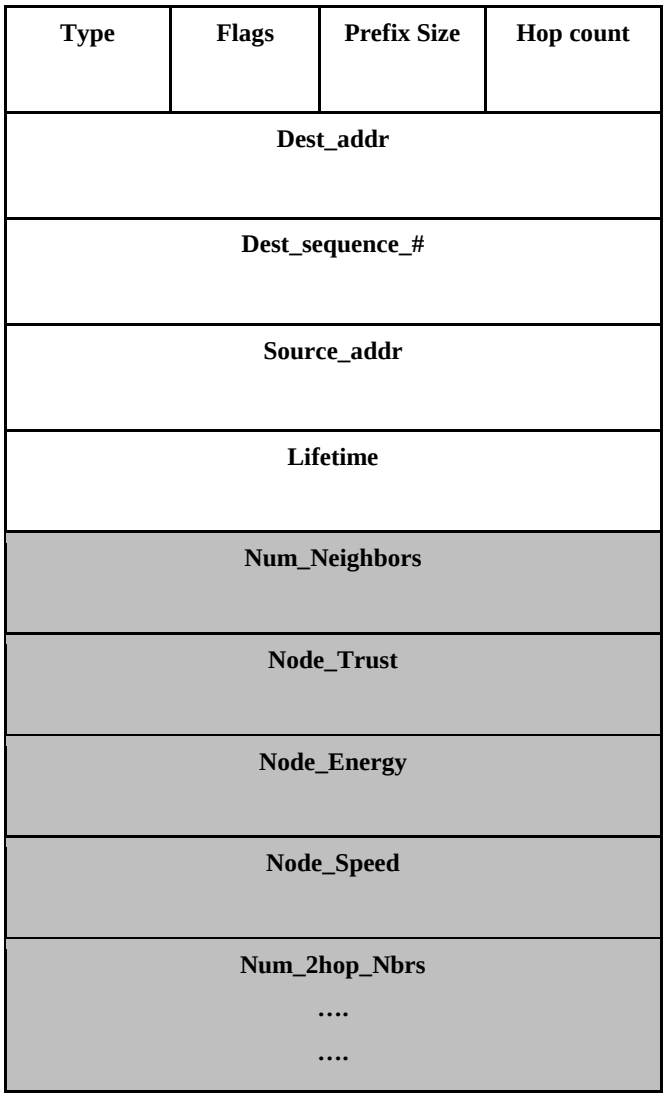


Figure 5.3 Hello packet format of Trusted AODV

The Appendix-E shows the C++ source code used to modify and implement the trusted AODV RREQ, RREP and modified hello packets in NS2. The code also includes the mutual authentication process of how trust information is carried between nodes to send trust requests and send trust replies.

5.3 Mobility

Mobility is implemented using Random Way Point mobility model in NS2 (NS2 documentation, no Date). The node speed is calculated in meter per second, varying between 5 (min) and 20 (max). Scenarios were created by applying values using 'Setdest' functionality in NS2, a built-in method for generating scenarios. Node destination and speed are provided as input to measure speed. The position of nodes is updated only when there is change in the destination. The current speed is calculated from previous value of speed and the mean speed given as input as described by NS2.

5.4 Energy

Energy model in NS2 is used to implement energy and access energy of nodes during simulation. It is a key element in Adhoc network. Node has an initial value in the energy model in NS2, which represents the level of energy a node has at initialisation (beginning of the simulation). It is known as initialEnergy_. For every packet a node sends and receives, it has an energy usage. The packet transmitted is txPower and received is called rxPower.

When the simulation starts, the energy_ is set to initialEnergy_ which is then decremented for every transmission and reception of packets at the node. No more packets can be received or transmitted by the node, when the energy level at the node goes down to zero. The energy is assigned in NS2 by using parameters showed in the table 5.2.

Attribute	**Description**	**Value**	**Initial/Default Value**
EnergyModel	Type of Energy Model	EnergyModel	NS2.33Built-in
rxPower	Power for receiving one packet	Power in watts (ex 0.2)	35.28e-3
txPower	Power for receiving one packet	Power in watts (ex 0.1)	31.32e-3
initialEnergy	Node energy at initialisation	Energy in joule	100
SleepPower	Power consumed during sleep state	Power in watts	144e-9

Table 5.2 Energy Model

5.5 Dynamic Threshold Simulation

Once all the nodes have calculated the average trust values for its neighbours, this value is used as input to calculate the average trust threshold. We have presented in detail, in analytical model section, various parameters and the mathematical formulas or equations to calculate those parameters. The dynamic threshold can be computed using the threshold values of all parameters. The average trust value is required for two reasons.

Firstly, it gives us a dynamic value based on network conditions at a particular time after the network is initialised. The threshold value is obtained by taking all five parameters into account therefore; it is referred to dynamic threshold value, as it is obtained dynamically. It can be used to measure the trust level of each node by comparing the static average trust value with dynamic threshold value.

Secondly, it is used to authenticate peer nodes prior to any data communication. All the neighbours of source and destination nodes trust values are requested by the corresponding peers. The destination requests average trust values of the source's neighbours and source requests the average trust values of the destination's neighbours. This enables both communication nodes to get first hand trust information about their corresponding peer and thus provides an additional layer of security. The additional layer is used to mitigate against some of the known form of attacks such as Blackhole, Greyhole and Wormhole. The details of these types of attacks are discussed in section 4. Figure 5.4 shows the flow diagram of dynamic threshold scheme.

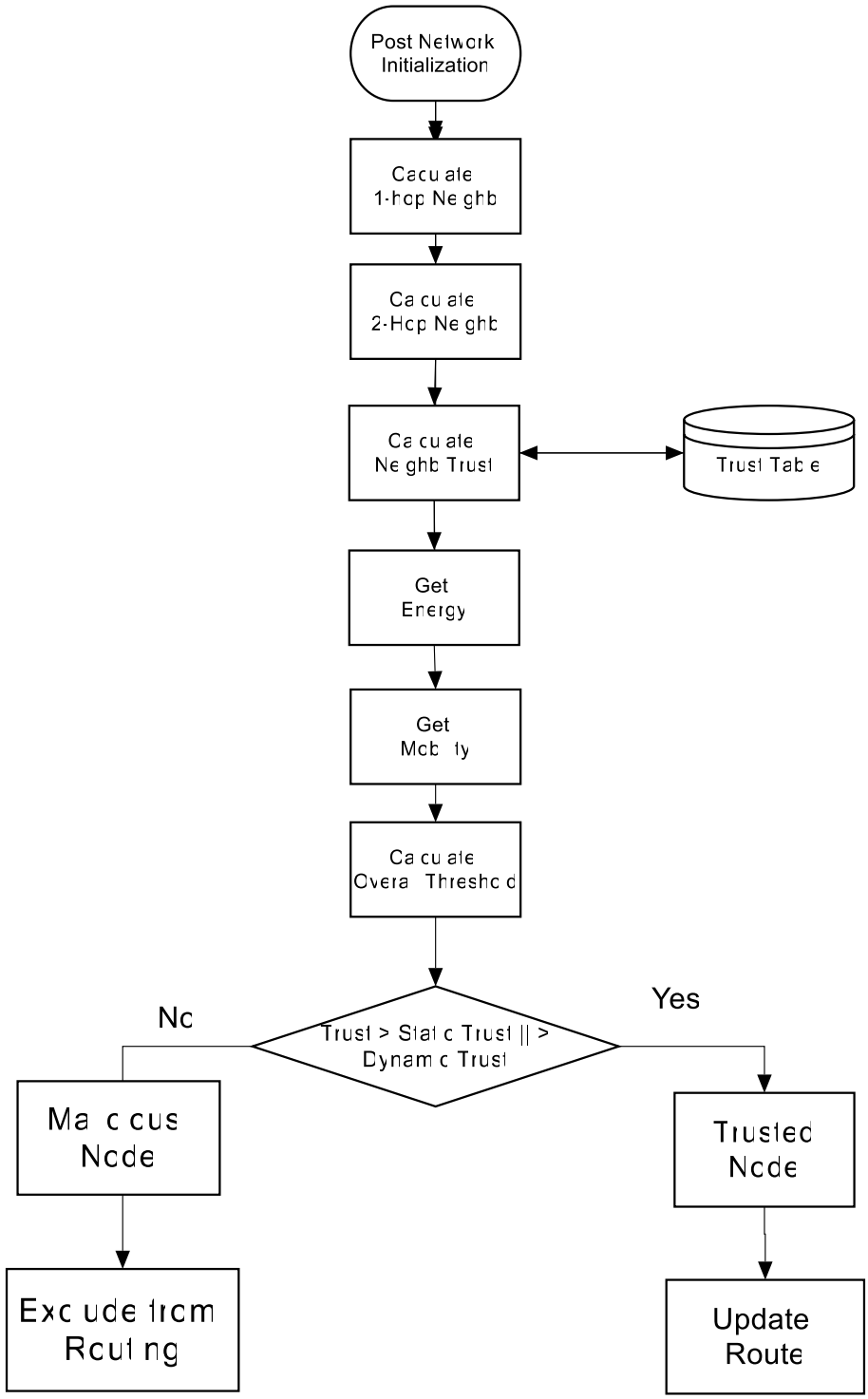


Figure 5.4 Dynamic trust threshold flow diagram

5.6 Conclusion

The details of how the proposed scheme is implemented using AODV protocol and the test results are presented in this chapter. We tested the proposed scheme using various parameters using NS2 simulation tool. The reason for the selection of NS2 and various underline resources used to implement it, is also discussed in detail in this chapter.

CHAPTER 6

Performance Evaluation

The performance of the proposed scheme is discussed in this chapter. The tests are conducted using normal AODV having varying number of malicious nodes running without using a security or trusted algorithm compared to AODV using the proposed dynamic trusted scheme.

The tests were conducted to check the performance of the standard AODV compared to Trusted AODV. Standard AODV was used as a reference to check whether the Trusted AODV can mitigate against malicious attacks and what are the performance impacts. In order to ensure that the scheme is tested in a diverse environment and with a variety of metrics, the simulation tests results were generated by taking different scenarios and parameters into account. The test results indicate that Trusted ADOV can successfully thwart packet dropping attacks however, there is a performance overhead.

6.1 Performance Metrics

The performance of Trusted AODV evaluated using the following metrics:

- **Throughput:** It is the amount of data (bit or packets) transferred between source and destination per period of time (seconds).

$$\text{Throughput} = \frac{Size\ of\ Data\ Received}{StopTime - StartTime} x \frac{8}{1000}$$ …..……Equation 6.1

- **Routing Overhead:** Routing overhead represents any control packets required by protocol to perform a specific task. It is therefore, the sum of all the control packets sent during the total simulation time.

  RO = Sum of the *Total Number of AODV packets sent* ……..Equation 6.2

- **Average End-to-end Delay:** It is the average time taken by packet to reach from source to the destination. This includes the delay caused by retransmission (delay at MAC level), buffering (during route discovery) and queuing delay (interface queues).

$$EED = 1/N \sum_{n=1}^{N} (r_n - s_n) \quad \text{sec.}$$

…….Equation 6.3

r_n **:** Time when data packet was sent

s_n **:** Time when data packet was received

N: Total number of data packets received

- **Packet delivery ratio:** The ratio at which packets are delivered in the network.

$$PDR = \frac{\sum_{\forall i \in D} TPR_i}{\sum_{\forall i \in D} TPS_k} x100 \qquad \text{……..……Equation 6.4}$$

The *TPRi* represents the total number of packets received by the DBR destination *i,* and *TPSk,* represents total packets sent by CBR source *k.*

Where:

S: Represents set of CBR source

D: Represents set of CBR destination

Packets parameters such as packets forwarded, received, sent and dropped are also presented to compare the performance of standard AODV with trusted AODV. Two scenarios having 20 and 50 nodes are generated using parameters listed in the tables 6.1, 6.2 and 6.3, 6.4 respectively, to compare and evaluate the metrics. The using NS2 simulator is used the standard and trust AODV. The details of the complete scenarios and test bed created are listed in the appendix A-D. The performance metrics are obtained and analysed when the Dynamic Trust scheme is run by varying the number of nodes and mobility to prove that the scheme can successfully implement a trusted framework to authenticate nodes.

6.2 Scenario-One (20 Nodes): Standard AODV VS Trusted AODV

- **Node Mobility Parameters**

Two scenarios are generated using different parameters. Scenario-one mobility and size parameters are listed in the table 6.1 to compare and evaluate the metrics obtained using NS2 simulator. Identical scenarios are run using standard AODV and dynamic trusted scheme in the presence of malicious nodes and the results obtained are presented in the section below.

The Random Waypoint Mobility (RWM) model was used to generate mobility. Parameters listed in table 6.1 were used to generate mobility in NS2. The complete file is attached in appendix B. In this scenario the normal AODV is used as a routing protocol without any trust scheme. In order to obtain and compare results compared to AODV using the proposed dynamic trust scheme, malicious nodes are added to the network. As described in the table 6.2, there are three malicious nodes introduced. Any data traffic that comes in the path of malicious nodes will be dropped. This type of attack is called Grayhole attack which is a type of Denial-of-Service where nodes drop data packets that it receives. The table 6.1 shows node properties and simulation system environment and table 6.2 shows all the parameters and their corresponding values.

Machine Specification					
Model	CPU	CPU's Speed	Memory	Memory Speed (Hz)	Operating System
HP Probook 450	Intel Core i5	2.20 GHz	8.0 GB	166 MHz	Ubuntu 16.04

Table 6.1 Simulation System Environment

	Node movement scenarios and Network size parameters						
Mobility model	Network Size (Node)	Malicious Nodes	Topology Size (m)	Transmit. Range (m)	Node's Speed (ms^{-1})	Pause Time (Seconds)	Simulation Time (Seconds)
RWP	20	3	400x400	250	5-20	0-100	180

Table 6.2 Node Movement and Network Size

Where:

Nodes: Total number of nodes in the network

Min Speed: Minimum speed.

Max Speed: Maximum speed a node can achieve. Represented in meter per second

Pause Time: The interval of time where the node stops any movements. It is represented in milliseconds.

Dimensions: The area of the network in x and y dimensions.

Number of Malicious Nodes: Node 1, 10 and 19 are malicious nodes in the network are.

- **Parameters Specifying Traffic Patterns**

The data parameters are shown in table 6.3, list all the parameters and their corresponding values used to run the simulation.

Conn No	Source Node	Sink Node	Application	Send Rate	Layer 4 Type	Packet Size	Max Pkts	Conn Time
1	1	2	CBR	0.2 Approx.	UDP	512	10000	2.556 Approx.
2	4	5	CBR	0.2 Approx.	UDP	512	10000	56.333 Approx.
3	4	6	CBR	0.2 Approx.	UDP	512	10000	146.9651 Approx.
4	6	7	CBR	0.2 Approx.	UDP	512	10000	55.634 Approx.

Table 6.3 Traffic Pattern Parameters 20 Nodes

Where:

Conn No: Represent the maximum number of connections

Sink Node: Representing node that receives the data.

Send Rate: The interval after which data is sent

Packet Size: The size of each data packet.

Max Pkts: Represent the maximum number of packets

Conn Time: Simulation time at which two nodes connect to exchange data.

6.2.1 Performance Scenario-One

- **Packet Statistics**

One of the scenarios tested was, varying the number of nodes in the network. The packet statistics include packet sent, received, forwarded and dropped. The key characteristic of the trust based schemes is that each node observes its neighbour, in other words, the nodes operate in promiscuous mode. All the packets that are sent, received, forwarded and dropped are observed by each node.

The figure 6.1 shows the packet statistics obtained when the network is running standard AODV compared to Dynamic Trusted AODV protocol and both having three malicious nodes as adversary to simulate an attack in the form of packet dropping. The results show total packets sent, total packets received, total packets dropped and packets forwarded.

The purpose of gathering packet statistics is to gain an insight into the network when its running using AODV with and without trusted algorithm, given the same set of conditions. For the purpose of the testing, malicious nodes are introduced in the network. These nodes drop any packets that they receive. The key purpose of packet statistics is to capture the malicious activities in terms of the number of packet dropped by malicious nodes. This is an indicator of how good or bad the response of trusted AODV is compared to standard AODV, as the later is used as a reference to test the proposed scheme. The response is measured by comparing the difference between the total number of total packets sent and received. The difference between them gives us the total number of dropped packets. The packets dropped indicate the network has misbehaving and malicious nodes. By comparing the results it is realized that ratio of packet drop is high in standard AODV protocol as compared to trusted AODV. This is due to the trusted algorithm, which makes the network more resistant to packet drop attacks. Therefore, it can be concluded that the trusted algorithm is performing by providing mitigation against Blackhole attack.

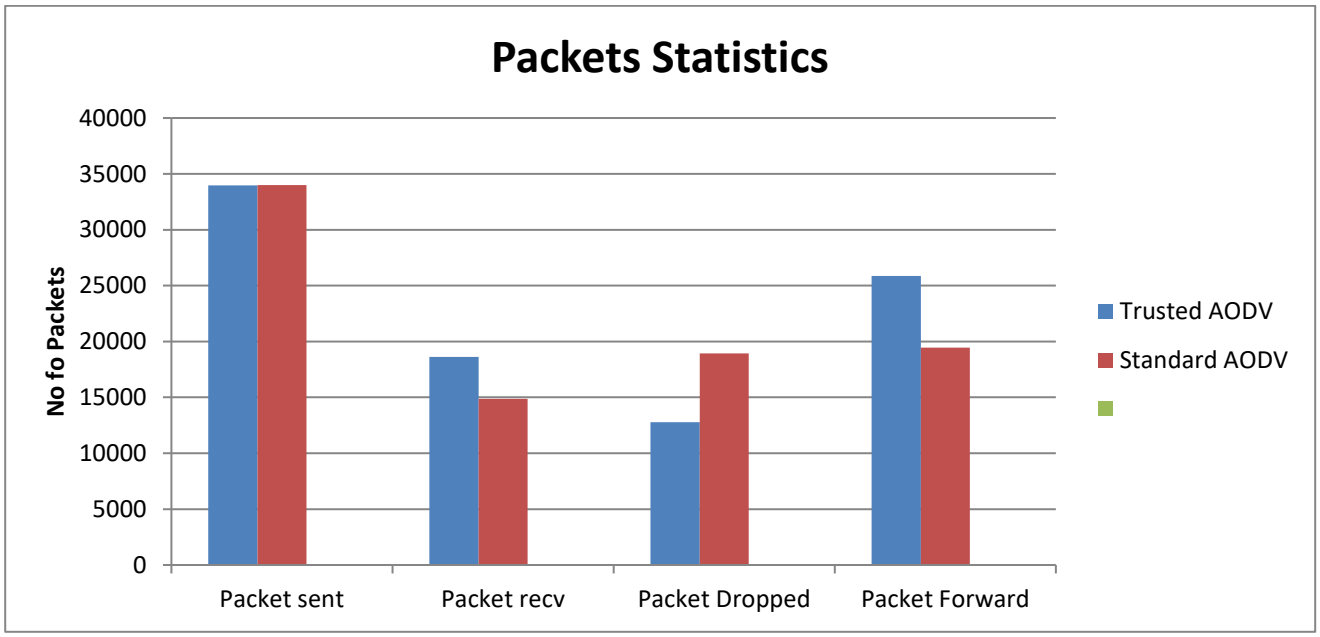


Figure 6.1 Packet Statistics

The packet statistics show the total number of packets sent by trusted and standard AODV. As shown in the figure 6.1, the number of packets received and forwarded by trusted scheme is relatively higher. The reason behind this is that, the trusted scheme is preventing malicious nodes from taking part in routing and also preventing packet dropping. For the same reason the number of packet dropped is lower in trusted AODV as compared to standard.

- **Throughput**

Using scenario-one, the Throughput obtained is shown in figure 6.2. When data is transmitted from one place to another in a network, throughput is the amount of data moved successfully from one point to another in a given period of time. Since, malicious nodes are introduced to the network therefore; the network throughput in case of standard AODV is as expected. Malicious nodes in the network are dropping any data packets when they traverse the network and as normal AODV has no protection against malicious nodes, as a result, it has low throughput as shown in the figure 6.2. On the contrary, trusted AODV has a relatively higher throughput compared to standard due to fact that the dynamic trust scheme is mitigating against malicious dropping packets.

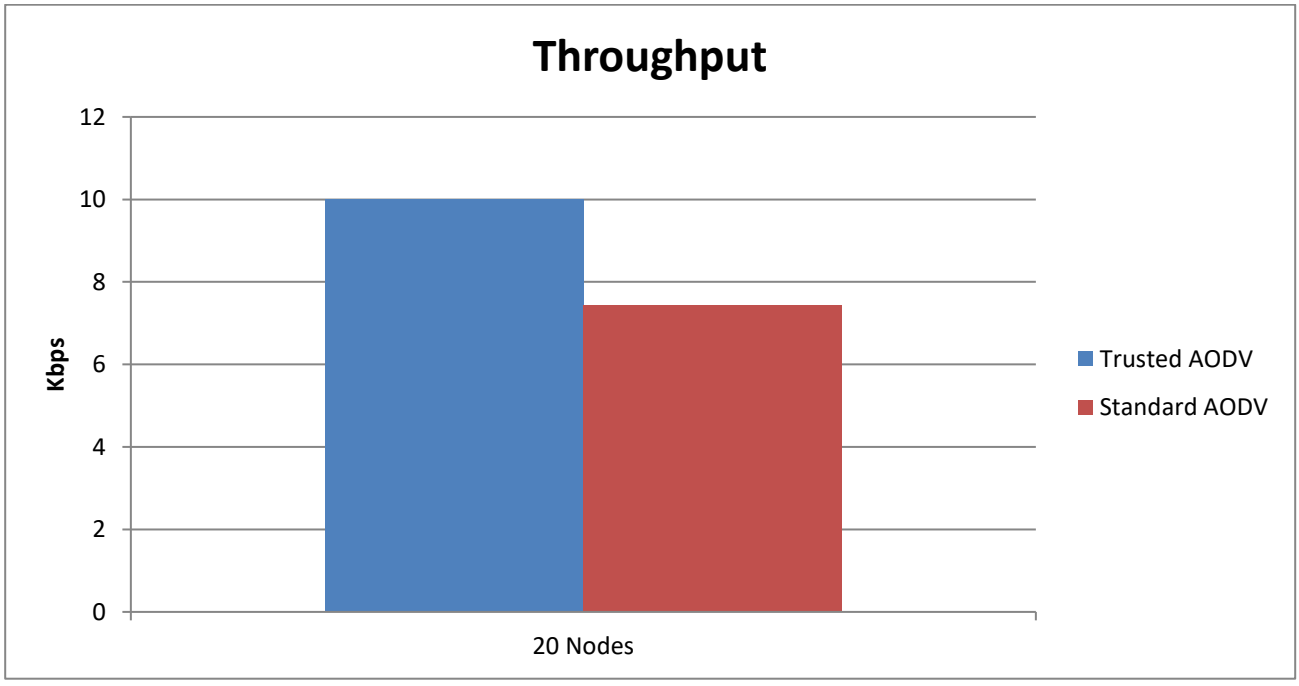


Figure 6.2 Network Throughput

The most important of all is the performance metric is the throughput, which is referred to as the number of packets successfully received per unit time. It is an important indicator of the performance and quality of network connection. The results show high throughput when trusted AODV is run against standard AODV. This indicates that even though the trusted algorithm has high overhead and end-to-end delay but it has high throughput.

- **Routing Overhead**

Routing overhead represent any control packets required by protocol to perform a specific task. The higher the number of these packets the lager the overhead become. These packets are required for network communication. Performance is critical for any organization and may be a priority but implementing security means slowing down and adding latency. It is therefore very important to measure the overhead of any scheme to find the right balance evaluate the reliance in AODV to carry the trust information and other metrics used to calculate the dynamic trust threshold.

The proposed scheme depends upon additional control packets to be sent between nodes to implement the trusted algorithm. The routing overhead produced are shown in the figure 6.3, it shows the overhead produced by standard AODV as compared to the trusted.

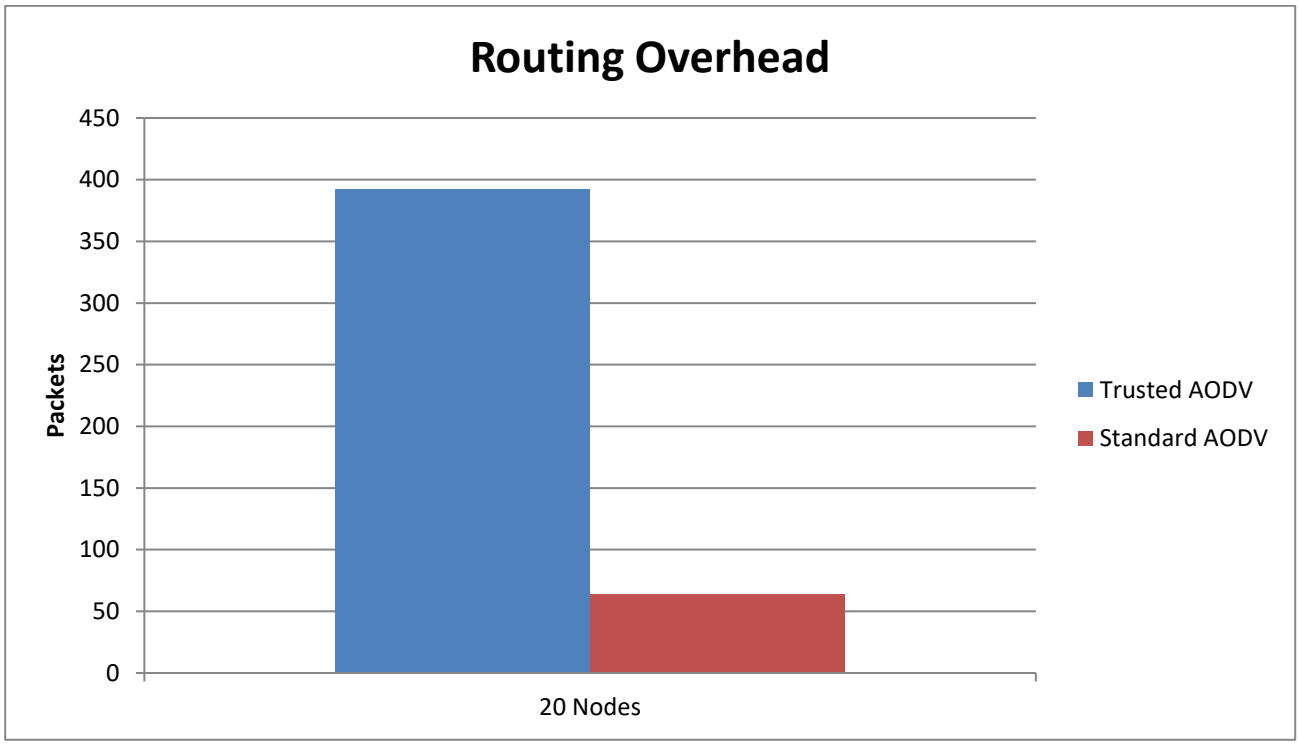


Figure 6.3 Routing Overhead

The dynamic trusted algorithm implemented using AODV as apparent from the packets statistics is showing positive results. But majority of the security techniques comes with some kind of performance overheads and trusted algorithms are not an exception.

- **Average End-to-End Delay**

Figure 6.4 shows Average End-to-End delay using parameters from scenario-one. The time taken by a packet to be transmitted from one point to another i.e. from source to the destination across the network is called End-to-End delay. This is the transmitting delay, propagation delay and queuing time of packets combined. The results show that there is slight higher delay due to trust based scheme running in the background and packets are routed to avoid any malicious nodes in the path. The standard AODV has less delay which is due to less overhead compared to trusted AODV.

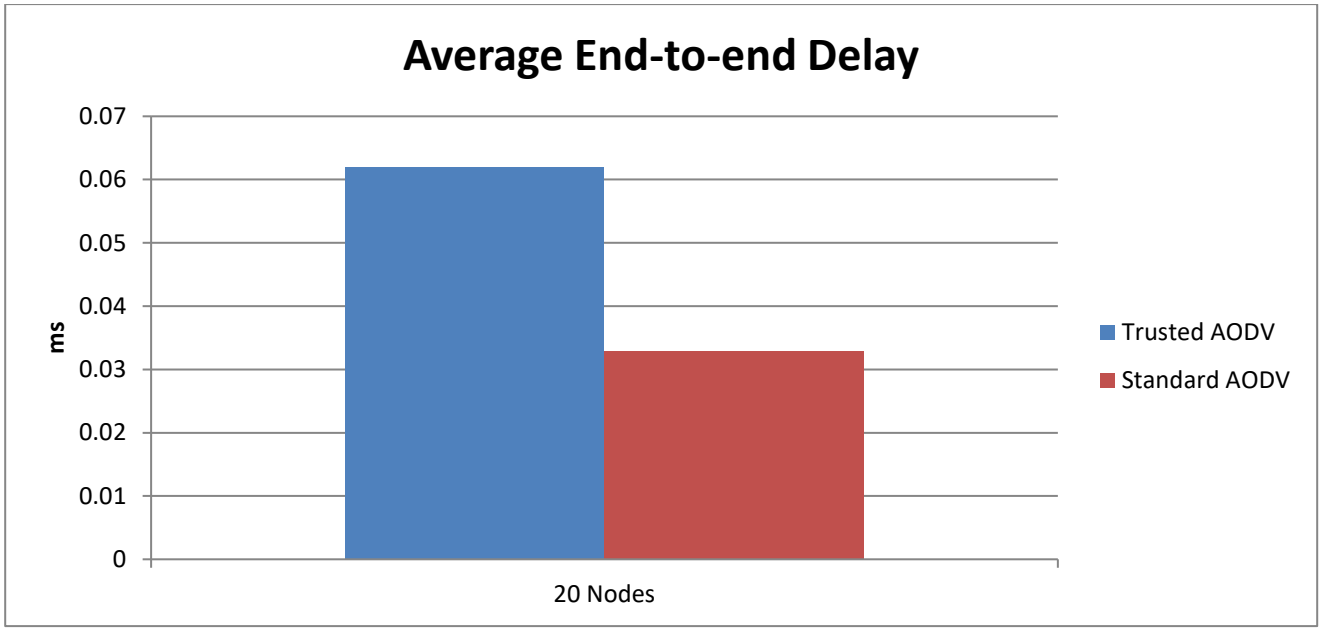


Figure 6.4 Ave End-to-End Delay

The overhead of the proposed trust scheme can also be represented by calculating the average End-to-end, as it is the time taken for a packet to travel from source to the destination. This is also an important performance metric because excessive delay can affect the throughput. Another consequence of a higher delay is that it could result in breach of TTL limit and ultimately the re-transmission of the packet. One of the functionality of the trust scheme algorithm is to identify the malicious nodes in the network. Once it is identified, the malicious nodes are excluded from routing, which means re-routing of packets thorough trusted nodes. This has a direct effect on time taken by packets to reach its destination. The running of trusted algorithm introduces the latency due to avoiding malicious nodes in its routing path and can increase the hop-count as results as well.

- **Packet Delivery Ratio**

The result for packet delivery ratio is shown in figure 6.5 below. This metric indicates the performance of the proposed trusted scheme after analysing all other performance metrics. This metric represent the ratio of the number of packets received by the destination to the number of packets sent by the destination. The packet delivery ratio is higher in secure AODV then standard as shown in the results section, which proves that the trust scheme is out performing. This is despite the relatively high overhead and end-to-end delay. It can therefore be concluded that although implementation of the trusted scheme comes with performance overhead but providing the necessary security to the MANET, which is the primary goal of this research.

The ratio at which the packets are delivered in the network is low for standard as compared to trusted AODV. The standard AODV has no defence against malicious nodes dropping data packets which has a direct adverse effect on packet delivery ratio.

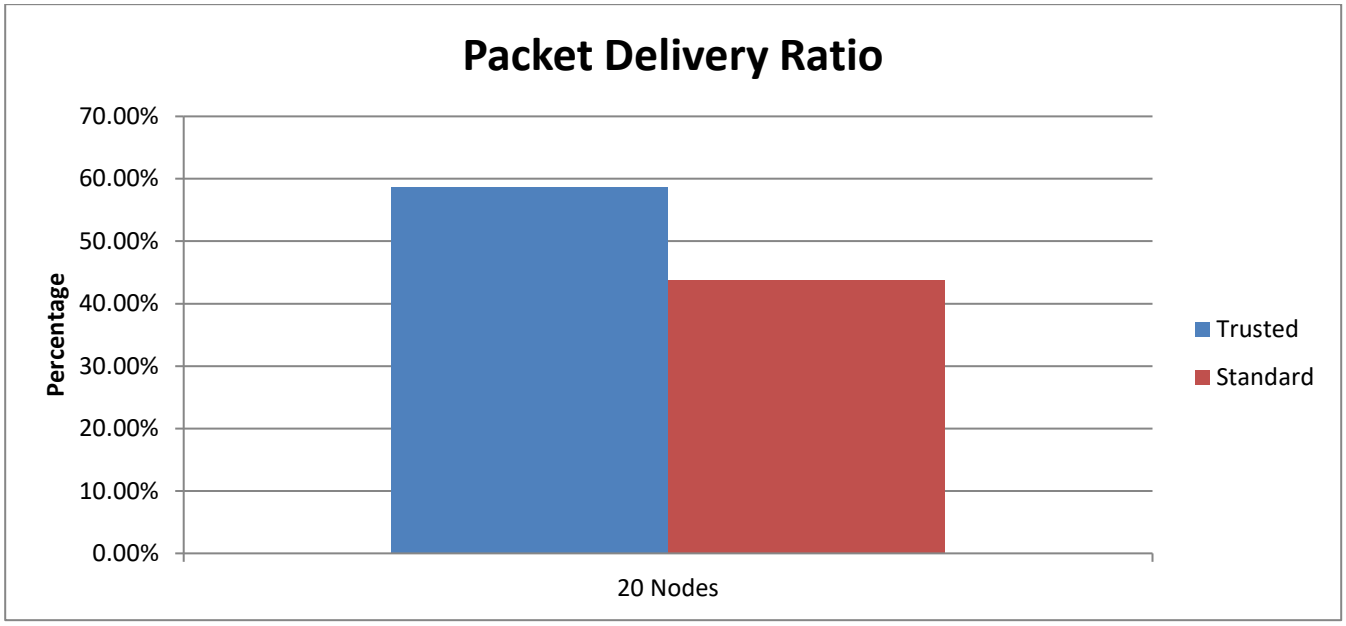


Figure 6.5 Packet Delivery Ratio

6.3 Scenario–Two (50 Nodes): Standard AODV vs Trusted AODV

- **Nodes Mobility Parameters**

In this scenario, the total numbers of nodes are increased to 50. Using identical parameters with increased number of nodes, the AODV protocol is run using Dynamic Trusted Scheme and normal AODV to compare the results. The individual results are shown below:

	Node movement scenarios and Network size parameters						
Mobility model	Network Size (Node)	Malicious Nodes	Topology Size (m)	Transmit. Range (m)	Node's Speed (ms^{-1})	Pause Time (Seconds)	Simulation Time (Seconds)
RWP	50	3	400x400	250	5-20	0-100	180

Table 6.4 Node Movement and Network Size

Nodes 15, 25 and 35 are malicious nodes in the network that drops data packet it receives. As shown in the table 6.4 the simulation area is 400x400 both x and y axis.

- **Parameters Specifying Traffic Patterns**

The data parameters are similar to what was selected in scenario-one apart from this scenario have larger number of nodes in the network. The number of nodes is increased to see the difference in

various statistics collected between AODV with and without Trusted Scheme. Number of maximum connections are 5 and CBR is used as an application layer protocol. Table 6.5 show all the parameters and their corresponding values in detail.

Conn No	**Source Node**	**Sink Node**	**Application**	**Send Rate**	**Layer 4 Type**	**Packet Size**	**Max Pkts**	**Conn Time**
1	1	2	CBR	0.1 Approx.	UDP	512	10000	2.556 Approx.
2	4	5	CBR	0.1 Approx.	UDP	512	10000	56.333 Approx.
3	4	6	CBR	0.1 Approx.	UDP	512	10000	146.9651 Approx.
4	6	7	CBR	0.1 Approx.	UDP	512	10000	55.634 Approx.
5	7	8	CBR	0.1 Approx.	UDP	512	10000	29.546 Approx.

Table 6.5 Data parameters for CBR application

6.3.1 Performance Scenario-Two

In the coming section, the results are analysed that are obtained using application data and mobility parameters of scenario-two. The statistics are again similar to what we had in scenario-one which includes packet data, throughput, routing overhead, end-to-end delay and packet delivery ratio.

- **Packet Statistics**

The packet statistics are shows in detail in figure 6.6 below. The results show a visible variation when the AODV is run using trusted scheme and without trust scheme. It can be observed from the results below that as the network grows and the number of nodes increases, the trusted scheme is still able to identify malicious nodes. As a result, the total number of packets forwarded is higher in standard ADOV with no trusted scheme running. The same pattern can be observed with packets dropped and packets received, where the number of packets dropped is higher and packets received is much lower in standard AODV compared to trusted one.

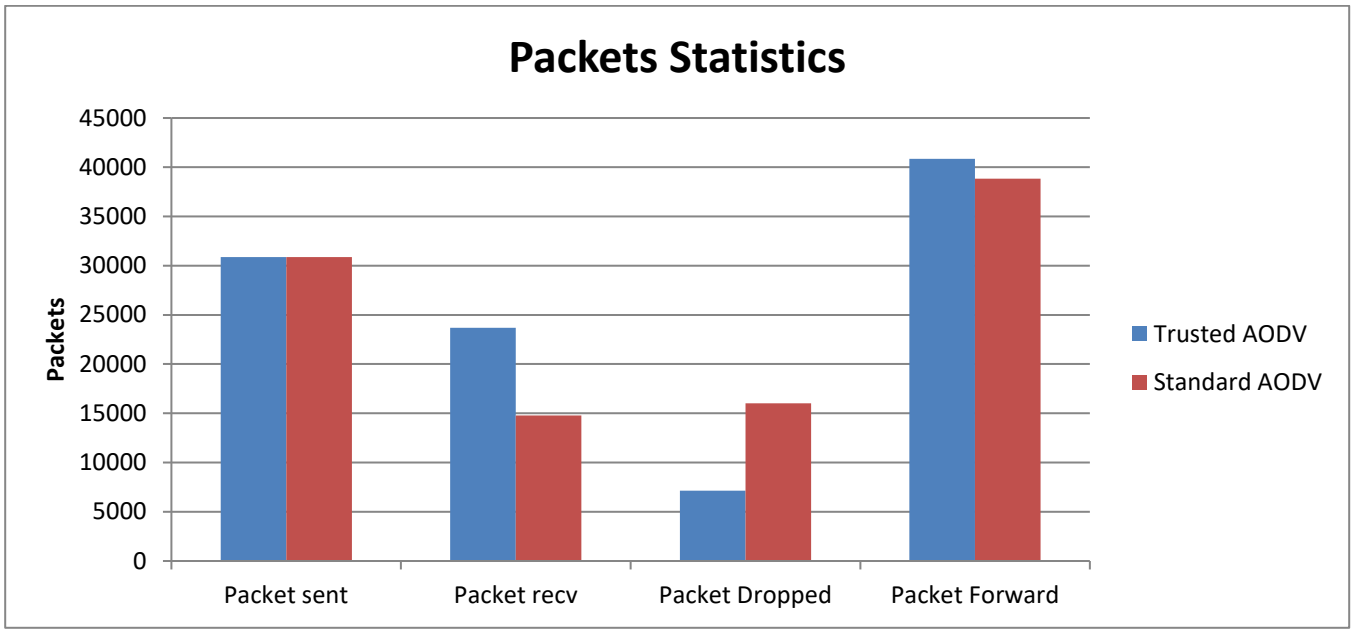


Figure 6.6 Packet Statistics

- **Routing Overhead**

Routing overheads reflect the use of number of packets generated using standard as compared to trusted AODV. Any control packets in the form of broadcast or unicast, whether it's a RREQ or RREP packets sent over the network, the result is an overhead. Thus, the higher number of control packets means a higher overhead. When the routing overheads of standard AODV are compared against trusted AODV, the trusted AODV as shown has higher number of control packets being exchanged.

The results in figure 6.7 show higher overhead due to the fact that the proposed trusted scheme relies on AODV protocol for its implementation. Thus, excess packets are required for the proposed trusted protocol to execute and provide the essential security.

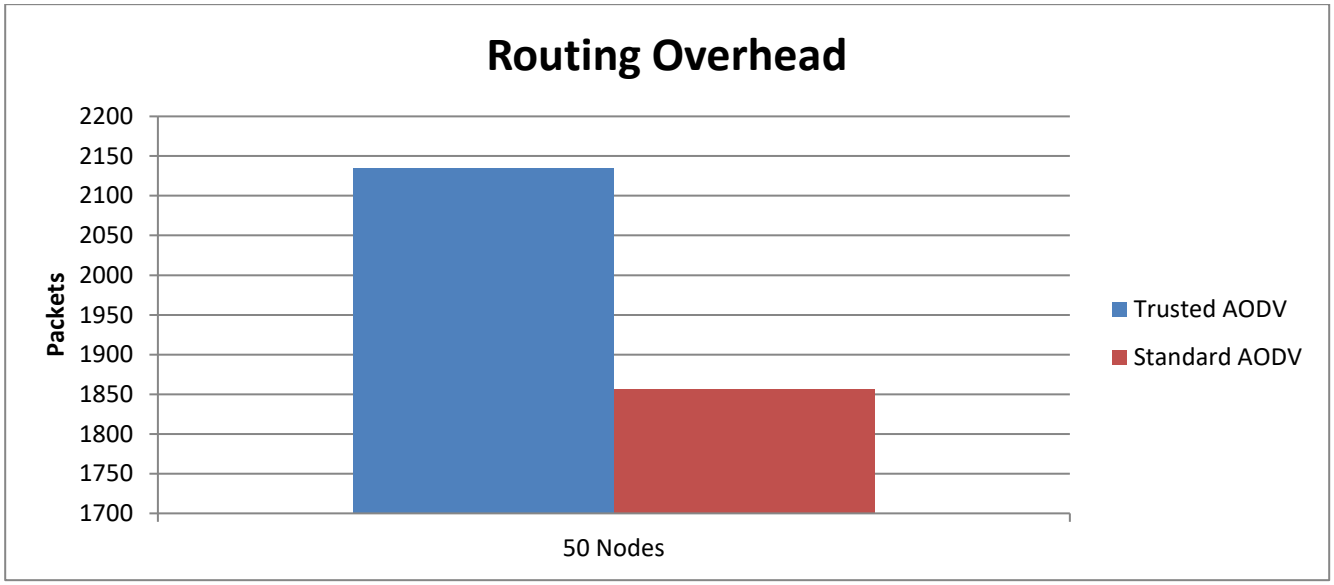


Figure 6.7 Routing Overhead

- **Throughput**

The figure 6.8 shows the throughput obtained using parameters of scenario-two. Malicious nodes are introduced to the network again in this scenario to compare the difference in throughput between trusted and untrusted AODV environment. The results show a significant drop in the throughput when there is no protection against malicious nodes. On the other hand, the throughput turns out to be relatively higher when the Dynamic Trusted Scheme is in place that enhances the network performance

by providing high security measures. The results presented in figure 6.8, shows that the trusted AODV resulted in about 35% overhead in the simulation where the network consists of 50 nodes.

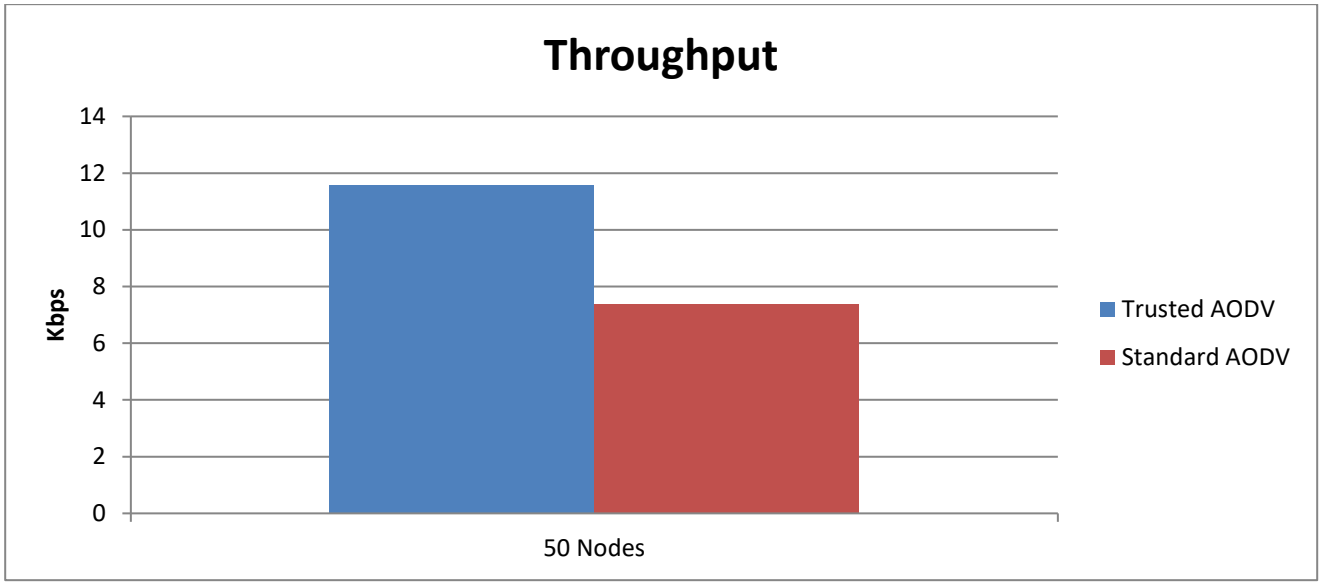


Figure 6.8 Network Throughput

- **End-to-End Delay**

The figure 6.9 shows Average End-to-End delay using parameters from scenario-two. The end-to-end delay is higher when AODV is run with the Trusted Scheme as compared to normal AODV. The higher delay is due to the overhead caused by the trusted scheme. The AODV performance is adversely affected in term of end-to-end delay but as a result the network is more secure. As shown in figure 6.9, it is 0.073 for normal and 0.085 for trusted AODV.

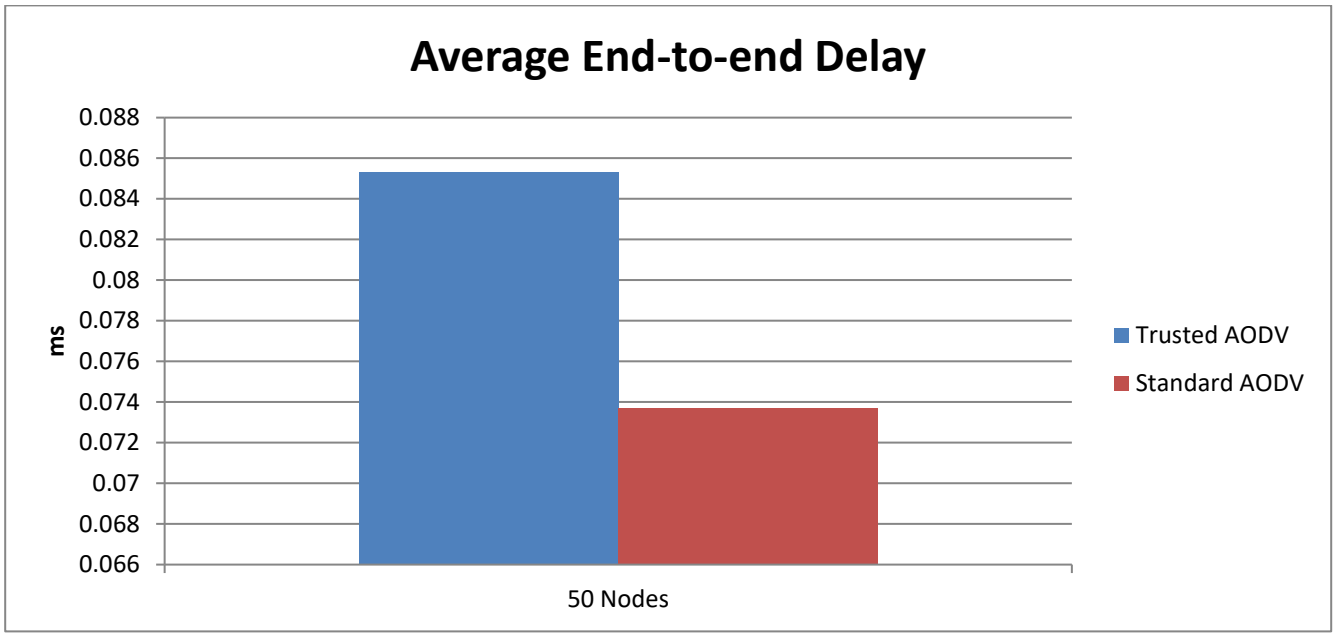


Figure 6.9 Ave End-to-End Delay

- **Packet Delivery Ratio**

The results for packet delivery ratio listed in figure 6.10, shows that trusted scheme is preventing malicious nodes from dropping packets. On the contrary, it can be observed that the packet delivery ratio in standard AODV is comparatively low, as there no measures in place, to counter the threats of malicious nodes attack.

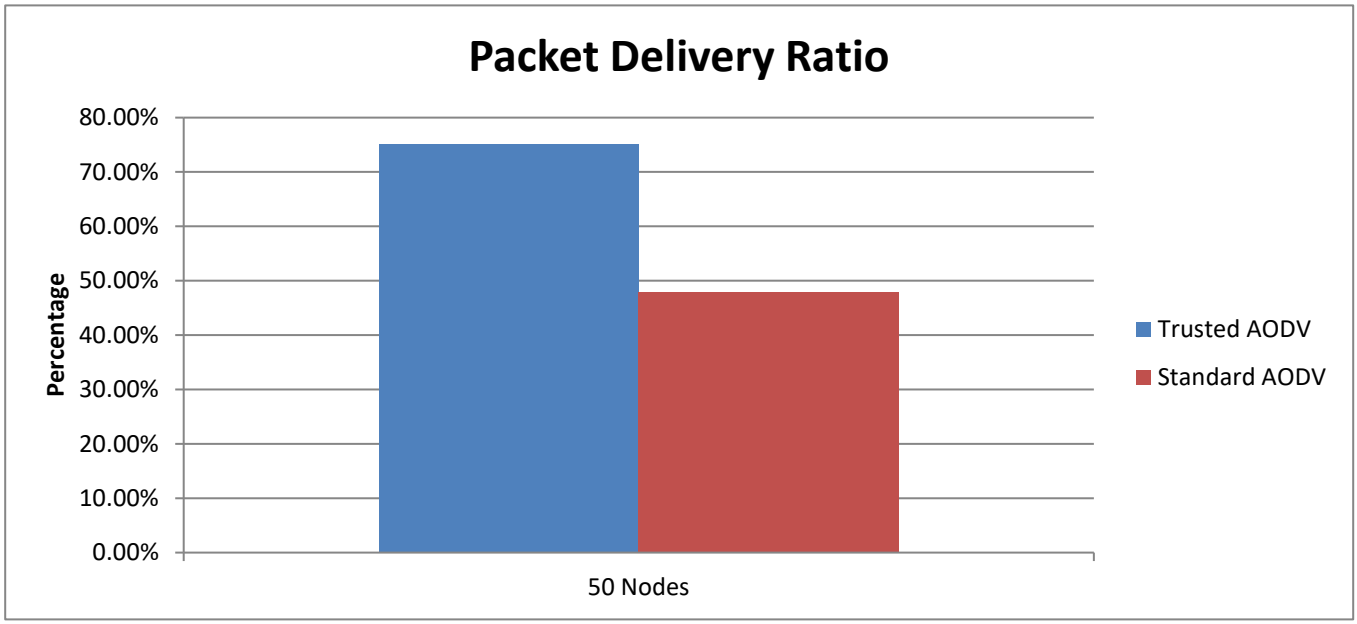


Figure 6.10 Packet Delivery Ratio

6.4 Static versus Dynamic Trust

As the detailed numerical analysis presented in chapter 4, demonstrated how the dynamic trust scheme can result in less false positives. In this section, some of the results obtained during simulations by varying number of nodes using static trust model and the proposed dynamic trust model are presented. Both schemes are run side by side and the throughput and packet delivery ratio is recorded to analyse and compare their performance. Both models are evaluated and the results are presented to demonstrate the following

- That the dynamic trust scheme not only results in less false positives but also shows better performance.

- To draw a comparison between the static and dynamic trust model, to prove that the trust calculated dynamically using the proposed trust algorithm can result in high throughput.

The graph below shows a higher throughput for dynamic model then static model. This also shows that although the static trust model is defending against malicious nodes in the network however, using a dynamic model can increase the detection rate, reduce false positives. The results also reflect the claim in form of a high throughput in figure 6.11.

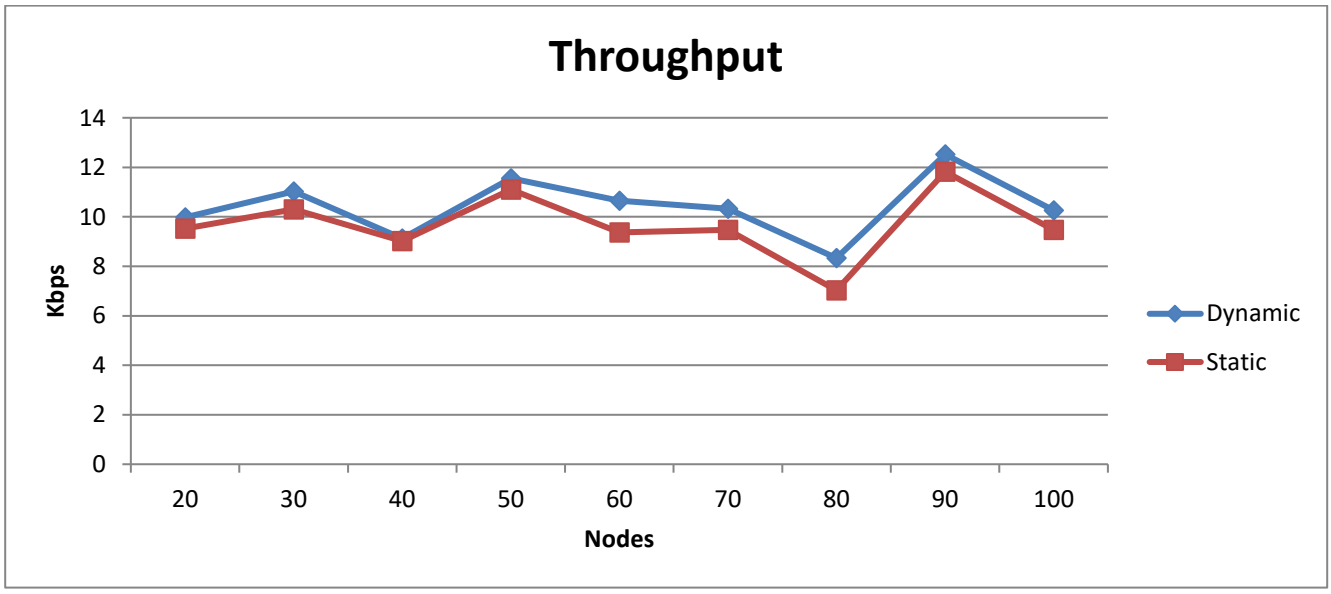


Figure 6.11 Static verses Dynamic Trust Model Throughput

Similarly, figure 6.12 presents the packet delivery ratio between static and dynamic model using varying number of node. Again the results show a higher percentage of packet delivered using dynamic model as compared to the static model.

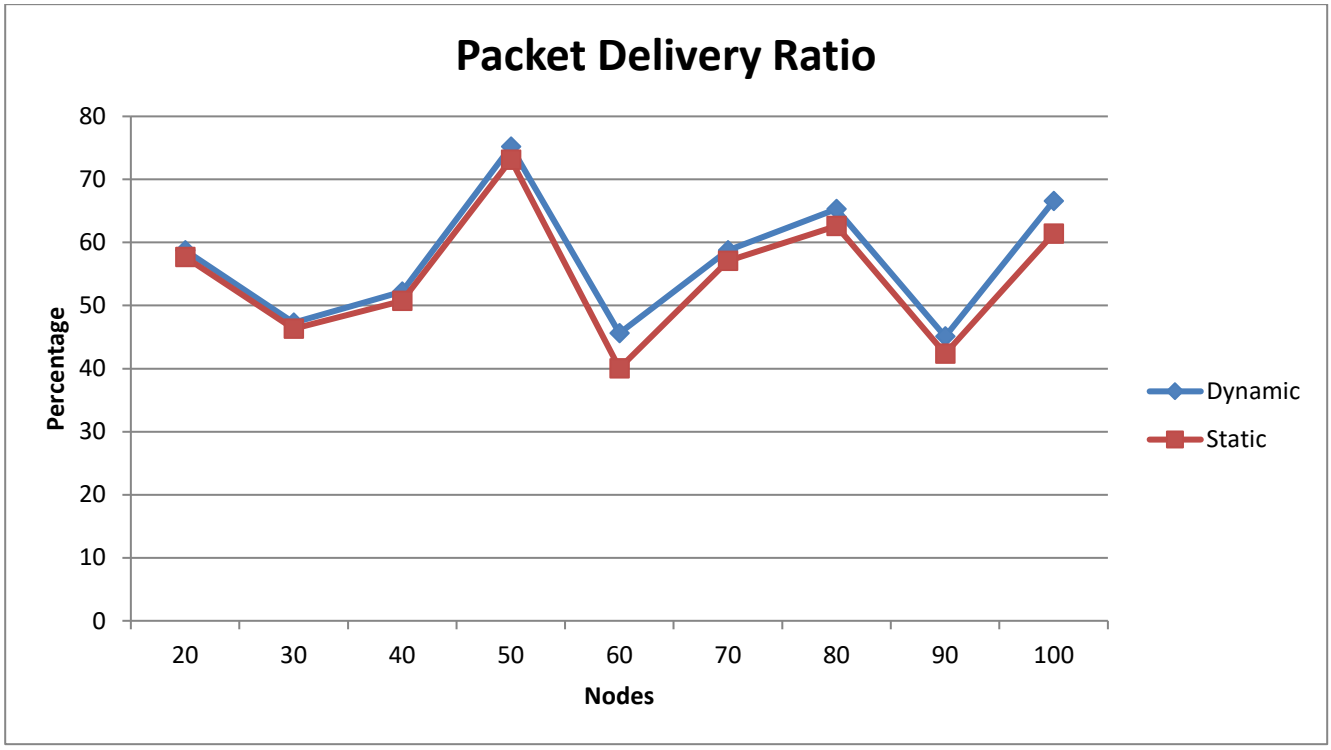


Figure 6.12 Static verses Dynamic Trust Model Packet Delivery Ratio

Although the scheme can be successfully implemented to give the desire results however, there are certain limitations in terms of the level of security it can provide. The scheme can defend against Blackhole and Greyhole attacks but it can provide limited protection against collaborative attacks such as Byzantine and Jellyfish attacks.

The reason for selection to use Elliptic Curve Diffie Hellman schemes is explained in the chapter 4. No end-to-end security can be fully achieved without using cryptographic algorithm. The use and implementation of cryptographic techniques has always been challenging in MANET. It is one of the key areas that have been extensively explored and many solutions have been proposed and the research continues till this time. ECDH is selected for data encryption as it suits MANET due to its efficiency and provides more security with smaller key size.

6.5 Phase-two Performance Evaluation

NS2.33 simulator is used to simulate phase-one of the proposed scheme is presented in this section along with phase-two, which represents the performance overhead of Elliptic Curve Diffie-Hellman key exchange is also presented. The merits and reason for the selection of the metrics are discussed in detail in the above section. The calculations when ECDH is implemented are presented in this section as well.

The metrics presented to test the performance of the proposed scheme is based on data packets and do not include the control and security message. Thus, the statistics represent only phase-one of the scheme, hence we explore overhead caused when ECDH is implemented in phase-two.

According to (Wong, Ramamurthy and Zou, 2006) and given the steps shown in section 4, there are 9 steps required to generate and exchange keys for ECDH algorithm. This means additional 9 packets are needed to the total number of packets in phase-one. The first step is peer nodes generate random number followed by generating their private and public keys. In the next step, each peer on the receipt

of public key from its corresponding peer computes shared key. Therefore, there is no significant effect on the throughput when ECDH is implemented.

Lastly, the overhead caused by ECDH, if EC key size of 160 bits is used between peer nodes, this key length gives us 80 bits of equivalent symmetric key security. Only one cycle is needed to exchange the keys as an encryption key of 80 bits will produce 160 bits (key size), where each data packet size is 512 bits.

This implies that the ECDH can be applied as key exchange algorithm to generate a shared secret key. The secret key produced is symmetric type and can be used to encrypt data between communicating peer nodes. The advance and more secure version of data encryption scheme such as AES and DES, explained in the literature review section, can be used to ensure secure data exchange. Data encryption can prevent against attacks such as Rushing (Rifquddin and Sukiswo, 2015) and Wormhole (Anju and Sminesh, 2014) that are common in MANET.

6.6 Conclusion

The performance Standard AODV is compared with the proposed trust scheme under varying conditions such as number of nodes, mobility and dimensions. The dynamic threshold scheme combined with mutual trust authentication and ECDH scheme resulted in some of the most promising systems. This is due to the reason that dynamic trust scheme makes the system self-configuring and robust, while mutual authentication avoids the difficult challenge of authentication without any centralised mechanism. And lastly, provide efficient key generation and exchange without using

expensive public key infrastructure and key distribution techniques. An important result of this thesis is that the area of trust based security in MANETs has been explored, where a methodology was adopted and modified to calculate trust dynamically that offers a platform to authenticate nodes, encrypt data and as a result provides an end-to-end security solution.

CHAPTER 7

CONCLUSION

In this chapter, the conclusion of the thesis, brief summary of the research findings, critical analysis and future research is presented. The dynamic trust based scheme is the novel way of defining trust in MANET by using various parameters most relevant to MANET conditions, to segregate malicious nodes. The novel method of mutual trust scheme provides the authentication between source and destination nodes for confidential data communication.

The dynamic nature of MANET makes the use of conventional security schemes such as secret and public key cryptography more challenging and prevents the design of one-size-fits-all solution (Kumar, and Mishra, 2012). Due to the lack of infrastructure in MANET, only the nodes in the network can be relied upon to observe and judge whether a particular node is trusted or compromised. Therefore, the scheme proposed in this research is robust and encompasses various aspects of security. To solve the security problem, specification are kept as general as possible and it is ensured that the scheme can be adapted to accommodate other protocols and it can be used for various applications. Hence, the scheme can be applied to a large number of applications using MANET.

The scheme not only allows the nodes to authenticate its-self but the security is implemented throughout the network and is scaled as the network grows through the efficient trust based scheme. This signifies that not only the security of individual nodes is important but the security of the network as whole is of paramount importance as well. The implications of multilayer security on the nodes in terms of performance is a factor that needs to be taken into consideration as the nodes have other

limitations as well and cannot be ignored. These limitations can change the way nodes behave and can massively impact the behaviour in an adverse way (Zhao et al 2012).

Security is essential for a number of reasons to keep valuables safe. We have started to see more security measures becoming mandatory at various levels in IT. In MANET, the security of nodes and securing communication between nodes is equally important, to safeguard the data (Perrig et al, 2002). MANET is the future of communication and can be used in a number of important applications. Its ability of nodes to form Adhoc network in the absence of any infrastructure, makes it popular research area. One of the biggest challenges faced by MANET is security (Hu et al 2002). MANET protocols were designed without taking security into considerations. Due to its distinct features such as lack of conventional security infrastructure, no centralised mechanism, constantly changing topology and open wireless medium makes MANETs more vulnerable to attacks. Therefore, unlike their wired counterpart, a different approach is needed to secure MANET (Eladi et al 2013).

7.1 Thesis Summary

The main goal of this thesis has been to design and implement a security algorithm using trust based schemes and to achieve the objectives of this research, that are dynamically implementing trusted algorithm, identifying trusted and malicious nodes, authenticate peer communicating nodes and data encryption between end nodes. The work carried out in this research describes the specific security issue faced by Adhoc networks and justification for having the trust based scheme that calculates the

trust dynamically, as a possible solution. This was proposed as a new concept after analysing several trust based schemes and comparing them.

In the initial stage of this research, MANET characteristics, protocols and other related features were thoroughly reviewed. In the second stage of the literature review, the network security was studied in detail (Juwad and Al-Raweshidy, 2008; Hu, John and Perrig, 2002; Perrig et al, 2002 ; Hu, Perrig and Johnson, 2005). This included general network and communication security and MANET specific security. Main part of the network and communication security included cryptographic techniques, trust based schemes, secure routing protocols and common security goals (Hinds et al, 2012). The common types of security attacks and the attacks that were more specific to MANET were analysed. Finally, part of the literature review was to conduct a detailed analysis of the completed work to secure MANET, using various trust and cryptographic techniques (Zhao et al 2012).

A number of trust based schemes have been designed that allow the detection of malicious nodes in MANET and mitigate various forms of security attacks specific to MANET. Majority of the trust schemes have used static and pre-determined values to calculate trust threshold in the network nodes (Balakrishnan, Deng, Varshney, 2005; Khan et al, 2015: Buttyan and Hubaux, 2000; Zhong et al 2003; Buchegger and Le-Boudec, 2002; Jhaveri, 2013). Their approach to trust calculation was that nodes listening in promiscuous mode to all packets sent and received by neighbours and then compared that trust against the static threshold. Although, this is the most common and popular approach but there could be a number of factors that can cause dropping of the packets by various nodes. These factors reflect a true MANET environment. It is important, that these factors are taken into consideration while calculating trust of nodes. First goal of this research was to identify those factors and apply them to

dynamically calculate the trust threshold. Secondly, using the trust as a framework, the proposed scheme applied a second layer of security by presenting a novel way of mutual authentication between peer nodes.

In this research, AODV routing protocol has been used to implement the dynamic trusted algorithm. A novel method has been introduced by adopting the research in (Khan et al 2015; Buttyan and Hubaux, 2000; Zhong et al 2003; Buchegger and Le-Boudec, 2002 and Jhaveri, 2013), in calculating the trust scheme. The first stage of the trust calculation involved working out the trust value of each node. All nodes listening to the neighbours nodes and work out their trust value. Once, every node has the trust value, a threshold value is needed, to work-out the trust of each node. The common approach that has been in used in the previous research is to use, a predetermined static value. A novel approach of calculating a dynamic trust threshold value was adopted that included parameters, most relevant to MANET environment. The parameters used are one-hop neighbours, two-hop neighbours, trust, energy and mobility. The dynamic trust threshold value of a node is compared with its actual trust value observed by neighbour node to calculate final trust. These parameters play a very crucial role in, how the nodes behave in MANET. The reason for the selection of these parameters and the justification for using dynamic scheme are discussed in detailed in chapter 3.

The trust framework is paramount for identifying malicious nodes in the network therefore; the dynamic trust scheme is used to implement that framework. Once the trust is established the second layer of security is invoked. This is referred to as mutual trust authentication, whereby peer communicating nodes request trust from the neighbour nodes. This step is invoked by source and destination nodes, whenever they initiate communication for the first time. According to this step,

when source receives a RREP, it requests trust values of destination node from the destination's neighbour nodes before initiating any data communication. The same process is repeated at the destination node when it receives a RREQ. Destination node also send a request to source's neighbour nodes, to get the trust value of source node, the trust value that is already calculated through the dynamic trust scheme.

7.2 Proposed Scheme Evaluation

There have been various trust based schemes designed to secure MANET. Some of the schemes have been discussed in detail in the literature review section, majority of them have been designed not taking MANET dynamic nature into account. The dynamic approach has been adopted to calculate the trust threshold mainly, to distinguish between misbehaving and malicious node. A misbehaving node in MANET is where the node is partially or not participating in routing process at all. Normally, the compromised node can engage in many types of malicious activities (Jhaveri, 2012). For the simulation, this activity is recorded in the form of dropping packets otherwise these nodes will not be detected. The trust value is mostly dependent upon neighbour observations in promiscuous mode and if no data is forwarded by the corresponding node, the neighbour marks it down. But this behaviour of not forwarding data could be for a number of reasons. It could be either due to MANET specific environment or could be a malicious node whose security is compromised. As each node is working as router than wrongly excluding a node due to its misbehaving can result in false positive and affect network connectivity (Khan et al 2015). If such a node happens to be a destination or source node than it will have even greater implications, as it could result in the breakdown of the whole communication

process. The dynamic trust calculation takes the combined static and dynamic approach to make a clear distinction between malicious and misbehaving node.

The parameters used to calculate trust dynamically are selected in-line with the MANET specific conditions. The parameters include one-hop neighbours, two-hop neighbours, trust, mobility and energy. The energy and mobility reflects the dynamic nature of MANET, as the nodes are mobile and the power resource is constantly depleting. The one-hop and two-hop is crucial information for trust calculation. This is also a constantly changing resource and plays an important role in determining the trust of a node dynamically. The details of all the metric and how they are calculated are presented in chapter 3 and 4 respectively. The analytical model presents the equations for calculating average trust, average trust threshold and dynamic trust threshold values. The dynamic trust threshold is derived by taking all the metrics into account and applying weights to each metric. The weights applied can be changed to fine tune the dynamic trust threshold depending on the applications.

The scheme is implemented in the MANET environment with no predetermined trust hence all nodes are treated as having no trust at all (Nikander, Ed. P., Kempf, J. and Nordmark, E. 2004). The scheme is compatible with any MANET routing protocol and can be implemented in the network using routing protocol other than AODV. The reason for the selection of AODV as a routing protocol is, it's being on-demand (Morshed et al 2010). In other words, it is a reactive routing protocol. There are two main types of Adhoc routing protocols called reactive and proactive. Proactive protocols are topology-based protocols that have high overheads and consume relatively more energy as compared to reactive protocols. This is due to the fact that the routing table needs to be deployed beforehand, that causes routing overheads and consumes more energy. The implication of using proactive protocol for this

research would have been that the node consuming energy at a faster rate. Energy being one of the metric for calculating the dynamic trust threshold, therefore, a protocol that consumes less power has been the primary choice (Mohandas, Silas and Sam, 2013). The AODV has been used for testing, as it's a reactive protocol it causes less overhead due to no routing tables are deployed beforehand. Hence this helped in reducing routing overhead and as a result it reduces the energy consumption and increasing the network lifetime (Gupta, and Sexena, 2010).

Determining the trust level of new nodes and allowing them to become part of the network, so that, they can take part in routing and communication, is still a challenging issue. A novel method for authentication is proposed that enabled nodes to prove their identity prior to exchanging data with each other. In the proposed security scheme, the trust based scheme and the trust values are utilized for authentication and DHEC is used for key exchange and data encryption (Wang, Ramamurthy and Zou, 2006).

These schemes have some distinctive characteristics that support MANET decentralized and resource constraint environment. The proposed scheme is developed assuming no prior trust and association between nodes. This represents pure MANET (Nikander, Ed. P., Kempf, J. and Nordmark, E. 2004) and is one of the fundamental characteristics of MANET. Given this feature of MANET, any conventional security method is hard to implement. For instance, the use of various cryptographic techniques, such as PKI cannot be applied without having some prior security in place. This would be that nodes have some form of prior trust; identity verification, cryptographic keys storage, nodes authentication and authorization, KDC or CA allocation and access are some of the fundamental security steps needed or considered before deploying nodes to afford a level of security. These

measures can address some of the security issues but does not represent pure MANET (Nikander, Ed. P., Kempf, J. and Nordmark, E. 2004). The proposed dynamic trust based scheme is implemented using pure MANET and this supports the primary goal of implementing security in MANET that has no prior security.

In this research, an efficient way to support existing nodes leaving and joining the network has also been proposed. This is due to the fact that the nodes are mobile and the topology is constantly changing so there is a provision for new nodes joining the network and go through the trust and authentication process as shown in figure 6.1 (Cho, Swami and Chen, 2011).

The proposed trust scheme achieves this by constantly updating the trust values regardless of whether it's a new or existing node. Additionally, it is self-organizing as the trust values are evaluated against a dynamic value and allow the network to adjust the trust value of nodes according to the network specific conditions.

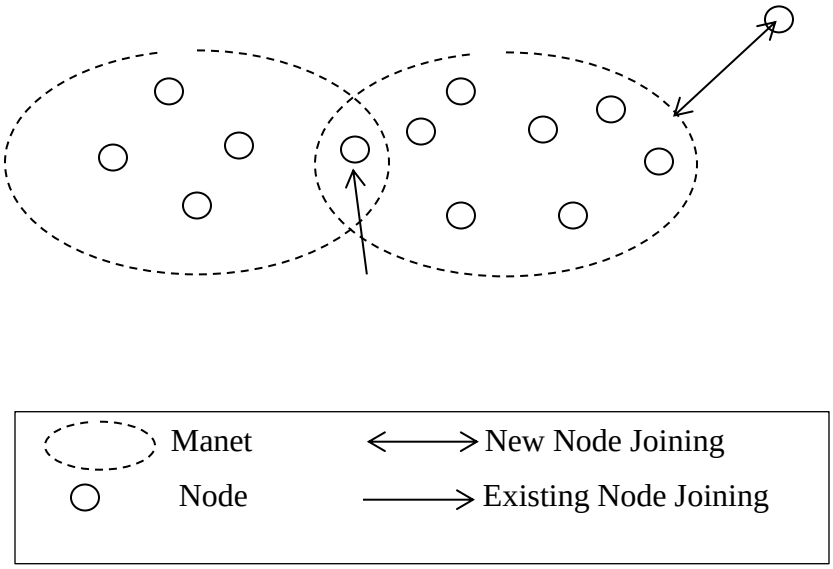


Figure 7.1 New and Existing Node Joining the Network

The mutual authentication scheme presented in the proposed trust model is a key contribution. In MANET, as other methods of authentication discussed in chapter 2, are hard to achieve but there is a need for authentication and it is paramount to obtain a degree of authentication before peer nodes initiate any data exchange (Zhao, et al 2012). The dynamic trust provides a platform, where there is no pre-established trust, to enable mutual trust authentication in the nodes. The trust from network initialisation to mutual authentication is shown in figure 7.2.

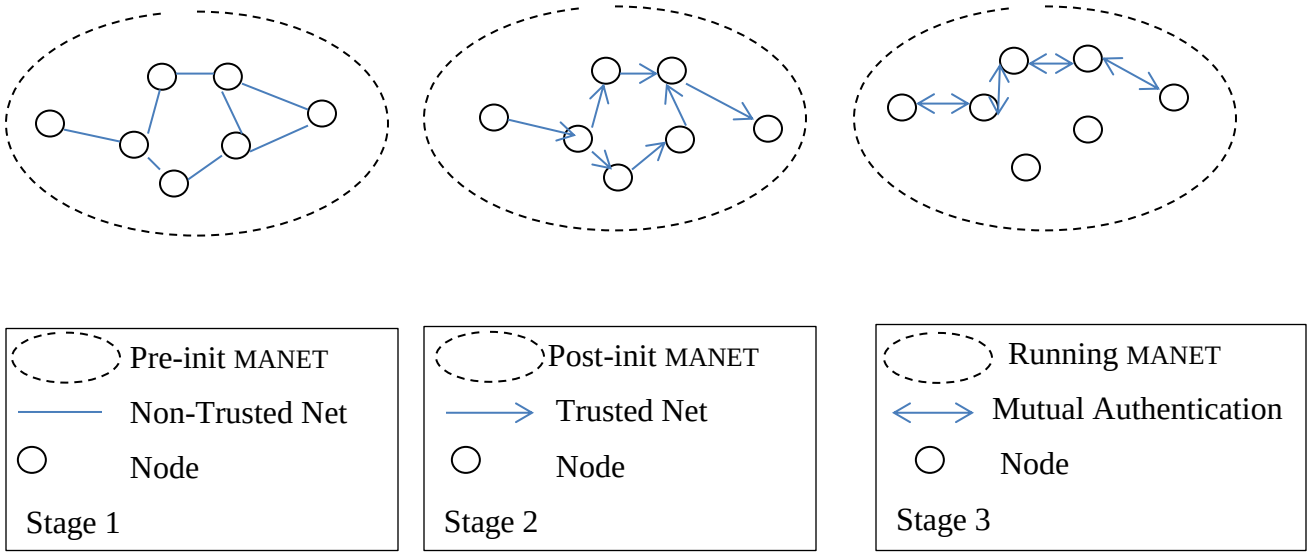


Figure 7.2 MANET Trust Stages

The authentication scheme uses the underline AODV protocol to request the trust values from neighbour nodes of the corresponding peer nodes. This is achieved at the expense of routing overhead. There is a trade-off between achieving the security goal and network performance. The scheme does receive a performance hit by implementing the security but this performance overhead does not come as a surprise. The level of routing overhead can be argued and there is always room for improvement in terms of optimising the overall scheme but the primary goal had been security. We believe that although the scheme has performance overheads but it can be implemented without severely crippling the whole network.

The scheme provides an efficient and secure mechanism for the distribution of keys between nodes over an insecure channel. The scheme offers encryption of data communication using keys that are

freshly generated by the communicating nodes. The keys are valid for the single session and fresh keys are generated for every new session initiation. This ensures that all the nodes, whether existing or new will undergo the process of trust evaluation and authentication.

The scheme also deals with the inherent key escrow property in the context of MANET. We believe that our novel concept addresses the issue of Key Escrow, because no certificate authorities (CA) are used to generate and distribute keys. The use of CA has inherent issue when it comes to MANET. The primary role of CA is to deal with PKI infrastructure and cryptographic keys management such as keys generation, distribution and revocation makes it a key escrow (Hinds et al, 2012). In MANET, CA is faced with physical security, malicious attacks, bandwidth consumption, energy, availability, access control, mobility and no centralized mechanism. Availability is referred to when node acting as CA and cannot be accessed due to network partitioning, out of range, low power or compromised for any other reason where it is unable to render services to a legitimate node's request (Zhao et al 2012).

One of the properties of the proposed trust based scheme is that it is self-organizing. From pre-network initialisation stage where there is no trust to when the network is fully initialised, something we referred to post-network initialisation in this thesis. At this stage, the trust algorithm is invoked and trust is established among nodes. This process is completely self-configuring. This is an important aspect and as the life of the network increases the scheme gets refined.

7.3 Contributions

The research deals with the question of calculating trust, using pure (Nikander, Ed. P., Kempf, J. and Nordmark, E. 2004) MANET features and how trust can be used to authenticate communicating peer nodes. This research proves, by demonstrating that the dynamic trust algorithm in MANET can be built unlike static trust, to identify malicious nodes. The proposed trust scheme not only reduces false positives but also has a higher throughput and packet delivery ratio.

A novel way of authentication, referred to as mutual authentication, has been demonstrated and proved in this research. According to this scheme, the communicating nodes validated each other by requesting trust from their corresponding peer's neighbour nodes. The trust values calculated as result of the dynamic scheme is used for authentication.

7.4 Research Challenges

There were various challenges faced during different stages while carrying out this research. The first stage was at literature review, when cryptographic and security schemes were analysed, it was realised that, a very deep level of understanding and mathematical background is needed to implement these algorithms. Learning these algorithms and achieving the required level of understanding was a daunting task. A lot of learning and practice was put in place, to understand and implement various cryptographic algorithms (Wang, Ramamurthy and Zou, 2006).

NS2 simulator is used to implement and test the scheme. It is an open source and Linux based simulation tool however, it can be run in Windows environment using Cygwin. Numerous operational, installation and performance issues were faced during the implementation stage of the research therefore; the idea of using Cygwin in Windows environment was eventually dropped. It was for the first time that Linux OS and Linux based applications were used. It was a huge learning curve to work on different versions of Linux operating systems to run the simulation tool.

Modifying AODV code to implement the trust algorithm was the most challenging out of all the challenges faced during the research and it was very time consuming as well. There were number of problems that were ran into during designing, learning, understand, debugging and troubleshooting C++ code. NS2 is written in C++ and TCL programming language used as interpretation language. It is used to write simulation script in NS2. Every instruction in TCL is a command for the simulation program (Morshed et al 2010). There was a dual challenge of learning TCL and C++ to understand and implement our algorithm in NS2.

7.5 Future Work

The detection of malicious behaviour is a challenging task (Annarasi and Sevanesh, 2014). There could be other types of metrics used to detect these nodes such as count the number of generated packets for instance route requests RREQ. Another metric that can be used is to check the response time of nodes when they receive packets. Lastly, an acknowledgment method can be used whenever a packet is sent. The receiving nodes always send an acknowledgment to the sender when a packet is received as shown

in figure 7.3. When two neighbour nodes communicate, the receiving node send a reply in the form of an acknowledgement to the sender node to confirm it has received the packet (Botkar and Chaudry, 2011).

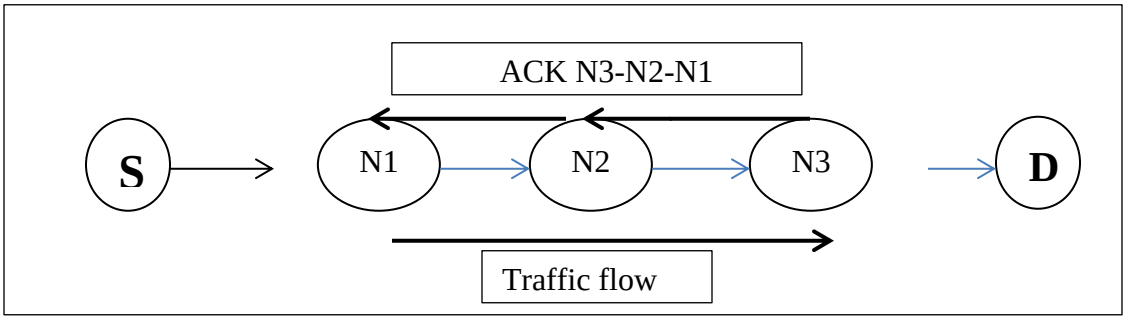


Figure 7.3 Acknowledgement Method

Part of the future work would be to enhance the scheme to protect against collaborative attacks. These types of attacks are called Byzantine attacks that are against the routing protocols, in which two or more colluding routers attempt to disrupt routing operation by modifying, fabricating or dropping packets (Yu et al 2001).

The mutual authentication scheme proposed in this thesis can be secured using pairwise shared keys authentication as well. This is type of cryptographic algorithm where the keys can be deployed at post-initialisation stage. The scheme can provide additional security in terms of authentication.

The use of Intrusion Detection and Prevention System commonly known as IPS and IDS, to protect mobile nodes against any malware or other types of active attacks could be implemented alongside the proposed scheme as future work (Botkar, S. and Chaudry, S. R. 2011). This would be to run IDS on each node to detect signature of the known attacks or anomaly detection system to look for any unusual behaviours. The combined use of trust based security and IDS schemes in MANET can be further explored as they are suitable for MANET environment (Al-Roubaiey et al 2010).

As the security threat landscape is getting more advanced and complicated, a single security tool does not provide security against all types of threats. Depending on the type of applications and the conditions in which MANET is implemented, identifying the threat model will define what type of security measures need to be enforced (Johnson et al 2011). Therefore, it's important to understand the threat landscape and devise a security plan and as referred to, within the security community, as applying a right tool from the security tool box.

REFERENCES

1. Al-Roubaiey, A., Sheltami, T., Mahmoud, A., Shakshuki, E., Moufta-King-Fahd, H. (2010) 'AACK:Adaptive Acknowledgment intrusion detection for MANET with node detection enhancement', *IEEE International Conference on Advanced Information Networking and Applications*

2. Anju, J. and Sminesh, C. N. (2014) 'An Improved Clustering-based Approach for Wormhole attack detection in MANET', *IEEE, 3rd International Conference on Eco-friendly Computing and communication systems.*

3. Asmuth, C. and Bloom J. (1983) 'A modular approach to key safeguarding' *IEEE transaction on Information Theory,* 29(2):pp 208-210.

4. Botkar, S. and Chaudry, S. R. (2011) ' An Enhanced Intrusion detection System using Adaptive Acknowledgment based Algorithm', *IEEE*.

5. Balakrishnan, K., Deng, J., Varshney, P. K. (2005) 'TWOACK: Preventing Selfishness in Mobile AdHoc Networks', *IEEE communication society*.

6. Buttyan, L. and Hubaux, J. P. (2000) 'Enforcing Service Availability in Mobile Ad-Hoc WANs', *Proc. MobiHoc*.

7. Buchegger, S. and Le-Boudec, J. Y. (2002) ‘Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Fairness in Dynamic Ad-Hoc Networks’, *Proc. MobiHoc*.

8. Buttyan, L. and Hubaux, J. P. (2000) ‘Enforcing Service Availability in Mobile Ad-Hoc WANs’, *Proc. MobiHoc*.

9. Buchegger, S. and Le-Boudec, J. Y. (2002) ‘Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Fairness in Dynamic Ad-Hoc Networks’, *Proc. MobiHoc*.

10. Cho, J., Swami, A. and Chen, I. (2011) ‘A survey on Trust management for Mobile Adhoc Networks’, *IEEE communications survey and tutorials*, vol 13, NO.4 Fourth Quarter.

11. Whitfield, D. and Martin, E. Hellman (1976) ‘New directions in cryptography [J]’, *IEEE.* Transactions on Information Theory.

12. Boneh, D. and Franklin, M. (2001) ‘Identity-based Encryption from the Weil Pairings’, *In Proceedings of CRYPTO 2001, Springer-Verlag*, LNCS 2139, 213-229.

13. Desmedt, Y. (1997) ‘Some recent research aspects of threshold cryptography’, *Proc of ISW’97 1st International Information Security Workshop, Springer-Verlag,* vol.1196 of LNCS, pp.158-173.

14. Desmedt, Y. and Frankel, Y. (1990) 'Threshold cryptosystems', *Springer Verlag, Proc of CRYPTO'89,* volume 435 of LNCS, pp.307-315.

15. Morshed, M. D., Ko, I. S., Lim, D., Rahman, M. H., Mazumder, R. M. R. and Ghosh, J. (2010), 'Performance evaluation of DSDV and AODV routing protocols in Mobile Ad-hoc Networks', *IEEE, 4th international conference on new trends in information science service science,* pp.399-403.

16. Kahn, R.E., Gronemeyer, S.A., Burch¯el, J. and Kunzelman, R.C (1978) 'Advances in Packet Radio Technology', *Proceedings of the IEEE,* vol. 66, no. 11, pp. 1468-1496.

17. Kumar, M. and Mishra, S. (2012) 'An Overview of MANET: History, Challenges and Applications', *Indian Journal of Computer Science and Engineering (IJCSE),* Vol. 3 No. 1.

18. XU, B., Hischke, S. and Walke, B. (2003) 'The Role of Ad hoc Networking in Future Wireless Communications', *IEEE, Proceedings of ICCT.*

19. Perkins, C., Belding-Royer, E. and Das, S. (2003) 'Ad hoc on-demand distance vector (AODV) routing', *RFC 3561,* Available at: http://www.ietf.org/rfc/rfc3561.txt

20. Clausen, T. and Jacquet, P. (2008) 'Optimized Link State Routing Protocol (OLSR)', *RFC 3626 (Experimental),* October 2003. Available at: http://www.ietf.org/rfc/rfc3626.txt.

21. Johnson, D., Hu, Y. and Maltz, D. (2008) 'The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*', RFC 4728 (Experimental),* February 2007. Available at: http://www.ietf.org/rfc/rfc4728.txt.

22. Hinds, A., Sotiriadis, S., Bessis, N. and Antonopoulos, N. (2012) 'Performance Evaluation of Security Algorithm for AODV MANET Routing Protocol', *IEEE, Third International Conference on Emerging Intelligent Data and Web Technologies*.

23. Juwad, M.F., and Al-Raweshidy, H. S. (2008) 'Experimental Performance Comparisons between SAODV and AODV', *IEEE, Second Asia International Conference on Modelling & Simulation*.

24. Zapata, M. and Asokan, N. (2002) 'Securing Ad hoc Routing protocols', *in proc. Of ACM Workshop on Wireless Security (WiSe),* Atlanta, GA.

25. ARAN '(A secure Routing Protocol for Ad hoc Networks)', Implementation, Available at: http://signl.cs.umass.edu/arand/

26. Hu, Y. C., Perrig, A. and Johnson, D. B. (2005) 'Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. Wireless Networks', 11(1-2), pp. 21–38.

27. Yi, S., Naldurg, P. and Kravets, R. (2005) 'Security - Aware Ad hoc Routing for Wireless Networks', *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, Pages 299-302.

28. Carter, S. and Yasinsac, A. (2002) 'Secure Position Aided Ad Hoc Routing', *Proc. IASTED Int'l Conf. Comm. and Computer Networks (CCN '02)*, pp. 329- 334.

29. Hu, Y., John, D.B. and Perrig, A. (2002) 'SEAD: Secure Efficient Distance Vector Routing For Mobile Ad Hoc Networks', *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)*.

30. Perrig, A., Canetti, R., Tygar, J. D. and Song, D. (2002) 'The TESLA Broadcast Authentication Protocol', *RSA Laboratories*, vol. 5, no. 2.

31. Gupta, M. and Kumar, S. (2015), 'PERFORMANCE EVALUATION OF DSR, AODV AND DSDV ROUTING PROTOCOL FOR WIRELESS ADHOC NETWORK', *IEEE International Conference on Computational Intelligence & Communication Technology*.

32. Singh, K. and Verma, A. K. (2015) 'Experimental Analysis of AODV, DSDV and OLSR Routing Protocol', *IEEE*.

33. Park, V. and Corson, S. (2001) "Temporally-Ordered Routing Algorithm (TORA) Version 1" , *RFC 2026*. Available at: https://www.ietf.org/proceedings/52/I-D/draft-ietf-manet-tora-spec-04.txt .

34. Murthy, S. and Garcia-Luna-Aceves, J.J. (1996) 'An Efficient Routing Protocol for Wireless Networks', *ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks*, pp. 183-97. Available at: http://www.ics.uci.edu/~atm/adhoc/paper-collection/aceves-routing-winet.pdf

35. Chai-Keong Toh, (1996) 'A novel distributed routing protocol to support Ad hoc mobile computing', *Proc. IEEE 15th Annual Int'l. Phoenix Conf. Comp. and Communication Mar. 1996*, pp. 480-86. Available at: http://www.ics.uci.edu/~atm/adhoc/paper-collection/toh-distributed-routing-ipccc96.pdf

36. Hass, Z. J., Pealman, M.R. and Samar, P. (2003) 'The Zone Routing Protocol (ZRP) for Ad Hoc Network', Available at: https://www.ietf.org/proceedings/55/I-D/draft-ietf-manet-zone-zrp-04.txt.

37. Zou, X., Ramamurthy, B., and Magliveras, S. S. (2005) '*Secure Group Communications Over Data Networks*', *Springer*.

38. Miller, V. S. (1986) ‘Use of elliptic curves in cryptography’, *Springer-Verlag, In Lecture notes in computer sciences; 218 on Advances in cryptology—CRYPTO 85. New York, NY, USA: New York, Inc*, pp. 417–426.

39. Koblitz, N. (1987) ‘Elliptic curve cryptosystems’, *Mathematics of Computation, vol.* 48, pp. 203–209.

40. Wang, Y., Ramamurthy, B. and Zou, X. (2006) ‘The Performance of Elliptic Curve Based Group Diffie-Hellman Protocols for Secure Group Communication over Ad Hoc Networks’, *IEEE*.

41. Hass, Z. J., Pealman, M.R. and Samar, P. (2003) ‘The Zone Routing Protocol (ZRP) for Ad Hoc Networks’, Available at: https://www.ietf.org/proceedings/55/I-D/draft-ietf-manet-zone-zrp-04.txt

42. Jhaveri, R. H. (2012) ‘MR-AODV: A Solution to Mitigate Blackhole and Grayhole Attacks in AODV Based MANET’, *IEEE, Third International Conference on Advanced ohnson, Computing & Communication Technologies.*

43. Yu, M. and Su, W. (2009) ‘A Secure Routing Protocol Against Byzantine Attacks for MANETs in Adversarial Environments’, *IEEE TRANSACTIONS ON VEHICULAR “Elliptic TECHNOLOGY,* VOL. 58, NO. 1.

44. Rifquddin, M. R. and Sukiswo. (2015) ‘Performance of AOMDV Routing Protocol Under Rushing and Flooding Attacks in MANET’, *IEEE, Proc. Of 2nd Int. Conference on Information Technology, Computer and Electrical Engineering (ICITACEE),* Indonesia, Oct l6-l8th

45. Davis, R. M. (1978) ‘The Data Encryption Standard in Perspective’, *IEEE*.

46. Nie, T. and Zhang, T. (2009) ‘A Study of DES and Blowfish Encryption Algorithm’, *IEEE. Routing A Project of Shandong Province Higher Educational Science and Technology Program* (No. J09LG10) 978-1-4244-4547.

47. Sathiamoorthy, J., Ramakrishnan, B. and Usha, M. (2015) ‘A Reliable and Secure Data Transmission in CEAACK MANETs using Distinct Dynamic Key with Classified Digital Signature Cryptographic Algorithm’, 978-1-4799-7623-2/15.

48. Sarkar, S., Kisku, B., Misra, S. and Obaidat, M. S. (2009) ‘Chinese Remainder Theorem-Based RSA-Threshold Cryptography in MANET Using Verifiable Secret Sharing Scheme’, *IEEE International Conference on Wireless and Mobile Computing, Networking and communications.*

49. Lu, J., Wei, Y., Fouque P. A. and Kim, J. (2012) ‘Cryptanalysis of reduced versions of the Camellia block cipher’, *IEEE, IET Inf. Secure.*, Vol. 6, Iss. 3, pp. 228 –238

50. Rizvi, S. A. M., Hussain, S. Z. and Wadhwa, N. (2011) 'Performance Analysis of AES and TwoFish Encryption Schemes', *IEEE, International Conference on Communication Systems and Network Technologies.*

51. Ragab, A. H.M., Ismail N. A. and Farag Allah, O. S. (2001) 'An Efficient Message Digest Algorithm (MD) One-way For Data Security', *IEEE*, Catalogue No. 01 CH37239 0-7803-7IOI-I/OI.

52. Thulasimani, L. and Madheswaran M. (2009) 'Security and Robustness Enhancement of Existing Hash Algorithm', *IEEE International Conference on Signal Processing Systems.*

53. Annarasi R. S. and Sevanesh, S. (2014) 'A Secure Intrusion Detection System for MANETs', *IEEE International Conference on Advanced Communication Control and Computing Teclmologies (ICACCCT)*

54. Johnson, A., Syverson, P., Dingledine, R. and Matthewson, N. (2011) 'Trust-based Anonymous Communication: Adversary Models and Routing Algorithms', Available at: http://freehaven.net/~arma/anonymity-trust-ccs2011.pdf

55. Govindan, K. and Mohapatra, P. (2010) 'Trust Computations and Trust Dynamics in Mobile Adhoc Networks: A Survey', *IEEE*.

56. Elhadi M. Shakshuki, Kang, N. and Tarek R. S. (2013) 'EAACK—A Secure Intrusion-Detection System for MANETs', *IEEE Transactions On Industrial Electronics* , Vol. 60, No. 3.

57. Marti, S. Giuli, T. Lai, K. and Baker, M. (2000) 'Mitigating Routing Misbehavior in Mobile Ad Hoc Networks', *Proc. MobiCom*.

58. Nasser, N. and Yunfeng, C. (2007) 'Enhanced Intrusion Detection system for Discovering Malicious Nodes in Mobile Ad hoc Networks', *IEEE Communication Society subject matter expert for publication in the ICC 2007 proceedings*.

59. Jhaveri, R H. (2013) 'MR-AODV: A solution to Mitigate Blackhole and Greyhole attacks in AODV based MANETs', *IEEE, Third International Conference on Advance Computing and Communication Technology*.

60. Yu, M., Zhou, M. and Su, W. (2001) 'A Secure Routing Protocol Against Byzantine Attack in MANET in Adversarial Environment', *IEEE Transaction on Vehicular Technology* Vol.58, No.1.

61. Sukiswo, M. and Rifquddin, R. (2015) 'Performance of AOMDV Routing Protocol Under Rushing and Flooding Attacks in MANET', *IEEE, 2nd Conference of Information technology , Computer and Electrical Engineering (ICITACEE), Indonesia,* Oct 16 – 18th.

62. Gupta, A. K., Kumar, R. and Guptaa N. K. (2014) ‘A trust based secure gateway selection and authentication scheme in MANET’, *IEEE Contemporary Computing and Informatics (IC3I), International Conference.*

63. Anuj, J. and Sminesh, C. N. (2014) ‘An Improved Clustering-based Approach for Wormhole attack detection in MANET’, *IEEE, 3rd International Conference on Eco-Friendly Computing and Communication systems.*

64. Shamir, A. (1985) ‘Identity-based Cryptosystems and Signatures’, *In Proceedings of CRYPTO 1984, Springer-Verlag,* LNCS 196, pp. 47- 53.

65. Marin, R., Ruiz, P. M., Ros, F., Martinez, J. A. and Gomez, A. F. (2007) ‘Pre-authentication based enhanced for access control in hybrid MANET’, *IEEE, Computers and Communications.* ISCC 2007.

66. Khan, M. S., Midi, D., Khan, M. I. and Bertino, E. (2015) ‘Adaptive Trust Threshold Strategy for Misbehaving Node Detection an Isolation’ *IEEE Trustcom/ BigdataSE/ISPA.*

67. Rajesh, M. and Gnanasekar, M. (2016) ‘Consistently Neighbour detection for MANET’, *IEEE International Conference on Communication and Electronic Systems (ICCES).*

68. NS2 documentation, Available at: https://www.isi.edu/nsnam/ns/doc/node225.html

69. Khuswaha, P. (2017) 'Towards the Equivalence of Diffie-Hellman Problem and discrete logarithm problem for elliptic curves used in practice', *IEEE, Asia Security and Privacy (ISEASP).*

70. "IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications". Available at: ieeexplore.ieee.org.

71. Gupta, P. and Sexena, P. (2010) 'Energy consumption in Wireless Adhoc Network', *IEEE, 3rd International conference on emerging trendsin engineering and technology,* pp. 831-835.

72. IEEE 802.11 Standard. Available at: https://en.wikipedia.org/wiki/IEEE_802.11

73. IEEE Std 802.15.4-2011 8.1.2.2". Available at: Standard.ieee.org

74. Tottanesce, I., Anton, C., Ionescu, L. and Garagata, D. (2012) 'Elliptic Curve Cryptosystem Approach', *IEEE International Conference on Information Society.*

75. Nikama, P. D. and Raut, V. (2015) 'Improoved MANET security using Elliptic Curve Cryptography and EAACK', *IEEE International Conference on Computational Intelligence and Communication Networks.*

76. Gajbhiya, S., Karmakar, S. and Sharma, M. (2015) 'Diffie-Hellman Key Agreement with Elliptic Curve Discrete Logarithm Problem', *International Journal of Computer Application, Volume* 129-No 12.

77. Wong, Y. Ramamurthy, B. and Zou, X. (2006) 'The Performance of Elliptic Curve Based Group Diffie-Hellman Protocols for Secure Group Communication over Ad Hoc Networks', *IEEE Internatinal conference on Communication.*

78. Misic, J. (2008) 'Traffic and energy consumption of an IEEE 802.15.4 network in the presence of authenticated ECC Diffie-Hellman ephemeral key exchange', Computer Networks. Available at: www.elsevier.com/locate/comment.

79. Chang, Y. and Agarwal, D. P. 'Efficient Pairwise keys establishment and management in Wireless sensor Networks', *IEEE*, Internatinal conference on Mobile Adhoc and sensor systems.

80. Raja, L. and Baboo S.S. (2014) 'An Overview of MANET: Applications, Attacks and Challenges', l, *International Journal of Computer Science and Mobile Computing*, Vol.3 Issue.1, pp. 408-417

81. Sharma, S. B. and Chauhan, N. (2015), 'Security issues and their solutions in MANET', *IEEE International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*.

82. Sharma, K. S. and Sharma, V. (2016), 'Survey on security issues in MANET: Wormhole detection and prevention', *IEEE, International Conference on Computing, Communication and Automation (ICCCA)*, pp. 637 – 640.

83. Dodke, S., Mane, B. P. and Vanjale, M. S. (2016) *'A survey on energy efficient routing protocol for MANET' IEEE, 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 160 – 164

84. Sun, J. (2001), 'Mobile ad hoc networking: an essential technology for pervasive computing', *IEEE International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No.01EX479), Vol 3,* pp. 316 – 321.

85. Kaur, M., Rani, M. and Nayyar, A. (2014) 'A novel defense mechanism via Genetic Algorithm for counterfeiting and combating Jelly Fish attack in Mobile Ad-Hoc Networks', *2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence)*, pp. 359 – 364.

86. Mohandas, G., Silas, S. and Sam, S. (2013) 'Survey on routing protocols on mobile adhoc networks', *IEEE, 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s),* pp. 514 – 517.

87. Rashid, R. A., and Yusoff, R. (2006) 'Bluetooth Performance Analysis in Personal Area Network (PAN)', *2006 International RF and Microwave Conference, IEEE,* pp. 393 - 397

88. Liang, Y., Poor, H. V. and Ying, L. (2011) 'Secrecy Throughput of MANETs Under Passive and Active Attacks' *IEEE Transactions on Information Theory, Volume 57,* pp. 6692 – 6702.

89. Meddeb, R., Triki, B., Jemili, F. and Korbaa, O (2017), 'A survey of attacks in mobile ad hoc networks', *2017 International Conference on Engineering & MIS (ICEMIS),* pp. 1 – 7.

90. Nikander, E. P., Kempf, J. and Nordmark, E. (2004), *IPv6 Neighbor Discovery (ND) Trust Models and Threats* [Onine]. Available at: https://tools.ietf.org/html/rfc3756

91. Henderson, T. (2011), *NS2 Documentation* [Online]. Available at: https://www.isi.edu/nsnam/ns/doc/node644.html

92. Shaid, S. Z. M. and Maarof, M. A. (2014) 'Malware Behavior Image for Malware Variant Identification', *International Symposium on Biometric and Security Technologies (ISBAST),* pp. 238 – 243.

93. Najafabadi, M. M., Khoshgoftaar, T. M., Kemp, C., Seliya, N. and Zuech, R. (2014) 'Machine Learning for Detecting Brute Force Attacks at the Network Level', *IEEE 14th International Conference on Bioinformatics and Bioengineering,* pp. 379 – 385.

94. Liu, Q. S., Zhang, D. S. and Zhao, Y. (2013) 'STUDY ON FRAMEWORK OF DISTRIBUTED KEY MANAGEMENT FOR MANETS', *2013 International Conference on Information and Network Security (ICINS 2013),* pp. 1 - 5.

95. Bhanot, D and Chaudhary, A. (2017) 'Analyzing the Effects of Hello Packets in AODV', *International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; IC Value: 45.98; Volume 5 Issue VIII.* Available at: www.ijraset.com

96. Xu, Y. and Liu, F. (2017) ‘Hybrid Key Management Scheme for Preventing Man-in-Middle Attack in Heterogeneous Sensor Networks’, *3rd IEEE International Conference on Computer and Communications, 10.1109/CompComm.2017.8322777.*

97. Zhong, S., Chen, J. and Yang, Y.R. (2003) ‘Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks’, *Proc. INFOCOM* .

APPENDIX A: AODV 20 NODES TCL FILE

```
Phy/WirelessPhy set freq_ 2.472e9
Phy/WirelessPhy set RXThresh_ 2.62861e-09; #100m radius
Phy/WirelessPhy set CSThresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]
Phy/WirelessPhy set bandwidth_ 11.0e6
Mac/802_11 set dataRate_ 11Mb
Mac/802_11 set basicRate_ 2Mb

set val(chan) Channel/WirelessChannel ;
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail/PriQueue ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 30 ;
set val(nn) 20 ;
set val(rp) AODV ;
set val(x) 400 ;
set val(y) 400 ;
set val(stop) 100 ;
set val(energymodel)    EnergyModel ;
set val(initialenergy)    100 ;

set ns_ [new Simulator]
set tracefd [open AODV_20.tr w]

set winFile [open CwMaodv_20 w]
```

```
set namtracefd [open namwrls.nam w]

$ns_ trace-all $tracefd
$ns_ use-newtrace

$ns_ namtrace-all-wireless $namtracefd $val(x) $val(y)

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

$ns_ node-config -adhocRouting $val(rp) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -channelType $val(chan) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF \
                 -movementTrace OFF \
        -energyModel $val(energymodel) \
        -initialEnergy $val(initialenergy) \
        -rxPower 35.28e-3 \
        -txPower 31.32e-3 \
```

```
        -idlePower 712e-6 \
        -sleepPower 144e-9

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ node-config -initialEnergy [expr int(rand()*50)+50]
    set node_($i) [$ns_ node]
    $ns_ at 0.0 "[$node_($i) set ragent_] start_monitoring 55"
    $ns_ at 0.0 "[$node_($i) set ragent_] num_nodes $val(nn)"
 }

# Adding malicious nodes
$ns_ at 0.0 "[$node_(10) set ragent_] malicious"
$ns_ at 0.0 "[$node_(19) set ragent_] malicious"
$ns_ at 0.0 "[$node_(1) set ragent_] malicious"
#$ns_ at 0.0 "[$node_(8) set ragent_] malicious"

#setting different initial energy levels for nodes
#$ns_ node-config -initialEnergy expr {50+int(rand()*50)}
#set node_(1) [$ns_ node]

#$ns_ node-config -initialEnergy 60
#set node_(2) [$ns_ node]

#$ns_ node-config -initialEnergy 70
#set node_(8) [$ns_ node]

set god_ [God instance]

source scen-20-test.tcl
source cbr-20-4-10.tcl
```

```
for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ initial_node_pos $node_($i) 10
 }

for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ at $val(stop) "$node_($i) reset"
}

$ns_ at $val(stop) "stop"

proc stop {} {
global ns_ tracefd namtracefd
$ns_ flush-trace
close $tracefd
close $namtracefd
exec nam namwrls.nam &
exit 0
}


$ns_ run
```

APPENDIX B: AODV 20 NODES CBR FILE

```
#
# nodes: 20, max conn: 4, send rate: 0.10000000000000001, seed: 1
#
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(19) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.01000000000000001
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(14) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.01000000000000001
```

```
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 10.333118917575632 "$cbr_(1) start"
#
# 4 connecting to 6 at time 146.96568928983328
#
set udp_(2) [new Agent/UDP]
$ns_ attach-agent $node_(2) $udp_(2)
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(17) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.01000000000000001
$cbr_(2) set random_ 1
$cbr_(2) set maxpkts_ 10000
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 20.96568928983328 "$cbr_(2) start"
#
# 6 connecting to 7 at time 55.634230382570173
#
set udp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(15) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(18) $null_(3)
set cbr_(3) [new Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set interval_ 0.01000000000000001
```

```
$cbr_(3) set random_ 1
$cbr_(3) set maxpkts_ 10000
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 55.634230382570173 "$cbr_(3) start"
#
#Total sources/connections: 3/4
#
```

APPENDIX C: AODV 50 NODES TCL FILE

```
Phy/WirelessPhy set freq_ 2.472e9
Phy/WirelessPhy set RXThresh_ 2.62861e-09; #100m radius
Phy/WirelessPhy set CSThresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]
Phy/WirelessPhy set bandwidth_ 11.0e6
Mac/802_11 set dataRate_ 11Mb
Mac/802_11 set basicRate_ 2Mb

set val(chan) Channel/WirelessChannel ;
set val(prop) Propagation/TwoRayGround ;
set val(netif) Phy/WirelessPhy ;
set val(mac) Mac/802_11 ;
set val(ifq) Queue/DropTail/PriQueue ;
set val(ll) LL ;
set val(ant) Antenna/OmniAntenna ;
set val(ifqlen) 30 ;
set val(nn) 50 ;
set val(rp) AODV ;
set val(x) 400 ;
set val(y) 400 ;
set val(stop) 100 ;
set val(energymodel)    EnergyModel ;
set val(initialenergy)    100 ;

set ns_ [new Simulator]

set tracefd [open AODV_50.tr w]
```

```
set winFile [open CwMaodv_50 w]

set namtracefd [open namwrls.nam w]

$ns_ trace-all $tracefd
$ns_ use-newtrace

$ns_ namtrace-all-wireless $namtracefd $val(x) $val(y)

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)


$ns_ node-config -adhocRouting $val(rp) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -channelType $val(chan) \
                 -topoInstance $topo \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF \
                 -movementTrace OFF \
        -energyModel $val(energymodel) \
```

```
        -initialEnergy $val(initialenergy) \
        -rxPower 35.28e-3 \
        -txPower 31.32e-3 \
        -idlePower 712e-6 \
        -sleepPower 144e-9


for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ node-config -initialEnergy [expr int(rand()*50)+50]
    set node_($i) [$ns_ node]
    $ns_ at 0.0 "[$node_($i) set ragent_] start_monitoring 60"
    $ns_ at 0.0 "[$node_($i) set ragent_] num_nodes $val(nn)"
 }

# Adding malicious nodes
$ns_ at 0.0 "[$node_(15) set ragent_] malicious"
$ns_ at 0.0 "[$node_(25) set ragent_] malicious"
$ns_ at 0.0 "[$node_(35) set ragent_] malicious"


set god_ [God instance]

source scen-50-1-test.tcl
source cbr-50-5-10.tcl

for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ initial_node_pos $node_($i) 10
 }

for {set i 0} {$i < $val(nn) } {incr i} {
```

```
 $ns_ at $val(stop) "$node_($i) reset"
}

$ns_ at $val(stop) "stop"

proc stop {} {
global ns_ tracefd namtracefd
$ns_ flush-trace
close $tracefd
close $namtracefd
exec nam namwrls.nam &
exit 0
}
$ns_ run
```

APPENDIX D: AODV 50 NODES CBR FILE

```
#
# nodes: 50, max conn: 5, send rate: 0.10000000000000001, seed: 1
#
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(23) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(25) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.01000000000000001
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
```

```
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.01000000000000001
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1) start"
#
# 4 connecting to 6 at time 146.96568928983328
#
set udp_(2) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(2)
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(6) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.01000000000000001
$cbr_(2) set random_ 1
```

```
$cbr_(2) set maxpkts_ 10000
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 46.96568928983328 "$cbr_(2) start"
#
# 6 connecting to 7 at time 55.634230382570173
#
set udp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(7) $null_(3)
set cbr_(3) [new Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set interval_ 0.01000000000000001
$cbr_(3) set random_ 1
$cbr_(3) set maxpkts_ 10000
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 55.634230382570173 "$cbr_(3) start"
#
# 7 connecting to 8 at time 29.546173154165118
#
set udp_(4) [new Agent/UDP]
```

```
$ns_ attach-agent $node_(29) $udp_(4)

set null_(4) [new Agent/Null]

$ns_ attach-agent $node_(27) $null_(4)

set cbr_(4) [new Application/Traffic/CBR]

$cbr_(4) set packetSize_ 512

$cbr_(4) set interval_ 0.01000000000000001

$cbr_(4) set random_ 1

$cbr_(4) set maxpkts_ 10000

$cbr_(4) attach-agent $udp_(4)

$ns_ connect $udp_(4) $null_(4)

$ns_ at 29.546173154165118 "$cbr_(4) start"


#

#Total sources/connections: 4/5
```

APPENDIX E: SOURCE CODE

Send Trust Request

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


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

```
ih->saddr() = index;

ih->daddr() = IP_BROADCAST;

ih->sport() = RT_PORT;

ih->dport() = RT_PORT;


// Fill up some more fields.

rq->rq_type = AODVTYPE_TREQ;  //  The type of packet is defined here,  that is Trust Request

rq->rq_hop_count = 1;

rq->rq_bcast_id = bid++;

rq->rq_dst = dst;

rq->rq_dst_seqno = (rt ? rt->rt_seqno : 0);

rq->rq_src = index;

seqno += 2;

assert ((seqno%2) == 0);

rq->rq_src_seqno = seqno;

rq->rq_timestamp = CURRENT_TIME;
```

//Some code omitted for brevity

Send Trust Reply

```
void
AODV::sendTrustReply(nsaddr_t ipdst, u_int32_t hop_count, nsaddr_t rpdst,
        u_int32_t rpseq, u_int32_t lifetime, double timestamp, int dest_id) {
Packet *p = Packet::alloc();
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_reply *rp = HDR_AODV_REPLY(p);
aodv_rt_entry *rt = rtable.rt_lookup(ipdst);
#ifdef DEBUG
fprintf(stderr, "sending Reply from %d at %.2f\n", index, Scheduler::instance().clock());
#endif // DEBUG
 assert(rt);
 rp->rp_type = AODVTYPE_TREP; // The type of packet is defined here,  that is Trust Reply
 //rp->rp_flags = 0x00;
 rp->rp_hop_count = hop_count;
 rp->rp_dst = rpdst;
 rp->rp_dst_seqno = rpseq;
 rp->rp_src = index;
 rp->rp_lifetime = lifetime;
 rp->rp_timestamp = timestamp;
```

```
// trust value
printf ("Trust value of [%d] for [%d] .... is [%.2f] \n", index, dest_id, trustTable[index][dest_id]);
rp->node_trust = trustTable[index][dest_id];

// ch->uid() = 0;
ch->ptype() = PT_AODV;
ch->size() = IP_HDR_LEN + rp->size();
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_INET;
//printf("Here.....11\n");
ch->next_hop_ = rt->rt_nexthop;
//printf("Here.....22\n");
ch->prev_hop_ = index;          // AODV hack
ch->direction() = hdr_cmn::DOWN;
ih->saddr() = index;
ih->daddr() = ipdst;
ih->sport() = RT_PORT;
ih->dport() = RT_PORT;
ih->ttl_ = NETWORK_DIAMETER;
```

//**Some code omitted for brevity**

Send Hello

```
*
  Neighbor Management Functions
*/
void
AODV::sendHello() {
Packet *p = Packet::alloc();
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_aodv_reply *rh = HDR_AODV_REPLY(p);
iNode = (MobileNode *) (Node::get_node_by_address(index));
iEnergy = iNode->energy_model()->energy();
node_speed = iNode->speed();
two_hop_nbr = 0;
std::map<nsaddr_t, int>::iterator it = nbr_2hops.begin();
while (it != nbr_2hops.end() ){
 //    printf ("Neighbor Node [%d].... having Num of Nbrs [%d] \n", it->first, it->second);
       two_hop_nbr += it->second;
       it++;
   }
```

```
#ifdef DEBUG
fprintf(stderr, "sending Hello from %d at %.2f\n", index, Scheduler::instance().clock());
#endif // DEBUG


 rh->rp_type = AODVTYPE_HELLO;
 //rh->rp_flags = 0x00;
 rh->rp_hop_count = 1;
 rh->rp_dst = index;
 rh->rp_dst_seqno = seqno;
 rh->rp_lifetime = (1 + ALLOWED_HELLO_LOSS) * HELLO_INTERVAL;
 // my header data
 rh->num_nbr = num_nbr;
 rh->node_trust = total_avg_trust;
 rh->nbr_Energy = iEnergy;
 rh->nbr_Speed = node_speed;
 rh->nbr_2hop_nbrs = two_hop_nbr;
 strcpy (rh->trust_vector, aodv_trust_vector);
 // ch->uid() = 0;
 ch->ptype() = PT_AODV;
 ch->size() = IP_HDR_LEN + rh->size();
 ch->iface() = -2;
```

```
ch->error() = 0;

ch->addr_type() = NS_AF_NONE;

ch->prev_hop_ = index;          // AODV hack


ih->saddr() = index;

ih->daddr() = IP_BROADCAST;

ih->sport() = RT_PORT;

ih->dport() = RT_PORT;

ih->ttl_ = 1;


Scheduler::instance().schedule(target_, p, 0.0);


}
```

// **Some code omitted for brevity**