



# Masters Project Final Report

## March 2018

<b>Project Title</b>	Secured Ad Hoc Routing Protocol for Wireless Sensor Networks		
<b>Student Name</b>	W. M. S. S. Silva		
<b>Registration No. &amp; Index No.</b>	2014MIS021/ 14770218		
<b>Supervisor's Name</b>	Dr. Kasun De Zoysa		
<b>Please Circle the appropriate</b>	<b>Masters Program</b>	<b>Type</b>	
	<b>MIS</b>	<b>Research</b>	<b>Implementation</b>
<b>For Office Use Only</b>			

**SECURED AD-HOC ROUTING  
PROTOCOL FOR WIRELESS  
SENSOR NETWORKS**

W. M. S. S. Silva

2018



# SECURED AD-HOC ROUTING PROTOCOL FOR WIRELESS SENSOR NETWORKS

A dissertation submitted for the Degree of Master of  
Science in Information Security

W. M. S. S. Silva

University of Colombo School of Computing

2018



## Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/ institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name : W. M. S. S. Silva

---

Signature :

Date : 19.03.2018

This is to certify that this thesis is based on the work of  
Ms. W. M. S. S. Silva, under my supervision.

The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by :

Supervisor Name : Dr. Kasun De Zoysa

---

Signature :

Date : 19.03.2018

## Abstract

*The wireless sensor networks are collection of large numbers of tiny sensor devices with wireless communication capabilities. The sensor devices autonomously form networks through which sensor data is transported. The sensor devices are often severely resource constrained. An on-board battery or solar panel can only supply limited amounts of power. Moreover, the small physical size and low per-device cost limit the complexity of the system.*

*Wireless sensor network has issues compare with wired networking infrastructure. Among them range, power, processing power, memory and cost are major concerns. If, considered large nodes it is prone to failure, easy to be compromised, long term reliability cannot be considered. Due to highly increases of nodes there is a limitation. The wireless Ad-hoc network has list of challenges. For instance limited wireless transmission, routing overhead, battery constrains, Asymmetric links, time varying wireless link characteristic, broadcast nature of the wireless medium, packet losses due to transmission errors, Mobility-induced route changes, Potentially frequent network partitions and Ease of snooping on wireless transmissions (security issues).*

*In this project, the author proposes a solution to provide security in the route discovery process of mobile ad hoc networks in terms of confidentiality, integrity and authenticity. The proposed security solution is primarily designed for battle field surveillance systems however, it can be applied to any MANET application which seeks for confidentiality, integrity and authenticity.*

*The proposed solution is developed on top AODV; an existing MANET routing protocol by using 'Network Simulator 2 (NS2)' simulation environment. The author has used several case studies to perform a qualitative analysis on security achievements of the proposed solution and performed a comparison between the existing AODV protocol and the proposed solution in terms of security aspects. Furthermore, she has carried out a quantitative analysis to measure the performance of both protocols and performed a comparison between both protocols in terms of network performance.*

*The proposed solution has been succeeded in achieving its goal as a routing protocol which provides confidentiality, integrity and authentication in route discovery process. It can be used as a foundation to build a more secure and scalable routing protocol by improving and adding new features.*

# Table of Contents

Declaration.....	i
Abstract .....	ii
List of Figures.....	vi
List of Tables .....	vii
List of Acronyms .....	viii
Chapter 1: Introduction .....	1
1.1. Mobile Ad-Hoc Networks.....	1
1.2. Brief overview of project .....	2
1.3. Motivation .....	3
1.4. Goals of the project.....	4
1.5. Scope of the project and assumptions made.....	5
1.5.1. Security .....	5
1.5.2. Use of protocols and algorithms .....	5
1.5.3. Behavior of the MANET routing.....	5
1.5.4. Establishing a shared secret.....	5
1.6. Overview of the report .....	5
Chapter 2: Literature Study .....	7
2.1. MANET routing algorithms .....	7
2.1.1. Proactive routing algorithms.....	7
2.1.2. Reactive routing algorithms.....	8
2.1.3. Hybrid routing algorithms .....	8
2.1.4. Selection of the base routing protocol.....	8
2.2. AODV protocol and route discovery process.....	9
2.3. Security attributes for MANETs.....	12
2.3.1. Confidentiality .....	12
2.3.2. Integrity .....	12
2.3.3. Availability .....	12
2.3.4. Authentication .....	12
2.3.5. Authorization .....	12
2.3.6. Non repudiation .....	12
2.4. Types of security attacks in MANETs.....	13
2.4.1. Internal attacks.....	13
2.4.2. External attacks.....	13
2.4.3. Passive attacks .....	13
2.4.4. Active attacks .....	13
2.4. Attempts at MANET security.....	16
2.4.1. Secure Ad-hoc On-Demand Distance Vector Routing .....	16
2.4.2. A Trust Model Based Routing Protocol for Secure Ad hoc Networks.....	16
2.4.3. Prevention and Elimination of Gray Hole Attack in Mobile Ad hoc Networks by Enhanced Multipath Approach .....	17
2.4.4. A Framework for Detecting Malicious Nodes in Mobile Ad hoc Network.....	17

2.4.5. Mitigation of Collaborative Black Hole Attack Using TRACEROUTE Mechanism with Enhancement in AODV Protocol.....	17
2.4.6. Towards a Flexible Security Management Solution for Dynamic MANETs ...	18
2.4.7. Enhancing Security Features and Performance of AODV Protocol under Attack for MANET .....	18
2.4.8. Enhancing MANET Security using Secret Public Keys .....	19
2.4.9. Improved PKI Solutions for Mobile Ad hoc Networks .....	19
2.4.10. A Hybrid Cryptography Model for Managing Security in Dynamic Topology of MANET.....	19
2.4.11. A Guard Node (GN) based Technique against Misbehaving Nodes in MANET	20
Chapter 3: Design .....	21
3.1. Overall architecture.....	21
3.2. Simulation environment .....	21
3.3. Proposed security solution .....	22
3.3.1. Working mechanism of the proposed solution .....	22
3.3.2. Pseudo code of the proposed solution .....	24
3.3.3. Flowchart for the security solution .....	25
Chapter 4: Implementation – A Secured Ad hoc Routing Protocol for Wireless Sensor Networks.....	28
4.1. Implementation on the simulation environment – NS2 .....	28
4.1.1. Assigning a common secret key .....	29
4.1.2. Generating a random string .....	30
4.1.3. Creating the string table .....	30
4.1.4. Implementing cryptographic functions .....	31
4.1.5. Implementing the security solution in AODV.....	32
4.1.6. Compiling source code.....	33
4.1.7. Simulating and analyzing .....	33
Chapter 5: Testing, Evaluation and Validation.....	34
5.1. The plan.....	34
5.2. Evaluation of security aspects .....	34
5.2.1. Case Studies.....	35
5.3. Evaluation of performance .....	35
5.3.1. Performance matrices.....	35
Chapter 6: General discussion .....	37
6.1. On the security aspects of the proposed solution .....	37
6.1.1. Case Study 1: Eavesdropping .....	37
6.1.2. Case Study 2: Man-in-the-Middle .....	37
6.1.3. Case Study 3: Bogus Information.....	38
6.2. On the performance of the proposed solution .....	38
6.2.1. End to End Delay .....	39
6.2.2. End to End Delay .....	39
6.2.3. Normalized Routing Overhead.....	40
6.2.4. Average Throughput .....	41
Chapter 7: Conclusion and future work .....	42

7.1. Problems faced .....	42
7.1.1. Defining a boundary .....	42
7.1.2. Maintaining a tradeoff between security and performance .....	42
7.1.3. Evaluation.....	42
7.2. Lessons learned.....	43
7.2.1. Understanding MANET operations .....	43
7.2.2. Understanding concepts of information security .....	43
7.2.3. Getting familiar with new technologies .....	43
7.3. Deviations from original project plan .....	43
7.3.1. Evaluation techniques .....	43
7.3.2. Software version changes.....	44
7.4. Extensions and future work.....	44
7.4.1. Introducing more security attributes .....	44
7.4.2. Improving network performance .....	44
7.5. Conclusion.....	44
Appendix A: TCL Scripting .....	46
A.1. Brief Overview of TCL .....	46
A.2. TCL Script Used – explanation .....	46
Appendix B: Trace Files.....	49
B.1. Brief Overview of Trace Files .....	49
B.2. Old Trace File Format .....	49
Appendix C: AWK Files .....	51
C.1. AWK Scripting Overview .....	51
C.2. AWK Scripts to Analyze Performance .....	51
Appendix D: AODV Modifications .....	55
D.1. AODV::sendRequest(...) Modifications .....	55
D.2. AODV::recvRequest(...) Modifications .....	56
D.3. AODV::sendReply(...) Modifications .....	57
D.4. AODV::recvReply(...) Modifications .....	58
Appendix E: Readings of the Performance Matrices .....	60
E.1. End to End Delay .....	60
E.2. Packet Delivery Ratio .....	60
E.3. Normalized Routing Overhead .....	60
E.4. Average Throughput .....	61
References .....	62



## List of Figures

Figure 1: Sample Wireless Sensor Network Topology .....	1
Figure 2 : Message format of RREQ .....	9
Figure 3 : Message format of RREP .....	10
Figure 4 : Message format of RERR.....	11
Figure 5: Testing environment.....	21
Figure 6: Sample TCL Script.....	28
Figure 7: Sample piece of a trace file.....	29
Figure 8: Assigning a key value to a variable .....	29
Figure 9: Generating a random string .....	30
Figure 10: Creating the sting table .....	31
Figure 11: Initialization of encryption and decryption .....	31
Figure 12: Sample code from encryption process .....	32
Figure 13: Sample code from hashing .....	32
Figure 14: End to end delay vs. number of nodes .....	39
Figure 15: Packet delivery ratio vs. number of nodes.....	40
Figure 16: Normalized routing overhead vs. number of nodes .....	40
Figure 17: Average throughput vs. number of nodes.....	41
Figure 18: Snap shot of an old format trace file .....	49

## List of Tables

Table 1: Project timeline .....	3
Table 2 : Common attacks in MANETs.....	14
Table 3 : Simulation parameters .....	22
Table 4: Description of the fields.....	50
Table 5: End to end delay simulation results.....	60
Table 6: Packet delivery ration simulation results .....	60
Table 7: Normalized routing overhead simulation results .....	60
Table 8: Average throughput simulation results .....	61

## List of Acronyms

<b>Acronyms</b>	<b>Explanation</b>
MANET	Mobile Ad-Hoc Network
IP address	Internet Protocol address
DoS	Denial of Service
TCP	Transmission Control Protocol
PKI	Public Key Infrastructure
CA	Certification Authority
TA	Trusted Authority
PGP	Pretty Good Privacy
UDP	User Datagram Protocol
RREQ	Route Request
RREP	Route Reply
RERR	Route Error

# Chapter 1: Introduction

## 1.1. Mobile Ad-Hoc Networks

Wireless Sensor networks are collection of number of mobile nodes with wireless communication capabilities. This is a special type of Ad Hoc Network [1]. They forms a temporary network without rely on a per-existing infrastructure, where there is absence of a central access point. This is an advancement of wireless technologies like IEEE 802.11/Wi-Fi [2], Bluetooth [3] and etc. The mobile nodes autonomously form network through which sensor data is transported. These devices consider of limited resources, with small size, limited battery power and etc. These type of networks are useful in disaster management, emergency search and rescue, battlefield surveillance and other communication application due advantages shown in these network as easy installation and upgrade more flexibility, low cost and maintenance [4].

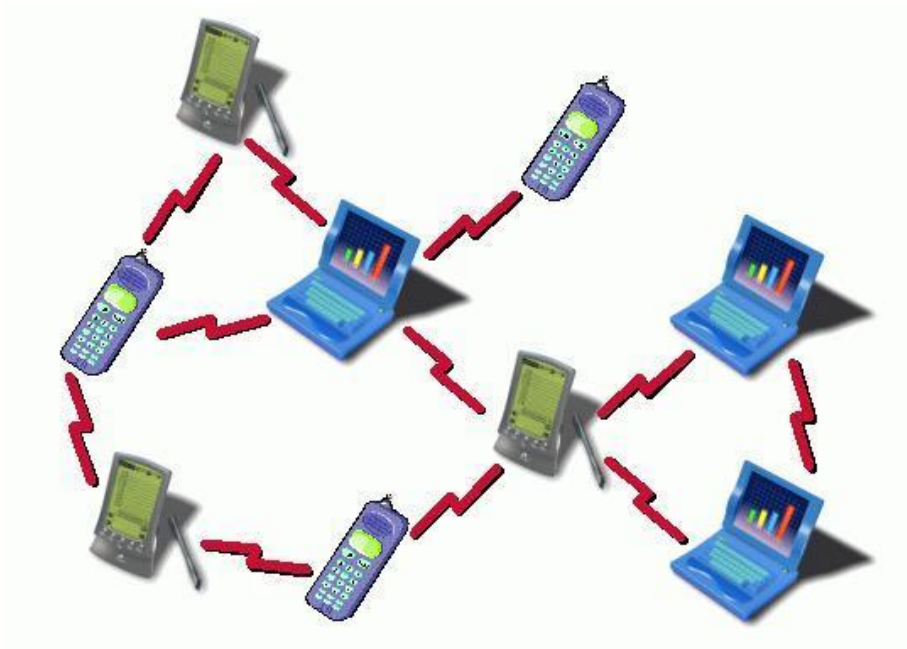


Figure 1: Sample Wireless Sensor Network Topology

Compared with the other networks Wireless Sensor networks show distinct characteristics such as [5]:-

- Dynamic Topology
- Node Mobility
- Distributed Operation
- Shared Media
- Multi-hop peer-to-peer routing
- Low bandwidth & low data rate
- Power consumption constrains for nodes using batteries or energy harvesting
- Communication failures and etc.

Due to this nature they have issues compared with the wired Network. Major issues faced by Wireless Sensor Networks are Architecture, Security, Quality of Service, Power, Memory, Range and Processing power.

And since the mobile nodes in a mobile ad-hoc network operates without a central access point all the networking functions such as routing and packet forwarding are done by the mobile nodes themselves. Therefore these kinds of networks are highly vulnerable to attacks.

## **1.2. Brief overview of project**

In this project, the author attempts to minimize these attacks by embedding a security solution for an existing MANET routing protocol. This solution particularly targets military surveillance like systems where confidentiality, integrity and authenticity of the data are considered as the first priority. However, the proposed security solution can be applied to any MANET based system which seeks for confidentiality, integrity and authentication via MANETs. The main deliverable of this project is “a secured ad-hoc routing protocol for wireless sensor networks”.

The following timeline gives a snapshot of the progress of the project.

Table 1: Project timeline

Tasks	April	May	June	July	Aug.	Sep.	Oct.	Nov.	Dec.	Jan.	Feb.	March
Project Initialization												
Prepare the Project Proposal												
Evaluating the Existing Solutions												
Designing and implementing the Prototype												
Evaluating the Model												
Improvisation												
Prepare the Final Model												
Prepare the Final Thesis												

The project was initiated in late July of 2017. Prior commencing on the design, existing work related to secure MANET routing was thoroughly investigated (refer Chapter 2). Based on these investigations the requirements of security enhanced routing protocol for MANETs were identified and defined. Afterwards, the solution was designed (refer Chapter 3) and implemented (refer Chapter 4). Finally, the proposed solution was tested and evaluated (refer Chapter 5).

Since this is a research project, it does not follow the development pattern of a conventional software development project.

### 1.3. Motivation

Wireless sensor networks are highly demanding as well as highly evolving area due to its great usage in critical scenarios such as disaster management, military surveillance, emergency search and rescue where typical communication methods are not applicable.

However, these mobile ad-hoc networks are generally more prone to security threats than wired networks due to the higher degree of flexibility [6]. Ordinary security solutions used in wired networks are ineffective and inefficient for MANETs due to its highly dynamic and resource constrained nature.

Since the mobile nodes in a mobile ad-hoc network operates without a central access point all the networking functions such as routing and packet forwarding are done by the mobile nodes themselves. Therefore these kinds of networks are highly vulnerable for attacks.

When considering military surveillance systems, security of the data transmitted over the medium is one of the major concerns. Such systems basically requires to ensure that the information is available only for the intended/ authorized parties which is known as ‘confidentiality’, ensure that the information being transferred is not subjected to any unauthorized alterations referred as ‘integrity’ as well as ensure and confirm a user’s identity, in other terms ‘authentication’.

The main reason that the author chose this project is to come up with a security solution which basically provide confidentiality, integrity and authentication for data transmission in mobile ad-hoc networks so that, it can be used in military surveillance systems which will help greatly in reducing the costs involved in setting up a traditional network, maintaining the network, training officers as well as to help in reducing the time involved in setting up and maintaining the network.

Even though the author mainly targets military surveillance systems, this solution is applicable for any commercial/non commercial system which interested in providing confidentiality, integrity and authentication for data transmission process.

## 1.4. Goals of the project

The primary goal of this project, as stated in the project proposal is to “implement a security solution for wireless sensor networks” so that it can be applied to battlefield surveillance systems and any system which seeks for secure communication via MANETs. This goal contains several sub-goals;

- To provide confidentiality in data transmission
- To provide integrity in data transmission
- To provide authentication in data transmission

## **1.5. Scope of the project and assumptions made**

### **1.5.1. Security**

This project only attempts at providing confidentiality, integrity and authentication for the data transmission process of a specific MANET routing protocol. This project is not a general investigation on security of MANETs.

### **1.5.2. Use of protocols and algorithms**

Even though many algorithms and protocols such as AODV, AES, H-MAC are used in this project, it does not intend to individually develop these algorithms and protocols in a general way.

### **1.5.3. Behavior of the MANET routing**

In this project the author will not consider all MANET routing protocols in general, but restrict herself to a specific routing protocol which is, AODV. AODV is a MANET routing protocol which is having least control overhead and least complexity.

### **1.5.4. Establishing a shared secret**

In this project the author has used a hard coded secret key as a shared secret. However in real-world, an algorithm such as Diffie Hellman can be used to establish a shared secret between two parties.

## **1.6. Overview of the report**

Chapter 2 reviews the background and existing literature related to the project. First of all it will brief the audience about existing MANET routing algorithms. After that the chapter will highlight the reasons of selecting the base routing protocol for this project. Then it will explain the route discovery process of the selected routing protocol. Next objective of this chapter is to give the audience an idea about security issues in MANETs. Finally the chapter ends with existing attempts at MANET security.

Chapter 3 describes the design of the proposed solution. Chapter 4 describes the implementation and usage of the simulation environment.



Chapter 5 describes the evaluation process of security aspects and evaluation process of performance with the explanations of performance matrices.

Chapter 6 discusses certain topics related to the project such as security aspects and performance of the proposed solution.

Chapter 7 states the problems faced and lessons learned during the project, deviations from the original project plan as well as extensions and future work with critical appraisal of the system.

## Chapter 2: Literature Study

### 2.1. MANET routing algorithms

Because of the infrastructure-less nature of Mobile Ad-Hoc Networks, routing plays an important role in it. There are different types of routing algorithms proposed for MANETs to serve different types of network requirements. This section describes these routing algorithms along with their advantages and disadvantages.

One of the famous MANET routing algorithm categorizations is based on the information that particular algorithm collects in order to form the routing table. These types of algorithms can be categorized into two categories;

- A. Shortest path algorithms
- B. Link state algorithms

Where shortest path algorithms use distance information to form the routing table while link state algorithms use information relevant to connectivity to form the routing table.

Another interesting categorization is based on when the routing tables are formed. There are three main algorithm types under this category. They are;

- A. Proactive routing algorithms
- B. Reactive routing algorithms
- C. Hybrid routing algorithms

#### 2.1.1. Proactive routing algorithms

Proactive routing algorithms find and maintain routes for all source-destination pairs in advance even if they are not required. Proactive routing algorithms periodically send and receive routing information to maintain up-to-date routing information on each and every node in the network.

Advantages of proactive routing algorithms:

- Fast connection establishment time since path is already available

Disadvantages of proactive routing algorithms:

- Large amount of control overhead
- Uses only bidirectional links
- Suffers from count to infinity problem

Destination Sequenced Distance Vector (DSDV) [7] and Optimized Link State Routing (OLSR) [8] are two well known proactive routing protocols.

### 2.1.2. Reactive routing algorithms

Also known as on-demand routing algorithms, because they establish routes between nodes only when it is required by the source nodes to transfer data packets to a specific destination. In reactive routing algorithms, routing table is not periodically updating. It chooses the routes that are already being used or setup. If it requires a route to a destination for which it does not have routing information, it starts a route discovery process. Route discovery process will be explained in the next section.

Advantages of reactive routing algorithms:

- Reduced control overhead

Disadvantages of reactive routing algorithms:

- Route setup latency

Ad-hoc On-demand Distance Vector (AODV) [9] is one of the famous and widely used reactive routing protocols. Other than AODV, Dynamic Source Routing (DSR) [10] and Temporally Ordered Routing Algorithm (TORA) [11] are other examples for reactive routing protocols.

### 2.1.3. Hybrid routing algorithms

Hybrid routing algorithms combine features of both proactive and reactive routing algorithms. These algorithms use proactive approaches to gather the initial routing information. Afterwards they act according to the demand to serve nodes.

Location Aided Routing (LAR) [12] and Zone Routing Protocol (ZRP) [13] are such well known hybrid routing protocols.

### 2.1.4. Selection of the base routing protocol

Reactive routing protocols have a major advantage over proactive routing protocols, which is less control overhead. Since reactive routing protocols

designed to find a route on demand basis the control overhead in the network is less compared to the proactive routing protocols. Since they generate a small amount of control overhead, the usage of network bandwidth is again less compared to the proactive routing protocols.

Therefore, in this research the author has selected AODV, a reactive routing protocol as the base routing protocol to implement the security solution. Furthermore, as listed below AODV has several advantages over other reactive routing protocols.

- Simplicity
- Low computational complexity
- Less control and memory overhead
- Loop freedom
- Maturity of the protocol

## 2.2. AODV protocol and route discovery process

The Ad hoc On-demand Distance Vector (AODV) protocol is a reactive protocol which allows dynamic, self starting multi hop routing between participating mobile nodes who wants to establish and maintain an ad-hoc network.

There are three basic message types in AODV;

### 1. Route Requests (RREQs)

Message format

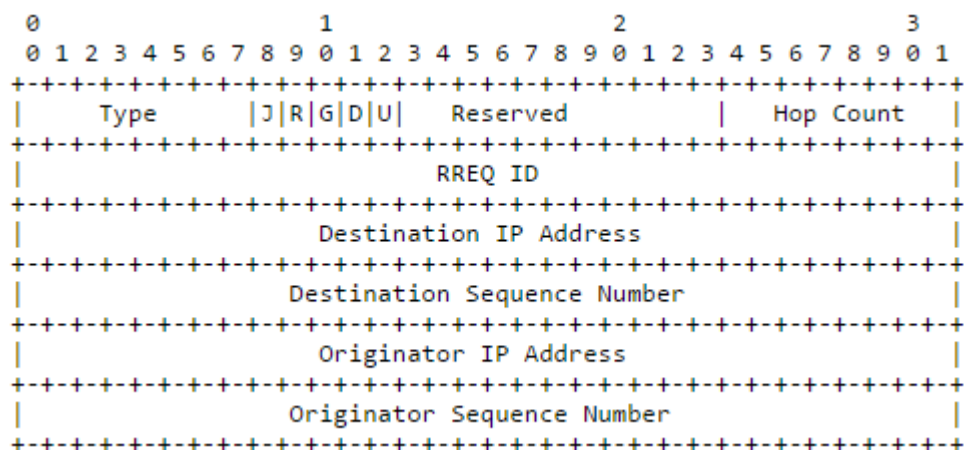


Figure 2 : Message format of RREQ

Figure 2 shows the format of a RREQ packet. It contains following fields;

Type : Type of the message

- Flags : Five types of flags
- Reserved: Ignored on reception
- Hop count : Total number of hops from the originator to the node which handles the request
- RREQ ID : Sequence number which used to identify a specific RREQ
- Destination IP address : IP address of the destination
- Destination sequence number : The latest sequence number received by the originator for any route towards the destination
- Originator IP address : IP address of the originator/source
- Originator sequence number : The current sequence number to be used in route entry which points towards the originator of the RREQ

## 2. Route Replies (RREPs)

Message format

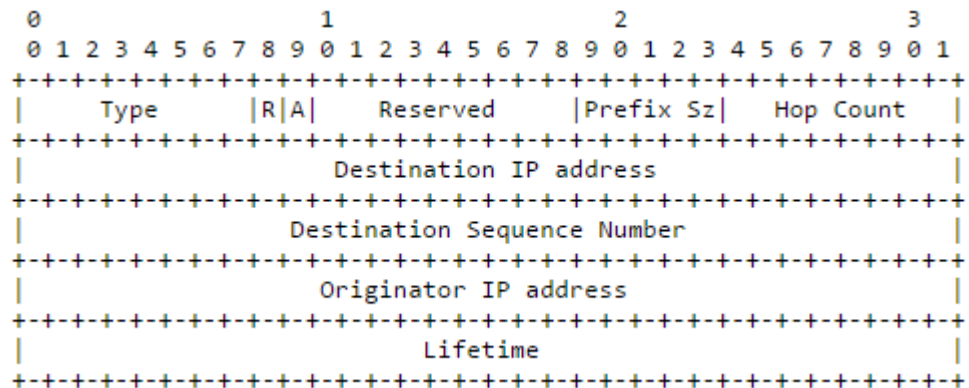


Figure 3 : Message format of RREP

Figure 3 shows the format of a RREP packet. It contains following fields;

- Type : Type of the message
- Flags : Two types of flags
- Reserved: Ignored on reception
- Prefix Sz: If set to non zero, this value specifies that the indicated next hop may be used for any node with the same prefix
- Hop count : Total number of hops from the originator to the node which handles the request

- Destination IP address : IP address of the destination
- Destination sequence number : The latest sequence number received by the originator for any route towards the destination
- Originator IP address : IP address of the originator/source
- Lifetime : Valid time of the RREP

### 3. Route Errors (RERRs)

Message format

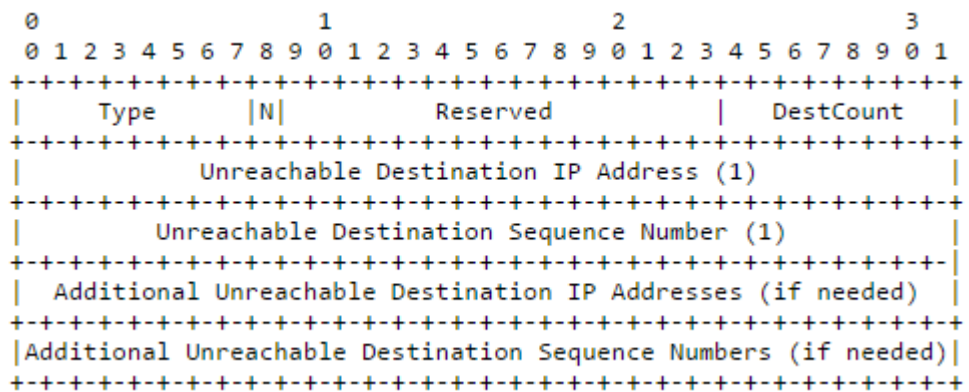


Figure 4 : Message format of RERR

Figure 4 shows the format of a RREQ packet. It contains following fields;

- Type : Type of the message
- Flags : one types of flags
- Reserved: Ignored on reception
- DestCount : Total number of unreachable destinations in the route
- Unreachable destination IP address : IP address of the unreachable host
- Unreachable destination sequence number : The sequence number in the routing table entry for the unreachable host

When a source has a need to send packets to a destination whose route is unknown, first of all it broadcasts a RREQ. Including all the fields mentioned in the previous section. When a node receives a RREQ, it checks whether it is the destination if not, it checks its routing table for a fresh enough route. If it has a fresh enough route it will

reply to the RREQ with a RREP. If not, after incrementing the hop count it will rebroadcast the RREQ. And it also maintains a route back to the source node for a certain amount of time.

When a node receives a RREQ and it has a fresh enough route to the destination it will send a RREP to the source. The intermediate node will increment the hop count and forwards the RREP to the source via the reverse route.

RERR is sent to the neighbors when a link failure in an active route is detected.

## **2.3. Security attributes for MANETs**

There are several important requirements to achieve when providing security in MANETs, which will be discussed in the following section.

### **2.3.1. Confidentiality**

Confidentiality guarantees that a message cannot be viewed in its original form by any unauthorized person. Most of the services which provide confidentiality use encryption methods to provide confidentiality.

### **2.3.2. Integrity**

Integrity guarantees that no modification, addition or deletion is done to the message intentionally or accidentally.

### **2.3.3. Availability**

Availability ensures that the network resources are available to the authorized users regardless of the state of the network.

### **2.3.4. Authentication**

Authentication ensures that the parties in a communication channel are genuine not impersonators. In here, parties need to prove their identities to access network resources.

### **2.3.5. Authorization**

Authorization ensures that the different types of users are assigned with different access rights.

### **2.3.6. Non repudiation**

With non repudiation, sender cannot disavow about sending a message.

## **2.4. Types of security attacks in MANETs**

Due to some distinct characteristics of mobile ad-hoc networks such as absence of predefined infrastructure and boundary, no centralized control facility, resource constrained nature, changing scale, etc. they are more prone to security attacks than wired networks. These attacks can be external or internal.

### **2.4.1. Internal attacks**

These types of attacks start from the nodes which are available inside the network. As an example a node may broadcast bogus routing information to other nodes in the network. Most of the time internal attacks are more difficult to handle when compare to external attacks.

### **2.4.2. External attacks**

These types of attacks usually generate from outside the network. They prevent the network from carrying out normal communication and produce additional overhead in the network.

### **2.4.3. Passive attacks**

Passive attacks do not modify the data transmitted within the network. It rather involves unauthorized users secretly listening to the network traffic. Passive attacks do not modify the routing process and it does not generate additional network overhead but attempts to achieve important information by looking at the traffic. Since the network operations do not get affected, it is difficult to detect passive attacks.

### **2.4.4. Active attacks**

Active attacks are intended to interferes the flow of a network transmission and hence prevent message flow between nodes. It can be internal or external. As an example active external attack can be executed by a resource which does not belongs to the network. And an active internal attack can be a



malicious node which belongs to the network. Active attacks allow attackers to modify the traffic, deny the network services or create network congestions.

Table 2 summarizes some of the common attacks faced by wireless sensor networks with their attack type and the security attributes which are violating.

Table 2 : Common attacks in MANETs

Attack	Description	Attack Type	Violated Security Attribute
Denial of service (DoS)	The attacker tries to flood the victim network with a large amount of packets.	Internal/ External Active	Availability
Man-in-the-middle (MITM)	Through a valid route the attacker comes inside the network and tries to sniff packets going through it.	Internal Passive/ Active	Confidentiality Integrity
Wormhole attack	The attacker tries to disrupt routing by short circuiting the network.	External Active	Availability Confidentiality
Sybil attack	The attacker tries to fake multiple identities at the same time by pretending to be consisted by multiple nodes.	Internal Passive	Authentication
Eaves Dropping	The attacker tries to obtain some confidential information during the communication. These information can be passwords, locations, keys, etc...	Internal Passive	Confidentiality

Bogus information	Attacker generates and transmits false routing information to other nodes.	Internal Active	Integrity Authentication
Blackhole attack	The attacker tries to fool the legitimate nodes saying that it has the optimum route to destination which cause other nodes to send their traffic via the attacker. The attacker can misuse or drop the packets.	External Passive/Active	Availability
SYN flooding attack	The attacker tries to create a huge amount of half opened TCP connections with the victim node without completing the three way handshake.	External Active	Availability
Session hijacking attack	The attacker tries to spoof the IP address of destination node and get the correct sequence number which is expected by the target node and launch a DoS attack on the victim.	External Active	Availability
Sinkhole Attack	Malicious node advertises bogus routing information and receives whole network traffic. After that it modifies the secret information and tries to attract the secure data from neighboring nodes.	Internal Active	Authentication Integrity Confidentiality

## 2.4. Attempts at MANET security

Following section discussed the current and proposed research works done by the researchers past few years. Mainly it discussed security issues of ad-hoc networks, which is still a major challenge in MANET.

In Mobile Ad-hoc Networks (MANETs), providing security is a challenging task. Many researches are taken place on ad-hoc networks at both the MAC and Routing layers. Meanwhile, security issues are considered in both MAC and Routing layers for ad-hoc networks.

In MANET many secure routing protocols were proposed during the past and recent years. These protocols ensure various security aspects. They include various strategies such as cryptography, trust models and etc. These strategies can improve the security level of routing process in MANETs. However, they have their own pro and cons.

### 2.4.1. Secure Ad-hoc On-Demand Distance Vector Routing

Secure Ad-hoc on Demand Distance Vector (SAODV) [14] is an extension to AODV. It protects the route discovery process by providing authentication, integrity and non-repudiation. SAODV uses asymmetric cryptography to provide authentication and digital signatures to provide integrity as well as non-repudiation. In SAODV, each mobile node has a public, private key pair from a suitable asymmetric crypto system. Furthermore it uses hash chains to authenticate the hop count of the AODV routing. However, simulation experiment carried out [15] shows that SAODV is significantly more expensive protocol because of the added asymmetric cryptography, increased message size and inability of intermediate nodes to respond to RREQs.

### 2.4.2. A Trust Model Based Routing Protocol for Secure Ad hoc Networks

Trusted Ad-hoc on Demand Distance Vector (TAODV) [16] uses trust relationship among nodes which allows better routing decisions and detects uncooperative nodes. Basically there are four modules in this trusted model; basic routing protocol, trust model, trusted routing protocol and self-organized key management mechanism. In TAODV nodes can flexibly choose whether and how to apply cryptographic operations. Therefore computational can be reduced without using certificates at every routing operation. However, TAODV information itself in each packet is not secured. Furthermore, a

detailed simulation evaluation is required in terms of message overhead, security analysis and tolerance to mobile attacks.

### **2.4.3. Prevention and Elimination of Gray Hole Attack in Mobile Ad hoc Networks by Enhanced Multipath Approach**

The authors of [17] suggested a mechanism to eliminate the gray hole affects by finding all the malicious nodes which are present in the network and send broadcast to whole network for elimination of malicious nodes. Black hole and gray hole attacks are type of a DoS attacks which greatly affects the network integrity as well as network availability. The proposed solution introduced a packet update scheme in which it fetches information from the intermediate nodes for the enquiry of the suspected nodes in the network. The simulation results show that the proposed solution can improve the network performance in terms of throughput around 234544 bits per second and in terms of end to end delay around 45%. However, it will be more useful if the solution uses unicasts instead of broadcasts which uses higher level of resources so that it will save resources as malicious nodes will be only detected through partial multicasting process.

### **2.4.4. A Framework for Detecting Malicious Nodes in Mobile Ad hoc Network**

In [18], the researchers proposed a technique which can be used to detect and prevent the malicious nodes presence in a particular network. Malicious nodes are the nodes which do not cooperate with other nodes and they act selfishly by reserving the resource for its own use. This decreases the network performance in terms of throughput and packet delivery ratio. In order to prevent these malicious nodes has to be identified and that specific route has to be prevented from routing. In this proposed solution the malicious nodes are detected prior to the routing process using consensus based algorithm and then that route is prevented from transferring data between nodes. Simulation results show that this can increase the network performance in terms of throughput.

### **2.4.5. Mitigation of Collaborative Black Hole Attack Using TRACEROUTE Mechanism with Enhancement in AODV Protocol**

The author of [19] proposed a solution to detect collaborative co-operative black hole attacks in MANET. The existing AODV routing protocol has been modified by introducing a new field which allows finding the optimal, secure

and reliable path to a destination. The proposed solution uses mechanism of 'traceroute' to detect the source of the collaborative black hole attack and breaks the collaboration by eliminating the route between those malicious nodes. The simulation results show that the proposed solution works more efficiently when detecting malicious nodes with minimal false positives and no true negatives. Thus increases the network performance in terms of packet delivery ratio. However the proposed mechanism generates constant overheads in route maintenance due to the improvements and cryptography. Furthermore, it is required to introduce a mechanism which can be used to identify and label all the nodes which are actively involved in co-operation for the purpose of dropping packets.

#### **2.4.6. Towards a Flexible Security Management Solution for Dynamic MANETs**

The researchers of [20] proposed a scalable, flexible and application oriented framework named Adaptive Secured Multipath for Ad hoc networks (ASMA) which is able to manage security depending on the application requirement and the network conditions. The solution is based on a structure known as macrograph combining both dynamic trust management and multipath routing. Whereas the macrograph structure is capable of estimating transmission security to provide the assurance that communication is established only when they satisfied the application's security requirements. Simulation results show that the proposed protocol outperforms AOMDV, dividing by three the packet loss rate in networks including 20% of malicious nodes, while causing only 3% of additional loss in safe networks.

#### **2.4.7. Enhancing Security Features and Performance of AODV Protocol under Attack for MANET**

In [21], the authors proposed a security mechanism which integrates digital signature and hash chain to protect the AODV routing protocol against both malicious and untrusted nodes. Hash chains has been used to authenticate the hop count of RREQ and RREP packets so that it allows each and every receiver node to verify that the hop count has not been modified by the attacker. Whereas digital signatures has been used to protect the integrity of the data in the RREQ and RREP packets. Furthermore, the simulation results show marginal performance difference under attack when compared with the original AODV protocol.

#### **2.4.8. Enhancing MANET Security using Secret Public Keys**

The authors of [22] present a secure way for MANET nodes to authenticate each other and to secure data sent by each other using PKI scheme. The solution assumes that each node is able to generate its own public/private key pair using RSA algorithm and each node is equipped with a smart machine which is capable of generating unforgeable identity information. The proposed mechanism uses four keys security scheme. This scheme uses RSA to enhance the speed of the network. And, it does not use distributed CA or TA services in order to avoid the huge amount of control overhead. Currently, the research is going on to prove the efficiency of the proposed mechanism and to analyze the overhead of its behavior by simulating it.

#### **2.4.9. Improved PKI Solutions for Mobile Ad hoc Networks**

In [23], the authors proposed an improved PKI solution for mobile ad hoc networks which provides an intelligent way to identify a set of CA nodes and distribute the system secret over those nodes using threshold cryptography. These chosen nodes are responsible for collectively providing the CA functionality for MANETs using threshold cryptography. In order to minimize the usage of limited resources in mobile nodes, the researchers employed a simple communication protocols for mobile nodes to receive certification services. However the authors did not present a real world or simulation results to support the theory.

#### **2.4.10. A Hybrid Cryptography Model for Managing Security in Dynamic Topology of MANET**

The authors of [24] proposed security architecture for small-scale networks utilizing a seniority-based trust model and PGP type certification service building a specific PKI. In order to construct the model, the authors have made the following main assumptions;

- Each node has a unique ID and is capable of discovering available senior member nodes of the network
- Communication with senior member nodes is more reliable than that of junior member nodes
- Mobility is centralized by using a maximum node moving speed
- Each senior node is capable in detecting misbehaving nodes locally
- All nodes are maintaining a seniority table
- Two nodes having offline certificate holder are used to centralized

The simulation results show that the proposed mechanism is easy to deploy and efficient for small networks. However the scalability issue and a construction of certification revocation scheme are yet to be addressed.

#### **2.4.11. A Guard Node (GN) based Technique against Misbehaving Nodes in MANET**

In [25], the authors proposed a solution based on assigning ‘guard nodes’, which are responsible for overhearing and reporting misbehaving nodes. One of the existing MANET routing protocols; DSR has been used as the base protocol to implement the proposed solution. The idea behind this solution is to delegate the responsibility of overhearing of a misbehaving node into an independent node known as ‘guard node’ rather than the sender. The guard node is capable of overhearing the data transmission from source to the destination covering the entire path excluding the source and the destination nodes. These special nodes are selected by the source node and the selection process is based on the returned routes during the route discovery process and the cached routes. The results show that the proposed solution performs better than DSR when the nodes move in high speed. However, the packet delivery ration for both protocols affected by almost the same rate when the percentage of the malicious nodes increases. Furthermore, the proposed solution has a higher amount of routing overhead because of the embedded security features.

## Chapter 3: Design

### 3.1. Overall architecture

In this project the author focused on implementing a secured routing protocol for wireless sensor networks on top of NS-2 simulation environment. The author has installed NS-2 on top of Ubuntu, a Linux based operating system.

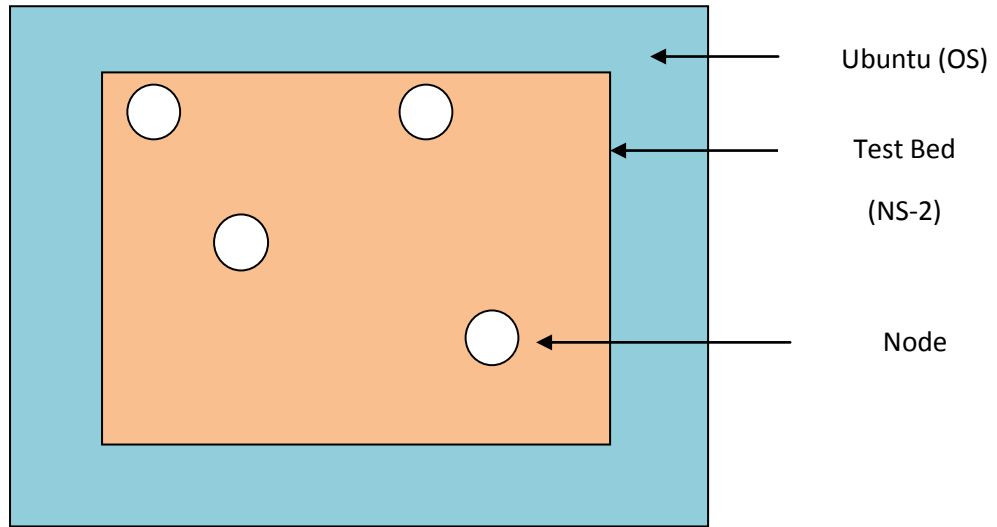


Figure 5: Testing environment

NS-2 is an open source simulation environment that runs on top of Linux/Windows operating systems. It is a discrete event simulator targeted at network research including wired and wireless areas. It supports simulation of routing, multicast protocols and IP protocols such as TCP, UDP over wired and wireless networks. It has many characteristics that make it a very useful tool. Some of those characteristics are shown as follows;

- Support for a wide range of protocols
- Flexibility of the tool
- Capability of graphically detailing network traffic
- Support for various algorithms in routing and queuing

### 3.2. Simulation environment

NS-2 makes use of TCL scripts to create a network topology and uses a tool known as NAM (Network Animator) to visualize the network topology. Another interesting



feature is that it records information regarding packet transmission in a trace file which greatly helps researchers to analyze the data transmission in depth.

In this project author has prepared the simulation environment using TCLs to test and validate the results. The author has simulated the proposed solution using 802.11 MAC protocol with 11Mbps bandwidth. For simulations the author has considered 10,20,30,40 and 50 mobile nodes separately in a mobility area of 1000 x 1000 meters. The simulation time is 1000 seconds. Refer Table 3 for a detailed description of the simulation parameters used.

Table 3 : Simulation parameters

<b>Parameter</b>	<b>Value</b>
Topology	1000m x 1000m
Bandwidth	11 Mbps
MAC protocol	802.11
Number of nodes	10,20,30,40,50
Mobility model	Propagation/ TwoRayGround
Pause time (s)	0.0
Routing protocol	AODV
Queue type	Queue/ DropTail/PriQueue
Radio transmission	100m
Simulation time (s)	1000
Traffic type	CBR
Password assigned	123

### 3.3. Proposed security solution

#### 3.3.1. Working mechanism of the proposed solution

All the known nodes in the network are assigned with a common key, using AES symmetric key cryptographic algorithm. Node should be configured with a key to provide Authentication. Nodes which aware of this key are consider as authenticated nodes.

When node requires a specific route to find, it sends a RREQ. In this proposed algorithm, author plans to add a new field to RREQ packet. It will generate a random string from the sequence number of the RREQ packet. And then, in a table called string table the sequence number and random string generated will be stored. After that, the source IP Address, sequence number and random string will be encrypted using the common key, which will be called as encrypted string, the encrypted string will put into the new field of the RREQ. Then the node will broadcast the RREQ packet.

When a node receives a RREQ, it generally checks whether it's the destination or has the path to the destination. In the proposed algorithm, it will perform an authentication. In order to decrypt the encrypted field node must know the common key. If it doesn't know, it is not an authenticated node it needs to forward it to other nodes. Once decrypted by the key, node will check whether the sender IP address is same with the decrypted field's IP address. If they match, node will be authenticated. The sender will be considered as trustworthy and processes the RREQ. Else it will drop the packet. This Process of dropping if RREQ is not authenticated will be able to prevent intruders from going inside the network and will likely reduce the unwanted traffic in the network.

Node sends a RREP if that node is a destination or has the specific path to the destination. In the proposed algorithm, a new field will be added to RREP packet. Received node IP address (source IP address) and random string from the decrypted RREQ will input to H-MAC hashing algorithm. This process will ensure that it knows the common key of the network. Then the hash value and random string will be encrypted using the common key. This encrypted value will put into the new field and sends back to the Sender of the RREQ. Use of hashing will prevent the Man in the Middle attacks.

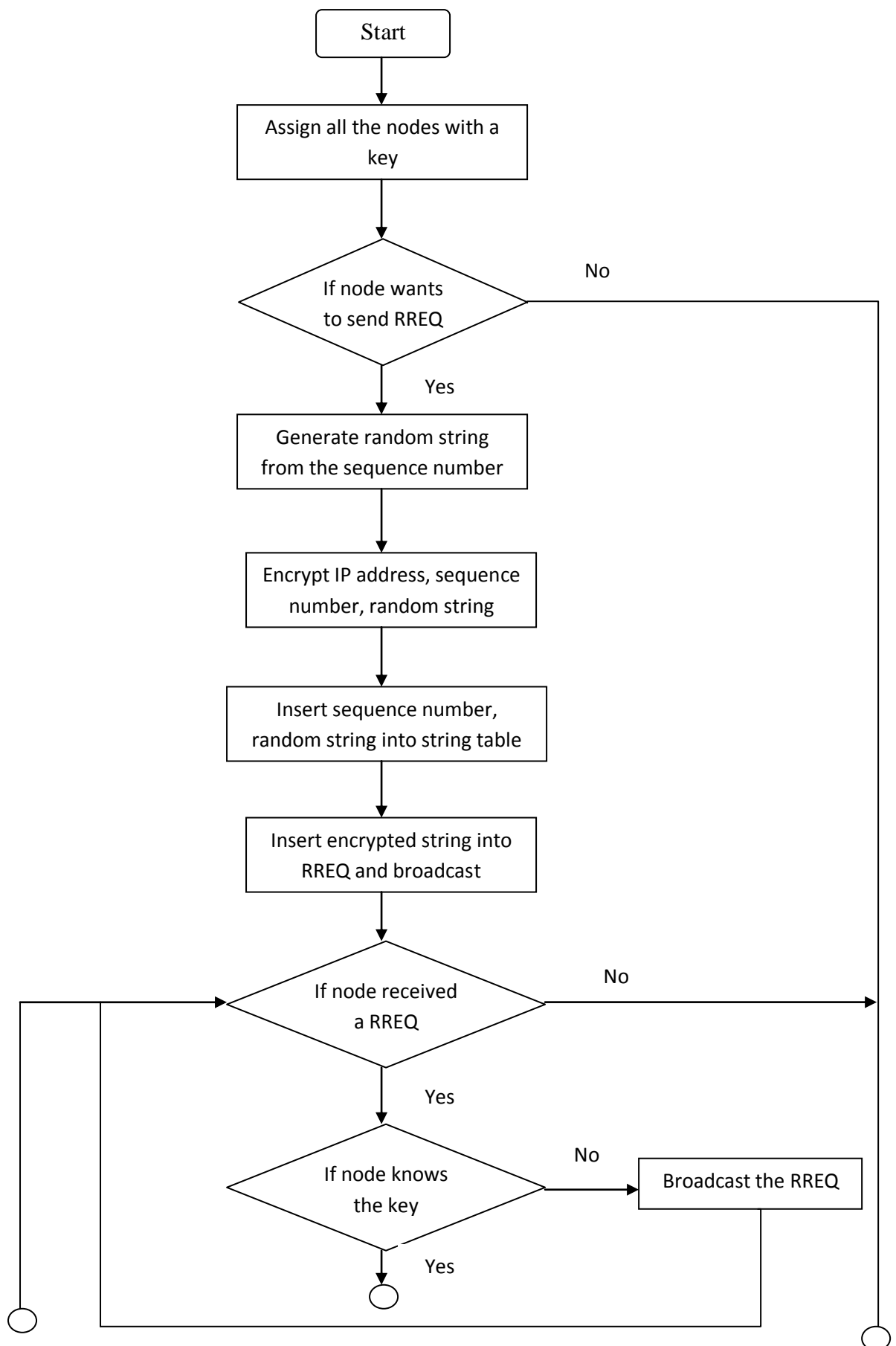
When a node receives a RREP, It will decrypt the encrypted field in the RREP using the common key. Then it will check the string table to identify whether it has generated any random string like the one in the received RREP. If so, it will process the hashing and check whether the two hash values are same. If so, accept the reply, which means the encrypted string is successfully decrypted at the receiver node therefore that node know the common key and it's an authenticated node so the sender node of RREQ will update the routing table. If not match, the packet will be dropped.

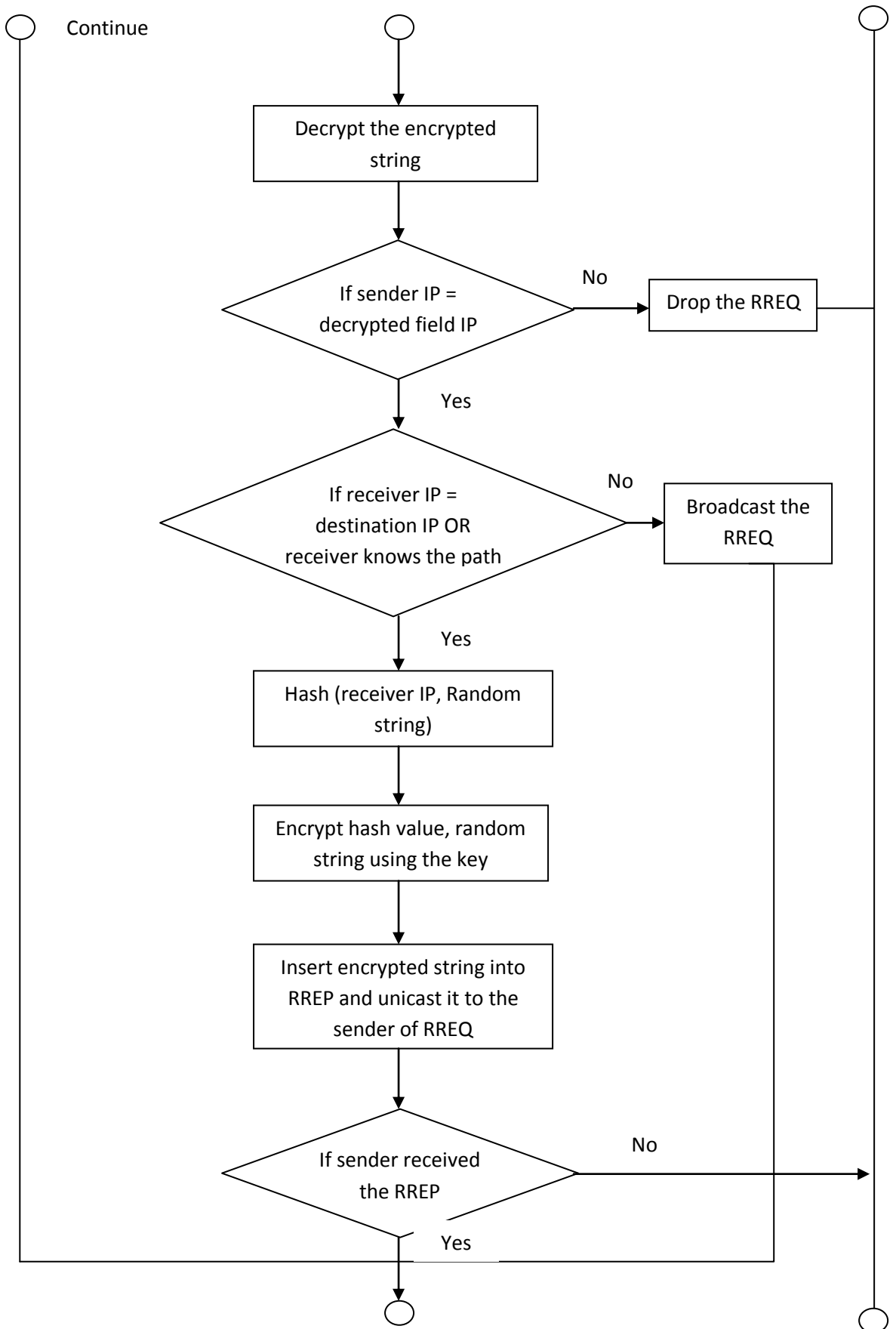
Diffie Hellman algorithm can be used to securely exchange the secret key between nodes.

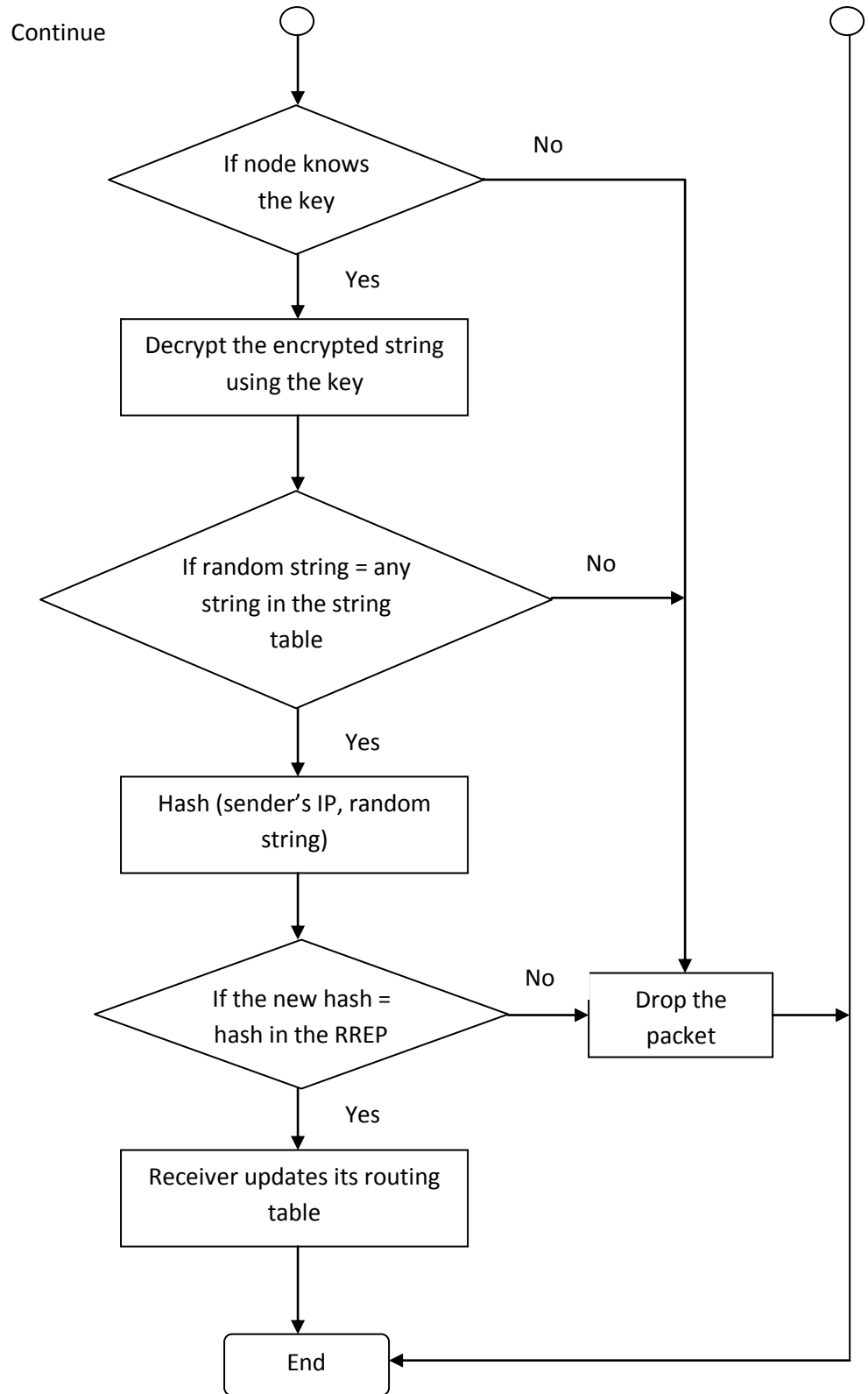
### 3.3.2. Pseudo code of the proposed solution

- Assign a common secret key 'K' to all the intended nodes
  - When a node needs to find a specific route to a destination
    - Generate a random string from the sequence number of RREQ
    - Save random string + sequence number in the table 'string table'
    - Encrypt (source IP+ sequence number+ random string) using 'K'
    - Add a new field to RREQ and insert the encrypted string into it
    - Broadcast the RREQ to network
  - When a node receives a RREQ
    - If (it has the common secret 'K')
    - Decrypt the encrypted string using 'K'
    - If (sender's IP = decrypted field's IP)
      - Node is authenticated
      - If ((receiver's IP = destination IP) || (receiver has a route to the destination))
        - Hash (receiver's IP + random string of the decrypted RREQ)
        - Encrypt (Hash value + random string) using 'K'
        - Add a new field to RREP and insert the encrypted value into it
        - Send the RREP to the sender of the RREQ
      - Else
      - Forward the RREQ to neighbor nodes
    - Else
    - Drop the packet
  - Else
  - Forward the RREQ to neighbor nodes
- When a node receives a RREP
  - If (it has the common secret 'K')
  - Decrypt the encrypted string using 'K'
  - If (random string = any string in the 'string table')
  - Hash (sender's IP + random string)
  - If (the new hash value = hash value in the RREP)
    - Accept the RREP
    - Receiver updates its routing table
  - Else
  - Drop the packet
- Else
- Drop the packet
- Else
- Drop the packet

### 3.3.3. Flowchart for the security solution



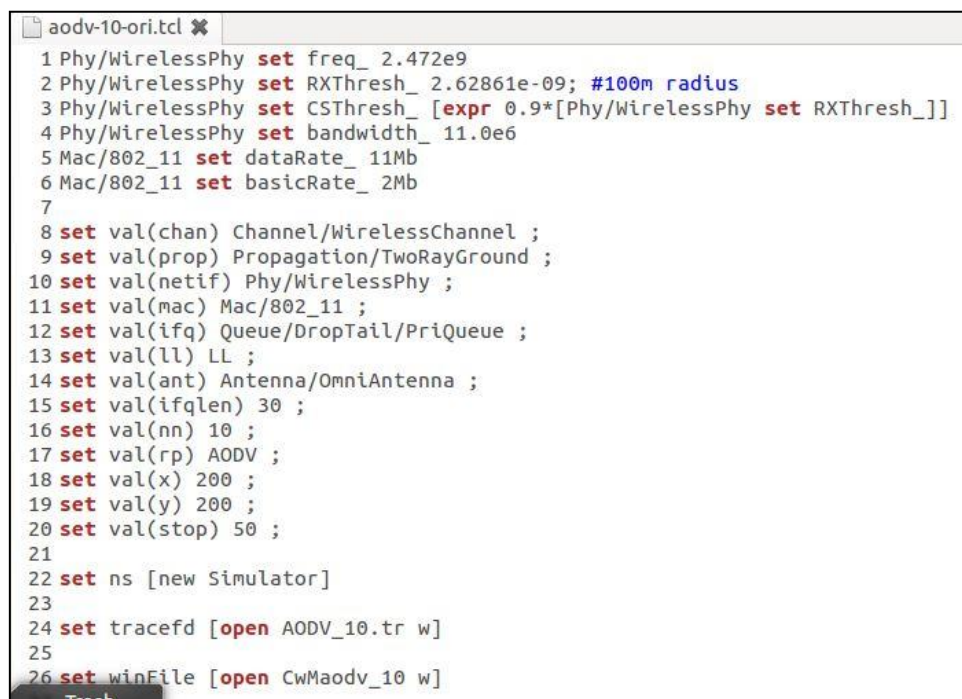




## Chapter 4: Implementation – A Secured Ad hoc Routing Protocol for Wireless Sensor Networks

### 4.1. Implementation on the simulation environment – NS2

The proposed security solution was developed on NS2, which allows researchers testing and validating the solution on various scenarios with different number of mobile nodes. The algorithm of the proposed solution was written in C++ by modifying the existing AODV algorithm. The simulation environment has been created using TCL scripts (refer Figure 6). Appendix A gives an overview of the TCL scripts which has been used.



```
aodv-10-ori.tcl ✖
1 Phy/WirelessPhy set freq_ 2.472e9
2 Phy/WirelessPhy set RXThresh_ 2.62861e-09; #100m radius
3 Phy/WirelessPhy set CSThresh_ [expr 0.9*[Phy/WirelessPhy set RXThresh_]]
4 Phy/WirelessPhy set bandwidth_ 11.0e6
5 Mac/802_11 set dataRate_ 11Mb
6 Mac/802_11 set basicRate_ 2Mb
7
8 set val(chan) Channel/WirelessChannel ;
9 set val(prop) Propagation/TwoRayGround ;
10 set val(netif) Phy/WirelessPhy ;
11 set val(mac) Mac/802_11 ;
12 set val(ifq) Queue/DropTail/PriQueue ;
13 set val(ll) LL ;
14 set val(ant) Antenna/OmniAntenna ;
15 set val(ifqlen) 30 ;
16 set val(nn) 10 ;
17 set val(rp) AODV ;
18 set val(x) 200 ;
19 set val(y) 200 ;
20 set val(stop) 50 ;
21
22 set ns [new Simulator]
23
24 set tracefd [open AODV_10.tr w]
25
26 set winFile [open CwMaadv_10 w]
```

Figure 6: Sample TCL Script

NS2 records information regarding packet transmission in a trace file (refer Figure 7). Appendix B describes the format of a trace file in detail. Trace files are raw data files which allows analyzing the data transmission in depth. In order to make that analysis process more efficient, the author has used AWK scripts. AWK is an interpreted programming language. It is a very powerful and specifically designed language for text processing. In this project the author has used the GNU AWK which is written and maintained by the Free Software Foundation (FSF). Appendix C gives an overview of the AWK scripts which has been used in the project.

```

r 0.100660094 _2_ RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 199.999584 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]]
s 0.100660094 _2_ RTR --- 0 AODV 44 [0 0 0 0] [energy 199.999584 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [2:255 0:255 30 0] [0x4 1 [2 4] 10.0000000] (REPLY)
r 0.100660197 _1_ RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 199.999584 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]]
r 0.100660232 _3_ RTR --- 0 AODV 48 [0 ffffffff 0 800] [energy 199.999584 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [0:255 -1:255 30 0] [0x2 1 1 [2 0] [0 4]]

```

Figure 7: Sample piece of a trace file

AODV source code which is used in this solution has been already developed according to the RFC 3561: “Ad hoc On-Demand Distance Vector (AODV) Routing” and as the RFC suggests, by default algorithm itself disables the use of hello packets. And for the sake of simplicity, the author has left the default configuration as it is.

When creating the simulation environment all the nodes are pre-assigned with a common key. In this project, in order to provide cryptographic functions such as encryption, decryption, hashing, etc. the author has made use of Openssl 1.0.0.a package.

#### 4.1.1. Assigning a common secret key

The assignment of shared secret key has been achieved in NS2 via TCL scripts. In the proposed solution each node is assigned with a common secret key which has to be used to achieve privacy for their communication. As an example, when two nodes needs to communicate with each other without revealing the content to unintended nodes, first of all these two nodes need to exchange a shared secret key which is only known to themselves. After that the sender can use this shared secret key to encrypt the message. The receiver can use the same key to decrypt it. Since the key is only known to these two nodes, others cannot decrypt and see the content of the message and hence confidentiality of the data transmitted can be achieved.

This goal has been achieved by assigning a key value to a variable which is available in MobileNode.h file which is inside the ns2.35 configuration folder as shown in Figure 8. This key value has been set through TCL script at run time of the simulation.

```

inline void getPassword(double *pw){
    *pw = password;
}

```

Figure 8: Assigning a key value to a variable



### 4.1.2. Generating a random string

This is achieved through random.h file as shown in Figure 9. Random.h file use a seed value to generate random strings. In this project the author uses the seed value to ensure that a particular RREQ has been sent using the same sequence number in a particular node will have the same random number in the encrypted string.

```
static const char alphanum[] = "0123456789" "!@#%$%^&* " "ABCDEFGHIJKLMNOPQRSTUVWXYZ" "abcdefghijklmnopqrstuvwxyz";
int stringLength = sizeof(alphanum) - 1;

char genRandom() {
    return alphanum[rand() % stringLength];
}

void genRandomString(char *newStr, const unsigned int size, int seed)
{
    srand (seed);
    for(unsigned int i=0; i < size; i++)    {
        newStr[i]=genRandom();
    }
}

void printRandomString (char str[], int len)
{
    for (int i = 0; i < len; i++)
    {
        printf ("%c",str[i]);
    }
    printf ("\n");
}
```

Figure 9: Generating a random string

### 4.1.3. Creating the string table

This table stores sequence number and the random string generated for each RREQ. Purpose of maintaining this table is to validate the RREP received for a specific RREQ. When a node receives a RREP it will check the string table to see whether it has ever generated a RREQ with such random string.

In order to achieve this goal, it is required to modify the aodv\_packet.h file. In NS2 this has been achieved by introducing Chal\_Info class as shown in Figure 10, which belongs to the aodv.cc file. By default the Chal\_Info class can store up to twenty strings sent from a particular node. However it can be increased or decreased according to the requirements.

```

ch = new Chal_Info[20];
for(int i=0; i<20; i++)
{
    ch[i].seqno = 0;
    ch[i].chal = new char[20];
    for(unsigned int j=0; j < 20; j++)    {
        ch[i].chal[j]='\0';
    }
}
position =0;
prevSeq = seqno;

```

Figure 10: Creating the sting table

#### 4.1.4. Implementing cryptographic functions

In NS2 Security.h file is responsible for encryption and hashing functions as shown in Figure 11, Figure 12 and Figure 13. It contains a set of methods which required for encryption, decryption and hashing. It makes use of the standard openssl library for that purpose.

```

int INITIALIZEAES(unsigned char *KeyData, int KeyLength, unsigned char *Salt, EVP_CIPHER_CTX *en_ctx, EVP_CIPHER_CTX *de_ctx)
{
    int rc;
    int Rounds = 10;
    unsigned char aes_key[32], aes_iv[32];
    //////////////////////////////////////
    //
    // Cipher Block chaining mode , SHA1
    rc = EVP_BytesToKey(
        EVP_aes_256_cbc(), // Cryptographic mode
        EVP_sha1(), // SHA1
        Salt, // a fuzzifier
        KeyData,
        KeyLength,
        Rounds, // more rounds
        aes_key,
        aes_iv
    ); // return buffers

    EVP_CIPHER_CTX_init(en_ctx);
    EVP_EncryptInit_ex(en_ctx, EVP_aes_256_cbc(), NULL, aes_key, aes_iv);

    EVP_CIPHER_CTX_init(de_ctx);
    EVP_DecryptInit_ex(de_ctx, EVP_aes_256_cbc(), NULL, aes_key, aes_iv);
    return 0;
}

```

Figure 11: Initialization of encryption and decryption

```

char *AES_Encrypt(char *msgtoencrypt,char *key){
    EVP_CIPHER_CTX en_ctx;
    EVP_CIPHER_CTX de_ctx;

    unsigned int Salt[] = {12345, 54321};
    unsigned char *KeyData;
    char *ciphermsg;
    int KeyDataLen;

    KeyData = (unsigned char *)key;
    KeyDataLen = strlen(key);
    // printf("Msg to be Encrypted : %s , Key : %s \n",(unsigned char *)msgtoencrypt,key);
    if (INITIALIZEAES(KeyData, KeyDataLen,(unsigned char *)&Salt, &en_ctx, &de_ctx))
    {
        printf("Couldn't initialize AES System\n");
        return 0;
    }

    int inlen;
    inlen = strlen(msgtoencrypt)+1;

    ciphermsg =(char *)Encrypt(&en_ctx, (unsigned char *)msgtoencrypt, &inlen);
    // printf("\n Encrypted Msg : %s", ciphermsg);
}

```

Figure 12: Sample code from encryption process

```

void calHash(char *key,const unsigned char ibuf[],unsigned char obuf[])
{
    HMAC(EVP_sha1(),key,strlen(key),ibuf, strlen((const char*)ibuf),obuf,NULL);
    return;
}

```

Figure 13: Sample code from hashing

#### 4.1.5. Implementing the security solution in AODV

The proposed solution has been developed and implemented in route discovery process of AODV protocol. The main configuration file which modified is aodv.cc. Following functions has been modified in order to embed the proposed solution.

- AODV::sendRequest(...) – responsible for sending RREQ
- AODV::recvRequest(...) – responsible for receiving RREQ
- AODV::sendReply(...) - responsible for sending RREP
- AODV::recvReply(...) - responsible for receiving RREP

The modifications which have been done in the above functions are presented in Apendix D.

In addition to that, the author has defined several variables in aodv.h file which supports the implementation of the security solution.

#### 4.1.6. Compiling source code

After doing required modifications in the source code files it is necessary to run the following commands in the terminal to apply the modifications.

- Make clean
- Make
- Make install

All the above commands are required to execute with super user privileges to apply the changes.

#### 4.1.7. Simulating and analyzing

If there are no errors present during the compilation process next step is to start the simulation process using the TCL scripts. In order to run the tcl script in NS2 it is required to issue the following command in the terminal;

- Ns <<simulation\_file.tcl>> <<number\_of\_nodes>>

After the simulation, NS2 generates a trace file which contains information about the packet transmission. In this project the author has used these trace files to measure the performance of the solution. To achieve this goal AWK scripts has been used. AWK script can extract only the required data from a given trace file to perform calculations. The following command has to be issued in order to run an AWK script against a trace file;

- Awk -f <<awk\_script\_file.awk>> <<trace\_file.tr>>

## Chapter 5: Testing, Evaluation and Validation

### 5.1. The plan

As the first step, the author has implemented the existing AODV routing protocol on top of NS2. It has been subjected to testing, evaluation and validation as bellow;

- Simulations have been carried out separately for a MANET which has 10, 20, 30, 40 and 50 mobile nodes in it.
- A qualitative analysis has been carried out to evaluate the security of the existing AODV protocol.
- Performance of the existing protocol has been measured as a quantitative analysis process to evaluate the current level of performance.

The proposed security solution has been tested, evaluated and validated as the second step.

- As previous section, simulations have been carried out for 10, 20, 30, 40 and 50 mobile nodes.
- A qualitative analysis has been performed to evaluate the security aspects of the proposed solution.
- Performance of the proposed solution has also been measured as a quantitative analysis to evaluate the adoptability of the solution.

### 5.2. Evaluation of security aspects

When considering the evaluation process of security aspects, it is not practical to carry out a quantitative analysis to measure the security or on the other hand insecurity. Therefore, according to the supervisor's guidance it has been decided to perform a qualitative analysis on both existing AODV protocol and proposed solution to evaluate the security aspects.

To achieve this, the author makes use of several case studies and explains how each protocol (existing AODV and proposed solution) behaves on a particular scenario so that evaluation can be done based on the behavior of the protocol.

### 5.2.1. Case Studies

The author has used real-world case studies with imaginary characters. Each case study addresses a unique information security related threat. In order to make it understandable for audience, the author has made the case studies simple as possible without damaging to its purpose. As an example even though the solution is applicable for MANETs which has more than 2-3 nodes, case studies has been designed with 2-3 mobile nodes to maintain the simplicity. The same case studies can be applied to MANETs with large amount of nodes.

In the case studies, the author has used imaginary names and each name represents his/her mobile node. As an example, “Alice sends a RREQ” can be interpreted as “the mobile node used by Alice sends a RREQ”. The case studies are presented in Chapter 6: General Discussion under the sub heading 6.1. On the security aspects of the proposed solution.

## 5.3. Evaluation of performance

A quantitative analysis has also been performed on both AODV protocol and proposed solution to measure the performance of each protocol. Afterwards, a comparison will be carried out between performance matrices of each protocol.

To achieve this goal, the author has used four performance matrices. The results are represented in the next chapter using graphs. In addition to that, the author has compared these results which received for both protocols in terms of the four performance matrices which are described in the next section.

### 5.3.1. Performance matrices

The following performance matrices have been used in this project to evaluate and get an idea about the performance of both AODV protocol and the proposed solution.

- |                          |  |
|--------------------------|--|
| End to end delay         | : The total amount of time taken by a data packet to reach its destination.  |
| Packet delivery fraction | : The ratio of the total number of packets received at the destination and the total number of packets sent by the source. |

Normalized routing overhead : The ratio of control packets sent and the number of packets delivered at the destination.

Throughput : The total amount of packets transfer through a link for a given period of time.

## Chapter 6: General discussion

### 6.1. On the security aspects of the proposed solution

#### 6.1.1. Case Study 1: Eavesdropping

##### **With AODV protocol without a security solution:**

Alice and Bob want to communicate with each other secretly. Alice broadcasts a RREQ to the network to find Bob. Eve is a third party who interests about Alice and Bob's conversation. She listens into this communication to obtain some sensitive information which can be used to intercept future communications between them.

##### **With the proposed security solution:**

Alice and Bob exchange a shared secret key "K" between them. Alice encrypts her IP address, sequence number and a random string using "K" and broadcasts the RREQ with the encrypted string to the network. Upon receiving the RREQ, Bob decrypts it using "K". Even though Eve wants to obtain the information in the encrypted string, she is unable to decrypt it because the shared secret "K" is known to Alice and Bob only.

Therefore, it can be observed that the proposed security solution provides confidentiality in route discovery process as expected.

#### 6.1.2. Case Study 2: Man-in-the-Middle

##### **With AODV protocol without a security solution:**

Alice and Bob want a secret communication channel between them. Eve is a third party who interests about this communication and wants to pretend to be Alice hence wants to remove Alice from the conversation. Therefore, after receiving Alice's RREQ, Eve changes the source IP address to her IP address and sends it to Bob. Bob continues the communication with Eve thinking that it is Alice.

##### **With the proposed security solution:**

Even though Eve modifies the source IP address she cannot modify the source IP address which is inside the encrypted string without the shared secret "K". Therefore, when Bob receives the RREQ he decrypts it, then he checks



whether the source IP address is equal to the decrypted field's source IP address. If they do not match, Bob drops the RREQ.

This case study proves that the proposed solution provides confidentiality and authentication to the route discovery process.

### 6.1.3. Case Study 3: Bogus Information

#### **With AODV protocol without a security solution:**

Alice and Bob want to communicate secretly. As an initial step Alice sends a RREQ to Bob. Eve is a third party who wants to disturb the communication. After receiving the RREQ, Eve creates a RREP with the encrypted string stolen from the Alice's RREQ and tries to pretend that she is the requested destination. Upon receiving this RREP, Alice updates her routing table with this wrong information.

#### **With the proposed security solution:**

Alice receives the RREP sent by Eve (as Bob) and decrypts the encrypted string using the shared secret "K". She checks whether the random string in the encrypted sting is available in her string table. In this case yes, because Eve did not change it. Then Alice hashes the sender's IP address and random string and checks whether its equal to the hash value in the RREP. At this point they are not equal. Because the hash value presented in the RREP has been created by Eve's end and she does not have any idea about the correct random string. Because Eve is unaware of the shared secret "K". If this happens Alice drops the RREP of Eve.

Therefore, it can be observed that this solution achieved the goal of providing integrity to the route discovery process.

## 6.2. On the performance of the proposed solution

After implementing the existing AODV protocol and the proposed solution in the simulator, number of simulations has been carried out in order to measure the performance of both AODV and the proposed solution. Readings of the performance matrices are included in Appendix E. The following section illustrates the results in a graphical manner.

### 6.2.1. End to End Delay

Figure 14 shows that end to end delay increased (as an average case) as the number of nodes increased in both protocols. The reason is the bandwidth reduction with the increment of nodes in the network. The proposed mechanism achieve slightly less end to end delay compare to AODV in the first three scenarios with number of nodes changing as 10, 20, 30. However, in last two scenarios with number of nodes changing as 40, 50, end to end delay of the proposed solution increased in a considerable amount.

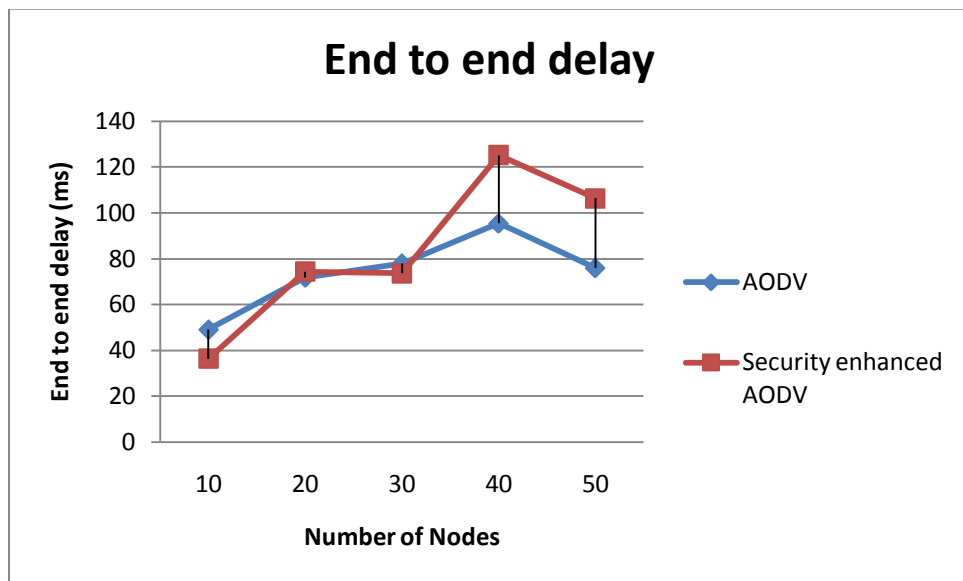


Figure 14: End to end delay vs. number of nodes

### 6.2.2. End to End Delay

Figure 15 shows a decrement of the packet delivery ratio with increment of the number of nodes in both protocols. The reason behind this is unsuccessful route repairs as the number of nodes increases. The simulation results show that the proposed algorithm has the ability to achieve very much similar result to that of the AODV algorithm in average cases (number of nodes 10, 20, 50).

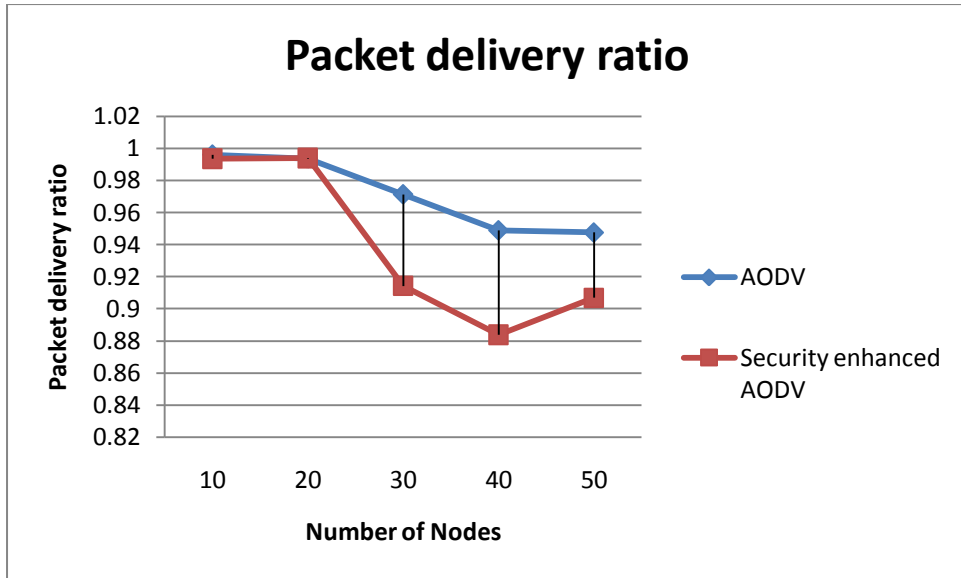


Figure 15: Packet delivery ratio vs. number of nodes

### 6.2.3. Normalized Routing Overhead

Results in the Figure 16 show that the normalized routing overhead of AODV protocol is slightly equal throughout the entire graph. In the proposed algorithm, normalized routing overhead of first two cases with number of nodes 10 and 20, are slightly equal to that of the AODV protocol. However, with the increase of the number of nodes as 30, 40 it gradually increases. This is due to the high amount of control overhead in the proposed security solution as the number of nodes increases.

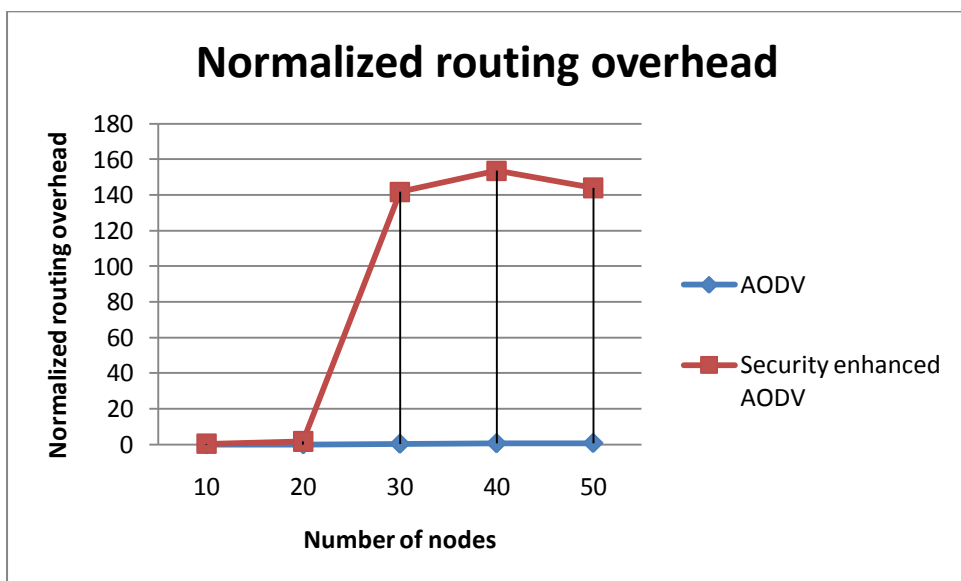


Figure 16: Normalized routing overhead vs. number of nodes

### 6.2.4. Average Throughput

According to the results in Figure 17, it can be observed that in both AODV and the proposed solution the average throughput decreases as the number of nodes increases. The reason is high control overhead as the number of nodes increases. And also the results show that the average throughput of the proposed algorithm is comparatively lower than that of the AODV protocol. The reason is the increment of the control overhead with the security implementation.

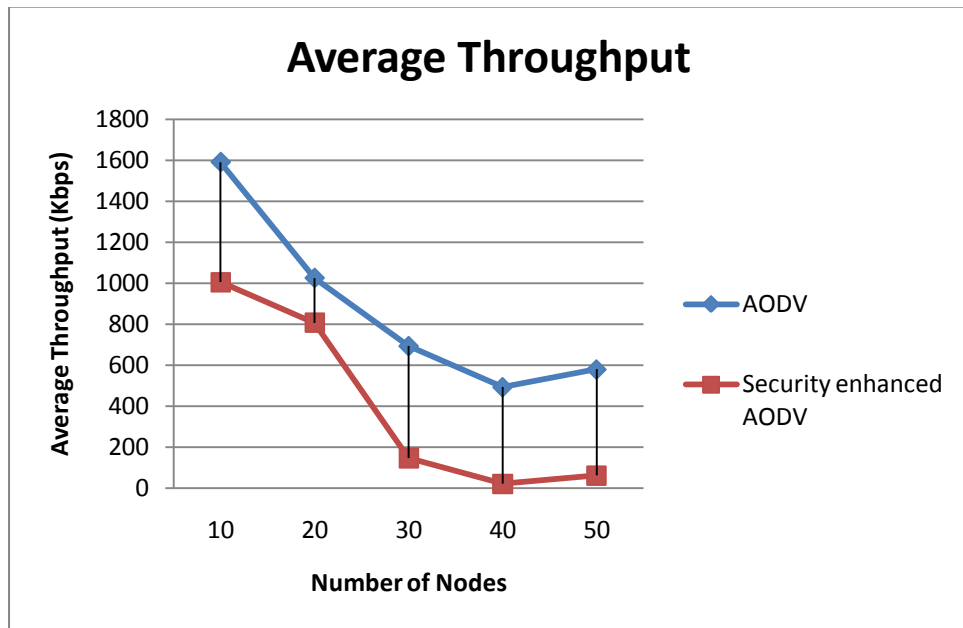


Figure 17: Average throughput vs. number of nodes

## **Chapter 7: Conclusion and future work**

### **7.1. Problems faced**

#### **7.1.1. Defining a boundary**

Both “Mobile Ad hoc Networks” and “Information Security” are topics which expanded in a vast area. The idea behind “A Secured Ad hoc Routing Protocol for Wireless Sensor Networks” is highly abstract and high-level. Even though the idea of embedding security attributes like confidentiality, integrity and authentication seems fairly simple task, implementing such attributes in a wireless network which tends to frequently change its network topology, does not contain a central coordination point, has comparably less network bandwidth due the shared medium and suffers from limited battery power was a very exciting but a challenging task.

#### **7.1.2. Maintaining a tradeoff between security and performance**

The most difficult part was to maintaining the main job of a MANET, which is providing data transmission while providing confidentiality, integrity and authentication to the data transmitted over the network. However, by trial and error the author has come up with the final solution described in this project.

#### **7.1.3. Evaluation**

Coming up with an evaluation method which can be used to evaluate security aspects was another challenging task. Because a quantitative analysis process may not be the best idea to measure the security or the insecurity of the solution. However, the author was advised by the project supervisor as well as the project evaluation penal (during the interim presentation), to carry out a qualitative analysis using case studies to support the project outcomes.

## **7.2. Lessons learned**

### **7.2.1. Understanding MANET operations**

Unlike conventional wired and wireless networks mobile ad hoc networks have unique and distinct characteristics which on the other hand make it one of the hottest topics in research community. While conducting this project the author has gained vast amount of knowledge in MANETs.

### **7.2.2. Understanding concepts of information security**

Even though the author is familiar with the information security concepts in general, this was her first attempt at applying security attributes into “mobile ad hoc networks” which enables her to get a more realistic and in depth understanding about it. Furthermore, it allows her to understand the limitations of embedding security into MANETs and how they can be improved.

### **7.2.3. Getting familiar with new technologies**

The author has gained lot of knowledge and experience in new computer technologies and their applications. As an example she has gained vast amount of experience in NS2 simulation environment. Furthermore, she had the opportunity to learn programming/ scripting languages such as C++, TCL scripting and AWK scripting.

## **7.3. Deviations from original project plan**

### **7.3.1. Evaluation techniques**

The original project proposal only included the quantitative analysis process which can be used to measure the performance. However it was later decided that it will be more practical to perform a qualitative analysis to evaluate the security of the solution.

### **7.3.2. Software version changes**

The author's original plan was to install and configure the latest release of NS2 which is version 2.35. But during the installation process it was unable to run the simulator and it has been found out that this particular version is not compatible with the operating system in use which is Ubuntu 12.04. Therefore, the author has installed and configured NS2 version 2.34 on the same operating system.

## **7.4. Extensions and future work**

### **7.4.1. Introducing more security attributes**

This project is focused on embedding confidentiality, integrity and authentication into the route discovery process of AODV protocol. It can be extended by introducing other important security attributes such as availability, authorization, non repudiation, etc...

### **7.4.2. Improving network performance**

According to the simulation results it can be observed that the proposed solution does not have a significant impact on the network performance when applied to a network with small amount of nodes. However, when the number of nodes increases the network performance significantly degrades. Therefore, research work can be carried out to improve the proposed solution to serve networks with large amount of nodes.

## **7.5. Conclusion**

This project presented a modified version of an existing AODV routing protocol. The proposed solution has been specifically designed for military surveillance systems or any system which seeks for confidentiality, integrity and authentication in their communication channel.

The proposed solution has been succeeded in achieving its goal as a routing protocol which provides confidentiality, integrity and authentication in route discovery

process. It can be used as a foundation to build a more secure and scalable routing protocol by improving and adding new features.



## Appendix A: TCL Scripting

### A.1. Brief Overview of TCL

TCL stands for Tool Command Language. It is a combination of a scripting language and its own interpreter which gets embedded to the application. TCL was originally designed by John Ousterhout of the University of California.

TCL was originally developed for Unix systems. Later, instances were created to serve Windows, DOS and OSX. TCL is very much similar to other Unix shell languages such as Bourne Shell (Sh), Korn Shell (sh) and C Shell (csh).

It allows programs to interact with other programs and also it act as an embeddable interpreter. The following section lists down its features;

- It is a free and opensource tool
- It supports Windows, Mac OS X, and almost every Unix platform
- It helps in reducing the development time
- Simple language which makes it easy to learn
- It includes a powerful set of networking functions

### A.2. TCL Script Used – explanation

#### **Defining the environmental settings;**

```
Phy/WirelessPhy set freq_ 2.472e9
```

```
Phy/WirelessPhy set RXThresh_ 2.62861e-09; #100m radius
```

```
Phy/WirelessPhy set CStresh_ [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
```

#### **Setting up values for the environment variables;**

```
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) 10 ;
set val(rp) AODV ;
set val(x) 200 ;
set val(y) 200 ;
set val(stop) 50 ;
set ns [new Simulator]
set tracefd [open AODV_10.tr w]
set winFile [open CwMaadv_10 w]
set namtracefd [open namwrls.nam w]

```

### **Node Configurations;**

```

$ns node-config -adhocRouting $val(rp) \
-lType $val(l) \
-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 "EnergyModel" \
-initialEnergy 200

```

### **Creating nodes and assigning them with a shared secret;**

```

for {set i 0} {$i < $val(nn)} {incr i} {
set node_($i) [$ns node]
}
for {set i 0} {$i < $val(nn)} {incr i} {
$ns at 0.0 "$node_($i) password 123"
}

```

### **Assigning starting locations for the nodes;**

```

$node_(0) set X_ 1.0
$node_(0) set Y_ 50.0
$node_(0) set Z_ 0.0

```

```
$node_(1) set X_ 60.0
$node_(1) set Y_ 50.0
$node_(1) set Z_ 0.0
```

```
$node_(2) set X_ 25.0
$node_(2) set Y_ 65.0
$node_(2) set Z_ 0.0
```

.....

```
$node_(9) set X_ 130.0
$node_(9) set Y_ 5.0
$node_(9) set Z_ 0.0
```

### **Initializing communication;**

```
$ns at 10.0 "$node_(2) setdest 135.0 65.0 8.0"
$ns at 10.0 "$node_(4) setdest 140.0 70.0 8.0"
$ns at 10.0 "$node_(8) setdest 125.0 100.0 8.0"
```

### **Finalizing communication;**

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

$ns run
```

# Appendix B: Trace Files

## B.1. Brief Overview of Trace Files

Trace file is a file written by an application, in this case NS2 to store overall network information. In order to create a trace file it is required to define it in the relevant TCL script as follows;

```
set tracefd [open AODV_10.tr w]
```

There are two types of trace files;

- I. New trace file format
- II. Old trace file format

In this project, the author will generate old trace file format.

## B.2. Old Trace File Format

Figure 10 is a snap shot of a trace file created according to the old trace file format.

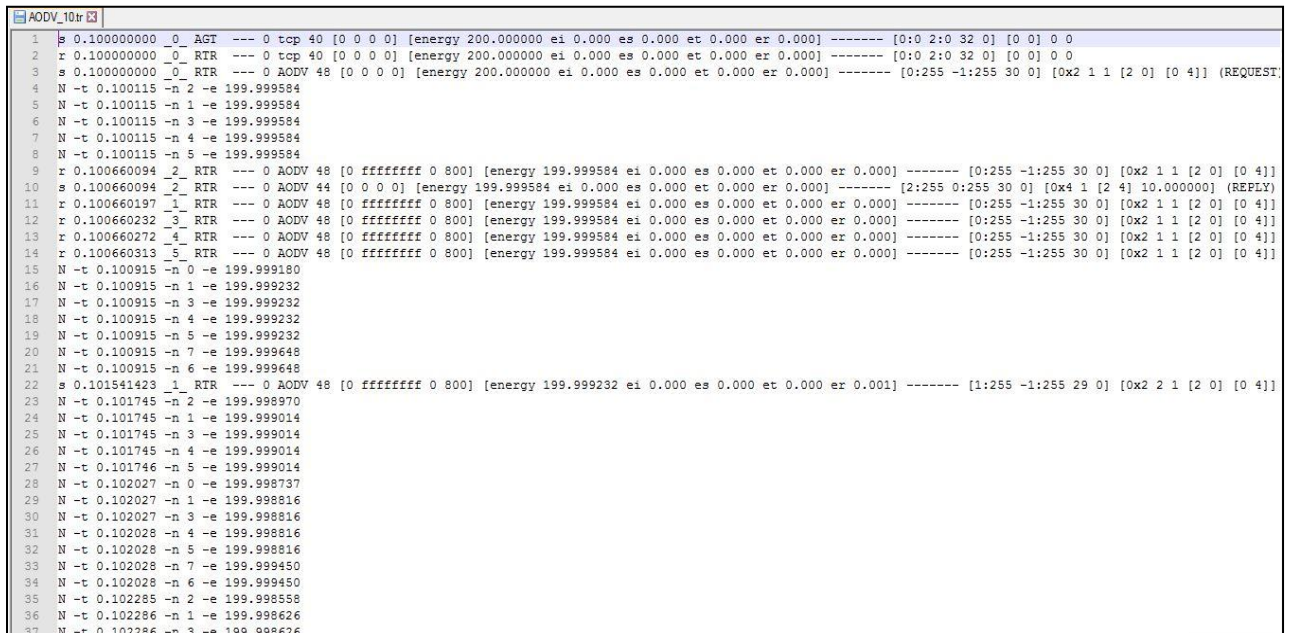


Figure 18: Snap shot of an old format trace file

There are twelve data fields in this trace file. The following table gives a brief idea about the fields in this trace file.

Table 4: Description of the fields

<b>Column No.</b>	<b>Description</b>
1	States whether this is a sent (s), received (r) or dropped (d) packet
2	Relative time the event has taken place
3	Node ID
4	States the layer which the event occurred; AGT – Application layer RTR – Routing layer MAC – MAC layer LL – Link layer IFQ – Interface Queue PHY – Physical layer
5	Delay over the underlying wireless channel
6	States the protocol type
7	Packet size
8	Duration of the packet, destination's MAC address, source's MAC address, MAC type
9	Energy level of the node
10	States information about the source node
11	Flag set
12	States the packet type

## Appendix C: AWK Files

### C.1. AWK Scripting Overview

AWK is a powerful method of processing or analyzing raw text files. AWK can be used in two ways;

- I. AWK command can be executed from the command line
- II. AWK can be executed as a program which known as AWK scripts

### C.2. AWK Scripts to Analyze Performance

This section explains the methods of calculating the network performance in terms of end to end delay, packet delivery ratio, normalized routing overhead and average throughput using AWK scripts.

#### **End to end delay:**

```
BEGIN {
seqno=-1;
dp=0;
rp=0;
cnt=0;
}
{
if($4=="AGT" && $1=="s" && seqno<$6)
{
seqno=$6;
}
else if(($4=="AGT")&&($1=="r"))
{
rp++;
}
else if($1=="D"&&$7=="tcp")
{
dp++;
}
#end_end delay
if($4=="AGT"&&$1=="s")
{
start_time[$6]=$2;
}
else if(($4=="AGT")&&($1=="r"))
{
end_time[$6]=$2;
}
```

```

else if(($1=="D")&&($7="tcp"))
{
end_time[$6]=-1;
}
}
END{
for(i=0;i<=seqno;i++)
{
if(end_time[i]>0)
{
delay[i]=end_time[i]-start_time[i];
cnt++;
}
else
{
delay[i]=-1;
}
}
for(i=0;i<=seqno;i++)
{
if(delay[i]>0)
{
ssdelay=ssdelay+delay[i];
}
}
ssdelay=ssdelay/(cnt+1);
printf( "average end to end delay= %.2f" ,ssdelay*1000 " ms");
print "\n";
}

```

### **Packet delivery ratio:**

```

BEGIN {
    sendLine = 0;
    recvLine = 0;
    fowardLine = 0;
}

$0 ~/^s.* AGT/ {
    sendLine ++ ;
}

$0 ~/^r.* AGT/ {
    recvLine ++ ;
}

$0 ~/^f.* RTR/ {
    fowardLine ++ ;
}

END {

```

```

    printf "cbr s:%d r:%d, r/s Ratio:%.4f, f:%d \n", sendLine, recvLine,
    (recvLine/sendLine),forwardLine;
}

```

### Normalized routing overhead:

```

BEGIN{
recvd = 0;##### to calculate total number of data packets received
rt_pkts = 0;##### to calculate total number of routing packets
received
}
{
##### Check if it is a data packet
if (( $1 == "r" ) && ( $7 == "cbr" || $7 == "tcp" ) && ( $4=="AGT" )) recvd++;

##### Check if it is a routing packet
if (($1 == "s" || $1 == "f") && $4 == "RTR" && ($7 == "AODV" || $7 == "message" ||
$7 == "DSR" || $7 == "OLSR")) rt_pkts++;
}

END{
printf("#####
#####\n");
printf("\n");
printf("          Normalized Routing Load = %.3f\n", rt_pkts/recvd);
printf("\n");
printf("#####
#####\n");
}

```

### Average throughput:

```

BEGIN {
    recvdSize = 0
    startTime = 400
    stopTime = 0
}

{
    event = $1
    time = $2
    node_id = $3
    pkt_size = $8
    level = $4

# Store start time
if (level == "AGT" && event == "s" && pkt_size >= 512) {
    if (time < startTime) {
        startTime = time
    }
}
}

```



```

    }

# Update total received packets' size and store packets arrival time
if (level == "AGT" && event == "r" && pkt_size >= 512) {
    if (time > stopTime) {
        stopTime = time
    }

    # Rip off the header

    hdr_size = pkt_size % 512
    pkt_size -= hdr_size

    # Store received packet's size
    recvdSize += pkt_size
}

}

END {
    printf("Average Throughput[kbps] = %.2f\t\t
    StartTime=%.2f\t\tStopTime=%.2f\n", (recvdSize/(stopTime-
    startTime))*(8/1000), startTime, stopTime)
}

```

## Appendix D: AODV Modifications

### D.1. AODV::sendRequest(...) Modifications

```
int size = 10;
char newStr[ size ];
for(int i=0; i < 50; i++) {
    newStr[i]='\0';
}
memset(newStr,0,size);
genRandomString (newStr,size,seqno);
if(seqno != prevSeq )
{
    setChal(seqno,newStr,size,position,index);
    prevSeq = seqno;

    if(position <19)
        position++;
    else
        position =0;
}
CurrentNode = (MobileNode*) (Node::get_node_by_address(index));
((MobileNode *) CurrentNode)-> getPassword(&password);
int intPassword = (int)password;
char key[20];
sprintf(key, "%d",intPassword);
char ChallengeToEncrypt[50];
char *keyp = key;
for(int i=0; i < 50; i++) {
    ChallengeToEncrypt[i]='\0';
}
sprintf(ChallengeToEncrypt, "%d %d %s",index,seqno,newStr);
char *EncryptedChallenge;
EncryptedChallenge = AES_Encrypt(ChallengeToEncrypt,keyp);
char *finalMsg;
finalMsg = AES_Decrypt(EncryptedChallenge,key);
for(int i=0; i<50; i++)
{
    rq->challenge[i]=EncryptedChallenge[i];
}
printf("\n System Common Key is %s Unencrypted Encrypted String is %s
Encrypted String is %s Request send from node %i ",keyp,
ChallengeToEncrypt, rq->challenge,index);
```

## D.2. AODV::recvRequest(...) Modifications

```
printf("Encrypte String Received : %s from node %d \n",rq->challenge,rq->rq_src);
CurrentNode = (MobileNode*) (Node::get_node_by_address(index));
((MobileNode *) CurrentNode)-> getPassword(&password);

char key[20];
int intPassword = (int)password;
sprintf(key,"%d",intPassword);
char *finalMsg=NULL;
char *MsgToDecrypt = rq->challenge;
finalMsg = AES_Decrypt(MsgToDecrypt,key);

printf("Encrypted String Received to Node %d from %d. Once Decrypted the Encrypted
String is %s \n",index,rq->rq_src,finalMsg);

char *sourceInChal = NULL;
sourceInChal= strtok(finalMsg, " ");
char *seqInChal = NULL;
seqInChal = strtok(NULL, " ");
char *randInChal = NULL;
randInChal = strtok(NULL, " ");
int ChalSeq = 0;
if(seqInChal != NULL)
ChalSeq = atoi(seqInChal);

char sourceInMsg[20];
for(int i=0; i<20; i++)
{
    sourceInMsg[i]=NULL;
}
printf(sourceInMsg,"%d",rq->rq_src);

if(sourceInMsg[0] != NULL && sourceInChal != NULL)
{
    if(strcmp(sourceInMsg,sourceInChal)!=0)
    {
        printf("Authentication failed from node %d at node %d . Request Ignored
\n", rq->rq_src,index);
        Packet::free(p);
        return;
    }
    else
    {
```

```

        printf("Received Request Successfully Authenticated for node %d at node
        %d Request Accepted \n",rq->rq_src,index);
    }

}
else
{
    printf("\n Critical Error Packet Dropped");
    Packet::free(p);
    return;
}

```

### D.3. AODV::sendReply(...) Modifications

```

printf("\n This is the received Encrypted String to be Hashed using H-MAC = %s Seq No =
%d",recvdChal,SeqNo);

```

```

if(recvdChal != NULL)
{
    rp->seqno = SeqNo;
    CurrentNode = (MobileNode*) (Node::get_node_by_address(index));
    ((MobileNode *) CurrentNode)-> getPassword(&password);
    int intPassword = (int)password;
    char key[20];
    sprintf(key,"%d",intPassword);
    char *keyp = key;
    calHash(keyp,reinterpret_cast< const unsigned char*>(recvdChal),(unsigned
char*)rp->responce);
}
else
{
    printf("\n Hash (H-MAC) Calculating Error When sending Responce");
    return;
}

printf("\n This is the Responce to be sent : ");
for (int i = 0; i < 20; i++) {
    printf("%02x ", rp->responce[i]);
}

printf("\n");

```

## D.4. AODV::recvReply(...) Modifications

```
if (ih->daddr() == index)
{
    printf("Reply Received from node %d with seq No : %d , Hash (H-MAC) : ",rp-
    >rp_src,rp->seqno);

    for (int i = 0; i < 20; i++) {
        printf("%02x ", rp->responce[i]);
    }

    printf("\n");

    for(int i=0; i<20;i++)
    {
        if(ch[i].seqno==rp->seqno)
        {
            printf("\n Sequence Number exists");
            char *chalHash;
            chalHash = new char[20];
            for(unsigned int j=0; j < 20; j++)
            {
                //      ch[i].chal[j]='\0';
                chalHash[j] ='\0';
            }

            CurrentNode = (MobileNode*)
            (Node::get_node_by_address(index));
            ((MobileNode *) CurrentNode)-> getPassword(&password);
            int intPassword = (int)password;
            char key[20];
            sprintf(key,"%d",intPassword);
            char *keyp = key;
            calHash(keyp,reinterpret_cast< const unsigned
            char*>(ch[i].chal),(unsigned char*)chalHash);
            printf("\n This is our entry in the String Table : ");
            for(int i = 0; i < 20; i++) {
                printf("%02x ", rp->responce[i]);
            }
            printf("\n");
            if(strcmp((const char*)chalHash,(const char*)rp->responce)==0)
            {
                printf("Encrypted String is Successfully Authenticated RREP
                Accepted\n");
            }
        }
    }
}
```

```
                break;
            }
        else
        {
            printf("\n Authentication Failed for Reply doesn't match Encrypted
String");
            Packet::free(p);
            return;
        }
    }
    if(ch[i].seqno == 19)
    {
        printf("\n Authentication Failed for Reply Sequence Number not in String
Table");
        Packet::free(p);
        return;
    }
}
}
```

## Appendix E: Readings of the Performance Matrices

### E.1. End to End Delay

Table 5: End to end delay simulation results

No. of Nodes	AODV	Security enhanced AODV
10	49.15	36.41
20	71.84	74.37
30	77.92	73.68
40	95.41	125.19
50	75.99	106.28

### E.2. Packet Delivery Ratio

Table 6: Packet delivery ration simulation results

No. of Nodes	AODV	Security enhanced AODV
10	0.9961	0.9936
20	0.9937	0.994
30	0.9714	0.9143
40	0.949	0.8838
50	0.9476	0.9068

### E.3. Normalized Routing Overhead

Table 7: Normalized routing overhead simulation results

No. of Nodes	AODV	Security enhanced AODV
10	0.008	0.382
20	0.038	1.726
30	0.343	141.803
40	0.65	153.653
50	0.778	143.987

## E.4. Average Throughput

Table 8: Average throughput simulation results

<b>No. of Nodes</b>	<b>AODV</b>	<b>Security enhanced AODV</b>
10	1590.23	1004.31
20	1025.79	807.21
30	693.47	146.33
40	493.23	20.65
50	579.3	60.35



## References

- [1] M.Yasir Malik. “An Outline of Security in Wireless Sensor Networks: Threats, Countermeasures and Implementation” Wireless Sensor Networks and Energy Efficiency: Protocols, Routing and Management, DOI:10.4018/978-1-4666-0101-7.ch024.
- [2] IEEE 802.11<sup>TM</sup>: Wireless LANs [Online]. Available: <http://standards.ieee.org/about/get/802/802.11.html>
- [3] Kevin Dunne, Elaine Roche, David O’Loughlin, Lewis Rhatigan. Bluetooth for Ad-hoc Networking [Online]. Available: <http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group3/index.html> [Accessed March 25, 2016].
- [4] Joe Tillison (2011, August 8). An Introduction to wireless sensor network concepts [Online]. Available: <http://eetimes.com/design/microcontroller-mcu/4218586/An-introduction-to-wireless-sensor-network-concepts>
- [5] Nikola Milanova, MiroslawMalek, Anthony Davidson, VeljkoMilutinovic, “Routing and Security in Moblie Ad Hoc Networks”, IEEE Computer Society, 2004.
- [6] Jyoti Neeli, Dr. N.K. Cauvery, “Comparative Study of Secured Routing Protocols in Wireless Ad hoc Networks: A Survey”, International Journal of Computer Science and Mobile Computing, Vol. 4, Issue. 2, February 2015, pp. 225-229.
- [7] [RFC6126] J. Chroboczek, “The Babel Routing Protocol”, RFC 6126, April 2011.
- [8] [RFC3626] T. Clausen and P. Jacquet, “Optimized Link State Protocol (OLSR)”, RFC 3626, October 2003.
- [9] [RFC3561] C. Perkins, E. Belding-Royer and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing”, RFC 3561, July 2003.
- [10] [RFC4728] D. Johnson, Y. Hu, and D. Maltz, “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4”, RFC 4728, February 2007.
- [11] [Internet Draft 2001, expired] V. Park and S. Corson, “Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification”, Internet Draft, July 2001.
- [12] Young-Bae Ko and Nitin H. Vaidya. Location-Aided Routing (LAR) in Mobile Ad Hoc Networks [Online]. Available: <https://www.sigmobile.org/awards/mobicom1998-student.pdf>
- [13] [Internet Draft 2002, expired] Zygmunt J. Haas, Marc R. Pearlman and Prince Samar, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, Internet Draft, July 2002.
- [14] Manel Guerrero Zapata, “Secure Ad hoc On-Demand Distance Vector Routing”, Mobile Computing and Communication Review, vol. 6, Number 3.
- [15] R. Balakrishna, U. Rajeswar Rao, Dr. Geethanjali, M. S. Bhagyashekar, “Comparisons of SAODV and TADOV, DSR Mobile Ad hoc Network Routing

Protocols”, International Journal of Advanced Networking and Applications, Vol. 2, Issue. 1, pp. 445-451.

[16] Xiaoqi Li, Michael R. Lyu, Jiangchuan Liu, “A Trust Model Based Routing Protocol for Secure Ad hoc Networks”, IEEEAC, 2004, pp. 1150.

[17] Vaishali Mittal, “Prevention and Elimination of Gray Hole Attack in Mobile Ad-Hoc Networks by Enhanced Multipath Approach”, International Journal of Advanced Research in Computer Engineering and Technology, Vol. 4, Issue 5, May 2015.

[18] Y. Haripriya, K.V. Bindu Pavani, S. Lavanya, V. Madhu Viswanatham, “A Framework for Detecting Malicious Nodes in Mobile Ad hoc Network”, Indian Journal of Science and Technology, Vol. 8 (S2), January 2015, pp. 151-155.

[19] Nitin Khanna, “Mitigation of Collaborative Black Hole Attack using TRACEROUTE Mechanism with Enhancement in AODV Routing Protocol”, International Journal of Future Generation Communication and Networking, Vol. 9, No. 1 (2016), pp. 157-166.

[20] Vincent Toubiana, Houda Labiod, “Towards a flexible security management solution for dynamic MANETs”, IEEE 2008.

[21] Sunil J. Soni, Suketu D. Nayak, “Enhancing Security Features and Performance of AODV Protocol under Attack for MANET”, International Conference on Intelligent Systems and Signal Processing, 2013.

[22] Tameem Eissa, Shukor Abd Razak, M. D. Asri Ngadi, “Enhancing MANET Security using Secret Public Keys”, International Conference on Future Networks, 2009.

[23] Yue Ai, Fuwen Pang, “Improved PKI Solution for Mobile Ad Hoc Networks”, IEEE 2010.

[24] Maqsood Razi, Jawaid Quamar, “A Hybrid Cryptography Model for Managing Security in Dynamic Topology of MANET”, IEEE 2008.

[25] Farid Bin Beshr, Ahmed Bin Ishaq, Saeed Aljabri, Tarek R. Sheltami, “A guard Node (GN) based Technique against Misbehaving Nodes in MANET”, Journal of Ubiquitous & Pervasive Networks, Vol. 7, No. 1, 2016, pp. 13-17.

# MIS 3104 Individual Projects

## Recommendation for Thesis Evaluation

Project Details (to be filled by the student, use font size 16)			
<b>Student Name</b>	W. M. S. S. Silva		
<b>Registration No.</b>	2014MIS021	<b>Index No.</b>	14770218
<b>Project Title</b>	Secured Ad Hoc Routing Protocol for Wireless Sensor Networks		
<b>Supervisor Name</b>	Dr. Kasun De Zoysa		
<b>Date</b>	19/03/2018		
Supervisor Recommendation for the Thesis (use a separate sheet if needed)			
Supervisor's Recommendation			
<b>Thesis is</b>	<input type="checkbox"/> Not Ready	<input type="checkbox"/> Ready After Modification	<input type="checkbox"/> Ready
	<b>for Evaluation.</b>		
<b>Research/System</b>	<input type="checkbox"/> Not Ready	<input type="checkbox"/> Ready After Modification	<input type="checkbox"/> Ready
	<b>for Evaluation.</b>		
<b>Student's Signature</b>		<b>Supervisor's Signature</b>	
<b>Date:</b>		<b>Date:</b>	
<b>Office Use only.</b>			