# SECURE NEIGHBOUR DISCOVERY
# IN
# WIRELESS MESH NETWORK

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

**Computer Science & Engineering**

Under the Supervision of

**Mr. RAVINDARA BHATT**

Assistant Professor (Grade II), Computer Science & Engineering

By

**LOKENDRA SINGH RATHOD**

Enrollment no. – 111336

To

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

# Certificate

This is to certify that project report entitled **“Secure Neighbour Discovery in Wireless Mesh Network”**, submitted by **Lokendra Singh Rathod (Enrollment No. – 111336)** in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date:

Mr. Ravindara Bhatt

Assistant Professor Grade II

JUIT, WAKNAGHAT

# Acknowledgement

I would like to express my special thanks of gratitude to my project guide **Mr. Ravindara Bhatt** Sir as well as **Prof. Dr. Satya Prakash Ghrera**, Head Of Department, Department of Computer Science & Engineering and **Prof. R.M.K. Sinha**, Dean CSE and IT Dept who gave me the golden opportunity to do this wonderful project on the topic Secure Neighbour Discovery in Wireless Mesh Network, which also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to them.

Secondly I would also like to thank my family and my friends who helped me a lot in doing this project and how to become successful.

I am making this project not only for marks but to also increase my knowledge.

THANKS AGAIN TO ALL WHO HELPED ME.

Date: Lokendra Singh Rathod

Enrollment no. - 111336

# Table of Content

# List of Figures

# List of Tables

# Abstract

Wireless Networks are increasingly being used for data monitoring in commercial, industrial, and military applications. Security is of great concern from many different viewpoints: ensuring that sensitive data does not fall into wrong hands; ensuring that the received data has not been doctored; and ensuring that the network is resilient to denial of service attacks [1].

Neighbourhood Discovery in Wireless network is of great importance as this will give only the legitimate nodes and all other malicious node are removed from the network. The key points which are being addressed in this report are security issues and threats and their counter measures and we have shown a privacy preservation mechanism (Penalty Based Routing Algorithm) which will ensure that the data which is being sent from source node to destination node is not been altered and monitored and also the identity of the source node is not revealed even if the intermediate nodes through which the data is transmitted is malicious. The report also addresses the method for protecting the network from packet dropping attack using Trust Value Algorithm and Ceasar Cipher encryption algorithm for encrypting the data.

# CHAPTER 1: An Overview

## 1.1 Introduction

### 1.1.1 Wireless Mesh Network

A wireless sensor network is a wireless network made of numerous small sensor nodes. The sensor nodes are self-contained units consisting of a battery, radio and sensors and a processor with minimal computation power. Thus, they are resource starved devices with a minimal amount of memory, energy and computation power.

It is a communication network made of numerous radio nodes organised in a mesh topology. These nodes perform a variety of functions including sensing, communication and computation. They may be static nodes which stay in a fixed position throughout their lifetime or mobile nodes which may move to various locations depending on the function that they need to perform.

Wireless sensor networks are a particular class of wireless ad-hoc networks which do not need any fixed routing units or base stations in order to communicate. Each sensor node can communicate with nodes that are within its communication range. We call nodes that are within the communication range of a sensor node as its one-hop neighbours, or just neighbours. Intermediate nodes perform the routing operation when a sensor node wants to communicate with a node that is not within its communication range and the communication is usually by wireless RF links that have a low bandwidth [1].

### 1.1.2 Mesh Network Topologies

A mesh network topology is a decentralised design in which each node on the network connects to at least two other nodes. A big advantage of this decentralised topology is that if any one of the node in the entire network fails to operate properly or treated as dead than other nodes can still communicate properly to each other i.e. no single point of failure in mesh network topologies.

For self-healing and optimization of ongoing process between the nodes mesh topologies are used, which are discussed as below:-

**Point to Point:**

Point to point provides high performance, high speed interconnections and dedicated connection between the nodes. It is very simplest form of wireless communications that enables two nodes for communication with each other. It is not highly scalable and relatively it can be deployed quickly.

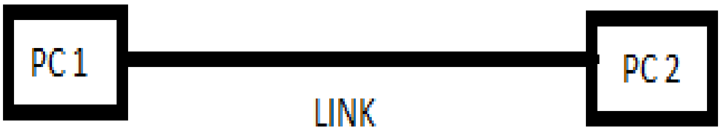


**Figure 1:** Point to point network.

**Point to Multipoint:**

In this type of topology we have more than one connection for a single node. By using multiple nodes a connection is being established between base station and other nodes. When a new user wants to enter in the existing network it can easily do that but user must be in the range of base station and subscriber requires only equipment for deployment at the user end, so this solution is best suited for backhaul operations like connection to main central site.

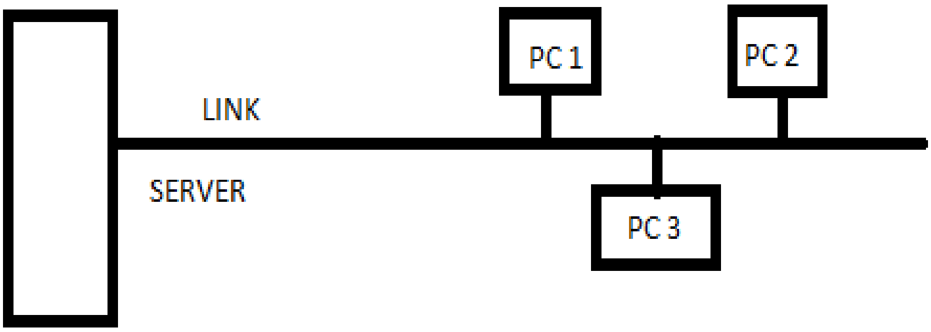


**Figure 2:** Point to Multipoint network.

**Multipoint to Multipoint:**

Data is routed between different nodes for the destination so for that a routed mesh topology is created for that purpose. Multiple access routers are deployed for maximum coverage and for high density, so all routers perform the functions for data through the network over multiple hops. User can join network anytime, anywhere in the entire mesh does not matter that the user is going to be connect through wireless or wired.

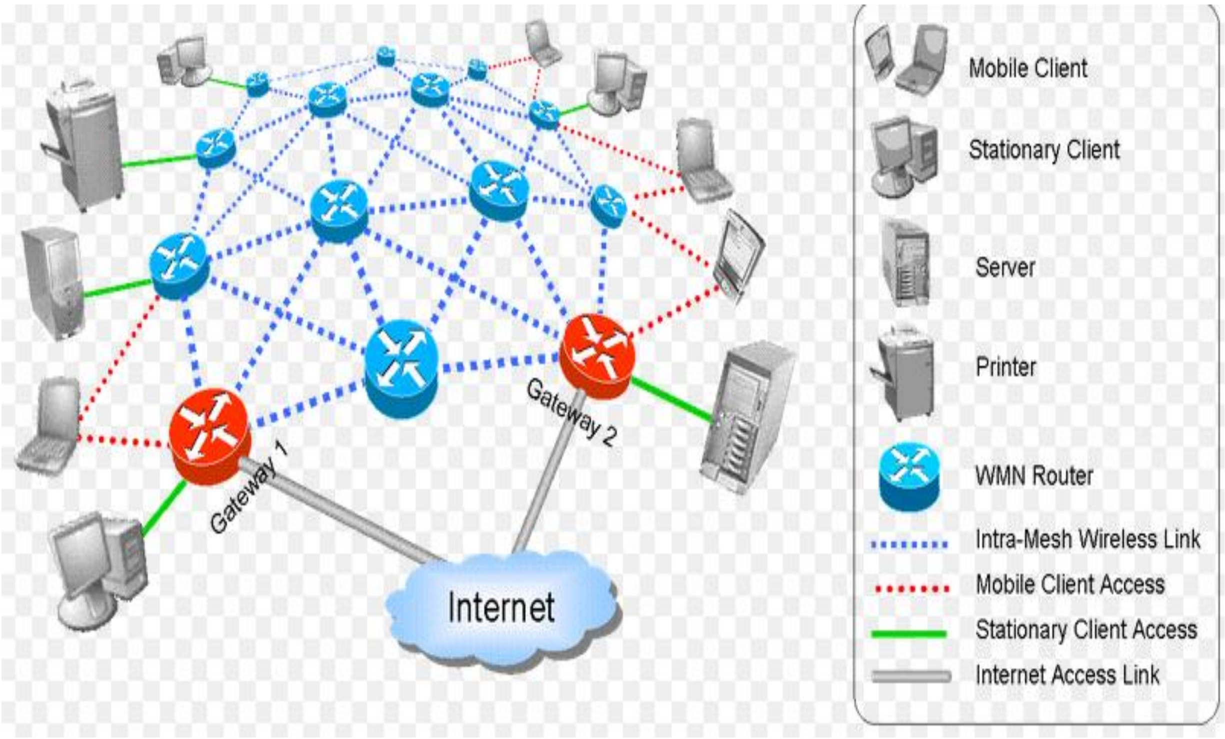


**Figure 3:** Wireless Mesh Network

### 1.1.3 Characteristics of Wireless Mesh Networks

→WMN‘s are considered to be a subclass of ad hoc networking

Routing nodes are stationary (unlike in Mobile Ad Hoc Networks, MANET‘s)

→WMN‘s have properties of an autonomic system:

- Self-Configuration
- Self-Healing (redundant, decentralized, no central point of failure)
- Self-Managment
- Self-Optimization

→High overall capacity:

- Spatial diversity
- Power management

**Important constraints:**

- Shared bandwidth & interference
- Number and location of nodes

## 1.1.4 Applications of Wireless Mesh Networks

**Broadband Internet Access (last mile)**

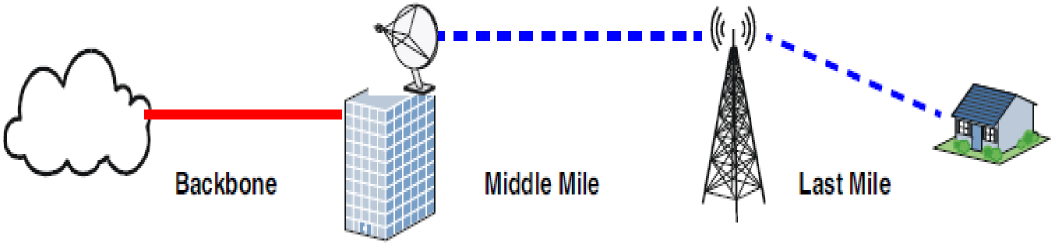


**Figure 4:** Wireless network application in Broadband Internet Access

→Wired infrastructure often too expensive for last and middle mile:

- Rural areas
- Weakly populated countries

→Cost factors in wired infrastructure:

- Number of endpoints
- Cable costs (length, unfriendly terrain)

→Issues in wireless solutions:

- Range and bandwidth (mesh networking is the key)
- Costs and maintenance requirements of hardware

### Community Mesh Network

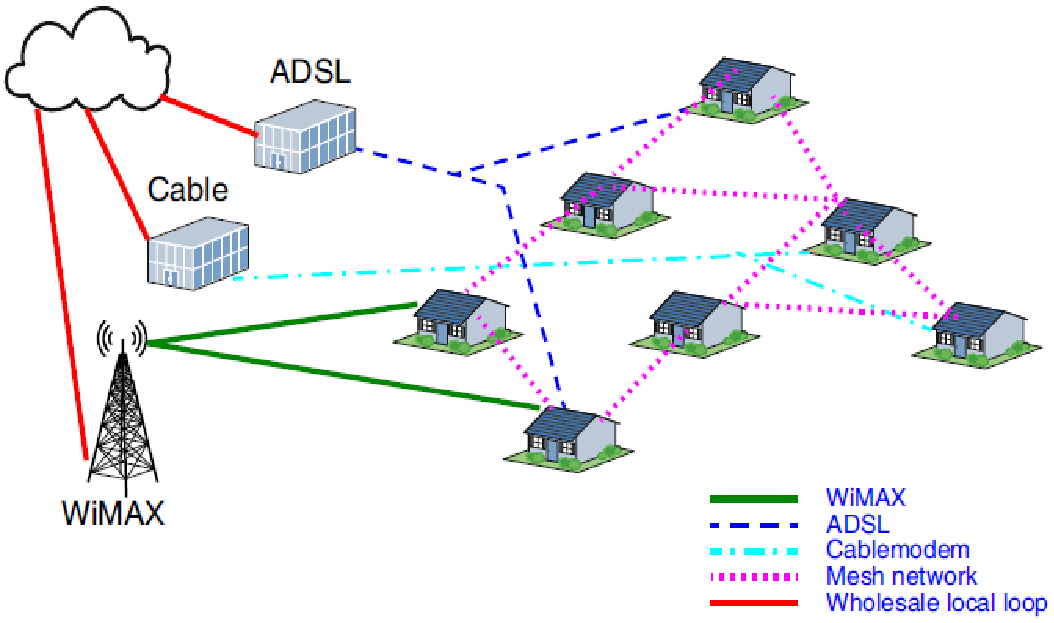


**Figure 5:** Mesh infrastructure owned by participants

### Industry Breakdown

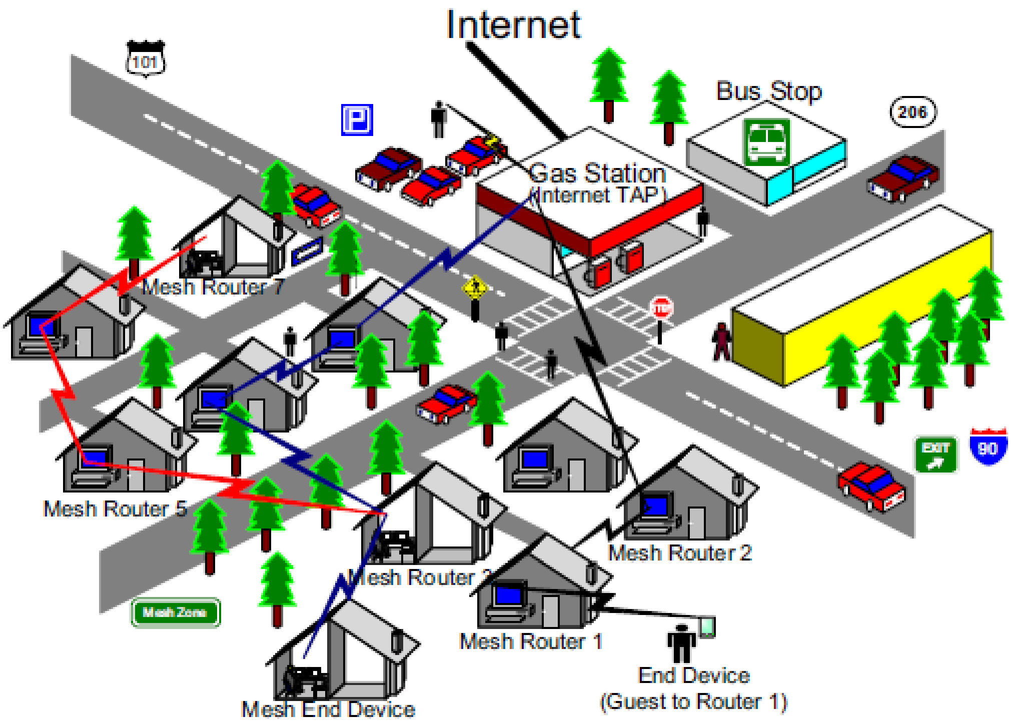


**Figure 6:** Wireless Mesh Network in Industry Breakdown

Wireless mesh based broadband access architecture:
→Master node: connects to wired internet
→Mini base station: mesh router rented to subscriber
→Franchising: system and business model rented to local service provider (usually as a “side business”)

## 1.1.5 Security Challenges in Wireless Mesh Networks

Security issues and the potential of WMNs cannot be ignored. In WMNs the understanding and properly addressing of these problems and challenges is very necessary. Due to dynamic change of network topology, distributed network architecture and shared wireless mediums WMNs lacks in security solutions. Attacks can occur on different protocol layers which can harm the network traffic and data. In wireless mesh there are different types of architecture which may uses different approaches for wireless mesh security purpose.

### 1.1.5.1 Basic Prevention

The primary issues which are very necessary for privacy preventions are as follows:-

**Data Confidentiality**

Its main purpose to prevent from eavesdropping and protect the data against the attacks. It is controlled by intermediate mesh routers/nodes. The algorithm by which one can protect the data from misbehaviours is message encryption so better the encryption technique better will be the confidentiality of that data. The best encryption technique today is Advanced Encryption Standard (AES).

**Traffic Confidentiality**

Traffic confidentiality is quite difficult to prevent against the attacks. For traffic confidentiality users must know that to whom they are communicating and their traffic patterns must be followed by the communicators. It is usually occurred by the attackers at mesh routers while traffic transfer. By following the key distribution mechanism WMNs can overcome on this type of attacks.

#### 1.1.5.2 Mesh Security

802.11s is a standard which will be followed in future for all kind of commercial mesh products. Right now mesh products are using different approaches for security and many of them may be derived from existing ad-hoc security mechanisms. 802.11s is a standard which will be based primarily on 802.11i security mechanisms.

The following security must be present in mesh network:

**Confidentiality**

In this the whole path should be protected and message should not be altered during the communication. Users must know each other for secure communication. The message and data information should not be disclosed. The data is only revealed to the intentional users and no illegitimate user can access or alter the data.

**Availability**

Insurance of authorized user actions can be done for secure communication. Provide the reliable delivery of data to the destination node. Protect the message and data against DoS (Denial of Service) attack in which the malicious node send floods of unwanted data to the receiver making him unreachable to get the data from authorized user.

**Authentication**

In WMNs authentication is very important because of change of shared medium. A proper mechanism should be followed for data sending and receiving. Users must know each other because it very necessary for reliable transmission of data. If user will not follow the any process then data may be infected or fabricated by anybody else which cause the problem in the network transmission.

**Authorization**

Users have the right to amend the data. If anybody wants to perform any task then there should be a proper process which ensures that the person have right to perform that task.

**Accounting**

If a user is using any service then there should be a process or method through which measurement of used resources can be done for billing information of specific user.

**Integrity**

Users cannot modify the data without having proper right to perform that task. If a user do not have right to perform any task then he/she cannot modify or change the message.

**Access Control**

User should ensure that only authorized actions can be performed, like if one cannot have authorization of changing the message then that user must be communicate with administrator for performing that task which he/she wants to perform.

### 1.1.6 Deployment

Sensor networks are used in commercial and industrial applications to monitor data that would be difficult or expensive to monitor otherwise. They could be deployed in wilderness areas, where they would remain for many years without the need to recharge or replace their power supplies. Since the environment that they monitor is generally hostile, it is usually not possible to deploy each node in a known location. Sensor nodes might therefore be scattered from a plane onto the region that they would be monitoring. Consequently, a sensor node, upon deployment, does not have knowledge of the nodes that are its neighbours [1].

### 1.1.7 Communication

Like wireless ad-hoc networks, sensor networks have two modes of communication.

1. Local Broadcast (One to Many)
2. Node to Node (One to One)

When a sensor node sends a packet by local broadcast, all its neighbours receive the packet. In the other case, the sensor node can send the packet to a specific node alone [1].

## 1.2 Motivation

The current neighbour discovery protocols are not 100% secure as the current NDP protocol assumes that the communication link is safe and reliable, which is not correct in reality.

Proposed system will provide:

- ➔ Better security even if the link is unreliable.
- ➔ It will be able to determine whether two nodes are neighbouring or non-neighbouring.

## 1.3 Problem Statement

The aim of the project is to design such a system which will:

- ➔ Prevent two non-neighbouring nodes from convincing themselves as well as their other actual neighbours that they are neighbours.
- ➔ Prevent the intermediate malicious nodes to drop/monitor the packets passing through it.
- ➔ Encrypt the data before passing so that no malicious node can access the data without the key.

## 1.4 Scope

The scope of this application can be very wide. If the system is designed accurately and exactly of what is thought to be design then it can solve a big problem of existing system i.e. the link reliability. This system will ensure that the source node and the destination node need not to worry about the intermediate node whether they are malicious or not because the data sent to them if passes through a malicious node the malicious node will not be able to gather much sufficient data to reveal their identity and also the full content of the transmitted data.

## 1.5 Organisation of the work

The report is organized as follows:

**Chapter 2** includes the Literature survey and gives us the full idea about the neighbour discovery phases and various cryptographic techniques.

**Chapter 3** gives various algorithms for searching the optimal path between source & destination nodes and protocols for transmitting the data or nonce (for discovering the neighbours).

**Chapter 4** presents the proposed work, flow charts and observations based on the proposed scheme.

**Chapter 5** concludes the report and discusses the future works that can be done to improve the proposed schemes.

# Chapter 2: Literature Survey

## 2.1 Secure Neighbour Discovery in Static Mesh Networks

Secure Neighbour Discovery in mesh network is of great importance since it requires only the secure node or non-malicious node in our network so that the chances of internal attack in the network is minimum.

This section suggests a protocol for secure one-hop neighbour discovery in WSNs in which the sensor nodes are static. One of the important characteristics of WSNs is that they are self- configuring, i.e., a large number of wireless nodes organize themselves to efficiently perform the tasks required by the application after they have been deployed. One-hop neighbours of a node are those which are within the radio communication range of the node. By secure neighbour discovery, we mean that for any node in the WSN, no node that is not within its one-hop communication range can become its neighbour. Malicious nodes that are within the communication range might not respond to Hello packets sent by certain nodes. If a node does not respond, it is only isolating itself and therefore cannot launch security attacks that are more devastating than when it responds to Hello packets [1].

### 2.1.1 Importance of Secure Neighbour Discovery

Knowledge of one-hop neighbours is essential for almost every routing protocol, MAC protocols and several other topology-control algorithms such as construction of minimum-energy spanning trees. Neighbour discovery is, therefore, a crucial first step in the process of self-organization of WSNs. Recently, neighbour discovery has also played a role in the security of wireless sensor networks, especially for mitigating control and data traffic attacks. Simple neighbour discovery has been found to significantly mitigate the wormhole attack in static sensor networks [2].

Since neighbour discovery is the first step performed by a sensor node upon deployment and since neighbour discovery requires a very small amount of time, it might be difficult for an adversary to compromise a lot of nodes before neighbour discovery is performed by the entire network. But the compromise of even a single node during neighbour discovery can prove significantly advantageous to the adversary to attack a variety of existing routing protocols.

Also, even external malicious nodes (nodes that do not possess the cryptographic keys) can significantly affect neighbour discovery protocols. They just need to relay packets between two non-neighbouring nodes and make them believe that they are neighbours. False neighbour discovery will also make protocols that trust on accurate neighbour discovery, like protocols that fight against wormhole attacks and certain routing protocols completely useless. To understand this, let us consider the following examples.

Let there be two legitimate sensor nodes, A and B, which are not within communication range of each other and an adversary M which is within communication range of both A and B, as shown in Figure 7. During neighbour discovery phase, M can fool A and B to believe that they are neighbours by relaying packets between them. After neighbour discovery, since A and B believe that they are neighbours, all communication between them gets controlled by the adversary M. If M colludes with another malicious node, the situation becomes worse. Colluding malicious nodes can make even legitimate nodes that are very far from each other to believe that they are neighbours. This is illustrated in Figure 8. Once a malicious node or a set of colluding malicious nodes make two non-neighbour legitimate nodes to believe that they are neighbours, they can easily create a wormhole and launch a variety of attacks against the data traffic flowing on the wormhole, such as selectively dropping the packets.

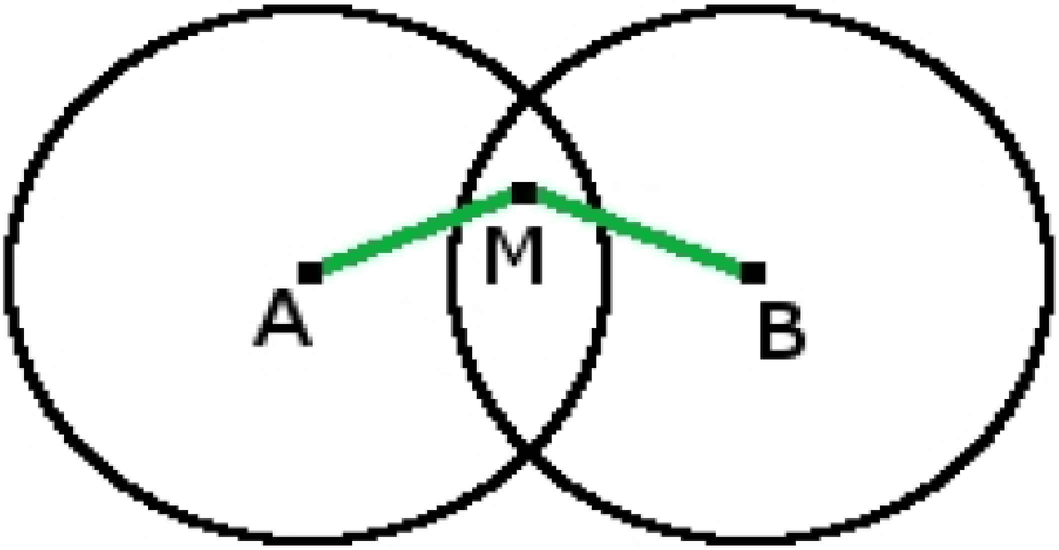


**Figure 7:** A malicious node M, fooling two legitimate non-neighbour nodes A and B to become neighbours. The communication range of A and B have been abstracted using circles of equal radii.

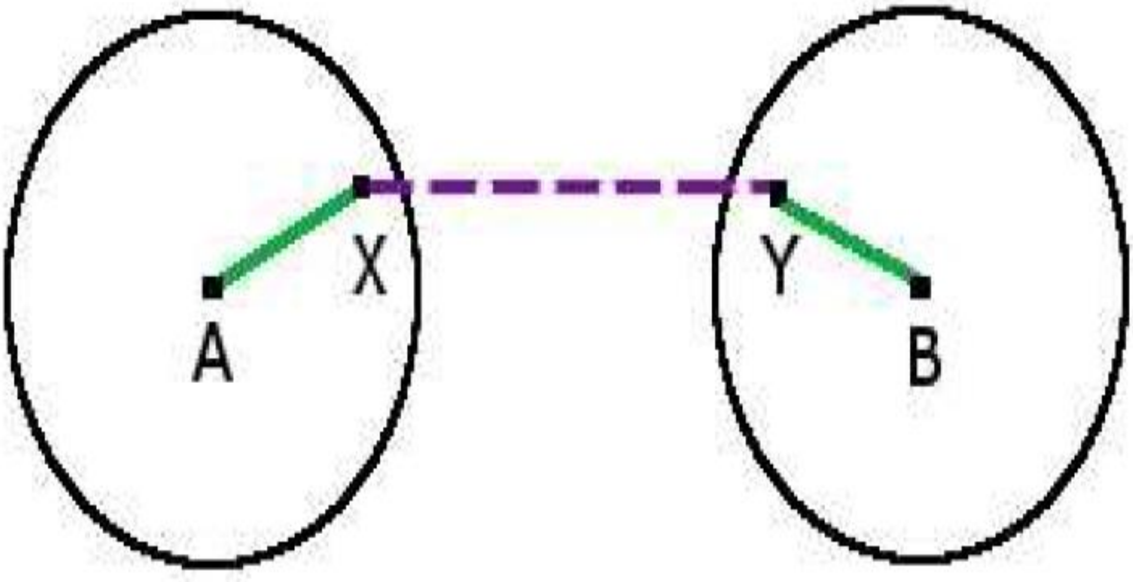


**Figure 8:** Two malicious nodes, X and Y, fooling two nodes A & B, which are far away, to become neighbours. The communication range of A and B have been abstracted using circles of equal radii.

Therefore, secure neighbour discovery is of immense importance in WSNs.

### 2.1.2 The Overhearing Technique

The advantage of using omnidirectional antennas is that, when a node sends a packet, all its neighbours can hear the node sending the packet. The identity of the node can be verified using existing cryptographic techniques. Such a technique can be used to verify whether or not a link exists between two nodes. In order for a node to verify whether a link exists between two nodes, it must be within the communication range of both the nodes. We call such nodes as verifiers.

In order to perform link verification, each node requires two pieces of information.

- Each node needs to find the nodes that claim to be its neighbours.
- Each node needs to know the neighbours of each of its neighbours.

Neighbour verification can then be performed to determine whether the nodes that claimed to be neighbours of a particular node are actually its neighbours.

But for the purpose of monitoring a link, for example, in a protocol like LiteWorp [3], each node also needs to determine the actual neighbours of each of its neighbours. In order for a node, X, to verify whether its neighbouring node, Y, is actually transmitting to one of the neighbours of Y (say Z), X also needs to know the neighbour list of Z. Then, X will know the verifiers of the link from Y to Z and can hence use their response to determine whether the link from Y to Z actually exists.

The neighbour discovery protocol is divided into two phases:

1. The Neighbour Discovery Phase
2. The Neighbour Verification Phase

#### 2.1.2.1 The Neighbour Discovery Phase

**Determination of the expected 1-hop neighbours:** In this phase, each node finds the nodes that claim to be its neighbours. Upon deployment, each node broadcasts a Hello packet and its node ID. Every node that hears this Hello packet sends back its ID and a reply containing a nonce which is authenticated using the key that is shared between the nodes.

**Determination of the expected 2-hop neighbours:** Once each node has found its expected list of neighbours, they need to know the neighbours of each of the nodes in this list to determine the verifiers. The verifiers will be used in the Neighbour Verification phase to decide whether two nodes are actually neighbours.

#### 2.1.2.2 The Neighbour Verification Phase

Once each node has completed the neighbour discovery phase, it can determine the verifiers for each of its links. Furthermore, it can also determine the links for which it is a verifier of and who the other verifiers of the link are.

In this phase, we need each node to explicitly announce the destination to which it sends the verification packet.

## 2.2 Cryptographic Technique

### 2.2.1 Definition

Cryptography (or cryptology; from Greek kryptós, "hidden, secret"; and gráphein, "writing", or -logia, "study", respectively) is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analysing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. It also provides more security to the data we want to deliver over a network for which we are not sure about its security so it make the data more secure for transmission over these type of networks.

- ➔ Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering.
- ➔ Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

### 2.2.2 Concepts & Items

- ➔ Plain Text and Cipher Text
- ➔ Key and Key Space
- ➔ Cryptosystem Services
  - • Confidentiality, Integrity, Authenticity, Non-repudiation, Access Control
- ➔ Cryptographic Methods
  - • Symmetric, Asymmetric
- ➔ Attributes of Strong Encryption
  - • Confusion, Diffusion

### 2.2.3 Encryption Algorithms

Encryption is the conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties.

The primary purpose of encryption is to protect the confidentiality of digital data stored on computer systems or transmitted via the Internet or other computer networks. But today these encryption techniques are used in all types of digital and non-digital types of communication. Modern encryption algorithms play a vital role in the security assurance of IT systems and communications as they can provide not only confidentiality, but also the following key elements of security:

- ➔ Authentication: the origin of a message can be verified.
- ➔ Integrity: proof that the contents of a message have not been changed since it was sent.
- ➔ Non-repudiation: the sender of a message cannot deny sending the message.

#### 2.2.3.1 Caesar Cipher

In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.

With a Caesar cipher, you replace each letter in a message with a letter further along in the alphabet. A Caesar cipher shifts the alphabet and is therefore also called a shift cipher. The key is the number of letters you shift. Caesar cipher is one of the oldest types of ciphers. It is named after Julius Caesar, who is said to have used it to send messages to his generals over 2,000 years ago.

To **encrypt** with a cipher wheel, turn the inner wheel counter-clockwise, shifting it the number of letters specified by the key. Then they match plaintext and ciphertext letters as shown on the wheel.

To **decrypt**, read the wheel backwards: after setting the wheel to the appropriate key, they begin with a ciphertext letter on the inner wheel and match it with the corresponding plaintext letter on the outer wheel.

The one advantage of this encryption technique is the ability to use this cipher without the need to send any information about the key to decrypt it.

#### 2.2.3.2 Vigenere Cipher

The Vigenère cipher, was invented by a Frenchman, Blaise de Vigenère in the 16th century. It is a polyalphabetic cipher because it uses two or more cipher alphabets to encrypt the data. In other words, the letters in the Vigenère cipher are shifted by different amounts, normally done using a word or phrase as the encryption key.

The maths behind the Vigenère cipher can be written as follows:

To encrypt a message: Ca = Ma + Kb (mod 26)

To decrypt a message: Ma = Ca – Kb (mod 26)

(Where C = Code, M = Message, K = Key, and where a = the ath character of the message bounded by the message, and b is the bth character of the Key bounded by the length of the key.)

The main disadvantage of this technique is that it is more prone to brute force attack in which the attacker tries every possible combination of key to decrypt the cipher-text until the original message is revealed.

#### 2.2.3.3 Data Encryption Standard (DES)

Originally designed by researchers at IBM in the early 1970s, DES was adopted by the U.S. government as an official Federal Information Processing Standard (FIPS) in 1977 for the encryption of commercial and sensitive yet unclassified government computer data. It was the first encryption algorithm approved by the U.S. government for public disclosure.

DES works by using the same key to encrypt and decrypt a message, so both the sender and the receiver must know and use the same private key.

The Data Encryption Standard is a block cipher, meaning a cryptographic key and algorithm are applied to a block of data simultaneously rather than one bit at a time.

To encrypt a plaintext message, DES groups it into 64-bit blocks. Each block is enciphered using the secret key into a 64-bit cipher-text by means of permutation and substitution. The process involves 16 rounds and can run in four different modes, encrypting blocks individually or making each cipher block dependent on all the previous blocks. Decryption is simply the inverse of encryption, following the same steps but reversing the order in which the keys are applied.

#### 2.2.3.4 Advanced Encryption Standard or AES

The origins of AES date back to 1997 when the National Institute of Standards and Technology (NIST) announced that it needed a successor to the aging Data Encryption Standard (DES) which was becoming vulnerable to brute-force attacks.

AES comprises three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively.

Symmetric or secret-key ciphers use the same key for encrypting and decrypting, so both the sender and the receiver must know and use the same secret key. All key lengths are deemed sufficient to protect classified information up to the "Secret" level with "Top Secret" information requiring either 192- or 256-bit key lengths.

There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys -- a round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of cipher-text.

In the next chapter, we will study about various existing protocols and algorithms related to neighbour discovery.

# Chapter 3: Route searching algorithms & protocols

## 3.1 Algorithm for searching routes (paths) between source & destination node

### i) Depth-first search

DFS starts the traversal from the root node and explore the search as far as possible from the root node i.e. depth wise. The only catch here is, unlike trees, graphs may contain cycles, so we may come to the same node again. To avoid processing a node more than once, we use a boolean visited array.

Time Complexity: O(V+E) where V is number of vertices in the graph and E is number of edges in the graph.

It can also be used for finding the paths from a single node to a single destination node by stopping the algorithm once the path to the destination node has been determined.

If there is no path from the source vertex to vertex v', then v's distance is infinite and its predecessor has the same special value as the source's predecessor.

Following are the problems that use DFS as a building block:

→For an unweighted graph, DFS traversal of the graph produces the minimum spanning tree and all pair shortest path tree.

→Detecting cycle in a graph.

→Path Finding.

→In Connectivity testing.

**Algorithm:**

```
dfs(vertex v,vertex x)

{
    visit(v);
    for each neighbor w of v
       while w!=x
             if w is unvisited
```

```
                {
                dfs(w);
                add edge vw to tree T
                }
}
```

## 3.2 Protocols for discovering the neighbours securely and transmitting the data to the neighbours safely:

- **PANA** (Protocol for carrying Authentication for Network Access)
  In (Cheikhrouhou et al., 2006), a security architecture has been proposed that is suitable for multi-hop WMNs employing PANA (Protocol for carrying Authentication for Network Access) (Parthasarathy, 2006). In the scheme, the wireless clients are authenticated on production of the cryptographic credentials necessary to create an encrypted tunnel with the remote access router to which they are associated. Even though such framework protects the confidentiality of the information exchanged, it cannot prevent adversaries to perform active attacks against the network itself.

- **LHAP** (Light-weight hop-by-hop access protocol)
  A light-weight hop-by-hop access protocol (LHAP) has been proposed for authenticating mobile clients in wireless dynamic environments, preventing resource consumption attacks (Zhu et al., 2006). LHAP implements light-weight hop-by-hop authentication, where intermediate nodes authenticate all the packets they receive before forwarding them. LHAP employs a packet authentication technique based on the use of one-way hash chains. Moreover, LHAP uses TESLA (Perrig et al., 2001) protocol to reduce the number of public key operations for bootstrapping and maintaining trust between nodes.

- **AAA** (authentication, authorization and accounting)

  In (Prasad et al., 2004), a lightweight authentication, authorization and accounting (AAA)infrastructure is proposed for providing continuous, on-demand, end-to-end security in heterogeneous networks including WMNs. The notion of a security manager is used through employing an AAA broker. The broker acts as a settlement agent, providing security and a central point of contact for many service providers.

# Chapter 4: Proposed work

**Step 1:** In this project, we have simulated a 2D mesh network contained the nodes as internetworking devices and edges as the link to other surrounding nodes (we can simulate the mesh network upto n nodes in our project).

Code snippet for generatind the 2D mesh is as follows:

```
public class GraphGenerator
{
    private Map<String, LinkedHashSet<String>> map = new HashMap();

    public void addNode(String node1)
{
        LinkedHashSet<String> adjacent = new LinkedHashSet<>();
        map.put(node1, adjacent);
    }

    public void addEdge(String node1, String node2)
 {
        LinkedHashSet<String> adjacent = map.get(node1);
        if(adjacent==null) {
            adjacent = new LinkedHashSet();
            map.put(node1, adjacent);
        }
        adjacent.add(node2);
    }

    public void addTwoWayVertex(String node1, String node2)
 {
        addEdge(node1, node2);
        addEdge(node2, node1);
    }
    public boolean isConnected(String node1, String node2)
{
        Set adjacent = map.get(node1);
        if(adjacent==null) {
            return false;
        }
        return adjacent.contains(node2);
}

    public LinkedList<String> adjacentNodes(String last)
   {
        LinkedHashSet<String> adjacent = map.get(last);
        if(adjacent==null) {
            return new LinkedList();
        }
        return new LinkedList<String>(adjacent);
    }
}
```

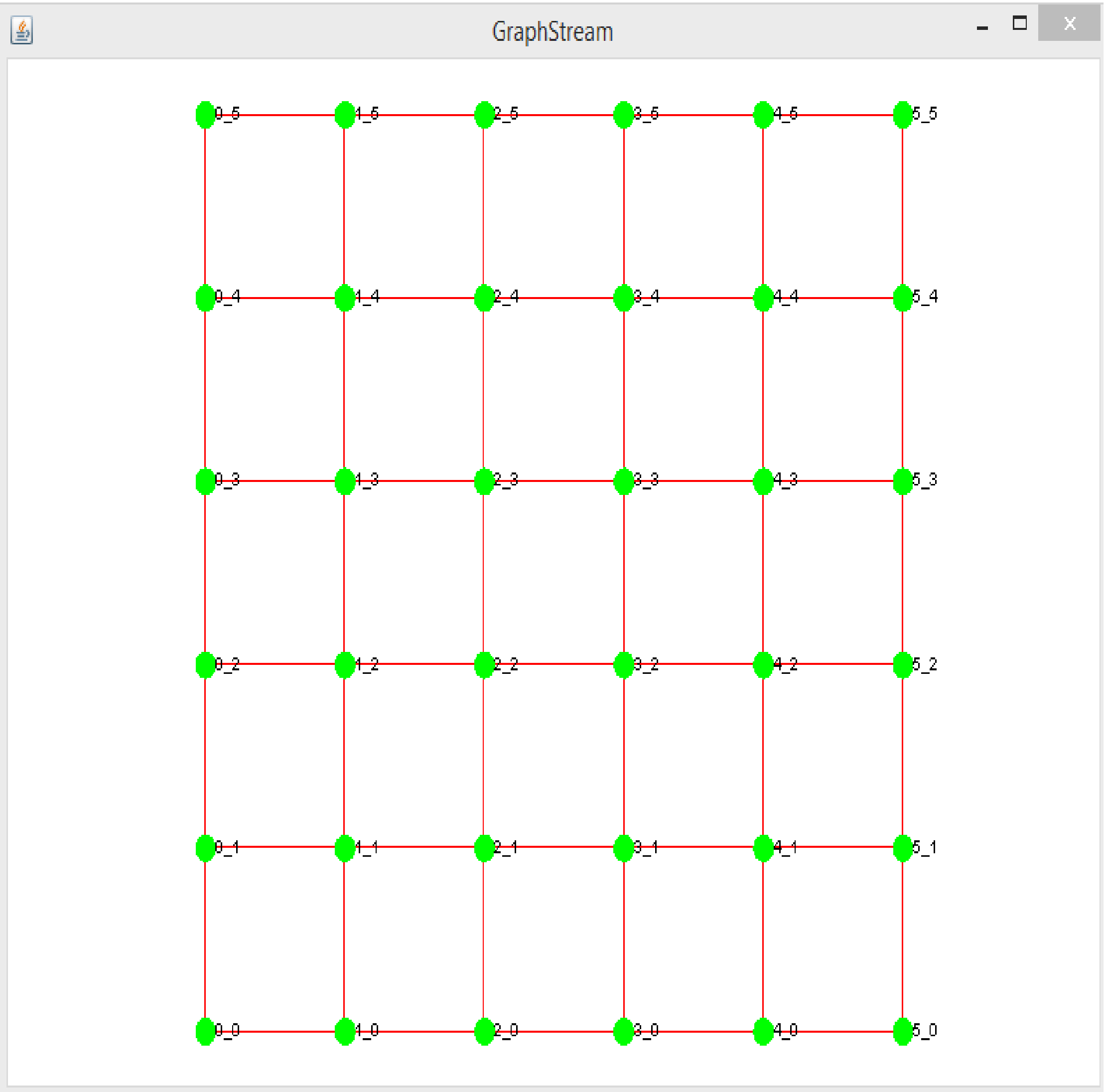


**Figure 9:** A 2D 6X6 mesh network with vertex's connected through edges used for the project

**Step 2:** In our simulation we have used Penalty-Based Routing Algorithm (*proposed by Yan Zhang, Jun Zheng, Honglin Hu in book "Security in Wireless Mesh Networks"*) for finding the disjoint multiple paths from source to destination node through which the data is to be transmitted.

## Penalty Based Routing Algorithm for privacy preservation

The penalty-based routing algorithm for traffic privacy preservation executes in three phases:

(i) path pool generation,

(ii) candidate path selection, and

(iii) individual packet routing.

First, in the *path pool generation phase,* a large number of paths from the gateway node (g) to the destination node (x) are identified. The paths are identified in such a way that the majority of them are disjoint paths, i.e., they do not have any common nodes. The set of these paths is denoted as Spaths. At the beginning of the first iteration, each node is assigned a penalty weight of 1. The shortest path algorithm is utilized to identify the shortest path between the gateway (g) and the destination node (x). Once, the first shortest path is identified, the penalty weights associated with each nodes on this route are increased arbitrarily so that these nodes do not become potential candidates in the next round of execution of the algorithm. The algorithm is executed iteratively to find the optimum paths in successive iterations. The iteration continues till the set Spaths has acquired sufficient number of candidate paths.

Second, in the *candidate path selection phase,* a subset Sselected is chosen from the set Spaths. The elements of Sselected are chosen randomly from the set Sselected. After a path is selected in the set Sselected, to reduce its probability of getting selected again in the next round, a suitable factor is employed. The use of this probability factor prevents selection of multiple identical paths in the set Sselected.

Third, in the *packet routing phase,* for each packet, one path from Sselected is randomly chosen and the packet is routed through that path. Every time a particular path is chosen for routing a packet, the value of a counter corresponding to that chosen path is increased by one. If the value of this counter for a path reaches a threshold value, then the entire Sselected set is assumed to have expired and a new Sselected is chosen by invoking the candidate path selection phase once again. Since each packet is routed through a randomly selected path, and the candidate paths are mostly disjoint, the probability that the packets are routed through the same path will be negligibly small. The algorithm seeks to achieve a trade-off between routing efficiency (i.e. routing through the shortest path between a source-destination pair) and traffic pattern privacy (routing through disjoint multi-paths).

**Algorithm:**

**/*Penalty-Based Shortest Path*/**

```
PBSP(Snode,Dnode)
        For each node v 2 V
                d[v] ← 1
                prev[v] ← 1
                visited[v] ← 0
        d[SNode] ← 0
        Repeat
                Get unvisited vertex v with the least d[v]
                If d[v] >= infinite, Then v unreachable
                Else visited[v] ← 1
                For all v's neighbors w
                        EdgePenalty = [pow(Y, (w.tag))] + B(v.tag)
                        If d[w] > d[v] + EdgePenalty
                        d[w] ← d[v] + EdgePenalty
                        prev[w] ← v
        Until visited[v] = 1, for every v belongs to V
```

**/*Generate Spaths for each g − x pair*/**

```
GenPath()
For all non-gateway nodes x
        For each node v 2 V
        v.tag ← 1
        Repeat
                PBSP(g, x)
                Get new g − x path $P_{new}$ from vector prev[]
                Store $P_{new}$ in $S_{paths}$
                For all nodes v on $P_{new}$
                v.tag ← v.tag + 1
    Until PathPoolSize paths found.
```

**/*Select Sselected for each g - x pair*/**

```
SelPath()
        Repeat
                rnd = rand() mod PathPoolSize
                select rndth path from Spaths
        Until SelPathNum paths selected
```

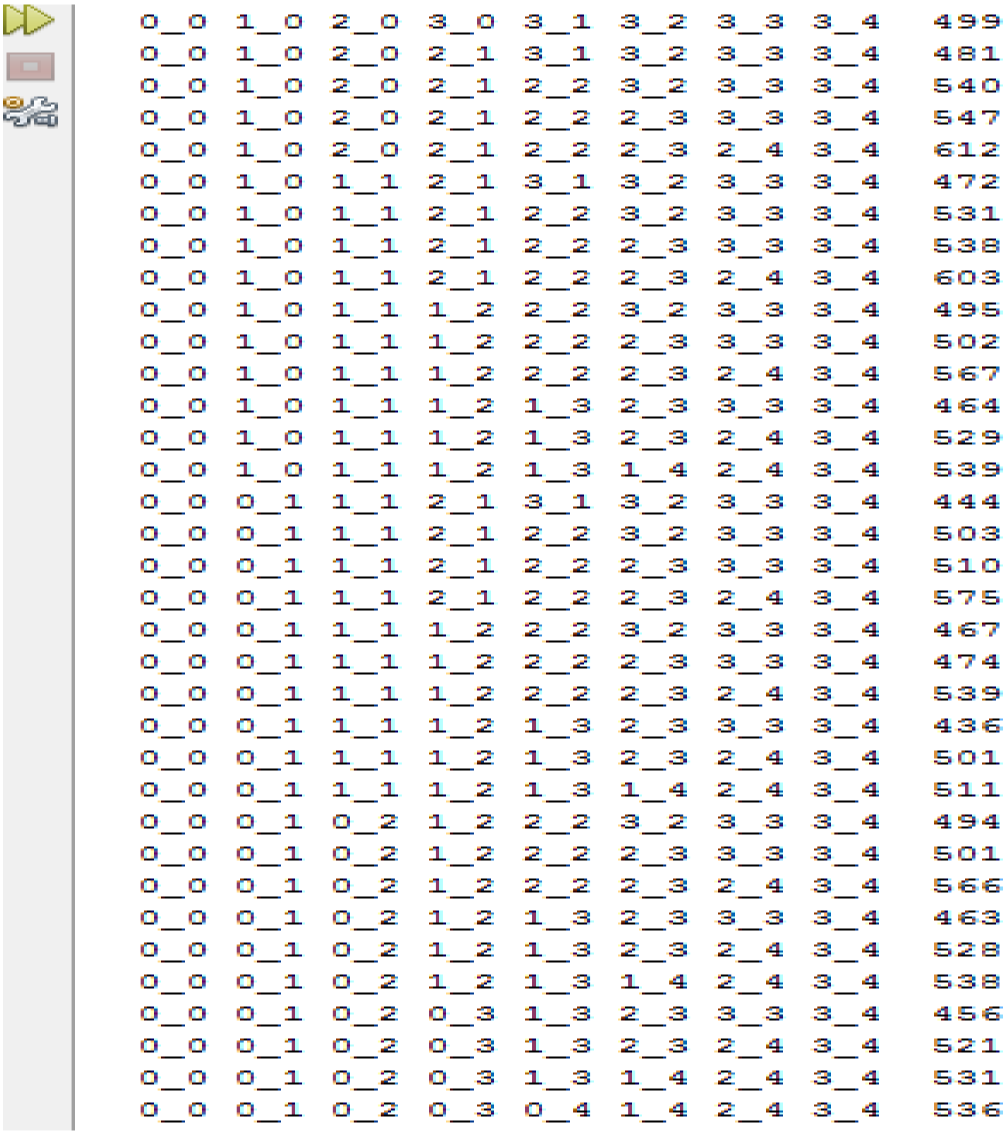
**/*Decide path for arriving packet*/**

```
RoutePkt(Snode,Dnode)
        Packets[Dnode]  ← Packets[Dnode] + 1
        rndpath = rand() mod SelPathNum
        route packet along the rndpathth path from Sselected
        If Packets[Dnode] > ReSelPathCnt
            Packets[Dnode]  ← 0
            SelPath()
```

| | |
|---|---|
| v,w | Node |
| v.tag | number of times v is included by a path |
| A | factor to slow down penalty rate |
| B | factor to avoid many identical paths in the beginning stages of path generation |
| Y | base of exponential penalty function |
| D[] | penalty vector for every node |
| Prev[] | vector to store Pnew in reverse order |
| Packets[] | vector to store the number of arrived packets for every node |
| Snode | Source node |
| Dnode | Destination node |

**Table 1:** Notations used in the algorithm for penalty based routing.

This algorithm prevents any intermediate nodes from getting the sufficient amount of data to be transmitted so that the node may not identify the identity of the sender and the content of the data other than the destination node.



```
0_0  1_0  2_0  3_0  3_1  3_2  3_3  3_4    499
0_0  1_0  2_0  2_1  3_1  3_2  3_3  3_4    481
0_0  1_0  2_0  2_1  2_2  3_2  3_3  3_4    540
0_0  1_0  2_0  2_1  2_2  2_3  3_3  3_4    547
0_0  1_0  2_0  2_1  2_2  2_3  2_4  3_4    612
0_0  1_0  1_1  2_1  3_1  3_2  3_3  3_4    472
0_0  1_0  1_1  2_1  2_2  3_2  3_3  3_4    531
0_0  1_0  1_1  2_1  2_2  2_3  3_3  3_4    538
0_0  1_0  1_1  2_1  2_2  2_3  2_4  3_4    603
0_0  1_0  1_1  1_2  2_2  3_2  3_3  3_4    495
0_0  1_0  1_1  1_2  2_2  2_3  3_3  3_4    502
0_0  1_0  1_1  1_2  2_2  2_3  2_4  3_4    567
0_0  1_0  1_1  1_2  1_3  2_3  3_3  3_4    464
0_0  1_0  1_1  1_2  1_3  2_3  2_4  3_4    529
0_0  1_0  1_1  1_2  1_3  1_4  2_4  3_4    539
0_0  0_1  1_1  2_1  3_1  3_2  3_3  3_4    444
0_0  0_1  1_1  2_1  2_2  3_2  3_3  3_4    503
0_0  0_1  1_1  2_1  2_2  2_3  3_3  3_4    510
0_0  0_1  1_1  2_1  2_2  2_3  2_4  3_4    575
0_0  0_1  1_1  1_2  2_2  3_2  3_3  3_4    467
0_0  0_1  1_1  1_2  2_2  2_3  3_3  3_4    474
0_0  0_1  1_1  1_2  2_2  2_3  2_4  3_4    539
0_0  0_1  1_1  1_2  1_3  2_3  3_3  3_4    436
0_0  0_1  1_1  1_2  1_3  2_3  2_4  3_4    501
0_0  0_1  1_1  1_2  1_3  1_4  2_4  3_4    511
0_0  0_1  0_2  1_2  2_2  3_2  3_3  3_4    494
0_0  0_1  0_2  1_2  2_2  2_3  3_3  3_4    501
0_0  0_1  0_2  1_2  2_2  2_3  2_4  3_4    566
0_0  0_1  0_2  1_2  1_3  2_3  3_3  3_4    463
0_0  0_1  0_2  1_2  1_3  2_3  2_4  3_4    528
0_0  0_1  0_2  1_2  1_3  1_4  2_4  3_4    538
0_0  0_1  0_2  0_3  1_3  2_3  3_3  3_4    456
0_0  0_1  0_2  0_3  1_3  2_3  2_4  3_4    521
0_0  0_1  0_2  0_3  1_3  1_4  2_4  3_4    531
0_0  0_1  0_2  0_3  0_4  1_4  2_4  3_4    536
```

**Figure 10:** Paths found from depth first search between source node 0_0 and destination node 3_4 with path weights sum.

**Flowchart:**

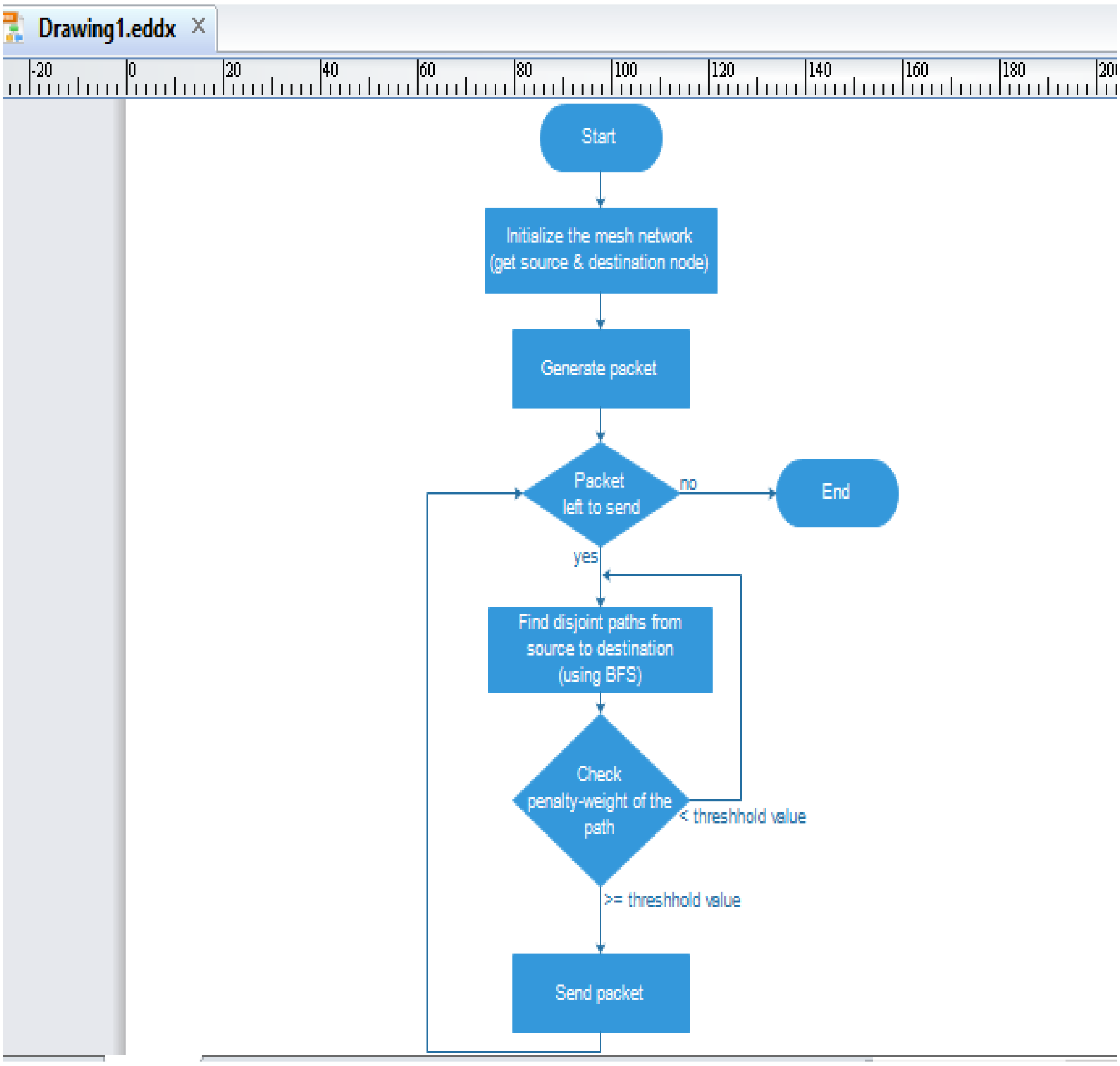


**Figure 11:** Flowchart of penalty based routing algorithm.

**Step 3:** In our simulation we have used Trust Value Algorithm (proposed by Rajendra Aaseri, Pankaj Choudhary & Nirmal Roberts in paper "TRUST VALUE ALGORITHM: A SECURE APPROACH AGAINST PACKET DROP ATTACK IN WIRELESS AD-HOC NETWORKS", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.3, May 2013).

## Trust Value Algorithm for protection against packet dropping attack

In the Packet drop attack, the attacker targets some nodes in the wireless network and then drop the packets sent towards the intended nodes. Attackers try to drop/delay the packets in the routine manner so it's very difficult to detect. The packet drop will further results into the high false positive rates and ultimately breaks the security of wireless networks. So, our problem is to detect the Packet drop attack and try to reduce the packet drop ratio so that it will result into law false positive rates.

If the number of packet drop nodes increases then the data thrashing would also likely to be boost. A malicious node can initiate the following two attacks:

**PACKET SINKING:** A malicious node slump all or a few of the packets that is believed to be forward. It can also sink the data produced by itself on behalf of some malicious intention for instance.

**PACKET AMENDMENT:** A malicious node alters the entire or a few of the data packets that is made-up to forward. It can also modify the data it produce to defend it from being recognized or to lay blame on former nodes.

The proposed algorithm is based on the trust values of individual nodes. Initially, all the nodes of wireless ad-hoc network have zero trust value.

**Algorithm:**

**[A] Initialization:**

1. Trust values of all the participating nodes are initializing with zero.
2. Initialize the threshold value of the trust value with 100.
3. Assumption: 1 trust value = 10 packets dropped.

**[B] Updating of trust values:**

**1.** If the packets are correctly transmitted from one node to another node:

(a) If the correctly transmitted number of packets is between 1 to 10, then trust values of the respective nodes will be incremented by one time.
Updated trust value = old trust value + 1;

(b) If the correctly transmitted number of packets are greater than 10, then the updated trust value will be:
Updated trust value = old trust value + (correctly transmitted packets / 10);

**2.** If the packets are dropped/delayed:

(a) The number of dropped or delayed packets is between 1 to 10, then trust value of that particular node is decremented by one.
Updated trust value = old trust value – 1;

(b) The number of dropped or delayed packets are greater than 10, then trust value of that particular node will be,
Updated trust value = old trust value – (Packet dropped or delayed / 10);

**3.** If the trust value of particular node is negative, then print “Invalid node”.

**[C] Isolating the Packet drop node from the network:**

**1.** If (Updated trust value <<< Threshold trust value)
Then the particular node is treated as malicious node (Black hole node)

**2.** If (Updated trust value > Threshold trust value)
Then the particular node is treated as legitimate node.

**3.** Stop comparing the trust values of nodes with threshold value.

**Flowchart:**

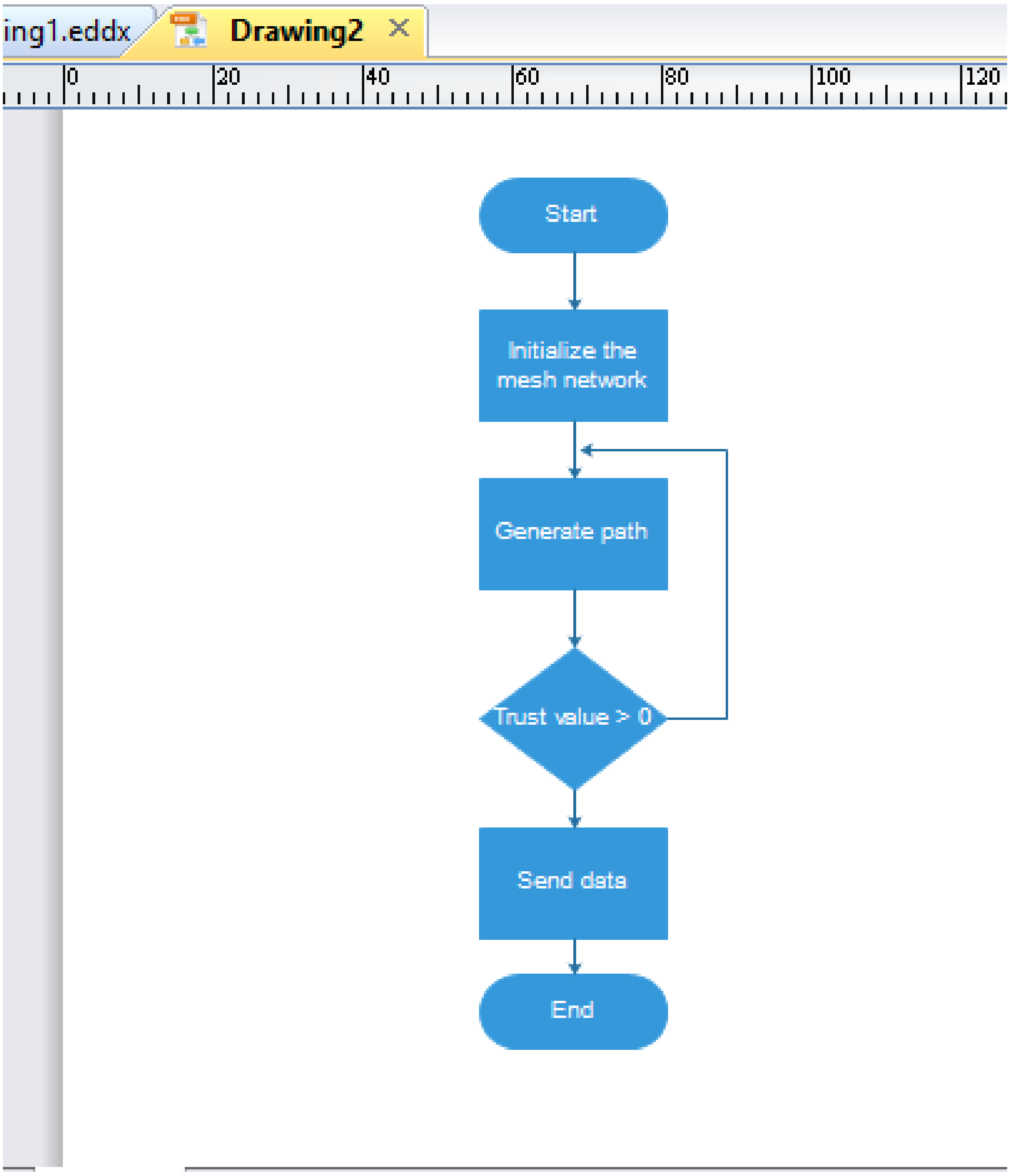


**Figure 12:** Flowchart of Trust value algorithm.

**Step 4:** In our simulation we have used ceasar cipher encryption technique (developed by Julius Ceasar) for encrypting the data before sending it to the destination.

The code snippet of **Ceasar cipher** is as follows:

```
public static String caesar(String value, int shift) {
    // Convert to char array.
    char[] buffer = value.toCharArray();
```

```
// Loop over characters.
for (int i = 0; i < buffer.length; i++)
   {
   // Shift letter, moving back or forward 26 places if needed.
   char letter = buffer[i];
   letter = (char) (letter + shift);
   if (letter > 'z')
    {
      letter = (char) (letter - 26);
   }
   else if (letter < 'a')
    {
      letter = (char) (letter + 26);
   }
   buffer[i] = letter;
}
// Return final string.
return new String(buffer);
```

Flowchart showing completed simulation:

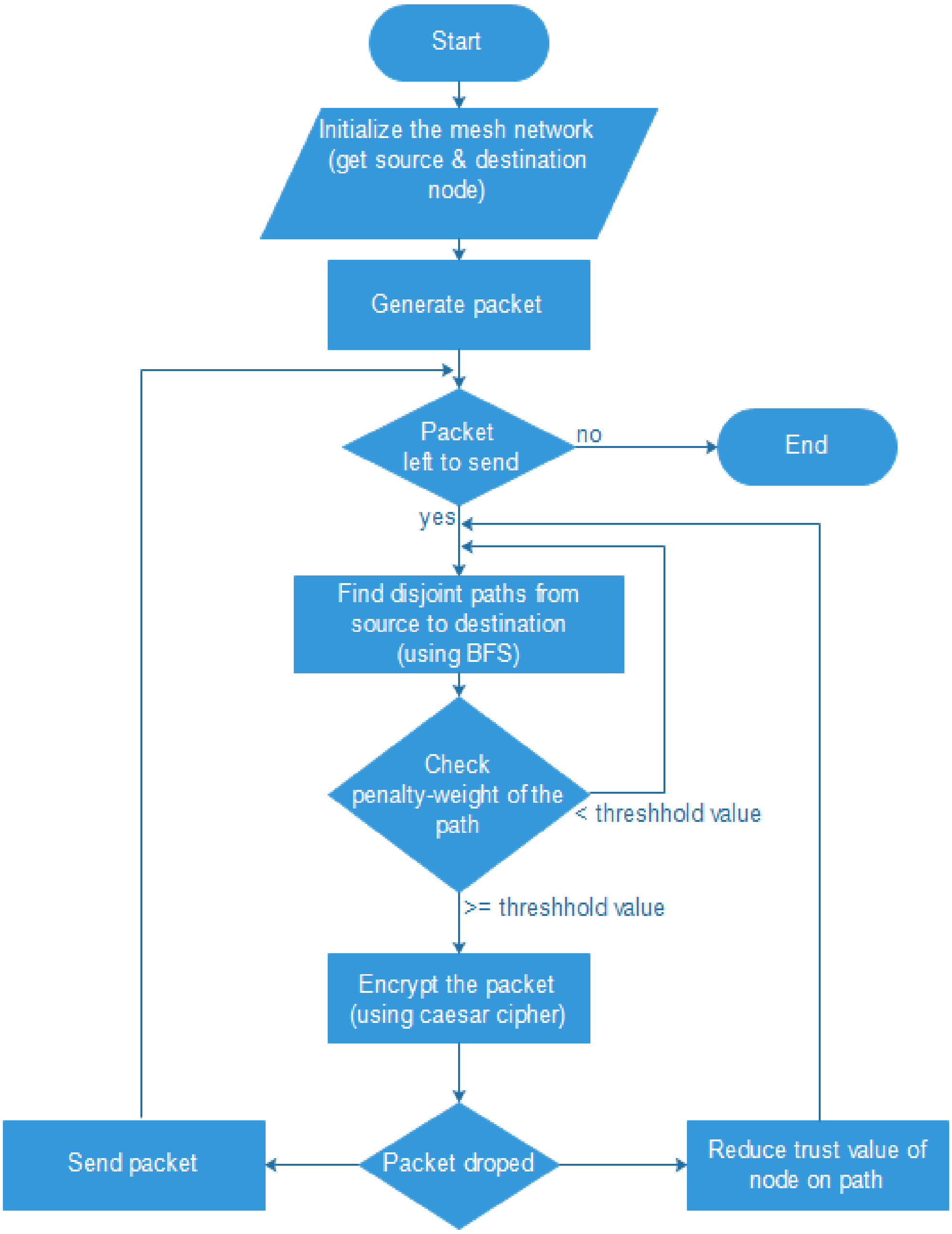

Start
Initialize the mesh network (get source & destination node)
Generate packet
Packet left to send
no
End
yes
Find disjoint paths from source to destination (using BFS)
Check penalty-weight of the path
< threshhold value
>= threshhold value
Encrypt the packet (using caesar cipher)
Send packet
Packet droped
Reduce trust value of node on path

**Figure 13:** Flowchart showing complete process of the proposed work.

# Chapter 5: Observation & Result

We base our simulation on a 2D Mesh network topology.

| | |
|---|---|
| Simulation tools Used | NetBeans IDE 8.0 (Build 201403101706) |
| Number of Nodes | 20,25,30,36 |
| Packet Size | 4 bit |
| Protocol | PBRP |
| Threshold trust value | 100 |

**Table 2:** Simulation Parameters

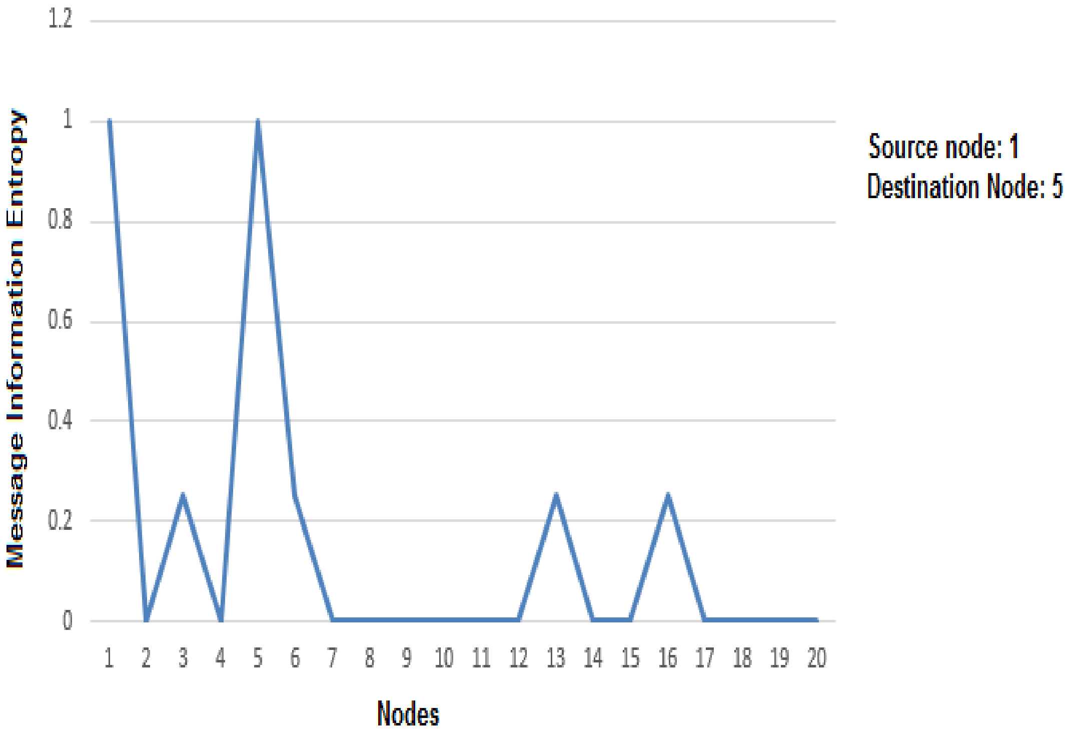


**Figure 14:** Graph showing how much information is available at the intermediate nodes and destination nodes when total nodes are 20

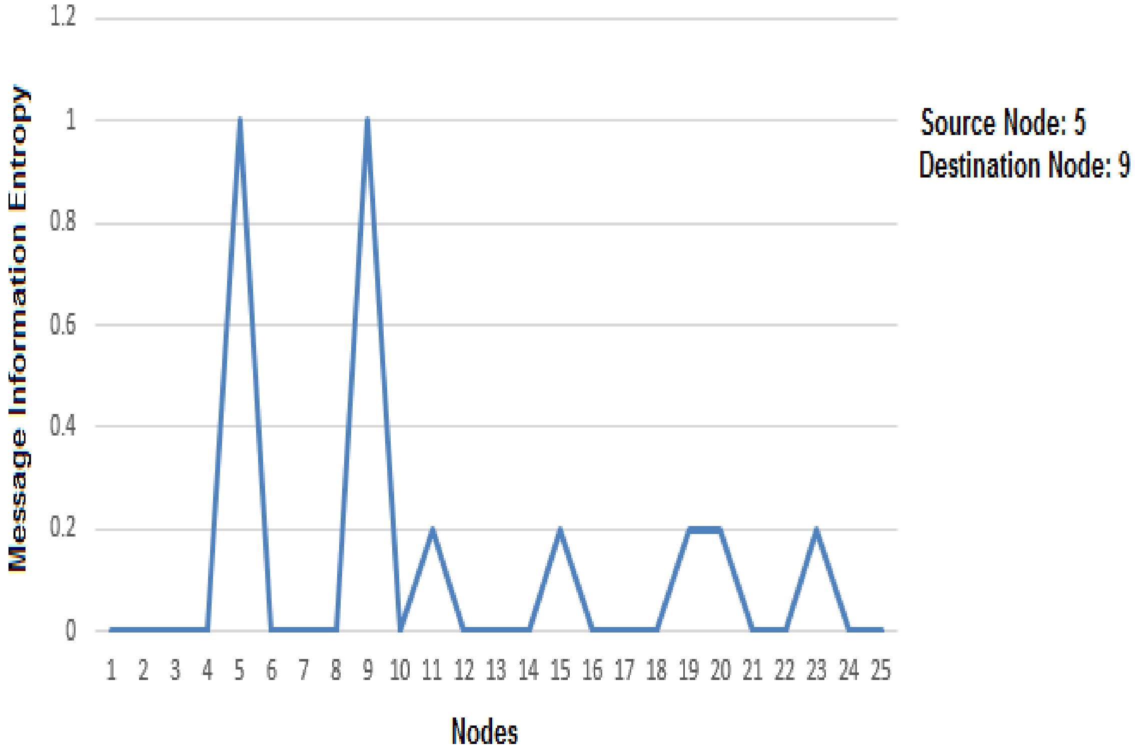


**Figure 15:** Graph showing how much information is available at the intermediate nodes and destination nodes when total nodes are 25

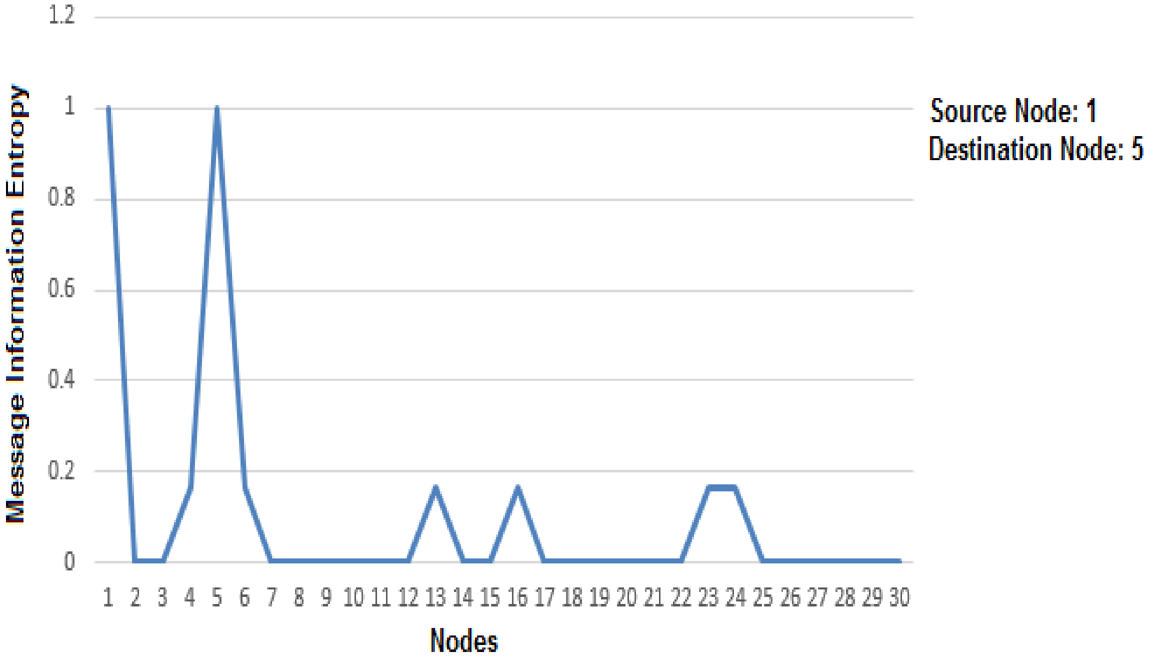


**Figure 16:** Graph showing how much information is available at the intermediate nodes and destination nodes when total nodes are 30.

# Analysis of simulation result:

| Number of nodes | Packet drop rate |
|---|---|
| 20 | 0.01 |
| 25 | 0.0137 |
| 30 | 0.032 |
| 36 | 0.057 |

**Table 3:** Analysis of the simulation result.

# Chapter 6: Conclusion

With the introduction of neighbour discovery, we argue that the secure transmission on wireless network are also necessary for secure neighbour discovery. Based on the emerging techniques we have described a way to securely transmit the data without bothering about link reliability.

The First planned phase which is responsible for the preservation of the privacy is completed and now the system is protected against privacy even if the intermediate nodes are malicious.

The Second phase which is responsible for protection against various attacks is completed in which protection against packet drop attack is done using Trust Value Algorithm and malicious node removal is done by penalty based routing protocol.

The third phase which is responsible for sending the encrypted data to the destination node by dividing the data and then encrypting it with ceasar cipher encryption technique is completed.

So, now the network is safe from interference of any type by any malicious node.

# References

[1] Saurabh Bagchi, Srikanth Hariharan and Ness Shroff: "Secure Neighbor Discovery in Wireless Sensor Networks" (2007). ECE Technical Reports. Paper 360.

[2] L. Hu and D. Evans. Using directional antennas to prevent wormhole attacks. In Network and Distributed System Security Symposium, 2004.

[3] I.Khalil, S. Bagchi, and N. B. Shro. Liteworp: A lightweight countermeasure for the wormhole attack in multi-hop wireless networks. In Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN'05), pages 612-621, 2005.

[4] Issa Khalil, Saurabh Bagchi, and Cristina Nina-Rotaru. Dicas: Detection, diagnosis and isolation of control attacks in sensor networks. In Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communication Networks (SECURECOMM'05), pages 89-100, 2005.

[5] D. Liu, P. Ning, S. Zhu, and S. Jajodia. Practical broadcast authentication in sensor networks. In The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous), pages 118-129, July 2005.

[6] Yan Zhang, Jun Zheng and Honglin HU. Security in Wireless Mesh Network. Edition 1st. 2008, Page no – 236

[7] Rajendra Aaseri, Pankaj Choudhary & Nirmal Roberts in paper “TRUST VALUE ALGORITHM: A SECURE APPROACH AGAINST PACKET DROP ATTACK IN WIRELESS AD-HOC NETWORKS”, International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.3, May 2013)

[8] http://ipv6.com/articles/research/Secure-Neighbor-Discovery.htm