Non-repudiation in Ad Hoc Networks

By

Tandel Purvi H.

Roll No: 09MCE027

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY
AHMEDABAD-382481

May, 2011

Non-repudiation in Ad Hoc Networks

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Tandel Purvi H.

Roll No: 09MCE027

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY, NIRMA UNIVERSITY

AHMEDABAD-382481

May, 2011

Declaration

This is to certify that

i) The thesis comprises my original work towards the degree of Master of Technology in Computer Science and Engineering at Nirma University and has not been submitted elsewhere for a degree.

ii) Due acknowledgement has been made in the text to all other material used.

Tandel Purvi H.

Certificate

This is to certify that the Major Project entitled "**Non-repudiation in Ad Hoc Networks**" submitted by **Tandel Purvi H. (09MCE027)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. K. P. Agrawal
Guide, Sr.Associate Professor,
Computer Science & Engineering Dept.,
IT, Nirma University.

Prof. Sharada Valiveti
Guide, Associate Professor,
Computer Science & Engineering Dept.,
IT, Nirma University.

Dr. S. N. Pradhan
Professor and PG-Coordinator,
Computer Science & Engineering Dept.,
IT, Nirma University.

Prof. D .J. Patel
Professor and Head,
Computer Science & Engineering Dept.,
IT, Nirma University.

Dr. K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad.

Abstract

With the phenomenal growth of the Internet and open networks in general, security services, such as non-repudiation, become crucial to many applications. In conventional networks, non-repudiation is achieved using protocols involving TTP. Non-repudiation in conventional networks is achieved using different protocols, but in ad hoc networks due to mobility problem it is infeasible to use trusted third party (TTP). There is a scope to implement a non-repudiation protocol, which satisfies non-repudiation requirements emerged by the application in a peer-to-peer network. In ad hoc network, non-repudiation can be achieved using two main methods : witness selection and proof of reception. Witness set selection can be done by setting transmission range in the topology and transmitting the data over the network. Some nodes will route packets for another node working as intermediate node. Node with number of packets routed for another node with higher probability can be an efficient intermediate node between the source and the destination nodes. Another approach to find witness node from the witness set is using data aggregation method. Witness node will work as TTP in the network and by using proof of reception method, source and destination node will transmit secure data. Non-repudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message. Due to infrastructure-less and self-organized characteristics, ad hoc network encounters different problems from infrastructure-based wired network, such as key management, power shortage, and security issues. Proof of reception is a key element for providing secure contract conclusion between members on a market place. According to nature of ad-hoc network, some requirements like data integrity, data confidentiality, non-repudiation of content for sender, non-repudiation of reception for recipient need to be taken care of. Same way attacks like denial by sender, denial by receiver, brute force attack and witness selection are the factors affecting non-repudiation in ad-hoc networks.

Acknowledgements

It gives me great pleasure in expressing thanks and profound gratitude to **Prof. K. P. Agrawal,** Sr. Associate Professor, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance, time to time suggestions and continual encouragement throughout the Major project.

I am heartily thankful to **Prof. Sharada Valiveti,** Associate Professor, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for her time to time suggestions and the clarity of the concepts of the topic that helped me a lot during this study.

I would like to extend my gratitude to **Dr. S. N. Pradhan,** Professor & M.Tech Coordinator, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for fruitful discussions and valuable suggestions during meetings and for his encouragement.

I am also grateful to **Prof. D. J. Patel**, Head of the Department, Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad for giving me an opportunity to perform the project under the premises of Institute of Technology, Nirma University, Ahmedabad.

I would like to thank **Dr. Ketan Kotecha,** Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for providing basic infrastructure and healthy research environment.

I would also like to extend my thanks to all my friends for their kind cooperation and motivation.

Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members, because of whom I am able to complete my dissertation work successfully.

- Tandel Purvi H.
09MCE027

Contents

List of Tables

List of Figures

Abbreviations

AODV ..... Ad-hoc On demand Distance Vector routing protocol
DoS ..... Denial-of-Service
GloMoSim ..... Global Mobile Information System Simulator
GUI ..... Graphical User Interface
NRD ..... Non-Repudiation of Delivery
NRO ..... Non-Repudiation of Origin
NRR ..... Non-Repudiation of Receipt
NRS ..... Non-Repudiation of Submission
OPNET ..... Optimized Network Engineering Tools
PARSEC ..... Parallel Simulation Environment for Complex systems
PCL ..... Parallel Calculation Laboratory
sSKA(m) ..... Message m signed by A's secret key
sSKB(m) ..... Message m signed by B's secret key
TTP ..... Trusted Third Party
UCLA ..... University of California in Los Angeles
VINT ..... Virtual InterNetwork Testbed

Chapter 1

Introduction

In an ad hoc network, there are no fixed infrastructure such as base stations or mobile switching centers. Mobile nodes that are within each other's radio range communicate directly via wireless links, while those that are far apart rely on other nodes to relay messages as routers. Node mobility in an ad hoc network causes frequent changes of the network topology. Due to these infrastructure-less and self-organized characteristics, ad hoc network encounters different problems from infrastructure-based wired network, such as key management, power shortage, and security issues [1].

The nature of ad hoc networks poses a great challenge to system security designers due to the following reasons:

a. The wireless network is more susceptible to attacks ranging from passive eavesdropping to active interfering;

b. The lack of an online CA or Trusted Third Party adds the difficulty to deploy security mechanisms;

c. Mobile devices tend to have limited power consumption and computation capabilities which makes it more vulnerable to Denial of Service attacks and incapable to execute heavy algorithms like public key algorithms;

d. There are more probabilities for trusted node being compromised and then being used by adversary to launch attacks on networks, in other words, we need to consider both insider attacks and outsider attacks in ad hoc networks, in which insider attacks are more difficult to deal with;

e. Node mobility enforces frequent networking reconfiguration which creates more chances for attacks [2].

1.1 Security goals in ad hoc networks

Security is an important issue for ad hoc networks, especially for the security-sensitive applications. To secure an ad hoc network, we consider the following attributes: availability, confidentiality, integrity, authentication, and non-repudiation [1, 2].

- **Availability** ensures the survivability of network services despite denial of service attacks. A denial of service attack could be launched at any layer of an ad hoc network. On the physical and media access control layers, an adversary could employ jamming to interfere with communication on physical channels.

- **Confidentiality** ensures that certain information is never disclosed to unauthorized entities. Routing information must also remain confidential in certain cases, because the information might be valuable for enemies to identify and to locate their targets in a battlefield.

- **Integrity** guarantees that a message being transferred is never corrupted. A message could be corrupted because of benign failures, such as radio propagation impairment, or because of malicious attacks on the network.

- **Authentication** enables a node to ensure the identity of the peer node it is communicating with. Without authentication, an adversary could masquerade

a node, thus gaining unauthorized access to resource and sensitive information and interfering with the operation of other nodes.

- **Non-repudiation** ensures that the origin of a message cannot deny having sent the message. Non-repudiation is useful for detection and isolation of compromised nodes. When a node A receives an erroneous message from a node B, non-repudiation allows A to accuse B using this message and to convince other nodes that B is compromised.

To achieve non-repudiation we need Trusted Third Party (TTP) so that we can guarantee the transaction between sender and recipient node in an ad hoc network. Suppose you send a registered mail, hence the recipient cannot deny that a letter was delivered. Similarly, a legal document typically requires witnesses to signing so that the person who signs cannot deny having done so.

1.2 Objective

Objective of my work is to implement non-repudiation in ad hoc networks using peer - to - peer mechanism having witness selection and proof of reception. We try to achieve the identified requirements and avoid the identified attacks by implementing proposed protocol to achieve non-repudiation in ad hoc networks.

1.3 Scope

The scope of the research is to achieve non-repudiation in ad hoc networks. These days, security has become very crucial factor in networks. In conventional networks, non-repudiation is achieved using protocols involving TTP. There is a scope to implement a non-repudiation protocol, which satisfies non-repudiation requirements emerged by the application in a peer-to-peer network.

1.4 Motivation

Motivation behind doing this research is the nature of ad hoc networks poses a great challenge to system security designers. With the phenomenal growth of the Internet and open networks in general, security services, such as non-repudiation, become crucial to many applications. Non-repudiation in conventional networks is achieved using different protocols, but in ad hoc networks, due to mobility problem, we can't use trusted third party (TTP).

1.5 Thesis Organization

The rest of the thesis is organized as follows.

Chapter 2, *Literature Survey*, describes non-repudiation services provided by protocol. It also covers an overview of non-repudiation protocols in conventional networks which uses Trusted Third Party(TTP) to achieve non-repudiation. A comparison of non-repudiation protocols is also shown. It also describes the probabilistic approach to achieve non-repudiation in ad hoc network.

Chapter 3, *Requirements and Threat Analysis*, describes identified requirements and threat analysis to achieve non-repudiation in ad-hoc networks after study of other non-repudiation protocol in conventional networks.

Chapter 4, *Study of GloMoSim Simulator*, describes comparative study of network simulators. It covers the basics of GloMoSim simulator and its two main components. It shows the description of GloMoSim architecture and the details about the visualization tool.

Chapter 5, *Proposed Algorithm*, describes the methods to implement non-repudiation in ad-hoc networks.

Chapter 6, *Implementation*, includes implementation of whole proposed protocol having witness node as Global Trusted Third Party.

Chapter 7, *Analysis of required results*, the results are analyzed in terms of identified requirements and threats.

Finally, in **Chapter 8**, *Conclusion and Future Work*, concluding remarks and future work is presented.

Chapter 2

Literature Survey

Non-repudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message. Non-repudiation is useful for detection and isolation of compromised nodes. Non-repudiation is usually achieved using public key cryptography. If A signs a message with their private key B can confirm the origin of the message by verifying the signature using A's public key. Similarly, given B's signature on the message, A can confirm receipt by verifying the signature using B's public key [3].

With the advent of digital signatures and public key cryptography, the base for non-repudiation services was created. A typical non-repudiation protocol can provide a number of different non-repudiation services, like non-repudiation of origin, non-repudiation of receipt and fairness, but the actual non-repudiation services provided by a protocol depend mainly on its application.

2.1 Non-repudiation services in ad hoc networks

- **Non-repudiation of origin (NRO)** provides the recipient with the evidence NRO which ensures that the originator will not be able to deny having sent the

message. The evidence of origin is generated by the originator and held by the recipient.

- **Non-repudiation of receipt (NRR)** provides the originator with the evidence NRR which ensures that the recipient will not be able to deny having received the message. The evidence of receipt is generated by the recipient and held by the originator.

- **Non-repudiation of submission (NRS)** is intended to provide evidence that the originator submitted the message for delivery. This service only applies when the protocol uses a TTP. Evidence of submission is generated by the delivery agent, and will be held by the originator.

- **Non-repudiation of delivery (NRD)** is intended to provide evidence that the recipient received the message. This service also only applies when the protocol uses a TTP. Evidence of delivery is generated by the delivery agent, and will be held by the originator.

- **Fairness** is achieved for a non-repudiation protocol if at the end of the protocol execution, either the originator has the evidence of receipt for the message m and the recipient has the evidence of origin of the corresponding message m, or none of them has any valuable information [4].

The fairness is an essential requirement of a transaction in any type of network. Keeping the fairness of the protocol is helpful for participants to complete procedures. No participant is willing to join in an unfair transaction either in the real world or in the electronic world. To achieve non-repudiation in conventional networks, protocols with or without TTP have been used. Most of them are used with TTP to solve the purpose.

2.2 Overview of Non-repudiation protocol in conventional network

Most of the practically fair non-repudiation protocols require the involvement of a TTP to some extent. The level of intervention can vary depending on the protocol and the requirements of the end users.

Kremer et al identified three main types of TTP [3]:

a. Inline

b. Online

c. Offline

An inline TTP (sometimes called a delivery agent) is involved in transmission of each protocol message. An online TTP is involved in each session of a protocol but not in every message transmission. An offline TTP is involved in a protocol only in case of incorrect behavior of a dishonest entity or in case of network failures [3].

2.2.1 Non-repudiation protocol without TTP

Although protocols without TTP were the first protocols proposed in the framework of fair exchange of secrets and digital contract signing, non- repudiation protocols without TTP were initially presented at the end of the 1990s. The first non-repudiation protocol without TTP was proposed in 1999.

Markowitch and Roggeman protocol [5]:

The goal of this protocol is to avoid the intervention of a TTP at the price of accepting the probabilistic version of fairness. The protocol has to be parameterized on the

basis of the most powerful entity's computing power. This iterative protocol is such that except at the last iteration, no entity is more privileged than another one during the protocol. The probabilistic non-repudiation protocol however does not need the involvement of a TTP to keep the fairness.

The transaction processes are:
Client → Provider: Request for a service
Provider → Client: Service
Client → Provider: Payment (acknowledgement)

The procedures are described in the following steps.
The recipient (Bob) determines the date D
Step 1. B → A sSKB(request,B,A,D)
The originator (Alice): checks D
chooses n
computes the signed f1, : : :, fn
Step 2. A → B sSKA (fn(m),A,B,D)
Step 3. B → A sSKB(ack1)
.
.
Step 2n. A → B sSKA(f1(m),A,B,D)
Step 2n+1. B → A sSKB(ackn)
m= fn(m) fn-1(m) f1(m).

NRO=NROi — i=1,,n, with NROi= sSKA (fi(m),A,B,D)
NRR=sSKB(ackn)

Where, NRO is Non-Repudiation of Origin, NRR is Non-Repudiation of Receipt, sSKB(m) is message m signed by B's secret key and sSKA(m) is message m signed

by A's secret key.

At any moment, if Alice or Bob receive an incorrect message, they stop taking part in the protocol. Moreover, if Bob does not directly answer Alice's messages by sending the corresponding NRR, Alice will suppose that Bob attempts to cheat and consequently she stops the protocol (by not sending the next value)[5].

2.2.2 Non-repudiation protocol with online TTP

The protocols based on an online TTP are such that the TTP does not act anymore as a delivery authority (as an intermediary for each transmission between the entities). However, an online TTP intervenes during each session of the protocol.

Zhou and Gollmann protocol [6]:

Zhou and Gollmann presented a non-repudiation protocol with online TTP. The idea of this protocol is to reduce the work of the TTP to a minimum. During the protocol, if an incorrect message arrives or if an awaited message does not arrive, the potential recipient stops the protocol.

Alice initiates the protocol by sending the request to Bob using session key k with a label identifying the protocol session and a time-out value before which the session key must be submitted to the TTP and after which it can be consulted, as well as the signed non-repudiation of origin evidence for the ciphered message. If Bob accepts the consultation time-out proposed by Alice, he sends his signed non-repudiation of receipt evidence for the ciphered message. Alice then sends to the TTP a signed copy of the session key. The TTP accepts during a session of a protocol only one submission from an entity and checks whether Alice's signature is valid and whether the time-out is not exceeded. After the time-out, Bob can get the session key and

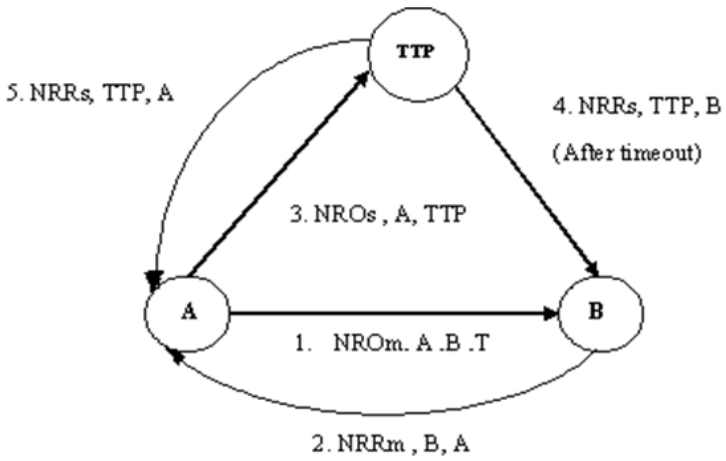

Figure 2.1: Zhou Gollmann Non-repudiation Protocol

the non-repudiation of origin evidence for this session key provided by the TTP. This evidence is necessary in order to build a complete non-repudiation of origin evidence for the message that Alice sends to him. In a similar way, Alice consults the TTP to complete her non-repudiation of receipt evidence for the message. Both Alice and Bob will fetch the session key and the corresponding evidence for this key at the TTP. This evidence serves to Bob as evidence of origin and to Alice as a proof that the key is accessible to Bob. The entities consult, at the proper time, a read-only public directory managed by the TTP. If one of the entity can't get the evidence at the TTP, while the other entity does, he will lose a possible future dispute on this subject [6].

2.2.3 Non-repudiation protocols with offline TTP

A TTP is said to be offline if it does not intervene in the protocol while no problem occurs. A problem could be an incorrect behavior of a dishonest entity or a network error. When such a problem occurs, Alice and/or Bob invoke the TTP to help them to finish the protocol run in a fair way. Non-repudiation protocol with offline TTP is proposed by Zhou-Gollmann [4, 7].

A Fair Non-repudiation Protocol [4, 7]:

The protocol is presented in figure 2.2 in Alice and Bob notation, where fNRO, fNRR, fSUB and fCON are labels used to identify the purpose of the messages.

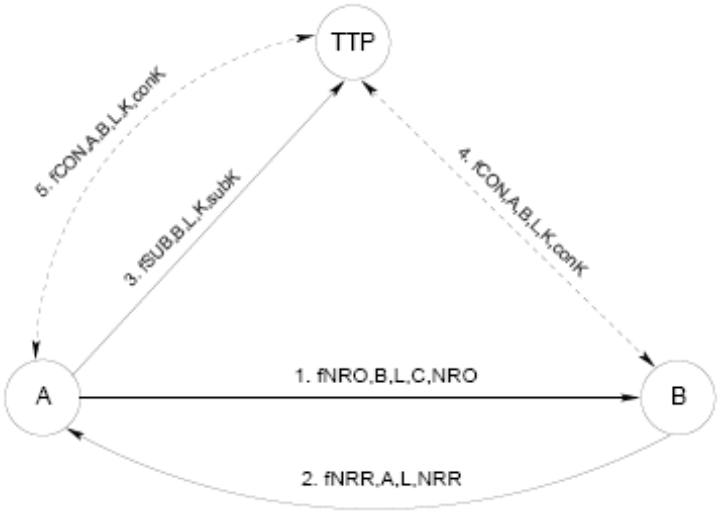

Figure 2.2: Fair Zhou Gollmann Non-repudiation Protocol [4]

a. A → B: fNRO.B.L.C.NRO

b. B → A: fNRR.A.L.NRR

c. A → TTP: fSUB.B.L.K.subK

d. B ↔ TTP: fCON.A.B.L.K.conK

e. A ↔ TTP: fCON.A.B.L.K.conK

Where,
A is Alice, the originator of the message M
B is Bob, the recipient of the message M
TTP is trusted third party

M is message sent from A to B
C is commitment, the message M encrypted under key K
L is a session identifier, a nonce
K is symmetric key defined by A
NRO is non-repudiation of origin, the message fNRO.B.L.C signed by A
NRR is non-repudiation of receipt, the message fNRR.A.L.C signed by B
subK is proof of submission of K, the message fSUB.B.L.K signed by TTP
conK is confirmation of K, the message fCON.A.B.L.K signed by TTP

The main idea of the FairZG protocol is to split the delivery of a message into two parts. First a commitment C, containing the message M encrypted by a key K, is exchanged between Alice and Bob. Once Alice has an evidence of commitment from Bob, the key K is sent to a trusted third party. Once the TTP has received the key, both Alice and Bob can retrieve the evidence conK and the key K from the TTP. This last step is represented by a double direction arrow in the Alice and Bob-style notation because it is implementation specific and may be composed by several message exchanges between the agents and the TTP.

In this scenario we assume that the network will not be down forever and both Alice and Bob have access to the TTP's shared repository where it stores the evidences and the key. That means the agents will eventually be able to retrieve the key and evidences from the TTP even in case of network failures [4, 7].

2.2.4 Comparison of non-repudiation protocols

In this section comparison among all the non-repudiation protocols is given as per the following table, where important information such as the degree of fairness that is reached, whether timeliness is respected or not, which kind of TTP is involved in the protocol and the channel requirements are compared.

Table 2.1: Comparision of non-repudiation protocols

Protocol	**Fairness**	**Timeliness**	**TTP involvement**
Markowitch-Roggeman	probabilistic	probabilistic	none
Zhou-Gollmann	strong	yes	online
A fair non-repudiation protocol	strong	yes	offline
Peer-to-peer mechanism	probabilistic	probabilistic	none

2.3 Probabilistic approaches for non-repudiation in ad-hoc networks

2.3.1 Non-repudiation mechanisms for peer-to-peer networks

Most peer-to-peer technology has one disadvantage today, only a few necessary security requirements like end-to-end transport security are available. Proof of reception is a key element for providing secure contract conclusion between members on a market place. The key principle is the involvement of other peers. These peers act as witnesses (see figure 2.3) and assist the non-repudiation protocol operations. In summary, the set of witnesses acts as a replacement for the trusted third party known from classical non-repudiation protocols. In figure 2.3, peers W1, W2 and W3 are selected as witness peers. These peers assist the proof of reception protocol between the peers A and B [8].

In this mechanism, TTP is not required because the involvement of other available peers which assist the protocol between sender and recipient will be acting as witnesses. Using **witness selection** and **proof of reception**, nodes communicates. One of the participants is able to compute witness peer set using brute-force attack, place malicious peers and manipulate the protocol. Sender and recipient both must be involved in the witness peer selection.

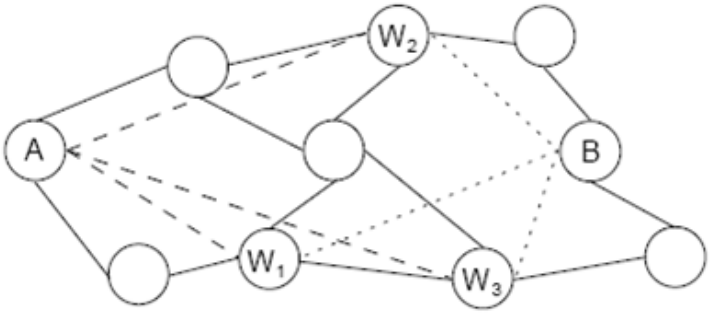

Figure 2.3: Involving other peers into the protocol [8]

Witness selection:

Both, sender and recipient, must be involved into the witness peer selection. To ensure this, the sender A requests a nonce value by sending a signed nonce request $S_A(H(H(O)), N_A)$ to the recipient B, including the hashed hash value of O and the nonce value N_A of A. B answers with a signed nonce response $S_B(S_A(H(H(O)), N_A), N_B)$ containing the original request and the nonce value N_B of B.

Proof of reception:

After exchanging nonce values, sender A computes the set of witness peers using the following formula.

$PeerID_{Pi} = H(i, S_B(S_A(H(O), N_A)N_B))$
Where, Peer ID_{Pi} is the selected witness node set.

Figure 2.4 shows the protocol for a proof of reception of document O of recipient B started by sender A. For simplification the figure only shows one witness peer Pi. By the integration of witness peers, there is no need for a global trusted third party. For each proof of reception, other witness peers are selected. This mechanism offers

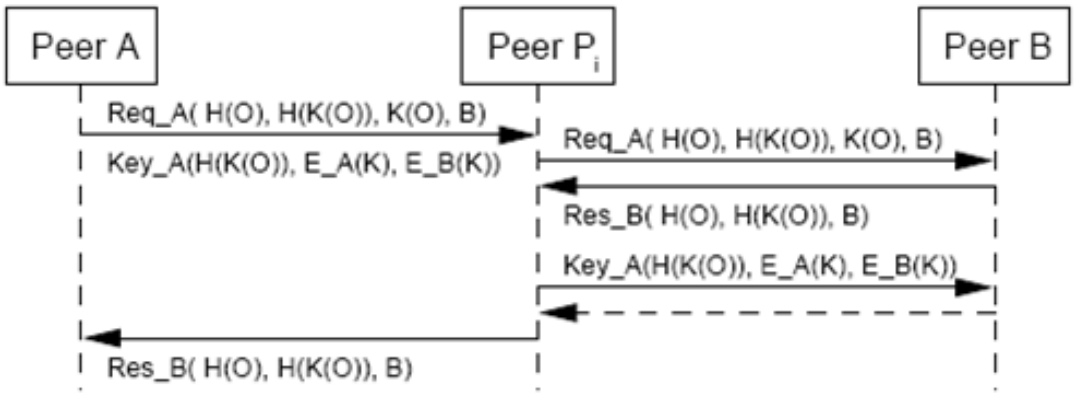

Figure 2.4: Communication protocol [8]

better scalability and better robustness [8].

2.3.2 Non-repudiation using Secure Data Aggregation

Data aggregation protocols aim to combine and summarize data packets of several nodes so that amount of data transmission is reduced. An example of the data aggregation scheme is presented in figure 2.5 where a group of nodes collect information from a target region. When the base station queries the network, instead of sending each node's data to base station, one of the nodes, called data aggregator, collects the information from its neighboring nodes, aggregates them (e.g., computes the average), and sends the aggregated data to the base station over a multi-hop path. As illustrated by the example, data aggregation reduces the number of data transmissions thereby improving the bandwidth and energy utilization in the network [9].

Security requirements:

There are mainly four security requirements in the network:

a. **Data confidentiality:**

Data confidentiality ensures that secrecy of sensed data is never disclosed to

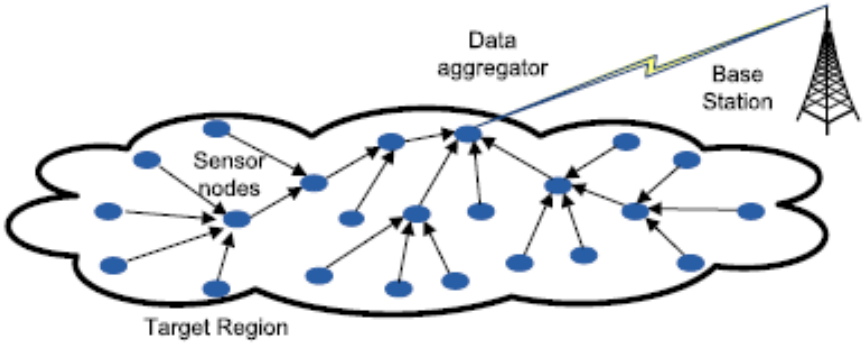

Figure 2.5: Data aggregation [9]

unauthorized parties.

b. **Data integrity and Data freshness:**
Data integrity guarantees that message being transferred is never corrupted. Data freshness protects data aggregation schemes against replay attacks by ensuring that the transmitted data is recent.

c. **Source authentication:**
Nodes need authentication mechanism to detect maliciously injected or spoofed packets. Without authentication, an adversary can masquerade a node.

d. **Availability:**
Availability guarantees the survivability of network services against Denial-of-Service (DoS) attacks [9, 10].

The main objective of data aggregation is to increase the network lifetime by reducing the resource consumption of nodes (such as battery energy and bandwidth).

There are several protocols that allow routing and aggregation of data packets simultaneously. These protocols can be categorized into two parts:

a. Tree-based data aggregation protocols

b. Cluster-based data aggregation protocols

1. Tree-based data aggregation protocols:

The simplest way to achieve distributed data aggregation is to determine some data aggregator nodes in the network and ensure that the data paths of nodes include these data aggregator nodes. The main issue of tree-based data aggregation proto-

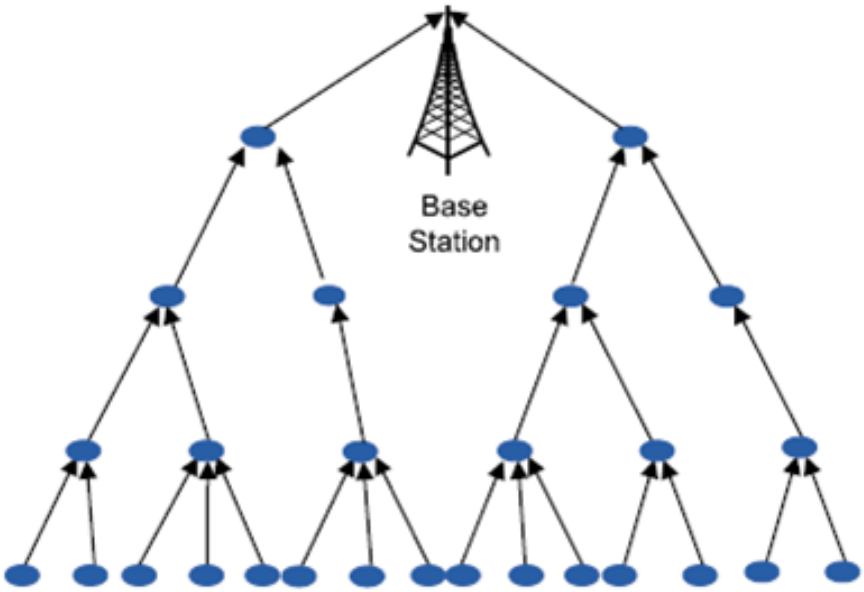

Figure 2.6: Tree based data aggregation [9]

col is the construction of an energy efficient data aggregation tree. In this protocol, parent selection is based on nodes' distance to the base station and their residual energy level. Data aggregation is performed during data forwarding phase. Using this approach amount of latency will be high. In the energy efficient tree, root node is the most efficient node amongst all node. This node is the strongest node by having more energy and lifetime.

2. Cluster-based data aggregation protocol:

To reduce the latency due to tree-based data aggregation, recent work on data aggregation tends to group nodes into clusters so that data are aggregated in each group for improved efficiency. The main issue of cluster based data aggregation protocol is the election of cluster head. In each cluster, cluster head is elected in order to aggregate data locally and transmit the aggregation result to the base station [9].

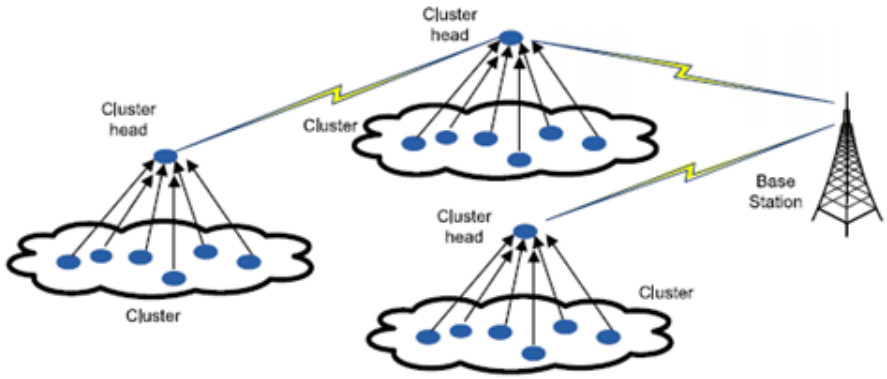

Figure 2.7: Cluster based data aggregation [9]

Aggregation scenario using sum function:

Data aggregation uses primitive functions, such as mean, average, addition, subtraction, and exclusive or to eliminate identical readings, and only unique results are to be forwarded, reducing the cost of data transmission. The network in figure 2.8 contains 16 nodes and uses SUM function to minimize energy consumption by reducing the number of bits reported to the base station. Nodes 7, 10-16 are normal nodes that are collecting data and reporting them back to the upper nodes whereas nodes 1-6, 8, 9 are aggregators that perform sensing and aggregating at the same time. In this example 16 packets traveled within the network and only one packet is transmitted to the base station. However, the number of traveling packets would

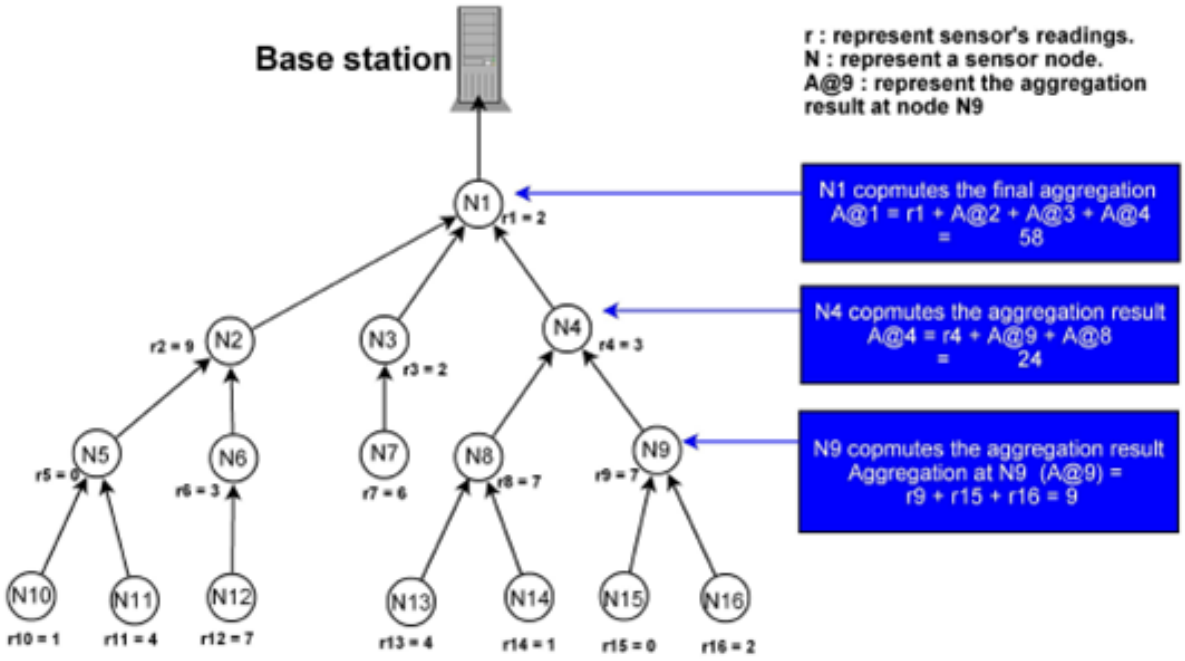

Figure 2.8: An aggregation scenario using sum function [10]

increase to 50 packets if no data aggregation exists. Base station gets data from only one aggregator so the number of bits reported to the base station reduced [10].

Using aggregation scenario we can select witness node to overcome the problems in ad hoc network like energy efficiency and bandwidth consumption. Non-repudiation can be achieved using two main techniques witness selection and proof of reception. By selecting witness node using data aggregation scenario, efficient result may be achieved.

2.4 Summary

In this chapter, some of the non-repudiation protocols for conventional networks are studied. Most popular protocols like Zhou-Gollmann and fair non-repudiation proto-

col using online or offline TTP provides strong fairness and timeliness property. Most of the non-repudiation protocols use TTP to guarantee the transaction between the sender and the recipient. Now to secure ad hoc networks, peer - to - peer mechanism can be used as there is no availability of TTP in communication. Another approach for selecting witness node is tree based or cluster based aggregator selection can be done in Ad-Hoc networks.

Chapter 3

Requirements and Threat Analysis

3.1 Requirements

We identified the following requirements for non-repudiation protocol for ad-hoc networks [8]:

- **Data integrity (R1):**
  Integrity guarantees that a message being transferred is never corrupted.

- **Data confidentiality (R2):**
  Confidentiality ensures that certain information is never disclosed to unauthorized entities. If other peers are involved, the confidentiality of the data exchanged between sender and recipient has to be guaranteed.

- **Authentication (R3):**
  Authentication enables a node to ensure the identity of the peer node it is communicating with [17].

- **Non-repudiation of reception for recipient (R4):**
  For a secure non-repudiation protocol it is crucial, that the reception can't be denied by the recipient.

- **Non-repudiation of content for sender (R5):**
  Non-repudiation of content is another requirement. This is necessary to avoid the fact that the sender can falsify the delivery of document by him.

3.2 Threat Analysis

In our threat analysis we pay attention to following attacks:

- **Denial by sender (A1):**
  Sender A could deny of having sent the document.
- **Denial by recipient (A2):**
  Recipient B could deny the reception of a dedicated document O from sender A.
- **Witness node selection (A3):**
  One of the participants tries to precompute the witness node set and place malicious nodes as witness nodes, which manipulate the protocol [8].
- **Brute-Force attack (A4):**
  Brute force attack is the systematic, exhaustive testing of all possible methods that can be used to break a security system. Attacker tries all possible keys in the keyspace to decrypt a ciphertext.

Chapter 4

Study of GloMoSim Simulator

Simulator is a valuable tool to verify and evaluate the performance of Mobile Ad Hoc networks. The goal for any simulator is to accurately model and predict the behavior of a real world environment. Developers are provided with information on feasibility and reflectivity crucial to the implementation of the system prior to investing significant time and money. Simulation-based testing can help to indicate whether or not these time and monetary investments are wise. Simulation is, therefore, the most common approach to develop or test new protocol for an Ad Hoc Network. The most commonly used Ad Hoc network simulators are NS2, GloMoSim and OPNET. GloMoSim particularly is a popular simulator tested in many previous works. Consequently, it is considered as a trustable simulator. This chapter will cover the GloMoSim Simulator and its architecture.

Nowadays, the popularity of the existing network simulators and in particular that of Ad Hoc Networks varies from one simulator to another. Besides this every simulator is based on its own methodology and models to simulate a real network [6].

4.1 Comparative study of network simulators

4.1.1 NCTUns

NCTUns is a network simulator and emulator able to simulate various protocols used in both wired and wireless IP networks. It is developed by Prof. S.Y. Wang in NCTU of Taiwan. It is a tool that provides a graphical user interface (GUI) written in C++. NCTUns is not as popular as GloMoSim. It has the following key features:

- It provides an easy-to-use GUI environment.
- It directly uses the real-life Linux's TCP/IP protocol stack to generate simulation results and can use a real-life existing UNIX application program as a traffic generator program.
- The programming is not supported by NCTUns. So, simulation parameters are set only by the graphical user interface. Consequently, the manipulation of protocol modules of every node has to be done manually, node by node, or all the nodes at the same time.
- To run simulation, it is necessary to use the graphical user interface. Thus, simulation is very slow especially when the network simulated contains many nodes.

4.1.2 GloMoSim

GloMoSim is a scalable simulation environment for wireless and wired network systems, developed by PCL (Parallel Calculation Laboratory) of UCLA (University of California in Los Angeles).It has the following key features:

- GloMoSim is a scalable simulation environment for wireless and wired network systems.

- It uses PARSEC (Parallel Simulation Environment for Complex systems) that is a C- based simulation language for sequential and parallel execution of discrete-event simulation models.

- GloMoSim is considered as a popular simulator because many researchers tested it.

- The updates of GloMoSim are not regular. The last update was in 2003.

4.1.3 NS-2

NS-2 is the second version of a network simulator tool developed by the Virtual InterNetwork Testbed (VINT) project [14]. It has the following key features:

- It is cheap (does not require costly equipment)

- Complex scenarios can be easily tested

- Results can be quickly obtained - more ideas can be tested in a smaller timeframe

- Controlled experimental conditions

- Disadvantages: Real systems too complex to model

GloMoSim was chosen because it is considered popular and has been tested by several researchers. GloMoSim uses data aggregation concept. GloMoSim provides higher scalability. GloMoSim is the open source simulator. GloMoSim provides layered architecture with easy plug-in capability. GloMoSim supports many Ad Hoc Networking protocols.

4.2 GloMoSim (Global Mobile Information System Simulator)

GloMoSim was developed in 1998 for mobile wireless networks. Global Mobile Information System Simulator (GloMoSim) is a scalable simulation environment for large wireless and wireline communication networks. GloMoSim uses a parallel discrete-event simulation capability provided by Parsec. GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multihop wireless communications using ad-hoc networking, and traditional Internet protocols [15].

GloMoSim Requires 2 components:

a. GloMoSim (Global Mobile Information Systems Simulation Library).

b. Parsec (Parallel Simulation Environment for Complex Systems) [7].

4.2.1 GloMoSim Library

GloMoSim is a scalable simulation library for wireless network systems built using the PARSEC simulation environment. Table 4.1 lists the GloMoSim models currently available at each of the major layers. GloMoSim also supports two different node mobility models. Nodes can move according to a model that is generally referred to as the "random waypoint" model. A node chooses a random destination within the simulated terrain and moves to that location based on the speed specified in the configuration file. After reaching its destination, the node pauses for a duration that is also specified in the configuration file. The other mobility model in GloMoSim is referred to as the "random drunken" model. A node periodically moves to a position

chosen randomly from its immediate neighboring positions. The frequency of the change in node position is based on a parameter specified in the configuration file [8].

Table 4.1: Models currently in the GloMoSim library

Layer	**Models**
Physical (Radio Propagation)	Free space, Two-Ray
Data Link (MAC)	CSMA, MACA, TSMA, 802.11
Network (Routing)	Bellman-Ford, FSR, OSPF, DSR, WRP, LAR, AODV
Transport	TCP, UDP
Application	Telnet, FTP

4.2.2 Parsec

PARSEC (for PARallel Simulation Environment for Complex systems) is a C-based simulation language developed by the Parallel Computing Laboratory at UCLA, for sequential and parallel execution of discrete-event simulation models. It can also be used as a parallel programming language. PARSEC runs on several platforms, including most recent UNIX variants as well as Windows. PARSEC adopts the process interaction approach to discrete-event simulation. An object (also referred to as a physical process) or set of objects in the physical system is represented by a logical process. Interactions among physical processes (events) are modeled by time-stamped message exchanges among the corresponding logical processes. One of the important distinguishing features of PARSEC is its ability to execute a discrete-event simulation model using several different asynchronous parallel simulation protocols on a variety of parallel architectures. PARSEC is designed to cleanly separate the description of a simulation model from the underlying simulation protocol, sequential or parallel, used to execute it. Thus, with few modifications, a PARSEC program may be executed using the traditional sequential (Global Event List) simulation protocol or one of many parallel optimistic or conservative protocols. In addition, PARSEC pro-

vides powerful message receiving constructs that result in shorter and more natural simulation programs [8].

4.3 Architecture of GloMoSim

The node aggregation technique is introduced into GloMoSim to give significant benefits to the simulation performance. Initializing each node as a separate entity inherently limits the scalability because the memory requirements increase dramatically for a model with large number of nodes. With node aggregation, a single entity can simulate several network nodes in the system. Node aggregation technique implies that the number of nodes in the system can be increased while maintaining the same number of entities in the simulation. In GloMoSim, each entity represents a geographical area of the simulation. Hence the network nodes which a particular entity represents are determined by the physical position of the nodes.

The basic structure of GloMosim is as follows:

/doc contains the documentation
/scenarios contains directories of various sample configuration topologies
/main contains the basic framework for GloMoSim
/bin contains the executables and the input/output files
/include contains common include files
/application contains code for the application layer i.e. files for traffic generation
/transport contains the code for the transport layer
/network contains the code for the network layer
/mac contains the code for the MAC layer, including 802.11b
/radio contains the code for the physical layer

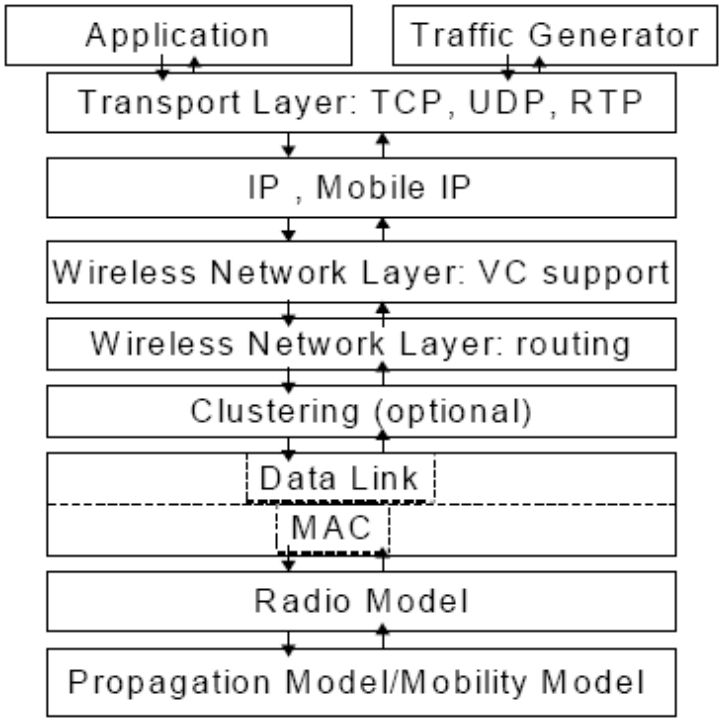

Figure 4.1: GloMoSim Archtecture

All of these files need to be compiled from the /main directory. To do this in Windows, run "makent" to create the executable "glomosim.exe".

4.4 The Visualization Tool

GloMoSim has a Visualization Tool that is platform independent because it is coded in Java. To initialize the Visualization Tool, we must execute from the java gui directory the following: java GlomoMain. This tool allows to debug and verify models and scenarios; stop, resume and step execution; show packet transmissions, show mobility groups in different colors and show statistics.

The radio layer is displayed in the Visualization Tool as follows: When a node transmits a packet, a yellow link is drawn from this node to all nodes within it's power

range. As each node receives the packet, the link is erased and a green line is drawn for successful reception and a red line is drawn for unsuccessful reception. No distinction is made between different packet types (ie: control packets vs. regular packets, etc).

Chapter 5

Proposed Algorithm

Non-repudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.

There are two main steps to achieve non-repudiation:

a. **Witness selection**

b. **Proof of reception**

In this mechanism, TTP is not required because the involvement of other available peers which assist the protocol between sender and recipient can act as witnesses. Using witness selection and proof of reception, nodes communicate. One of the participants is able to compute witness peer set using brute-force attack, place malicious peers and manipulate protocol. Sender and recipient both must be involved in the witness peer selection.

Using proof of reception method source and destination nodes will exchange the secure data with each other. If node fails in some condition to transmit secure data, other communicating nodes can claim and inform other nodes in the topology about malicious node.

5.1 Witness Selection

Nodes are taking part as Global Trusted Third Party in the ad-hoc networks. To select witness node, set the transmission range and send the packets from both source and destination nodes. Nodes in the transmission range will receive the packets. Nodes which take part in transmission are selected as witness nodes set. In this topology some nodes transfer data packets for other nodes. Using this behavior, packets routed with highest probability for other nodes can be achieved. We can find one efficient witness node among the selected witness node set using number of packets routed with higher probability for other nodes.

5.2 Proof of Reception Protocol

Proof of reception protocol will be used for secure data transmission between source and destination node [8].

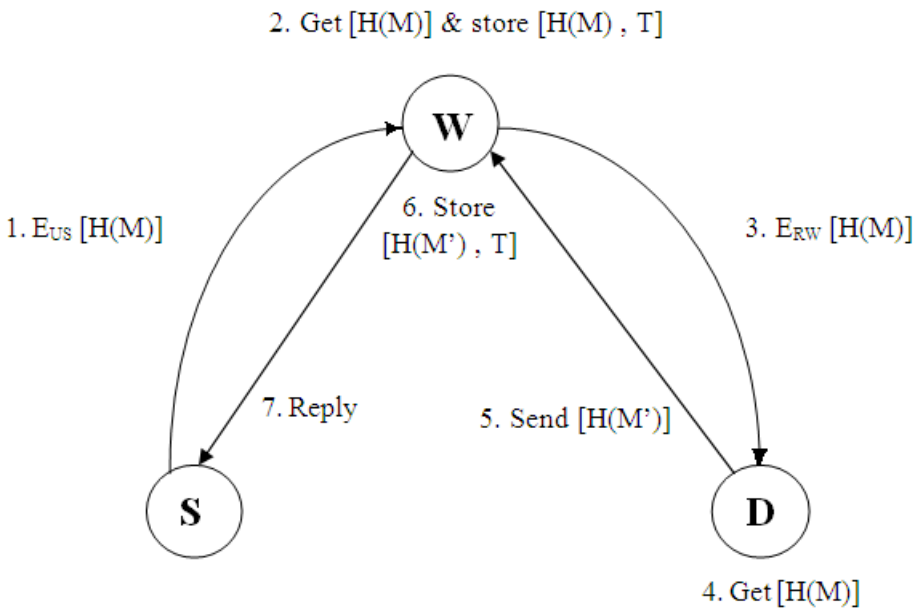

Figure 5.1: Proof of reception protocol

Where,
$[H(M)]$ is hash message of given message.
$E_{US}[H(M)]$ is encrypted hash message using public key of sender.
$E_{RW}[H(M)]$ is encrypted hash message using private key of witness node.
$[H(M')]$ is hash data of hashed message.
T is time-stamp.

Figure 5.1 shows the working of proof of reception protocol in ad-hoc networks. To secure the data, witness node will generate key pair using public key cryptography(RSA Algorithm). Using key pair, data will be encrypted and decrypted amongst source, destination and witness nodes.

Proof of reception works in following steps:

a. Encrypt the hash data using sender's public key and send it to the witness node.

b. Witness node will decrypt the hash data using private key. This process between source and witness node is **secrecy** in public-key cryptosystems [20]. Also store the hash data and time-stamp, so that if in future sender will deny having sent the data then witness node can claim that sender has sent the data earlier.

c. Now witness node will send encrypted hash data using private key to the destination node.

d. Destination node will decrypt it using public key. This process between witness node and destination node is **authentication** process in public-key cryptosystems [20].

e. Destination node will send hashed data of decrypted data by destination node, to the witness node. So that witness node will have the proof that destination node has received the data because it has the hashed data of received data.

f. Store the hashed data by destination node and time-stamp, so that if in future receiver denies having received the data, witness node can claim that receiver has received the data earlier.

This way proof of reception protocol works and we can achieve non-repudiation among the nodes. Suppose you might send registered mail, for example, so the recipient cannot deny that a letter was delivered. Similarly, a legal document typically requires witnesses to sign so that the person who signs cannot deny having done so.

Chapter 6

Implementation

6.1 Basics of AODV protocol

AODV is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-starting, and scales to large numbers of mobile nodes.

When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backward pointers to the source node in the route tables. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hop count, it may update its routing information for that destination and begin using the better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can re-initiate route discovery [18, 19].

6.2 Witness Selection

6.2.1 Witness node set selection

In this mechanism, TTP is not required because the involvement of other available peers which assist the protocol between sender and recipient act as witnesses. To select witness node, set the transmission range and send the packets from both source and destination nodes. Nodes in the transmission range will receive the packets. Nodes which takes part in transmission are selected as witness nodes set. In this network topology shown in figure 6.1 with the configuration parameters shown in table 6.1, node 9 is the source node and node 17 is the destination node.

Table 6.1: Configuration parameters

SIMULATION-TIME	5S
SEED	1
TERRAIN-DIMENSIONS	(1000, 600)
NUMBER-OF-NODES	25
MOBILITY	RANDOM-WAYPOINT
MOBILITY-WP-PAUSE	30S
MOBILITY-WP-MIN-SPEED	0
MOBILITY-WP-MAX-SPEED	15
ROUTING-PROTOCOL	AODV
APP-CONFIG-FILE	./app.conf

Set transmission range:

The radio range is the average maximum distance in usual operating conditions between two nodes. The transmitted power is the strength of the emissions measured in Watts (or milliWatts). A possible methodology to determine the transmission radio range in GloMoSim would be the following [15]:

a. Set the propagation pathloss model (PROPAGATION-PATHLOSS parameter).

b. Fix the received power of the destination antenna (RADIO-RX-THRESHOLD parameter).

c. Fix the distance and calculate the transmitted power according to the selected propagation pathloss model.

d. Set this value to the RADIO-TX-POWER parameter.

Table 6.2: Parameters to set transmission range

PROPAGATION-PATHLOSS	TWO-RAY
PROPAGATION-LIMIT	-111.0
RADIO-FREQUENCY	2.4e9
RADIO-TX-POWER	2.0
RADIO-RX-THRESHOLD	-81.0
RADIO-ANTENNA-GAIN	0.0

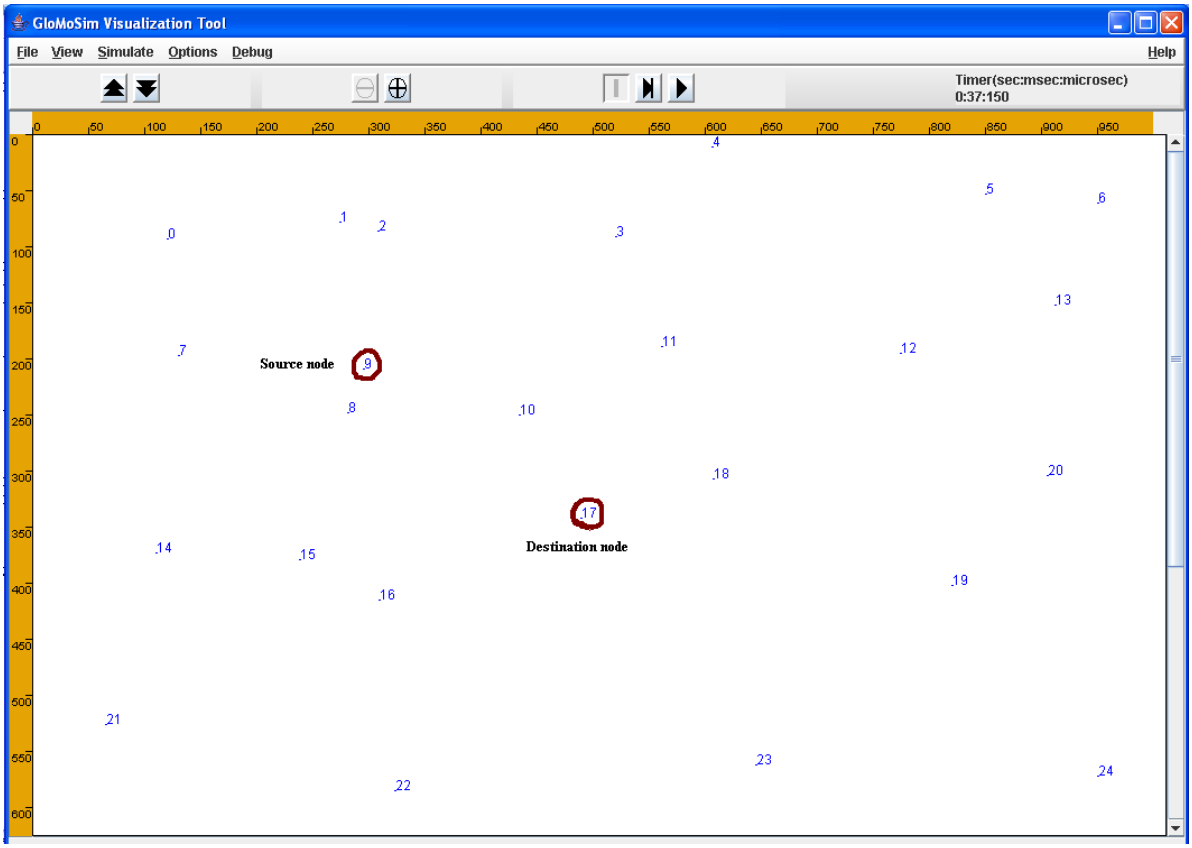

Figure 6.1: Topology indicating source and destination node

The propagation pathloss model indicates us to use the Two-Ray model formula. G_t = G_r = 0 dBm =1 (dimensional, relative to an isotropic antenna), thus:

$$P_r = P_t \frac{h_t^2 h_r^2}{d^4} G_t G_r$$

$$RadioRange \qquad d = \sqrt[4]{\frac{P_t h_t^2 h_r^2}{P_r}} = 140.508m$$

Where, P_r is the received power, P_t is the transmitted power (in Watts or milli-Watts), d is the distance between transmitter and receiver (in meters), G_t is the antenna gain at the transmitter and G_r is the antenna gain at the receiver (adimensional), h_t and h_r is the height of the transmitter and receiver antennas.

We get witness node set which are in the range of the transmission and transmit the data. This way we get witness nodes to be selected for the source and destination node in the network topology. We want witness node set because we want one efficient intermediate node among witness node set which works as TTP to achieve non-repudiation. Figure 6.2 represents selected witness nodes set.

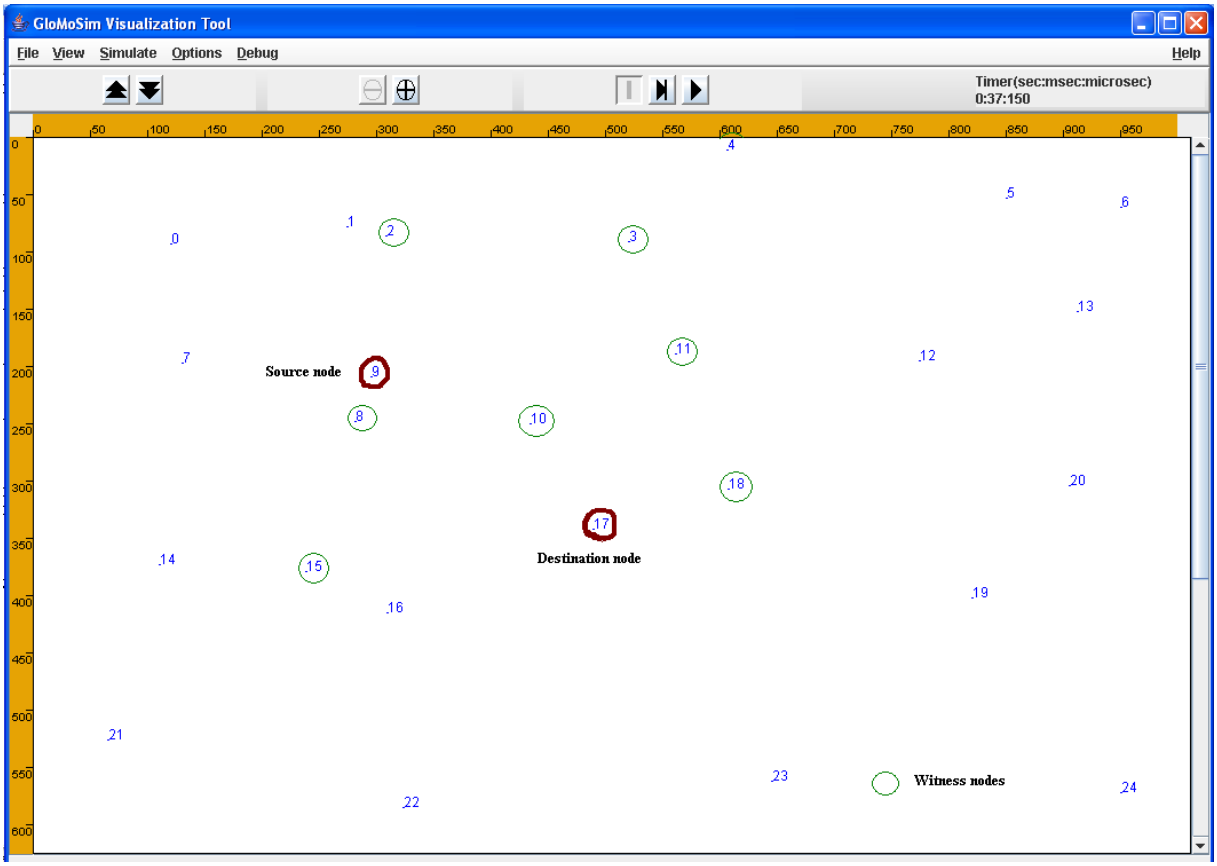

Figure 6.2: Selected witness nodes set

6.2.2 Efficient witness node selection

Now we have the witness nodes, but we need one efficient witness node among all witness node set. There is no need of global Trusted Third Party (TTP) because selected witness node now works as TTP.

Using witness node as TTP, source and destination node will transfer their secure

```
Node:      0, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:      1, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:      2, Layer:       NetworkIp, Number of Packets Routed For Another Node: 10
Node:      3, Layer:       NetworkIp, Number of Packets Routed For Another Node: 13
Node:      4, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:      5, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:      6, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:      7, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:      8, Layer:       NetworkIp, Number of Packets Routed For Another Node: 45
Node:      9, Layer:       NetworkIp, Number of Packets Routed For Another Node: 15
Node:     10, Layer:       NetworkIp, Number of Packets Routed For Another Node: 129
Node:     11, Layer:       NetworkIp, Number of Packets Routed For Another Node: 78
Node:     12, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     13, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     14, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     15, Layer:       NetworkIp, Number of Packets Routed For Another Node: 22
Node:     16, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     17, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     18, Layer:       NetworkIp, Number of Packets Routed For Another Node: 6
Node:     19, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     20, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     21, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     22, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     23, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
Node:     24, Layer:       NetworkIp, Number of Packets Routed For Another Node: 0
```

Figure 6.3: Simulation result of the given example using parameters shown in table 6.1 and table 6.2

information using proof of reception method. Each node transmits the routed packets. Using AODV in this scenario, some node will route the packets for another node. Node having maximum number of routed packets for other nodes is more trustable for the source and destination nodes. So maximum number of packets routed for other nodes can be the efficient intermediate node between the source and destination node. Figure 6.3 shows the results after the simulation of this example.

According to graph shown in figure 6.4, there are 25 nodes in the topology. Node 9 is the source node and node 17 is the destination node. Both node 9 and 17 send request to all the nodes in the topology. Nodes which come in the range of radio range will receive the request. Some nodes will route the packets for other nodes as

intermediate nodes. Node 2,3,8,9,10,11,15,18 are the selected witness node set which route packets for other nodes. In figure 6.4, node 10 is forwarding the packets with highest probability, so it becomes the efficient witness node. Figure 6.4 shows number of packets routed for other nodes corresponding to each node. This way the intermediate node means witness node which can work as the TTP is selected.

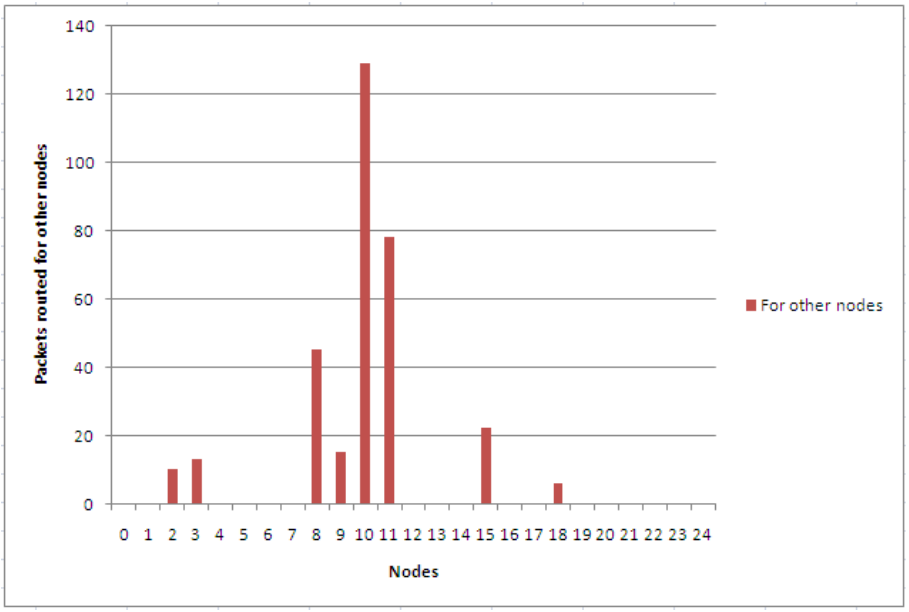

Figure 6.4: Graph of nodes vs packets routed for other nodes

```
Node:      10, Layer:      RoutingAodv, Number of Route Requests Txed = 64
Node:      10, Layer:      RoutingAodv, Number of Replies Txed = 15
Node:      10, Layer:      RoutingAodv, Number of CTRL Packets Txed = 79
Node:      10, Layer:      RoutingAodv, Number of Data Txed = 139
Node:      10, Layer:      RoutingAodv, Number of Data Packets Originated = 10
Node:      10, Layer:        NetworkIp, Number of Packet Attepted to be Sent to MAC: 218
Node:      10, Layer:        NetworkIp, Number of Packets Routed For Another Node: 129
Node:      10, Layer:        NetworkIp, Maximum no of Packets Routed For Another Node: 129
Node:      10, Layer:        NetworkIp, Selected Witness Node by source and destination node: 10
Node:      10, Layer:        NetworkIp, Number of Packets Delivered To this Node: 12
Node:      10, Layer:        NetworkIp, Total of the TTL's of Delivered Packets: 756
```

Figure 6.5: Statistic of particular node from ./Glomo.stat file

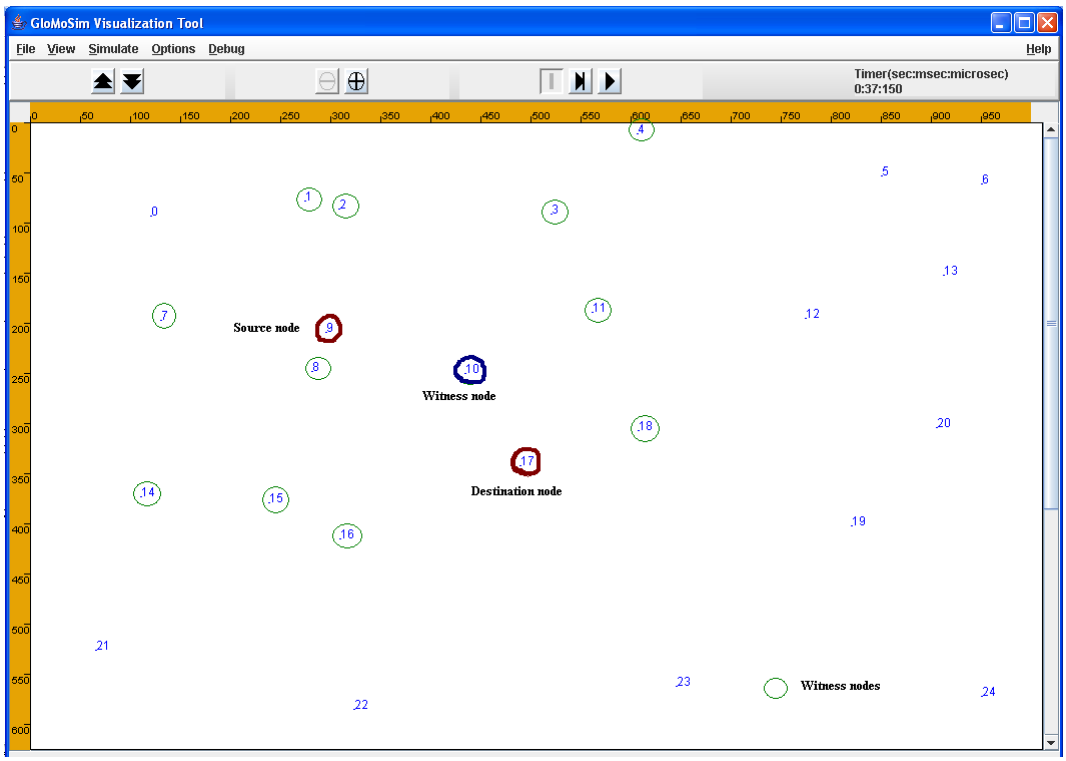

Figure 6.6: Source and destination with the selected witness node

Figure 6.5 shows the output statistics after the simulation in glomo.stat file which shows the maximum packets routed by each node for other nodes and the selected witness node. Figure 6.6 shows the resultant topology after simulation which has the witness node set and the selected witness node as TTP using the parameter packets routed for other nodes.

6.2.3 Testing and Analysis of results

In Random Waypoint Mobility Model (RWPM) a node randomly selects a destination from the physical terrain, and moves in the direction of that destination in a speed uniformly chosen between MOBILITY-WP-MIN-SPEED and MOBILITY-WP- MAX-SPEED parameters (defined in meter/sec) [15]. After it reaches its destination, the node stays there for a MOBILITY-WP-PAUSE time period. These are

some results by changing MOBILITY-WP- MAX-SPEED, so that witness node set selected will defer as per the speed given. Table 6.3 shows the experimented data for MOBILITY-WP-MAX-SPEED 5 to 300.

As we increase the speed of mobility, nodes in the network change the positions

Table 6.3: Simulated data for different mobility max speed up time

Mobility	**Witness set**	**Witness Node**	**Max. pkts for other nodes**
5	1 2 3 4 8 9 10 11 14 15 16 17 18	10	90
10	1 2 3 4 8 9 10 11 14 15 16 17 18	10	104
15	1 2 3 4 8 9 10 11 14 15 16 17 18	10	129
20	1 2 3 4 8 9 10 11 14 15 16 17 18	10	79
30	0 1 2 3 4 7 8 9 10 11 12 14 15 16 17 18	10	131
40	0 1 2 3 4 7 8 9 10 11 12 14 15 16 17 18	10	70
50	0 1 2 3 4 7 8 9 10 11 12 14 15 16 17 18	10	79
100	0 1 2 3 4 7 8 9 10 11 12 14 15 16 17 18	10	79
150	0 1 2 3 4 7 8 9 10 11 12 14 15 16 17 18	10	85
200	0 1 2 3 4 7 8 9 10 11 12 14 15 16 17 18 19	10	61
300	0 1 2 3 4 7 8 9 10 11 12 15 16 17 18 19	10	113

more frequently. So for each speed we get varied witness node set among which one efficient node become witness node. For the mobility speed 5 to 300, nodes set takes part in transmission changes but selected efficient witness node remains the same.

6.3 Key pair generation using RSA

In cryptography, **RSA** (which stands for Rivest, Shamir and Adleman who first publicly described it) is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations [1].

The RSA algorithm involves three steps: key generation, encryption and decryption.

6.3.1 Key generation

RSA involves a **public key** and a **private key**. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated in following way [2]:

a. Choose two distinct prime numbers p and q. For security purposes, integers p and q should be chosen at random, and should be of similar bit-length.

b. Compute $n = pq$.

c. Compute $\phi(n) = (p\text{-}1)(q\text{-}1)$.

d. Choose an integer e such that $1 < e < \phi(n)$ and $\gcd(e,\phi(n)) = 1$, i.e. e and ϕ(n) are co-prime.

e. Determine $d = e^{-1} \bmod \phi(n)$; i.e. d is the multiplicative inverse of e mod $\phi(n)$.

6.3.2 Encryption

Alice transmits her public key (n,e) to Bob and keeps the private key secret. Bob then wishes to send message **M** to Alice. He first turns **M** into an integer $0 < m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the cipher text c corresponding to

$$c = m^e (\text{mod } n).$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

6.3.3 Decryption

Alice can recover m from c by using her private key exponent d via computing. Given m, she can recover the original message $\mathbf{M}$ by reversing the padding scheme.

$m = c^d (\text{mod } n)$.

6.3.4 Key pair generation at witness node in this topology

At network layer, witness node selection has been selected and using RSA algorithm key pairs will be generated.

```
**********************************THIS NODE IS SELECTED AS A WITNESS NODE************
Node:      1, Layer:       NetworkIp, The chosen p and q values are:
Node:      1, Layer:       NetworkIp, Random Prime No. p is : 79
Node:      1, Layer:       NetworkIp, Random Prime No. q is : 197
Node:      1, Layer:       NetworkIp, The value of n = p * q is: 15563
Node:      1, Layer:       NetworkIp, The value of fi(n)=(p-1)*(q-1) is: 15288
Node:      1, Layer:       NetworkIp, The choosen relatively prime to fi(n), e : 5
Node:      1, Layer:       NetworkIp, The calculated d (e inverse) : 9173
Node:      1, Layer:       NetworkIp, The public key(encryption) is : (5,15563)
Node:      1, Layer:       NetworkIp, The plain text : 5210
Node:      1, Layer:       NetworkIp, Encrypted keyword : 12248
Node:      1, Layer:       NetworkIp, The private key(decryption) is : (9173,15563)
Node:      1, Layer:       NetworkIp, Decrypted keyword : 5210
```

Figure 6.7: Key pair generation at witness node

Public key will be known to both source and destination node while private key will be kept private at witness node. Source and destination node will encrypt data using public key and witness node will decrypt the data using private key.

6.4 Involvement of witness node in data transfer

To implement proof of reception protocol, one most important thing is to involve witness node in the data transmission between source and destination node. For that we have to set the route between source to witness and witness to destination node.

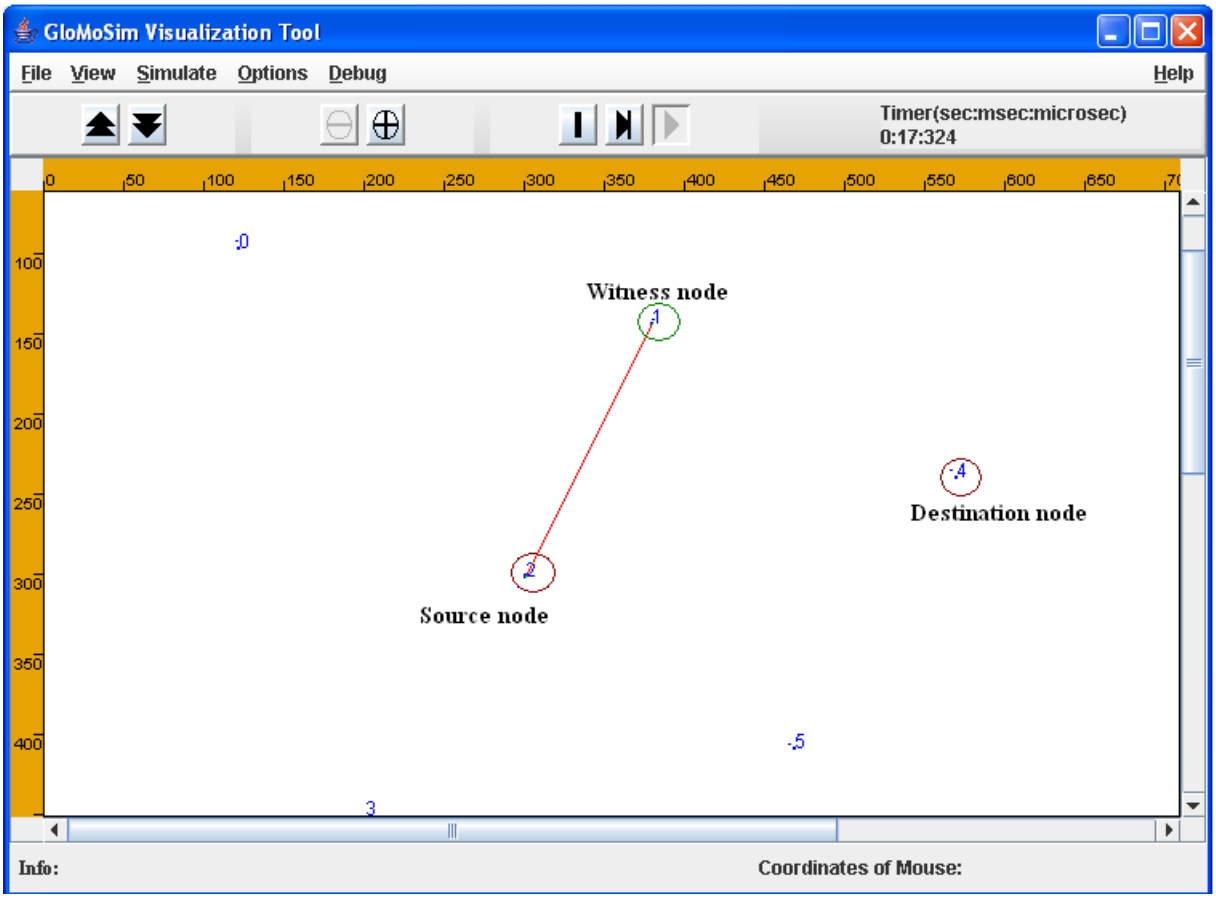

Figure 6.8: Data transmission between source node & witness node

In this topology, source node 2 and destination node 4 will send packets to all the nodes in the network. Using witness selection method, one efficient witness node will be selected which is node 1 in this topology.

```
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 2 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 4.
Data from source node 2 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 4.
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 2 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 4.
Data from source node 2 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 4.
Data from source node 2 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 4.
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 4 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 2.
Data from source node 2 is forwarded to witness node 1.
Data from witness node 1 is forwarded to destination node 4.
```

Figure 6.9: Involvement of witness node in data transfer

Figure 6.8 & 6.9 shows the established route between source to witness and witness to destination node. Figure 6.10 shows the number of packets routed from source to witness and number of packets routed from witness to destination node.

```
Node:      1, Layer:     RoutingAodv, Number of Route Requests Txed = 2
Node:      1, Layer:     RoutingAodv, Number of Replies Txed = 5
Node:      1, Layer:     RoutingAodv, Number of CTRL Packets Txed = 7
Node:      1, Layer:     RoutingAodv, Number of Routes Selected = 0
Node:      1, Layer:     RoutingAodv, Number of Hop Counts = 0
Node:      1, Layer:     RoutingAodv, Number of Data Txed = 24
Node:      1, Layer:     RoutingAodv, Number of Data Packets Originated = 10
Node:      1, Layer:     RoutingAodv, Number of Data Packets Received = 12
Node:      1, Layer:     RoutingAodv, Number of Data Packets witnessnode Received from source node= 14
Node:      1, Layer:     RoutingAodv, Number of Data Packets destination node Received from witnessnode= 14
Node:      1, Layer:     RoutingAodv, Number of Packets Dropped or Left waiting for Route = 0
Node:      1, Layer:       NetworkIp, Number of Packets Routed For Another Node: 14
Node:      1, Layer:       NetworkIp, Maximum no of Packets Routed For Another Node: 14
```

Figure 6.10: Number of data packets routed through witness node

6.5 Transferring Hash Data from source to destination

Proof of reception protocol, will be used for secured data transmission between source and destination node. But data will be transferred via witness node which acts as global trusted third party in the protocol. As data is not directly transferred from source to destination, there is the chance of data corruption by witness node. To achieve data integrity at witness node, we can use hash function. Hash functions are mostly used to speed up table lookup or data comparison tasks-such as finding items in a database, detecting duplicated or similar records in a large file. Hash functions are related to checksums, check digits, fingerprints, randomization functions, error correcting codes, and **cryptographic hash functions.**[3]

A cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the message and the hash value is sometimes called the message digest or simply digest [4].

The ideal cryptographic hash function has four main or significant properties:

- It is easy to compute the hash value for any given message
- It is infeasible to find a message that has a given hash
- It is infeasible to modify a message without hash being changed
- It is infeasible to find two different messages with the same hash

As of 2009, the two most commonly used cryptographic hash functions are MD5 and SHA-1. However, MD5 has been broken; an attack against it was used to break SSL in 2008. SHA-1 is the most widely used of the existing SHA hash functions, and is employed in several widely-used security applications and protocols because in SHA-1 collision is not found [3, 4].

6.5.1 Working of SHA-1

SHA-1 is used to compute a message digest for a message or data file that is provided as input. The message or data file should be considered should be a bit string. The computation is described using two buffers, each consisting of five 32-bit words and a sequence of eighty 32-bit words. The words of the first 5-word buffer are labeled a, b, c, d, e. The words of the second 5-word buffer are labeled h0, h1, h2, h3, h4. A single word buffer TEMP is also employed [5].

Figure 6.11 shows the flow chart of working of SHA-1 to generate hash data from given data. After processing M(n), the message digest is the 160-bit string represented by the 5 words h0 h1 h2 h3 h4. This way hash is generated for any given string. SHA-1 is popular hash function because it avoids collisions.

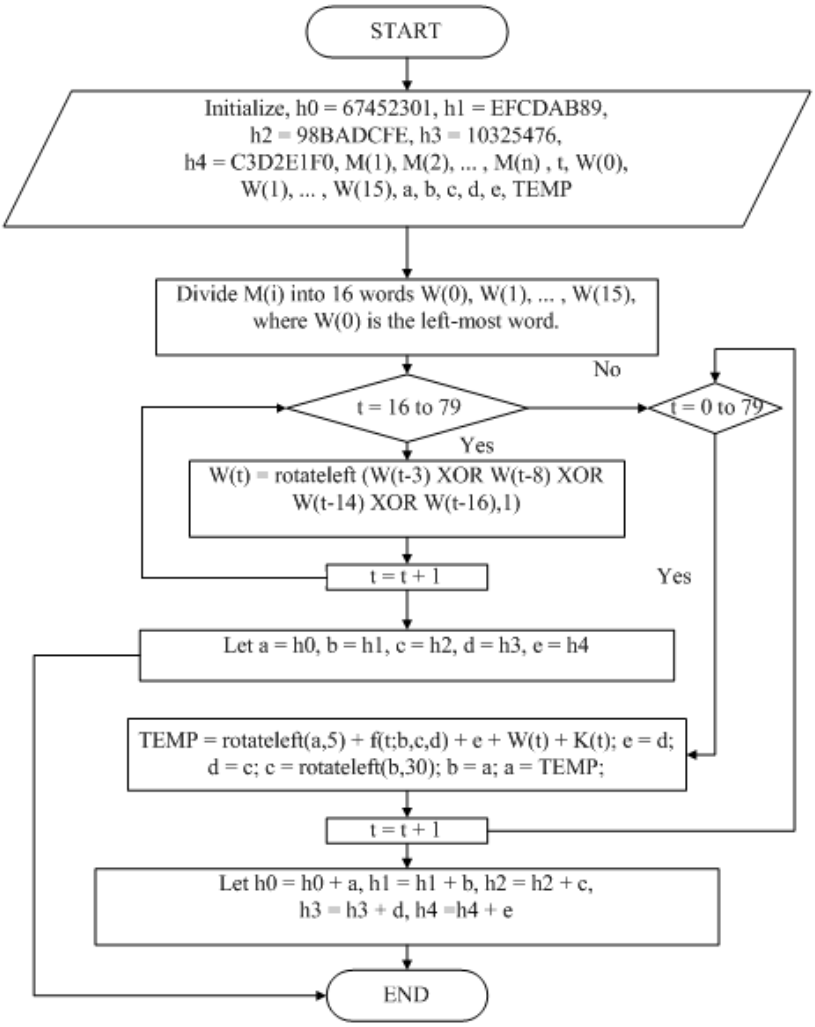

Figure 6.11: Flow chart of working of SHA-1

6.5.2 Implementation of Hash Data

In AODV routing protocol, data packet will be forwarded by RoutingAodvTransmitData function. Using this function data packet which contains message will be forwarded from source node to destination node. Message has a structure which has layerType, protocolType, instanceId, eventType, info, infoSize, packetSize, packet, payLoad, payLoadSize etc fields. Now we want to send hash data from source to destination node. For that we convert the message string into hash data using SHA1

function.

```
Hash data from source node 2 is forwarded to witness node 1.
data=====2378768
Hash data=====3232795515 3416869930 1884264650 2831382568 139492002
Hash data from witness node 1 is forwarded to destination node 4.
-----------------------------------------------------------------
Hash data from source node 4 is forwarded to witness node 1.
data=====2385648
Hash data=====1378710989 4041578104 3691396966 3773526941 3742260644
Hash data from witness node 1 is forwarded to destination node 2.
-----------------------------------------------------------------
Hash data from source node 2 is forwarded to witness node 1.
data=====2384944
Hash data=====191711308 3137163719 915282140 2007807704 2092171562
Hash data from witness node 1 is forwarded to destination node 4.
-----------------------------------------------------------------
Hash data from source node 2 is forwarded to witness node 1.
data=====2430976
Hash data=====3661210606 1584089869 844480495 2506102928 2950170377
Hash data from witness node 1 is forwarded to destination node 4.
-----------------------------------------------------------------
Hash data from source node 4 is forwarded to witness node 1.
data=====2449264
Hash data=====4219138587 1024884745 1331454547 4177356065 4181561984
Hash data from witness node 1 is forwarded to destination node 2.
-----------------------------------------------------------------
Hash data from source node 4 is forwarded to witness node 1.
data=====2428432
Hash data=====1473470756 2106984886 351305687 2893930906 431455702
Hash data from witness node 1 is forwarded to destination node 2.
-----------------------------------------------------------------
```

Figure 6.12: Glomo.stat result file showing hash data transmission between source and destination node

Due to integrity of data at witness nodes, we want to transmit hashed data from source to destination via witness node. As witness node doesn't have the hash to convert the data, witness node can't manipulate the data.

To transmit hashed data, we call RoutingAodvTransmitHashData function from the

RoutingAodvHandleData function. RoutingAodvHandleData function finds the path from source to destination via witness node. When witness node becomes the node address, we call function RoutingAodvTransmitHashData. In this function we call hash function SHA-1 and pass the data as a string. SHA-1 function will calculate the hash data and return to the RoutingAodvTransmitHashData function. This way hashed data will be transmitted from source to destination via witness node. At each time the message is passed to the function will be different, so corresponding to that message, hash data will be generated at each transaction before transmission. Figure 6.12 shows the glomo.stat result file with transmission of hashed data from source node to destination node via witness node 1. We can see that each time, the generated data will be different, so every time hash data is calculated and then transmitted among the nodes.

6.6 Proof of Reception Protocol

According to proposed proof of reception protocol, first step is to encrypt the hash data using public key and send it to the witness node. Only witness node is having the private key with it. Source and destination nodes have public key with them. Figure 6.13 shows the encryption of hash data using the public key and decryption using private key. We can see that after decryption using private key, we get the original hash data before the encryption.

This way we have performed all the steps mentioned in proposed Proof of Reception Protocol. For a particular request, the data is converted into the hash data. Then it is encrypted using public key and sent to the witness node. Witness node has generated one unique identifier to guarantee that sender has sent the data. Witness node decrypted the encrypted data using private key.

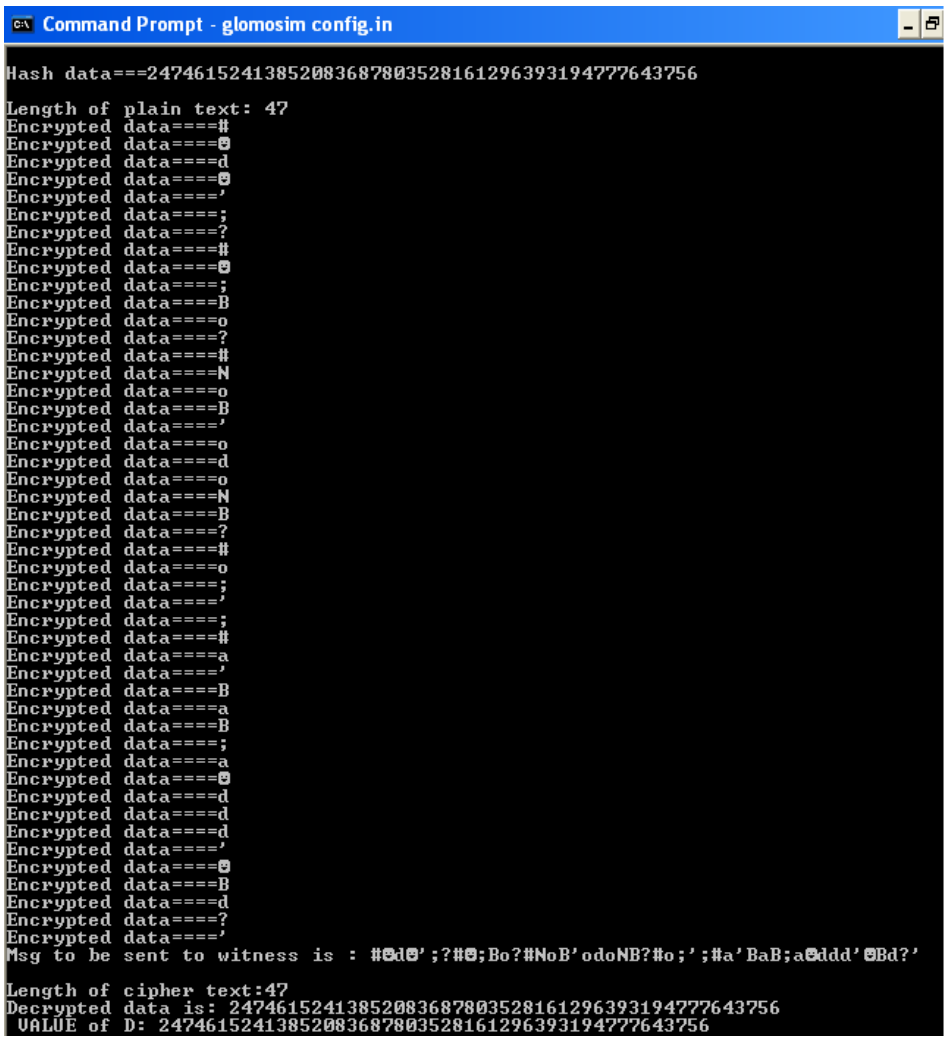

Figure 6.13: Encryption and decryption of hash data

This way secrecy will be achieved between source and witness node. Witness node, now encrypts the hash data using private key and sends it to the destination node. The destination node decrypts the hashed data using public key and authentication is done between them.

Now we want to guarantee the reception by destination so that destination can't deny having received the data. For that the destination node will hash the data of decrypted data and send it to the witness node. So witness node has the hashed data

sent by the destination node, hence witness node can claim that destination node has received the data. Figure 6.14 shows glomo.stat result file of implemented Proof of Reception Protocol for a single request.

```
Data=====2383344

Hash data=====247461524138520836878035281612963931947776437 56

Encrypted keyword to witness node using public key: 255311120

Encrypted hash data from source node 2 is forwarded to witness node 1.

Decrypted keyword at witness node using private key: 247461524138520836878035281612963931947776437 56

Hash data is stored at witness node.

Encrypted keyword to destination node using private key: 255351152

Encrypted hash data from witness node 1 is forwarded to destination node 4.

Decrypted keyword at destination node using public key: 247461524138520836878035281612963931947776437 56

The hashed msg by destination to witness is : 225621466534805766073947133826514245381924907220

Hash data is stored at witness node.
--------------------------------------------------------------------------------------------------------
```

Figure 6.14: Glomo.stat result file of implemented Proof of Reception Protocol for a single request

Chapter 7

Analysis of results obtained

We have transmitted data from source to destination node via the witness node. Witness node can manipulate the data in between. If we transmit hash data, witness node can't manipulate the data as witness node doesn't have the hash value to change the data. By transmitting hash data requirement R1 (data integrity) is fulfilled.

Hash data encrypted by sender node can only be decrypted by receiver node. This way requirement R2 (data confidentiality) is achieved.

Witness node will encrypt the hash data using private key and send it to the destination node. Destination node will decrypt the encrypted hash data using public key. This process between nodes is known as authentication in cryptosystem. Hence requirement R3 (authentication) is fulfilled.

Sender node will send encrypted hash data to the witness node. So witness node has the hashed data as proof that sender has sent the data. This way requirement R5 (Non-repudiation of content for sender) is fulfilled and attack A1 (Denial by sender) is avoided.

To prevent from Denial by recipient attack, destination node will send hashed data of

decrypted data by him using public key and sends it to the witness node. So if destination node has received the data then only the witness node will have the hashed data by destination node. This way requirement R4 (Non-repudiation of reception for recipient) is fulfilled and attack A2 (Denial by recipient) is avoided.

Here sender and receiver nodes take part in witness node set selection and further efficient witness node amongst them. One of the participants can't pre-compute the witness node set. By selecting witness node through both participants, attack A3 (Witness node selection) is avoided.

In this protocol, the generated key is sufficiently large enough that attacker can't decrypt the cipher text with all possible keys. If key size is 128 bits, the number of alternative keys are $3.4 * 10^{38}$ and time required for 1 decryption is $5.4 * 10^{24}$years. This way attack A4 (Brute-Force attack) is avoided [20].

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Most popular protocols like Zhou-Gollmann and fair non-repudiation protocol provide strong fairness and timeliness property. In ad hoc network Node mobility in an ad hoc network causes frequent changes of the network topology. So there is no fixed Trusted Third Party(TTP) in the ad hoc network. The key principle is the involvement of other peers. These peers act as witnesses and assist the non-repudiation protocol operations. In summary, the set of witnesses act as the replacement for the trusted third party known from classical non-repudiation protocols. We need TTP to achieve non-repudiation. Now to secure ad hoc networks, peer - to - peer mechanism can be used as there is no requirement of TTP in communication, because it will involve the other nodes as witness. To select a witness node, select witness node set using transmission range. In ad hoc network some nodes route packets for other node as an intermediate node. So intermediate node which routed packets with higher probability for other nodes can be efficient intermediate node between source and destination node. As we increase the mobility max speed, the set of witness nodes varies. But witness node selected using packets routed with higher probability for another node remains unchanged. So due to lack of trusted third party, we propose

the involvement of other nodes into the evidence process. As per the consideration, this witness selection is the stepping stone to achieve non-repudiation. Proof of reception is a key element for providing secure contract conclusion between members on a market place. Proof of reception protocol works and the requirements like data integrity, data confidentiality and authentication are satisfied. To achieve data integrity hash function (SHA-1) is applied to the normal data and then sent it to witness node. Hashed data is encrypted and decrypted at respective node with with their key which is generated using RSA algorithm to achieve the confidentiality of transmitted data. This design for non-repudiation avoids fraud by sender, fraud by recipient, provides robust selection of witness nodes, which acts as a replacement for trusted third party.

8.2 Future Work

Our Implementation performs non-repudiation for ad-hoc network, still there are certain optimizations which we can do in the witness selection procedure. Witness node can also be selected using secure data aggregation with cluster based data aggregator. This data aggregator procedure uses aggregator function to reduce the number of bits reported to base station, hence we can reduce the energy consumption. Using aggregation scenario we can select witness nodes to overcome the problems in ad hoc network like energy efficiency and bandwidth consumption. By selecting witness nodes using data aggregation scenario, better results may be achieved in terms of energy and bandwidth consumption.

Appendix A

List of publication

1. Purvi Tandel, Sharada Valiveti, K P Agrawal and K. Kotecha: **"Non-repudiation in Ad Hoc Networks"**,The 2010 International Conference on Future Generation Communication and Networking- MANET 2010, on December 13-15, 2010 in International Convention Center Jeju , Jeju Island, Korea, Communications in Computer and Information Science, 2010, Volume 120, pp. 405-415, (2010).

2. Paper titled **"Witness Selection to Achieve Non-repudiation in Ad-hoc Networks"** selected in the 2011 International Conference on Wireless and Optical Communications, may 21-22, Zhengzhou, China. The conference proceeding will be published by IEEE press in IEEE Xplore.

Web References

[1] http://en.wikipedia.org/wiki/RSA

[2] http://iki.fi/priikone/docs/rsa.pdf

[3] http://en.wikipedia.org/wiki/Hash function

[4] http://en.wikipedia.org/wiki/Cryptographic_hash_function

[5] http://en.wikipedia.org/wiki/SHA-1

[6] Maher Ben Jemaa , Nouha Baccour and Heni Kaaniche: "A comparative Study of two Ad Hoc Network Simulators".

[7] Akarapon Kunpisut: "Network Simulation Using GlomoSim".

[8] Lokesh Bajaj, Mineo Takai, Rajat Ahuja, Ken Tang, Rajive Bagrodia, Mario Gerla: "GloMoSim: A Scalable Network Simulation Environment".

References

[1] Zhou, L., Haas, Z.J.:"Securing Ad Hoc Networks", IEEE Network 13 (1999) 2430.

[2] Yu, S., Zhang, Y., Song, C., Chen, K.: "A security architecture for Mobile Ad Hoc Networks", Proceedings of Asia-Pasific Advanced Network(APAN), 2004.

[3] Robinson, P., Cook, N., Shrivastava, S.: "Implementing Fair Non-repudiable Interactions with Web Services", Proceedings of the 2005 Ninth IEEE International EDOC Enterprise Computing Conference (EDOC'05) 0-7695-2441-9/05 , (2005).

[4] Santiago, J., Vigneron, L.: "Study for Automatically Analysing Non-repudiation", ACI Securite SATIN and the IST-2001-39252 AVISPA project.

[5] Lin, Y.-C., Slay, J.: "Non-Repudiation in Pure Mobile Ad Hoc Network", Proceedings of 3rd Australian Information Security Management Conference, pp. 59-66, (2005).

[6] Kremer, S., Markowitch, O., Zhou, J.: "An Intensive Survey of Fair Non-Repudiation Protocols", Elsevier Science 23 April 2002.

[7] Zhou, J., Gollmann, D.: "A Fair Non-repudiation Protocol", Proceedings of the 1996 IEEE Symposium on Security and Privacy (SP '96) , pp.1081-6011/96 (1996).

[8] Conrad, M.: "Nonrepudiation mechanisms for Peer-to-Peer networks", ACM portal, (2006).

[9] Ozdemir, S., Xiao, Y.: "Secure data aggregation in wireless sensor networks: A comprehensive overview", Elsevier Computer Networks 53 , pp. 2022-2037 (2009).

[10] Alzaid, H., Foo, E., Nieto, J.G.: "Secure Data Aggregation in Wireless Sensor Network: a survey", Proc. 6th Australasian Information Security Conference (AISC 2008), Wollongong, Australia , CRPIT Volume 81 (2008).

[11] Bayya, A.K., Gupte, S., Shukla, Y.K., Garikapati, A.: "Security in Ad-hoc Networks", Term paper, CS 685-002 - Topics in Security in Mobile Computing Systems, (2003).

[12] Huang, S.-I., Shieh, S., Tygar, J.D.: "Secure encrypted-data aggregation for wireless sensor networks", Springer Science+Business Media, LLC 2009, Wireless Netw (2010) 16:915-927, (2009).

[13] Sang, Y., Shen, H.: "Secure Data Aggregation inWireless Sensor Networks: A Survey", IEEE Proceedings of the Seventh International Conference on Parallel and Distributed Computing,Applications and Technologies (PDCAT'06) 0-7695-2736-1/06, (2006).

[14] Gilberto Flores Lucio, Marcos Paredes-Farrera, E. Jammeh, M. Fleury, Martin J. Reed: "OPNET Modeler and Ns-2: Comparing the Accuracy of Network Simulators for Packet-Level Analysis using a Network Testbed", WSEAS Transactions on Computers, Issue 3, Volume 2:700-707 July (2003).

[15] Nuevo J.: "A Comprehensible GloMoSim Tutorial", March 4, 2004.

[16] Xiang zeng, Rajiv Bagrodia, Mario Gerla: "Glomosim: A library for parallel simulation of large-scale wireless networks", PADS '98 Proceedings of the twelfth workshop on Parallel and distributed simulation IEEE Computer Society Washington, DC, USA , (1998).

[17] Purvi Tandel, Sharada Valiveti, K P Agrawal and K. Kotecha: "Non-repudiation in Ad Hoc Networks", Communications in Computer and Information Science, 2010, Volume 120, pp. 405-415, (2010).

[18] C.E Perkins and E.M. Royer and S.R. Das: "Ad-hoc On Demand Distance Vector (AODV)routing",IETF Internet Draft, draft-ietf-manet-aodv-05.txt, March (2000).

[19] E.M. royer and C.E. Perkins: "Multicast Operation of Adhoc distance vector Routing Protocol",Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Net working, pp. 207-218, Seattle, WA, august, (1999).

[20] William Stallings: "Cryptography and network security- principles and practices", Pearson Education Inc. and Dorling Kindersley publishing Inc., (2008).

Index