

Master of Science Thesis

# Adaptive Secure Routing in Ad Hoc Mobile Network

November 01, 2004

Prepared and Submitted by:

Abu Raihan Mostofa Kamal

KTH ID: 771101-A754, E-mail: [icss-amk@dsv.su.se](mailto:icss-amk@dsv.su.se)

Supervisor

Jeffy Mwakalinga

[jeffy@dsv.su.se](mailto:jeffy@dsv.su.se)



ROYAL INSTITUTE  
OF TECHNOLOGY

**Royal Institute of Technology (KTH)**

**SecLab**

**Department of Computer and Systems  
Science (DSV).**

**Stockholm, Sweden**

*This thesis reflects 20 full-time workweeks*

## **Abstract**

Secure routing in ad hoc network is a daunting task because of some contradictions between the nature of the network and the associated applications. In this work various types of existing routing protocols have been extensively studied with a view to finding security vulnerabilities. It is followed by highlighting major security attacks on ad hoc on-demand distance-vector (AODV) routing protocol which is on the verge of being the default routing standard for ad hoc network. Both the security requirement of applications and limitations of the mobile nodes have been carefully considered in order to design a feasible solution to counter possible attacks. The uniqueness of the proposed solution lies with the fact that it ensures security as needed by the application which saves both energy and power. The proposition is actually a modification of AODV protocol. The solution uses several security modules which have been well designed prior to the functioning of the protocol. In fine, it demonstrates that the solution is capable to counter security attacks mentioned earlier. Finally the direction for future works has been discussed.

## **Foreword**

Firstly my heartiest gratitude goes to my supervisor Jeffy Mwakalinga who has rendered continuous and encouraging guidance throughout the entire thesis period.

I must pay gratitude to Mamun-or-Rahsid who is now continuing his Doctoral program in the same area at Networking Lab, Kyung Hee University, South Korea. Innovative discussion with him accelerated my work to great extent.

Thanks to my family especially my wife who is far away from me but always keeps closer contact with me during this research work. Her mental support works as a source of my inspiration.

Finally I like to thank my friends and well-wishers to helped me by rigorous reviews of my work and inspiriting suggestion.

## Table of Contents

<b>Chapter 1 Introduction .....</b>	<b>1</b>
1.1 Background.....	1
1.2 General Problem Statement.....	1
1.3 Aim of the Work .....	2
1.4 Scope of the work.....	2
1.5 Methods Used.....	2
1.6 Research Methodology.....	3
1.7 Target Audience .....	3
1.8 Reading Roadmap.....	3
1.9 Ad Hoc Networking – An Overview .....	3
1.9.1 Introduction.....	3
1.9.2 Ad Hoc Network Characteristics .....	4
1.9.3 Commercial Applications .....	4
1.9.4 Other Expected Applications .....	4
1.9.5 Goals in Ad Hoc Network.....	5
1.9.6 Security Goals in Ad Hoc Networking .....	5
1.9.7 Difficulties and challenges.....	6
<b>Chapter 2 Related Research .....</b>	<b>7</b>
2.1 Secure Efficient Distant Vector routing (SEAD) .....	7
2.2 Packet Leashes.....	7
2.3 Ariadne.....	8
2.4 Watchdog and Pathfinder .....	8
2.5 Secure AODV.....	9
2.6 Secure Link-State Protocol .....	9
2.7 Confidant.....	9
2.8 Other Secure Routing Protocols.....	10
<b>Chapter 3 Cryptographic Background .....</b>	<b>11</b>
3.1 Symmetric and Asymmetric Encryption.....	11
3.1.1 Symmetric Encryption .....	11
3.1.2 Asymmetric Encryption / Public Key Encryption .....	12

---

3.2	<i>Cryptographic Hash Functions</i> .....	13
3.3	<i>Digital Signature</i> .....	13
3.4	<i>Public Key Infrastructure (PKI)</i> .....	14
<b>Chapter 4 Routing in Ad Hoc Mobile Network</b> .....		<b>16</b>
4.1	<i>Reactive routing</i> .....	16
4.1.1	Destination-Sequenced Distance-Vector Routing (DSDV).....	17
4.1.2	The Wireless Routing Protocol (WRL) .....	17
4.2	<i>Proactive routing</i> .....	18
4.2.1	Ad hoc On-demand Distance Vector (AODV) routing .....	18
4.2.2	Dynamic Source Routing (DSR) .....	18
4.3	<i>Combination of reactive and proactive routing</i> .....	20
4.4	<i>AODV Protocol in Details</i> .....	20
4.4.1	Path Discovery .....	21
4.4.2	Reverse Path Setup .....	22
4.4.3	Forward Path Setup.....	22
4.4.4	Route Maintenance .....	23
4.4.5	Local Connectivity Management.....	23
<b>Chapter 5 Security Threats</b> .....		<b>24</b>
5.1	<i>Security flaws and attacks on routing protocol</i> .....	24
5.1.1	Traffic Redirection by Modification.....	24
5.1.2	Replay Attacks .....	25
5.1.3	Formation of Routing Loops.....	28
5.1.4	False Route Error .....	29
5.2	<i>Identification of major points of vulnerability</i> .....	29
<b>Chapter 6 Design and Solution</b> .....		<b>30</b>
6.1	<i>Assumptions and Scenario</i> .....	30
6.2	<i>General outline of our contribution</i> .....	31
6.3	<i>Building Blocks of the architecture</i> .....	33
6.3.1	Secure Neighbor Detection.....	33
6.3.2	Trust Factor Assignment.....	35
6.3.3	Workload of Node.....	37
6.3.4	Clustered PKI and hop-to-hop Encryption .....	37
6.4	<i>Modified Routing Protocol</i> .....	39
6.4.1	Operational Mode .....	40
6.4.2	Route Discovery.....	40
6.4.3	Route Selection .....	41

<b>Chapter 7 Discussion of Result and Future Work .....</b>	<b>44</b>
<i>7.1 Results.....</i>	<i>44</i>
<i>7.2 Direction for Future Works .....</i>	<i>44</i>
<b>References.....</b>	<b>45</b>
<b>Appendix : AODV Data types .....</b>	<b>49</b>

# Chapter 1

## Introduction

---

---

### 1.1 Background

Today modern civilization is bestowed with enormous advancement of Information Technology and Mobile Communication. Internet technology has added much ease and speed in all spheres of our life, from office job to personal entertainment. Recently mobile computing has enjoyed a tremendous improvement and enhancement. Excellent rise of processing power and computing power of mobile devices deserves the credit of such proliferation. There are situations where networking applications are badly needed even in absence of Internet connection, for example, in military applications and rescue operation in natural disaster. Furthermore, people using laptop computers may wish to initiate a conference without using the Internet access. Such scenarios depict the necessity of instant networking without any infrastructure more formally an *ad hoc* network. Such network is highly flexible and based on wireless transmission. In contrast of the applications of ad hoc network, it is evident to realize that secure service delivery in such network has become a major concern of the related researchers. Particularly secure routing has become an excellent topic of open research because of the extraordinary gap between the nature of ad hoc network and the security required by its applications.

### 1.2 General Problem Statement

In ad hoc environment much of the research has been done focusing on the efficiency of the network. Therefore there are quite a number of routing protocols that are excellent in terms of efficiency. Considering security has radically changed the situation, for all of the existing routing protocols are designed with an ambitious assumption that the participating players and the network environment are trusted. It highly contradicts with the reality. Most of the secure routing protocols have the following disadvantages:

- ❑ They use asymmetric encryption primitives that are too expensive for energy-constrained devices in ad hoc network.
- ❑ They require global clock synchronization
- ❑ They provide flat security services, i.e. do not cope with the security requirements of the applications

### **1.3 Aim of the Work**

The chief goal of this masters thesis is to provide a trusted solution for routing in ad hoc network. In order to accomplish this a group of related components (i.e. secure neighbor detection module, node characterization module and workload determination module) have been engineered. The routing protocol has been modified by relating these security components. Basically the work is based on the existing routing protocol, AODV. But it can be used as a general framework for any other routing protocol.

### **1.4 Scope of the work**

Recently ad hoc network has gained immense attention for both research and application. The inherent properties of ad hoc network and security requirements of its applications are often contradictory. Security services of Mobile Ad-hoc Network (MANET) can be broadly categorized as follows.

- ❑ Key Management
- ❑ Secure Routing
- ❑ Secure Data Transmission

Due to shortage of time, this work aims at to focus on secure routing and data transmission. Key management has been skipped here. But the assumptions regarding key distribution are quite easy to achieve. This work is basically designed to enhance the security property of existing AODV protocol with some additional modules such as trust level of nodes.

### **1.5 Methods Used**

Securing ad hoc routing is a daunting task because of its some opposite properties such as high mobility, energy-constraint nature of the nodes, no central administration. To accomplish the proposed solution following sequential methods have been used:

- ❑ Study the basic principal of ad hoc network
- ❑ Review of the existing routing protocols
- ❑ Identifying the major security flaws and attacks in routing
- ❑ Find out the major points of vulnerability
- ❑ Design of a set of related security components
- ❑ Redesign the existing protocol using the components



## **1.6 Research Methodology**

As a research methodology ‘**case study approach**’ has been used to finish the work in the desired way. Actually, at the initial stage extensive study in the area of secure routing in ad hoc network has been employed. The primary goal was to identify and establish the concrete boundary of its related work that, in turn, worked as the beginning point of this work. Here several cases have been used in two different ways. Firstly most of the relevant researches have been extensively studied. Chapter 2 has covered it. Again, all possible threats of routing protocol have been examined to find out major points of vulnerabilities.

## **1.7 Target Audience**

The reader should have the basic knowledge of information security. A fundamental understanding of cryptography is also recommended. A concise but concrete concept regarding ad hoc networking has been carried out in section 1.9. So anybody without any prior knowledge of ad hoc networking can well understand the required functionalities from this basic discussion.

## **1.8 Reading Roadmap**

This chapter discusses the aim of the study, general problem statement and methods used to the solution. It also covers the basic operations of ad hoc networking. Chapter two is a summary of the related research. Most of the relevant research has been concisely presented here. Chapter three covers the cryptographic background. Here some fundamental cryptographic primitives have been discussed. Readers who are familiar with information security and cryptography primitives may skip this chapter. Chapter four covers the existing routing protocols in ad hoc networking. In chapter five, major known attacks on routing protocol have been explained. Chapter six consists of our proposed solution. Finally chapter seven discusses the result and the direction of future works.

## **1.9 Ad Hoc Networking – An Overview**

This section is devoted to discuss the basic concepts of ad hoc networking.

### **1.9.1 Introduction**

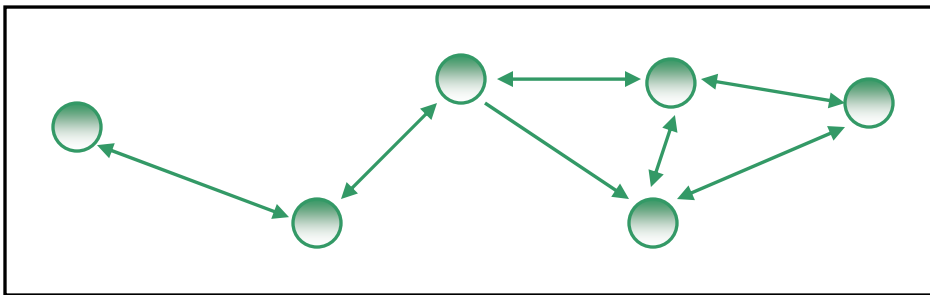
Ad hoc networking is a group of nodes or computers without any fixed infrastructure and connected by wireless communication. A node communicates with another distant node (i.e. out of radio range) by hop-by-hop basis. There are some unique and attractive features of mobile ad hoc network (MANET) as such:

- ❑ No fixed infrastructure
- ❑ Automatic self-configuration and maintenance
- ❑ Quick deployment
- ❑ No centralized administration
- ❑ Reduced administrative cost

### 1.9.2 Ad Hoc Network Characteristics

To achieve the attractive features mentioned above ad hoc network often contains the following network properties [1]:

- ❑ Peer-to-peer
- ❑ Multi-hop
- ❑ Dynamic
- ❑ Zero administration
- ❑ Low power
- ❑ Autonomous
- ❑ Auto-configured



**Figure 1.1** Ad hoc networks

But in reality in most cases it is not possible to strictly follow these properties.

### 1.9.3 Commercial Applications

There are some potential applications in ad hoc networking which can be described as follows:

- ❑ **Emergency Services**
  - Ambulance
  - Natural Disaster
  - Military and Police
- ❑ **Conferencing**
- ❑ **Home Networking**
- ❑ **Personal Area Network and Bluetooth**
- ❑ **Embedded Computing Applications**
  - Ubiquitous computers with short-range interactions
  - Automotive/PC interaction

### 1.9.4 Other Expected Applications

Besides the applications stated above it is expected in the near future that ad hoc networking will be more intensively used for different applications as such:

- ❑ Digital Battlefield Communications
- ❑ Movable Base-stations (for military applications)

- ❑ Range Extension for Cellular Telephone

### **1.9.5 Goals in Ad Hoc Network**

The concept of ad hoc network was founded to satisfy the following initial goals:

- ❑ Scalability
- ❑ To enable larger network
- ❑ Quick convergence
- ❑ Bi-directional communication
- ❑ Loop freedom
- ❑ Unicast

But with the rapid proliferation of ad hoc network in different applications for the last few years, the applications deserve some other properties for ad hoc networking:

- ❑ Security
- ❑ Multicast
- ❑ Quality of Service
- ❑ Smooth handovers
- ❑ Internet gateway operation
- ❑ Service discovery

### **1.9.6 Security Goals in Ad Hoc Networking**

Because of the sensitive applications of ad hoc network security is a vital factor for MANETs. Securing ad hoc network involves ensuring following attributes:

- ❑ Availability
- ❑ Confidentiality
- ❑ Integrity
- ❑ Authenticity
- ❑ Non-repudiation

Availability implies that the requested service is available even though there is potential problem in the system. Lack of availability ensures denial of service (DoS) attacks. In MANETs most of the security breaches are targeted to cause DoS attacks.

Confidentiality ensures that classified information is disclosed to only authorized persons. In many applications of MANETs like transformation of military secrets during war, confidentiality is a major concern.

Integrity implies that message to be transferred is not altered or tampered on the way. Message modification may be either intentionally or unintentionally. Unintentional modification occurs when there is an impairment of radio propagation. On the other hand, attackers often do intentional alteration of message by different attacks on the network.

Authentication ensures that a communicating entity is communicating with another legitimate entity. Without authentication an attacker can impersonate to be an authenticated node and thus gain control over the entire network.

Non-repudiation ensures that once a message has been sent it can not deny afterwards. It is particularly useful for detecting compromised node.

### **1.9.7 Difficulties and challenges**

Ad hoc network has some attractive features which are the major causes for rapid popularity in various applications. But at the same time, these features make it harder to achieve security. They can be summed up in the following way [2][3]:

- ❑ Ad hoc network uses wireless media for transmission. It is beneficial from the point of view that it can be deployed at any time and anywhere. But obviously it suffers from security flaws from wireless communication. Both active and passive attacks such as impersonation, eavesdropping, message redirection, traffic analysis can be performed by an adversary.
- ❑ In ad hoc network there is no central authority. Again, this feature is highly attractive but poses a major barrier to ensure security. Different security mechanisms such as Key Management, Node Authentication, Determination of Node Behavior etc without any central administration are really very difficult to achieve.
- ❑ Ad hoc network is highly dynamic in nature. Node joins and departures are performed without any prediction. Moreover, network topology is always changed in such network. Therefore any static security mechanism will not be applicable in MANETs. In other words, security primitives must be dynamically adjusted to cope with the network which is, of course, a daunting task.

In MANETs most of the nodes are considered to be constrained by power and computational capability. For example, hand held PDAs, Laptops are the best feasible nodes to form an ad hoc network. In order to ensure high degree of security robust encryption with large key (i.e. such as RSA) may be applied but in MANETs it becomes very expensive

## Chapter 2

### Related Research

---

---

Currently ad hoc network is gaining popularity for its attractive features and applications. Much of the research has been carried out to achieve high efficiency. Most of the related researches assume that the computing environment of ad hoc network is trust-worthy [4]. But in reality there is comparatively higher probability (compared with wired network) of being attacked by hostile opponents. Following few sections are devoted to highlight the state of the art in secure routing protocol in ad hoc network.

#### 2.1 Secure Efficient Distant Vector routing (SEAD)

Asymmetric encryption primitives are computationally expensive. In ad hoc network, on the other hand, nodes are commonly constrained by power and computational expensiveness. So, asymmetric encryption primitives can not be a good choice for MANETs. In [5], authors propose a proactive secure routing protocol called SEAD which relies on one-way hash chain for ensuring security. SEAD is based on a simple extension of DSDV called DSDV-SQ. DSDV protocol has the advantage in terms of efficiency as it stores only the next hop not the complete path. A sequence no of the destination has been added to prevent replay attack and formation of loops in the routing. In SEAD authentication is performed by hop-by-hop basis. It uses one-way hash chain to authenticate routing information (such as hop count). As it computationally infeasible to reverse engineer of a hashed value, it guarantees that the path length has not been tampered by any node on the way. Authors in [5] simulated the performance of SEAD with high mobility. It shows improved packet delivery. But it incurs very high overhead (about 5 times than DSDV-SQ). Therefore network congestion may be occurred.

#### 2.2 Packet Leashes

In [6] authors investigated the wormhole attacks on MANETs routing. The solution comes in the name of packet leases. They have proposed two types of leases: temporal and geographical. In order to build temporal lease, all nodes of the network must be tightly clock synchronized. The sending node includes the time of sending the packet within the packet. The receiving node receives it and calculates the receiving time. Then it computes the difference. Based the time calculated and the transmission speed (i.e. the speed of light) the receiving node decides whether the packet traveled too much or not, thereby rejects or accepts it. Another option of temporal lease is to use the expiration time of each packet. In geographical lease it is assumed that all nodes send the local position and time when it sent the packet along with the packet. The receiving node calculates the difference of these two parameters with its own fields (local time and receiving time). It also requires that all nodes must know the maximum speed of a node can travel. The solution needs extra and expensive hardware to provide geographical information.

Another major drawback is that it requires tight clock synchronization which is very hard to achieve in MANETs. Further it demands exact prediction of packet sending and receiving time.

### 2.3 Ariadne

A secure routing protocol called Ariadne has been proposed in [7]. Here the authors classify attacks into two categories: active and passive, but mainly discussed on active attacks. They identified the attack strength by two parameters such as  $x$ , no of compromised nodes and  $y$ , no of total nodes of the attacker. As a result any attacker is presented by **Active-x-y** or **Passive-x-y**. The strength is denoted by increasing order from Passive-0-1 to Active-x-y. To obtain node list authentication Ariadne employs three techniques: the TESLA protocol, digital signature and standard MACs. TESLA authentication is based on hash key-chain and it requires a global clock synchronization. A key publication interval is used to determine the time during which a message to be considered authenticated. A node shares two keys with every other node in HMAC authentication. An HMAC-signature is generated over the route discovery packet and appended with the original packet. The receiving node recalculates the HMAC using the sent key to the recipient and verifies that no node altered or tampered with the previous payload and also verifies the authenticity of the node. The whole authentication process is based on the assumption that the source node is communicating with a destination node and the destination node is not malicious. If the destination node is malicious authentication fails. The overhead incurred in TESLA-authentication protocol depends on the choice of delay length between key publications. If the delay is too short, packets are discarded and retransmission of packets increases. As a result it increases control overhead. For longer delay, overall network performance is decreased.

### 2.4 Watchdog and Pathfinder

In [8] authors propose a new scheme to assess node behavior and thus identify misbehaving nodes. In routing these nodes are excluded for better performance. Every node monitors its direct neighbor to verify that the packets are also forwarded in unaltered way. To do this it maintains a buffer, overhears the packets from the neighbor and compares with another copy cached in the buffer. If they are identical no action is taken, otherwise that node gets one bad remarks. When such bad remarks exceed a certain threshold it is declared as malicious. The monitoring has several limitations. Different possible collision may arise which falsely reports that a node is misbehaving even though it is not. In order to maintain a database of every other node it knows another module called *pathrater* is employed. If a *watchdog* reports a node to be a misbehaving a very high negative value is set by the *pathrater*.

## 2.5 Secure AODV

Manel Gerrero Zapata and N. Asokan [9] give a solution of securing AODV termed as Secure AODV (SAODV). The basic principal of SAODV protocol depends on the authenticating most of the fields of the RREQ/RREP packets and the use of hash chain to authenticate hop count field. A route request single signature extension (RREQ-SSE) is included with the RREQ packets. Based on the maximum diameter of the network the initiator estimates the maximum hop count, and generates a one-way hash chain of length equal to the maximum hop count plus one. Before sending the packet the initiator creates two signatures, one for RREQ packet and another for anchor of the hash chain. Both signatures are included in the RREQ-SSE. It also includes another value which is a function of actual hop count. It is termed as hop-count authenticator. For instance, if the chain of hash values  $h_0, h_1, h_2, \dots, h_n$  are generated such that  $h_i = H[h_{i+1}]$  then the hop-count authenticator  $h_i$  represents the hop count of  $N-i$ . To forward a RREQ packet in SAODV a node first authenticates the RREQ to check that each field is valid. Then it increments the value of hop count field and computes the hash value of hop count authenticator. Finally it rebroadcasts the RREQ along with the RREQ-SSE extensions. When the RREQ reaches the destination it validates the RREQ-SSE extensions and returns a RREP if the authentication is successful. In order to prevent false route error message (RERR) SAODV employs hop-to-hop signature generation scheme.

## 2.6 Secure Link-State Protocol

Panagiotis Papadimitratos and Haas [10] propose the Secure Link-State Protocol (SLSP). It relies on digital signatures and one-way hash chains to ensure the security of link-state updates. Actually it can be termed as Intrazone Routing Protocol in Zone Routing Protocol (ZRP). SLSP is an iterative protocol. It receives link state information by a periodic Neighbor Location Protocol (NLP). In NLP each node broadcasts a signed pair of IP address and corresponding MAC address. When it finds two different MAC addresses for a particular IP address it can instantly inform it to SLSP. Thus it ensures a reasonable level of security of MAC and IP addresses within its tow-hop radius.

## 2.7 Confidant

Confidant [11] is based on DSR. It consists of four related components: the monitor, the trust monitor, the reputation system and the path manager. When a node forwards a packet to another node, the monitor in the sending node checks whether it really forwards to the next node. If it does not, then it triggers an action by the reputation system. The reputation system maintains the local ratings of the nodes. The trust monitor handles the exchange of rating information from other nodes. The path manager selects the path based on the blacklist and ratings of the nodes.

## 2.8 Other Secure Routing Protocols

Baruch Awerbush and his colleagues [12] propose a secure routing protocol which prevents the Byzantine Failures. It has three related phases:

1. **Route discovery with fault avoidance**

It takes the source's weight list as input, performs flooding and uses cryptographic primitives. The weight list is computed from the past history of the node. Finally it finds the full least weight path from the source to destination.

2. **Byzantine fault detection**

It discovers the faulty links from the source link to the destination. An adaptive probing technique has been used to identify a faulty link after  $\log n$  faults occurred, where  $n$  is the length of the path. Cryptographic primitives and sequence numbers are used to protect the detection protocol from adversaries.

3. **Link weight management**

This phase maintains a weight list of links discovered by a fault detection algorithm.

J. Zhen and S. Srinivas in [13] highlight the same problem of wormhole attacks as discussed in [PL]. Basically they discuss the process of secure neighbor detection in AODV protocol. It uses Round Trip Time (RTT) for verification procedure.



## Chapter 3

### Cryptographic Background

---

As the aim of this work is to add security for ad hoc network it uses different security primitives. This chapter is devoted to discuss some basic cryptographic mechanisms. Readers who are with familiar with these may skip this section.

#### 3.1 Symmetric and Asymmetric Encryption

Encryption is the process of encoding a text so that its original meaning is lost. Decryption is the opposite process, a mechanism to reveal the original message from the encrypted one. The term encipher and decipher are used respectively. The original or unaltered version of the message is termed as **plain text** and the encrypted message is called **ciphertext**.

##### 3.1.1 Symmetric Encryption

It is the simplest but very efficient form of encryption. Here one secret is shared between the communicating parties (say our well known sender and receiver, Alice and Bob). The encryption and decryption procedures are mirror image of each other. Two parties communicating with symmetric encryption can be explained with the following diagram [14]:

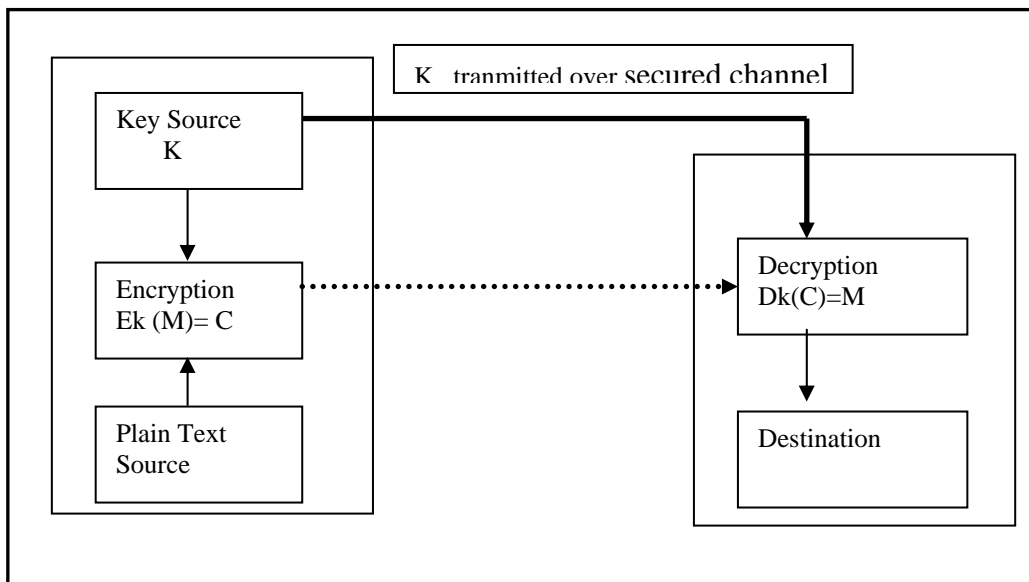


Figure 3.1 Symmetric Encryption [14]

The most challenging task in symmetric encryption is to distribute and manage the shared secret (Key). DES is an example of symmetric encryption.

### 3.1.2 Asymmetric Encryption / Public Key Encryption

Unlike the symmetric encryption it uses two separate keys for encryption and decryption. So keys come in pair called private-public key pair. The sender encrypts the message with his private key. Prior to this operation sender must send its corresponding public key to the receiver. On receiving the encrypted text, the public key is used to decipher the original plain text. The major advantage in asymmetric encryption lies with the fact that it incurs quite high computational expense for an attacker. But on the other hand its application is limited where both security and efficiency are deserved. RSA is a good example of public key encryption.

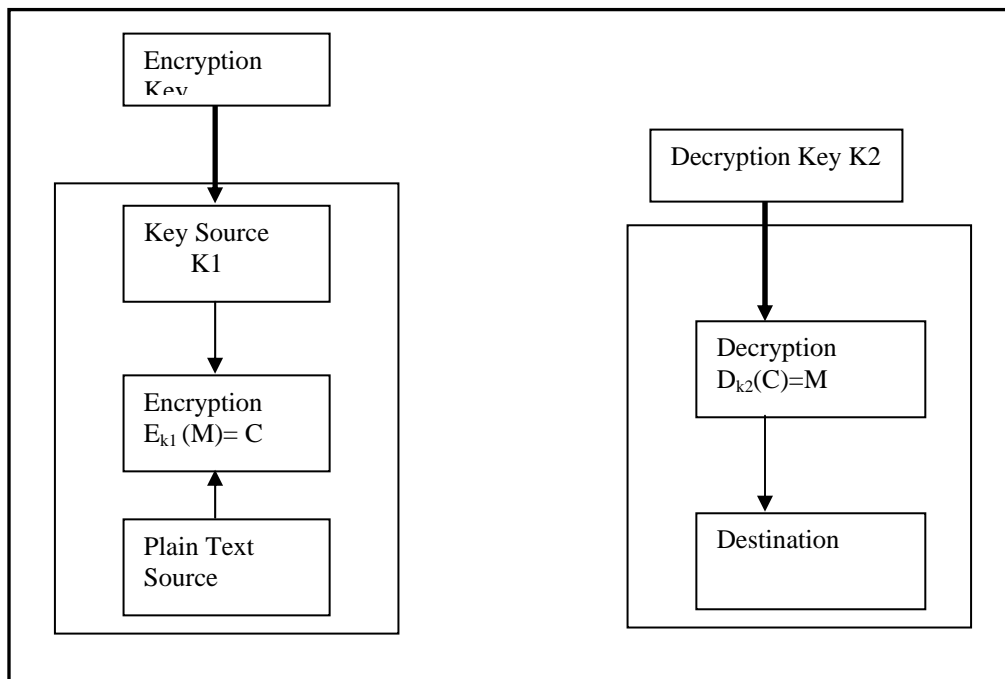


Figure 3.2 Asymmetric Encryption [14]

Another problem lies with asymmetric encryption, as it demands a huge number of key pair for a large network. A good comparative discussion between symmetric and asymmetric encryption can be found in [14] and [15]

## 3.2 Cryptographic Hash Functions

For secure communication it is required that data transmitted is not altered by any entity. Hash functions are the security primitives that ensure data integrity.

Hash function is often called one-way hash function, because it is quite difficult to compute the inverse function. For example, the cube function  $y=x^3$  it is quite easy to compute  $y$  given  $x$ . But the inverse function,  $\sqrt[3]{y}$  is much complicated to compute.

The most common use of hash function is digital signature and data integrity. It is also used for entity authentication [14]. With digital signature hash function is applied to the whole message. Then the hashed value is signed. On receiving the hash value is recomputed and verifies that the received signature is unaltered and from the original source. It saves both time and space as only the hashed value is signed instead of the whole message.

For integrity of data it is widely used. Sender computes the hashed value over the data and sends it along with the original message to the receiver. The destination entity recomputes the hash value from the transmitted message and compares with the hashed value (transmitted)[16].

Hash function can be public (without any key) or it can contain key. The most common hash functions are MD5 (Message Digest 5) and SHA (Secure Hash Algorithm)

## 3.3 Digital Signature

Digital signature is an important cryptographic primitives used for authentication, authorization and non-repudiation [14]. Digital signature has the best use of public key cryptography as discussed in section 3.2. An asymmetric encryption algorithm such as RSA can be used to create and verify digital signature. The simplest form of the protocol works as follows:

1. Alice encrypts the document with her private key, thereby signing the document
2. Alice sends the signed document to Bob
3. Bob deciphers the document with Alice's public key, thereby verifying the signature.

The strength of the digital signature lies with the fact that although the public-private key pair for asymmetric encryption is mathematically related, it is computationally infeasible to derive the private key from the corresponding public key.

Another fundamental process, termed a "hash function," is used in both creating and verifying a digital signature. It has been already discussed in section 3.2.

A digital signature must meet the following two properties [15]

- ❑ It must be unforgeable. If an entity signs a document  $M$  with signature  $S(M)$ , it is not possible for other entity to produce the same pair  $\langle M, S(M) \rangle$
- ❑ It must be authentic. If someone  $R$  receives a digital signature from  $S$ ,  $R$  must be able to verify that the signature is really from  $S$ .

In reality digital signature creation and verification are performed using the combination of hash function and asymmetric encryption.

To create a digital signature the sender first computes the message authentication code (MAC) or hash of the original message and append the code with the message. Then the hash code is encrypted using asymmetric encryption.

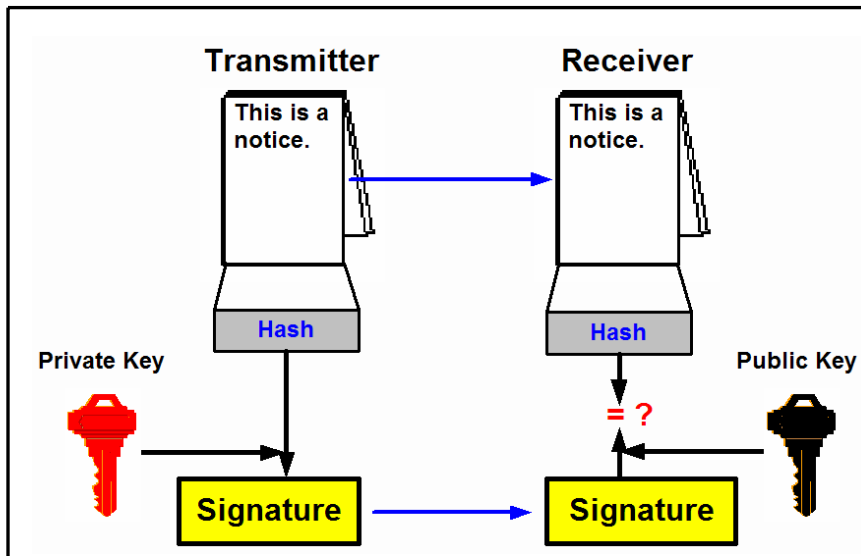


Figure 3.3 Digital Signature [17]

On the reception end the receiver uses the same hash algorithm to compute the hash code of the message, decrypts the encrypted message using the corresponding public key and compares the hash value. The process is illustrated in the figure 3.3

### 3.4 Public Key Infrastructure (PKI)

One of the major security flaws of the pure digital signature is that it is totally based on public and private keys. For example, suppose Alice and Bob are communicating. It is quite possible that the public key of Alice may be captured by some unwanted entity. Therefore the signed document can be decrypted by the attacker. Even worse case occurs when the attacker steals the key and impersonate as Bob with Alice. There is no assurance that the public key sent by Alice to Bob is really Alice's public key.

In order to solve the above problems the concept of public key infrastructure (PKI) has been introduced. It involves the central certification authority often termed as CA. Here Alice and Bob can securely communicate as follow: Alice requests to get the public key of Bob from the CA, and Bob also requests for Alice's public key. Now they can securely transfer document maintaining the integrity and authenticity of the document. Because the public key of Alice is now certified by CA who is trusted by everybody in the context.

The primary purpose of PKI is to distribute public key and certificates with security and integrity [18]. A PKI is a basement on which applications and network security components are built. The success of most of the e-commerce based applications is dependent on the PKI.

## Chapter 4

### Routing in Ad Hoc Mobile Network

---

One of the most exciting and challenging aspects of ad hoc network is the routing issue. Most of the routing protocols are designed for wired and structured network. It is often very hard to adopt these protocols for ad hoc network.

Broadly routing protocols can be classified into two groups: reactive and proactive. This is summarized in the following table [21]:

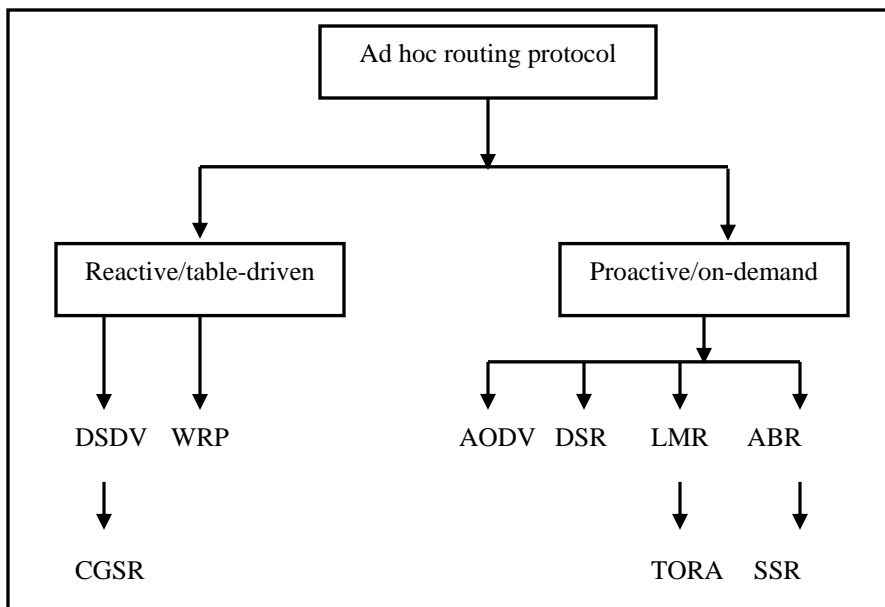


Figure 4.1 Classification of routing protocols

#### 4.1 Reactive routing

In reactive approach routing information is stored and maintained before the actual transmission begins. From application perspective it has the advantage of minimum initial delay as the desired route is already established. This strategy is also termed as 'table-driven' routing protocol [19].

### 4.1.1 Destination-Sequenced Distance-Vector Routing (DSDV)

This table driven approach is based on the classical Bellman-Ford routing mechanism [20]. Each node maintains the routing table with all possible destinations within the network and the no of required hops to reach the destination is also maintained in the table. Each destination assigns a sequence no in order to find out stale routes and prevent routing loops. For table consistency routing information are propagated to update routing table periodically. In order to decrease network traffic for updating routing table two sequential steps are followed. In the first phase, a *full dump* is maintained. Such packets contain all available routing information. Subsequently *incremental* packets are transmitted which carry only the changed routing information since the last full dump process.

### Clusterhead Gateway Switch Routing (CGSR):

Instead of using *flat* address like the previous one CGSR separates the nodes by maintaining different clusters. A distributed algorithm within the cluster selects a cluster head or authority. It has the advantage in terms of less communication information. But it has a heavy overhead when cluster head is changed very frequently. CGSR utilizes the DSDV as the underlying routing strategy. Each node maintains a cluster member table. Inter cluster communication is performed by clusterhead of each cluster [21].

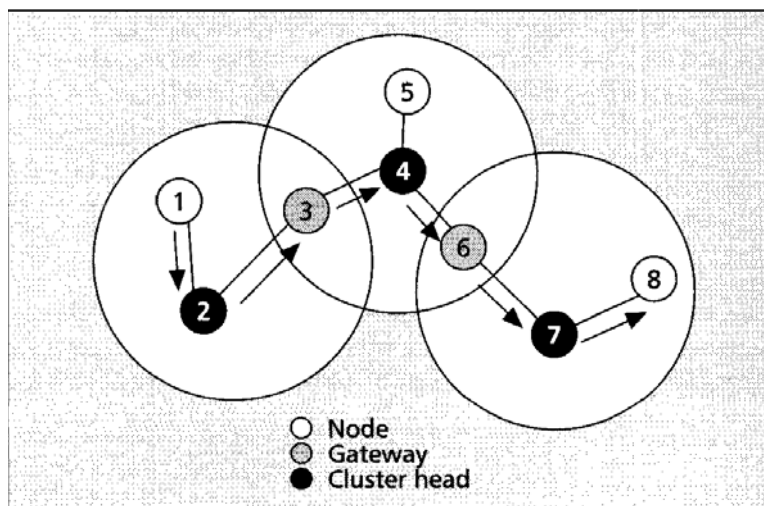


Figure 4.2 CGSR Routing [ 21]

### 4.1.2 The Wireless Routing Protocol (WRL)

In this routing protocol each node maintains the following tables:

- ❑ Distance table
- ❑ Routing table
- ❑ Link-Cost table
- ❑ Message retransmission list (MRL) table

Followings are the information contained for each entry in the MRL table [19]:

- ❑ Sequence no of update message
- ❑ Counter for retransmission
- ❑ Acknowledge-acquire flag
- ❑ A list of updates sent in the update message

Update messages are used to inform other nodes about link changes. Update message is exchanged only between two neighboring nodes. Each node learns about its neighbors by receiving acknowledgement or other message. If a node does not take part in exchange of information for a longer period of time, it maintains the neighborhood connectivity by sending *hello* message repeatedly.

## 4.2 Proactive routing

There is another attractive approach for routing called proactive or ‘source initiated on-demand’ routing protocols. Routes are created only when a source needs to communicate with another node whose path is not known to the source.

### 4.2.1 Ad hoc On-demand Distance Vector (AODV) routing

In AODV protocol a source broadcasts a RREQ request to its neighbors in order to communicate with another node if the source does not find a valid route to the destination in its routing table. Each node maintains a monotonically increasing counter called **broadcast ID**. Its value is increased in every issue of the RREQ packet. So, broadcast ID along with the IP address of the node uniquely identifies the RREQ in the entire network. Besides, the source attaches the destination sequence number that speaks about the freshness of the route information, the greater the number the fresher the information. Each intermediate node increments the hop count field in RREQ by one and broadcasts this RREQ until the RREQ reaches the destination or a node that has a higher destination sequence number than the one in the packet. Multiple replies (Route Replies - RREP’s) may be generated and transmitted along the reverse path. Each intermediate node increments the hop count in RREP and updates its routing table if the RREP has a higher sequence number of the destination or a shorter hop count. This continues until the RREP gets back to the source node [19].

### 4.2.2 Dynamic Source Routing (DSR)

DSR is an on-demand ad hoc network routing protocol consisting of two parts: *Route Discovery* and *Route Maintenance*. In DSR, when a node has a packet to send to some destination and does not currently have a route to that destination in its *Route Cache*, the node starts a Route Discovery procedure; this node is known as the *initiator* or *source* of the Route Discovery, and the destination of the packet is known as the *Discovery’s target or destination*. The initiator transmits a ROUTE REQUEST packet by broadcasting to its neighbors. In the ROUTE REQUEST packet the initiator specifies the target and a unique identifier from the initiator. Each node receiving the ROUTE



REQUEST, if it has recently seen this request identifier from the initiator, discards the REQUEST. Otherwise, it appends its own node address to a list in the REQUEST and rebroadcasts the REQUEST. When the ROUTE REQUEST reaches its target node, the target sends a ROUTE REPLY back to the initiator of the REQUEST, including a copy of the accumulated list of addresses from the REQUEST. When the REPLY reaches the initiator of the REQUEST, it caches the new route in its Route Cache [19].

Route Maintenance is the mechanism by which a node sending a packet along a specified route to some destination detects if that route has broken, for example because two nodes in it have moved too far apart. DSR is based on *source routing*: when sending a packet, the originator lists in the header of the packet the complete sequence of nodes through which the packet is to be forwarded. Each node along the route forwards the packet to the next hop indicated in the packet's header, and attempts to confirm that the packet was received by that next node; a node may confirm this by means of a link-layer acknowledgment, passive acknowledgment [9], or network-layer acknowledgment. If, after a limited number of local retransmissions of the packet, a node in the route is unable to make this confirmation, it returns a ROUTE ERROR to the original source of the packet, identifying the link from itself to the next node as broken. The sender then removes this broken link from its Route Cache; for subsequent packets to this destination, the sender may use any other route to that destination in its Cache, or it may attempt a new Route Discovery for that target if necessary [21].

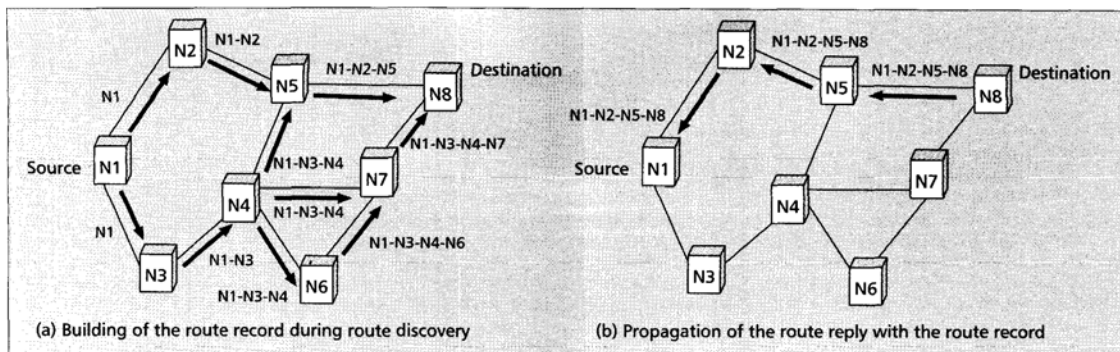


Figure 4.3 DSR [19]

### 4.3 Combination of reactive and proactive routing

Both purely pro-active and purely reactive routing protocols have their own advantages and disadvantages. The **Zone Routing Protocol (ZRP)** is a combination of both the strategies. In this hybrid scheme advantages of both paradigms have been adopted for optimal performance.

The ZRP is not an independent functioning protocol rather; it provides a foundation for other protocols. The separation of a nodes local neighborhood from the global topology of the entire network allows for applying different approaches - and thus taking advantage of each technique's features for a given situation. These local neighborhoods are called *zones* (hence the name); each node may be within multiple overlapping zones, and each zone may be of a different size. The "size" of a zone is not determined by geographical measurement, as one might expect, but is given by a radius of length  $p$ , where  $p$  is the number of hops to the perimeter of the zone.

By dividing the network into overlapping, variable-size zones, ZRP avoids a hierarchical map of the network and the overhead involved in maintaining this map. Instead, the network may be regarded as flat, and route optimization is possible if overlapping zones are detected. While the idea of zones often seems to imply similarities with cellular phone services, it is important to point out that each node has it's own zone, and does not rely on fixed nodes (which would be impossible in MANETs) [21]. Following figure shows an example of ZRP with  $p=2$  (Details are beyond the scope of this study).

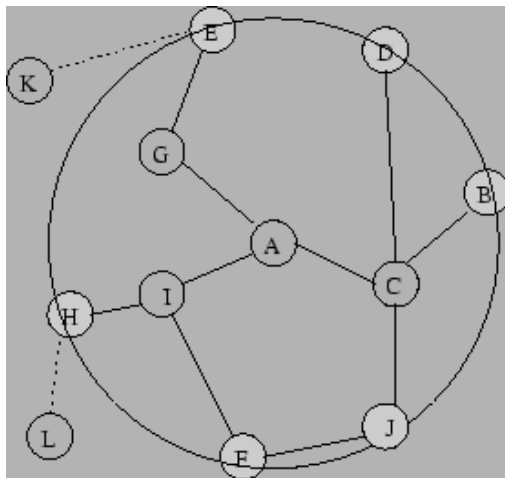


Figure 4.4 ZRP with  $p=2$  [21 ]

### 4.4 AODV Protocol in Details

The Ad Hoc On-Demand Distance-Vector (AODV) routing protocol is particularly designed for ad hoc wireless network. It provides very quick and efficient route establishment between communicating nodes. In most of the protocols the overhead is

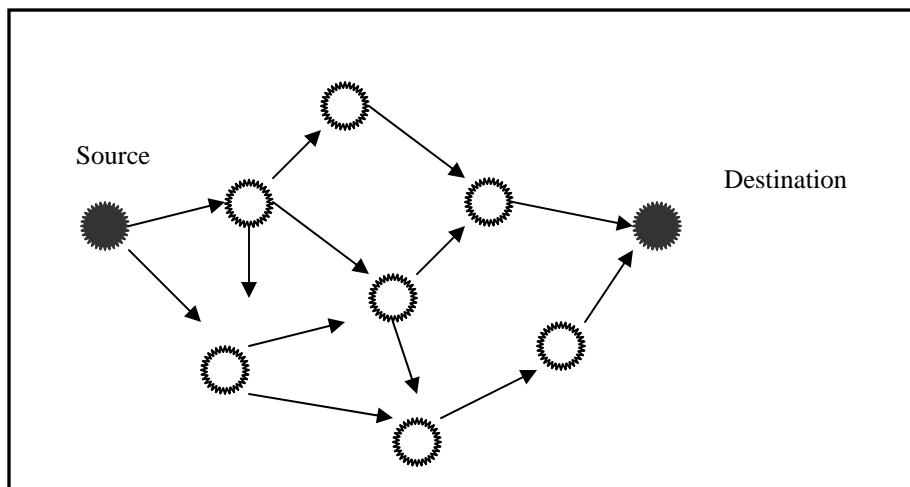
incurred by the fact that each packet transmitted contains the source route to the destination. AODV protocol eliminates this problem by maintaining only the next hop information to reach a particular destination. A monotonically increasing sequence number is used to prevent replay attacks and to ensure loop-free routing [21].

#### 4.4.1 Path Discovery

When a node needs to communicate with another node but does not have the routing information the source node then initiates the path discovery process. Every node maintains two counters [19]:

1. A node sequence number
2. A broadcast ID

The source node then broadcasts a route request (RREQ) packet to its neighbors. Each RREQ is uniquely identified by <IP address, Broadcast ID>. The value of broadcast ID is incremented every time a node issues a RREQ request.



**Figure 4.8** Propagation of RREQ request

A node when receives a RREQ request has two options:

- i. Sends a reply RREP back to the source
- ii. Increments the value of hop count filed and rebroadcasts the RREQ to its neighbors

Action i occurs under either of the two conditions: the node is the destination or the intermediate node knows the route to the destination of the RREQ (with equal or higher sequence number). It is possible for a node to get multiple copies of same RREQ packet. Duplicate packets are simply discarded. Otherwise, it records the following information that is used for reverse and forward path set up procedures discussed in the next sections.

- ❑ Destination IP address
- ❑ Source IP address
- ❑ Broadcast ID address
- ❑ Expiration Time for the reverse path route entry
- ❑ Sequence number of the source

#### 4.4.2 Reverse Path Setup

The RREQ packet travels from the source to several intermediate nodes and finally reaches the destination. In the mean time a reverse path from all nodes to the requesting node (source) is established as shown in the figure 4.6. To establish reverse path entry each intermediate node records the address of the neighbor node from which it got the first copy of the RREQ packet [19][21].

#### 4.4.3 Forward Path Setup

After traveling some nodes the RREQ packet will arrive at either destination node or any intermediate node. An intermediate node when first receives the RREQ checks its own routing table with destination specified in the packet. If so, it compares the destination sequence number with that of contained in the packet. If the packet's destination sequence number is larger than the sequence number in the local routing table the intermediate node will not respond to the RREQ. Then, it rebroadcasts the packet to its neighbors. The intermediate node can only respond when it has a route with a greater sequence number and if the packet has not been processed previously. For reply, an intermediate node unicasts a route reply packet (RREP) back to its neighbor from which it got the packet [21]. The information of RREP packet is explained in the figure.

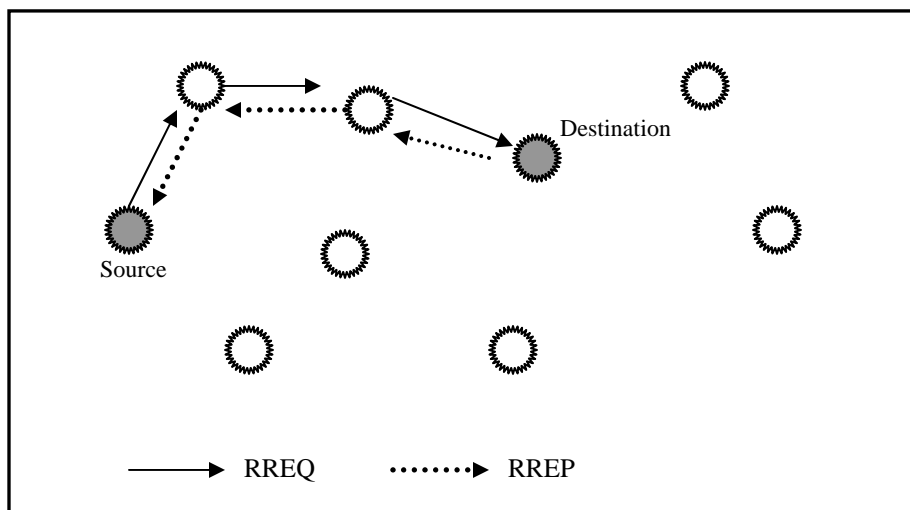


Figure 4.9 Propagation of route information

When an intermediate node receives a RREP, it establishes a forward path entry to the destination in its routing table. The forward path entry contains the following information:

- ❑ IP address of the destination
- ❑ The IP address of the neighbor from which the RREP came
- ❑ The hop count or distance to the destination

After processing the RREP the node forwards the reply towards the source. It is explained in the figure 4.9

It is possible for a node to get multiple copies of the same reply from different neighbors. It forwards the first copy of the request. It will process the next copy if that copy contains a greater sequence number or less hop count. Otherwise the packet is discarded.

#### **4.4.4 Route Maintenance**

After the successful route discovery process the route is maintained as long as the source node requires it. Node movement is very common in ad hoc network environment. If the node movement does not affect the discovered path no action is taken by the protocol. If the source node moves during an active session, it reinitiates the route discovery process. When the destination or any intermediate node moves a Route Error message (RERR) is sent back to the corresponding nodes. The error message is initiated by the node that is closest to the source [19].

#### **4.4.5 Local Connectivity Management**

Neighboring information is maintained by periodically broadcast message. Each time a node receives a broadcast from its neighbor it updates the lifetime for that node in the local routing table. If a node does not broadcast anything within the last *hello\_interval*, it then broadcasts a Hello packet to inform its neighbors to inform that it is still within its radio signal [19].

---

## Chapter 5

### Security threats

---

---

#### 5.1 Security flaws and attacks on routing protocol

In fact we consider AODV as the default routing protocol as it is presently going to be the acceptable standard for ad hoc network. So, we will highlight the major attacks on AODV or major flaws of this protocol. It is to be noted that it is not hard to transform similar type of attacks on other protocols, DSR for example.

Known attacks on AODV are as follows[13][22]:

- i. **Traffic redirection by modification**
- ii. **Replay attacks**
- iii. **Loop formation by spoofing**
- iv. **False Route Error**

In the following sections we discuss them in details.

##### 5.1.1 Traffic Redirection by Modification

In AODV protocol the main design issue is to achieve efficiency in ad hoc network environment. Expensive encryption is a feasible solution due to the energy-constraint property of the nodes participating in the network. In routing of packets, there are both mutable and immutable fields. Link to link encryption is not possible for mutable fields like hop count and destination sequence number. Therefore an attacker can easily modify them and cause different security problems in routing.

###### 5.1.1.1 Modification of Sequence Number

In AODV protocol a monotonically increasing sequence numbers to a particular destination maintains each route. Here any node may divert traffic through itself by advertising a better route to a destination, i.e. a sequence number better than the authenticated value. It could also cause DoD attack.

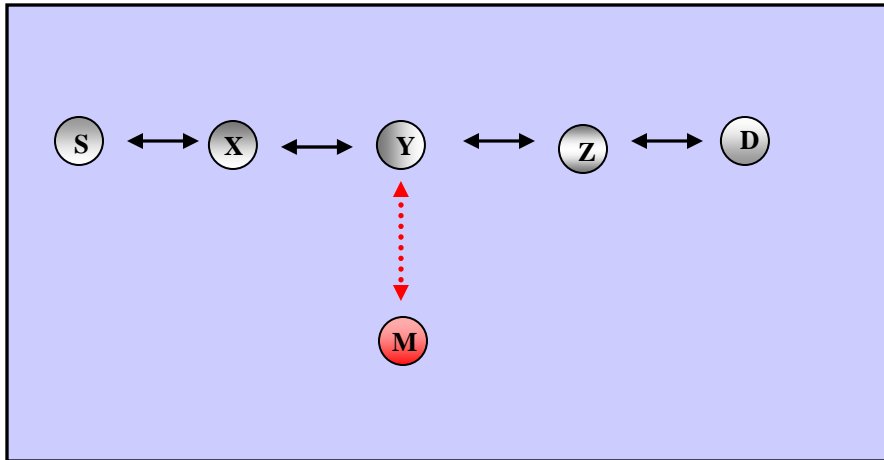


Figure. 5.1 Attacks using modification

Suppose node S in figure 5.1 sends a RREQ with destination D. A malicious node M can receive it and read the destination sequence number as it is not encrypted. So, M can unicast a RREP reply with greater destination sequence number to Y. Thus M can redirect all traffic to itself. Afterwards, the original copy of the RREP reply comes back to source S, but S already has received a RREP with greater sequence number. So S will drop the packet and a denial of service attack is launched.

### 5.1.1.2 Modification of Hop Count

In AODV protocol attacker can set the value of hop count field to zero so that it can later include itself with the route. Or it can set the value to infinity to exclude from the route.

## 5.1.2 Replay Attacks

There are 2 types of replay attacks in ad hoc network [13].

### 5.1.2.1 RREQ Flooding attack

In AODV protocol when a node wants to communicate with another node but does not have the route information it broadcasts a RREQ packet in an incremental way, which is bounded by the value of TTL in the IP header. The objective is to reduce flooding overhead. If it fails to receive any route information then it increments the broadcast diameter by a predefined value, the process continues until a valid route is discovered. The expansion of ring circulation is updated using *binary exponential back-off* algorithm which states that the radius is doubled by setting the appropriate value of TTL in each iteration. It is shown in the following figure.

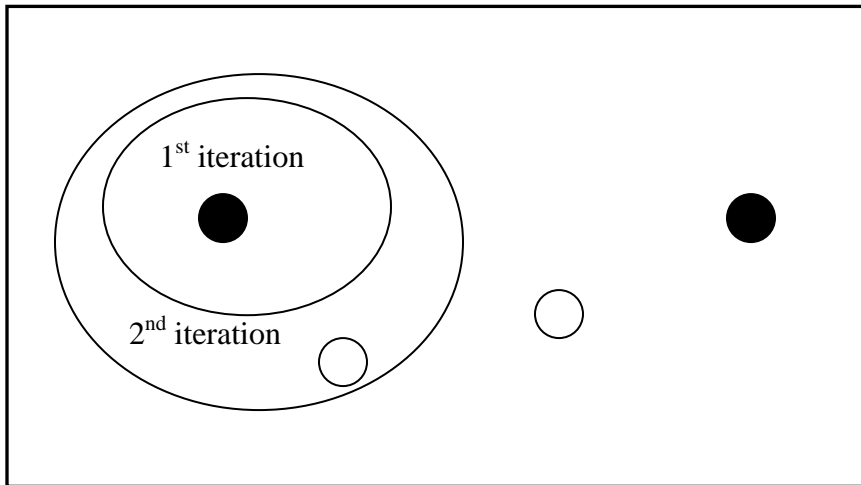


Figure 5.2 Ring Expansion for routing

Each node maintains a sequence number and RREQ\_ID to avoid the packets from being replayed. The higher the sequence numbers the fresher the information about the particular destination. It is easy to note that an attacker can record the RREQ of one node and circulate it to another area. If the new area is up-to-date no harm is caused, as it simply discards the packet. But the information of the nodes in the new area is not up-to-date it will cause extra unnecessary processing of packets which, in turns, causes a denial of service attack.

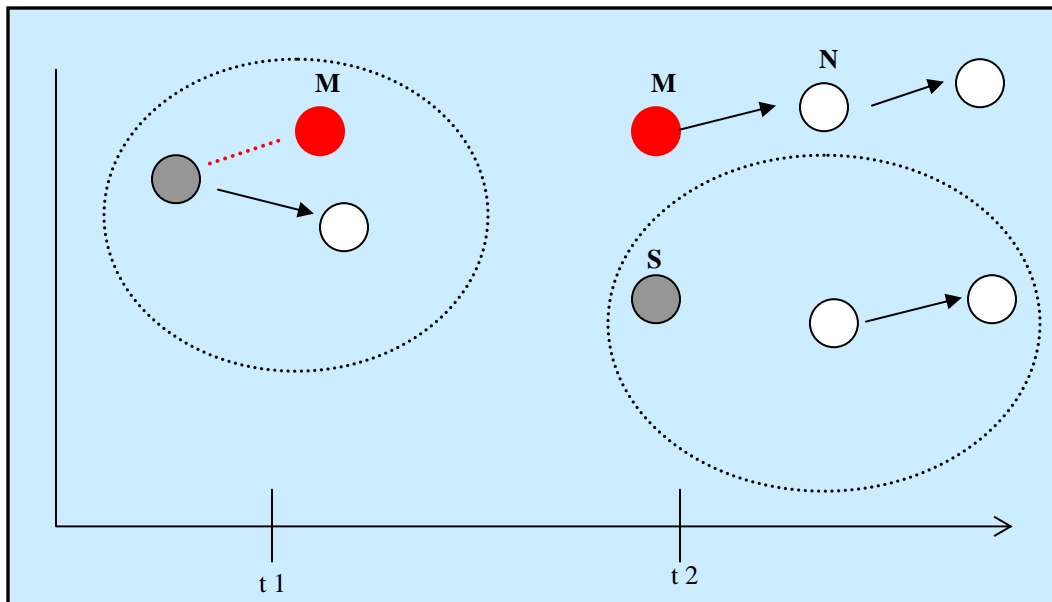


Figure 5.3 RREQ Flooding



In the figure 5.3 at time t1 attacker M overhears the RREQ sent by S to its neighbor. At time t2 node M replays the RREQ to another node N which is not yet informed about the freshest RREQ (outside of the radius). Therefore it will start processing the packet, so a false route discovery process is started to consume the resource and energy of the nodes.

### 5.1.2.2 Wormhole attack

It exploits the following two properties:

1. In AODV protocol when a node (source) needs to communicate with another node (destination) but the source does not have the route, it broadcasts RREQ to its neighbors. The process continues until an intermediate node having the fresh route to the destination is found (or the destination itself is found). To prevent unnecessary processing of same RREQ packet from different neighbors, each node processes the RREQ packets that **first arrives**, thereby ignores other copies
2. A direct (tunneling) link (wired/wireless) is faster than a general hop-by-hop propagation.

Usually it involves two attackers, one near the source and another near the destination. When a source broadcasts an RREQ packet the first attacker records it and transmits directly through a tunnel to the second attacker (who is near the destination). Any neighbor of destination receives the RREQ from the attacker it normally processes it. In the meantime the original RREQ comes to it by hop-by-hop propagation, it simply discards it. Because, already it has received the packet.

Thus can cause DoS attack. Further it bounds the source and destination to use the attacker nodes.

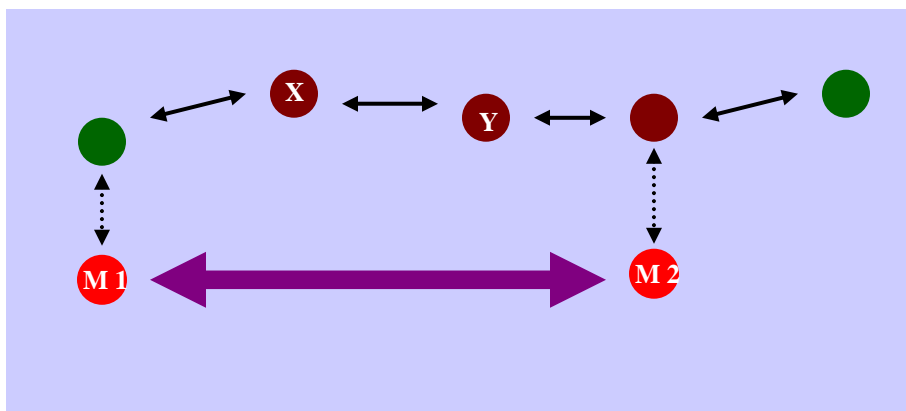


Figure 5.4 Wormhole Attack

Figure 5.4 explains the wormhole attack. S wants to communicate with D, so, it broadcasts a RREQ packet to its neighbor X. In wireless transmission it is quite trivial to be transparent for another node within the radio signal of the sending node. So an attacker M1 records the request and tunnels it through a fast link-to-link channel to another attacker M2 placed near the destination as shown in the figure. Obviously node Z will get the request first from M2 without any detection, this is because link-to-link communication is faster than multi-hop communication. So, Z processes the request. Thus the attackers force node S to use the route via M1 and M2 to reach D. Furthermore, when Z gets the original RREQ from its neighbor Y it will drop the packet as specified by the routing algorithm [6].

### 5.1.3 Formation of Routing Loops

This attack is based on the misinterpretation of identity (i.e. IP or MAC address). It can be best explained by the following example [22].

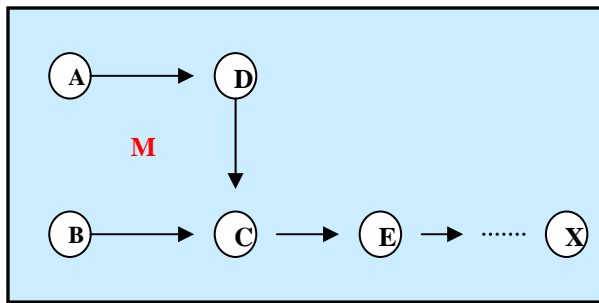


Figure 5.5 (a) Routing Loop[22 ]

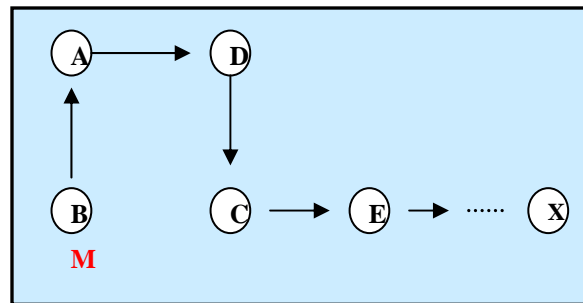


Figure 5.5 (b) Routing Loop[22 ]

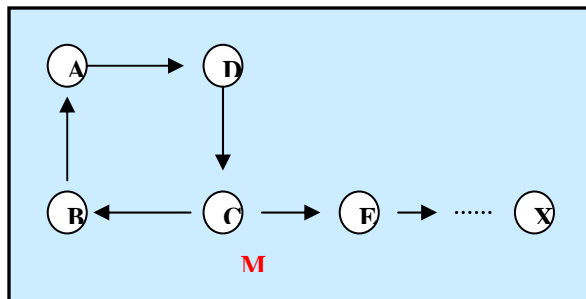


Figure 5.5 (c) Routing Loop[22 ]

After successful RREQ/RREP message exchange we assume that there exists a path among the five nodes with a remote node X as shown in the figure 5.5 (a). Here A can hear D and B, D can hear A and C, B can hear A and C, C can hear D, B and E, and finally E can hear C. An attacker M can learn about this topology by examining the RREQ/RREP messages for route discovery. In order to begin the attack M changes its

MAC address to the MAC address of A and places itself close to B but out of range of A as shown in figure 5.5(b). Then it sends a RREP reply with less hop count than sent by C. So, it redirects the route towards X through A as shown in figure 5.5(b). Now M impersonates B (i.e. copies the MAC address) and sends a RREP reply with less hop count than sent by E. So C changes its route towards X through B. Now a routing loop is formed and 4 nodes A,B,C,D are unreachable to X.

#### **5.1.4 False Route Error**

After a route from a source S to a destination D has been established, the route is maintained as long as it is needed by the source. If the source changes location, a new route discovery procedure is launched. When the destination or any intermediate node (i.e. any participating node of the route discovered) changes its location a route error message RERR is sent back to the active nodes of the path. This has been explained in chapter 4. An attacker M in figure 5.1 can copy the MAC address of Z and sends a RERR error message to Y. Y would assume the message is from Z and there is a link break between Z and D. So it will remove the corresponding entry from its own routing table and also forwards the message to its neighbor (towards S) X. X also deletes the entry from the routing table. In the same way S deletes the entry. If M repeats the process as soon as there is a route discovered from S to D it can successfully cause a permanently denial of service attack.

## **5.2 Identification of major points of vulnerability**

By summarizing the above attacks on routing protocol it is evident that “Secure Neighbor Detection” is the basic building block of our proposal. Because it is trivial to demonstrate that other building block of secure routing protocol such as “Authenticated Route Discovery” and “Authenticated Route Setup” can be constructed using our basic building block “Secure Neighbor Detection”

Again, the RREQ packet some fields are mutable (i.e. hop count) and some are non-mutable. Modification of some of these fields could cause several security attacks. In AODV protocol these fields are sent in plain text. Hence some lightweight mechanism for encryption/decryption must be adopted.

## Chapter 6

### Design and Solution

Ensuring security in ad hoc network is not an easy task. This is due to some unique and attractive properties which are often contradictory to security assurance. For instance in MANETs it is

- ❑ Wireless
- ❑ No fixed infrastructure
- ❑ Dynamic in nature
- ❑ No central authority

In this work we have investigated various security threats and vulnerabilities of existing routing protocols in ad hoc network. At the same time the present and expected applications of ad hoc network reveal that they require high degree of security. Ensuring complete security is not a feasible target to achieve. Because, there always exists an unavoidable gap between the intruders and the users. So our proposal aims at ensuring ‘*a probabilistic approach*’ for secure routing in MANET.

#### 6.1 Assumptions and Scenario

**Assumptions:** Our entire proposal is based on the basic operations of AODV protocol which is on the verge of being the default standard for routing protocol in MANET. The details of AODV protocol is discussed in section 5.1.

In fact this work adds several security modules with the existing protocol as shown in the figure:

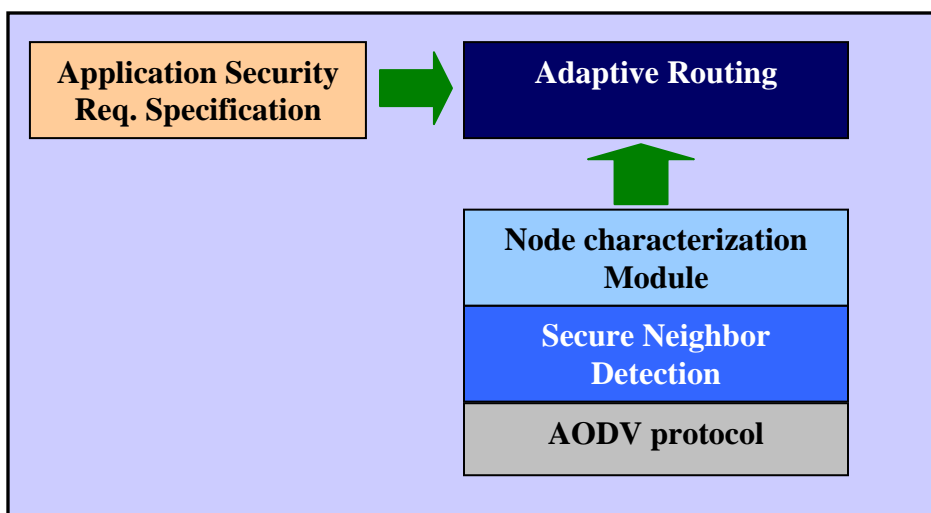


Figure 6.1 Conceptual Framework

**Scenario:** A group of researchers gather in a new place, their purpose is to exchange of different ideas, views and criticism of the related research. Apart from formal presentation and talks most of them are eager to communicate through a ‘*ad hoc and secure*’ network that will be dynamically formed.

We classify different applications with specific security requirements as follows:

Application	Security Requirement
Exchange of new and innovative ideas addressed to specific audience	Very High
Review of newly proposed idea	High
Review of the existing research in the related field	Low
Other exchange of non-classified information	N/A (Open to all)

Figure 6.2 Assumed security requirement of applications

## 6.2 General outline of our contribution

It is evident that existing AODV protocol is very efficient and suitable particularly for MANET. But our analysis shows that it suffers from different security flaws which is covered in [section 5.1](#)

In AODV protocol a source node wishing to communicate with a destination node first broadcasts a RREQ packet to its neighbors. Upon receipt the desired destination another reply packet RREP is sent back to the source. Each node maintains only the next hop information to reach to destination.

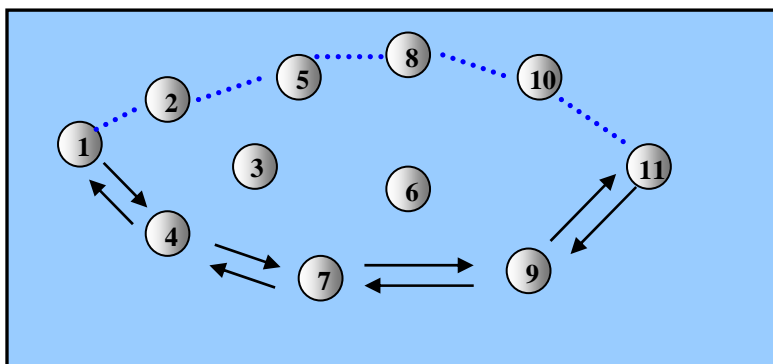


Figure 6.3 Trusted route discovery

For example, Node 1 broadcasts RREQ to its neighbors to communicate with node 11. In AODV protocol a reply is sent back either by an intermediate node or by the destination itself. It is illustrated in the figure. To reach the destination the source node now follows the following path **4---7----9**. We propose that to reach the destination (node 11) there might be another path i.e. **2---5---8-----10** which uses more hop counts but may be much reliable in terms of packet forwarding history.

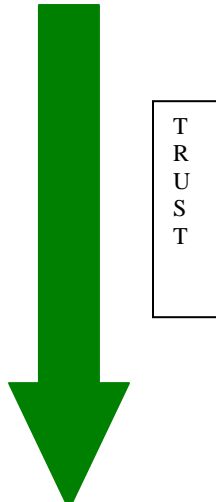
In the AODV protocol the route selection criteria are:

- ❑ **Hop count**
- ❑ **Destination Sequence Number**

Hop count determines how short the route is, and the sequence no of the destination speaks about the freshness of the route information. Therefore the route selection metric is clearly independent of the security level of the application and trust factor of the participating nodes.

In our proposed model the source node waits for a predefined period of time *t* to receive some other set of routes. Furthermore, each node maintains a local database of its neighbors with dynamically updated trust factor. For processing each RREQ packet each node assigns an additional field i.e. trust level of the node from which it just has got the RREQ packet. For simplicity we have defined the following trust levels:

<b>Trust Factor</b>	<b>Meaning</b>
-1	Malicious
0	Not trusted
1	Low trust level
2	
3	
4	Standard trust level
5	
6	
7	
8	High trust level
9	



**Figure 6.4** Assumed trust levels of nodes

Each node dynamically updates the trust level of its neighbors. It is explained the next section.

Besides, we have introduced more two parameters to determine the suitable route for any applications:

- ❑ Required security level of the application
- ❑ Workload of the node

In the existing secure routing protocol the encryption is independent of the required security level of the application. So, every node spends equal time for the expensive encryption mechanism that can be well minimized by our adaptive proposal. Route selection must be a function of the following parameters along with hop count.

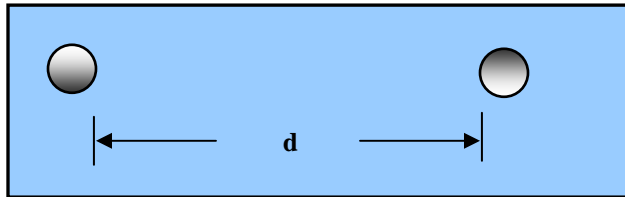
- ❑ Trust level of the next node, security level required by application and workload of the nodes

Next section explains the basic building blocks of the protocol.

### 6.3 Building Blocks of the architecture

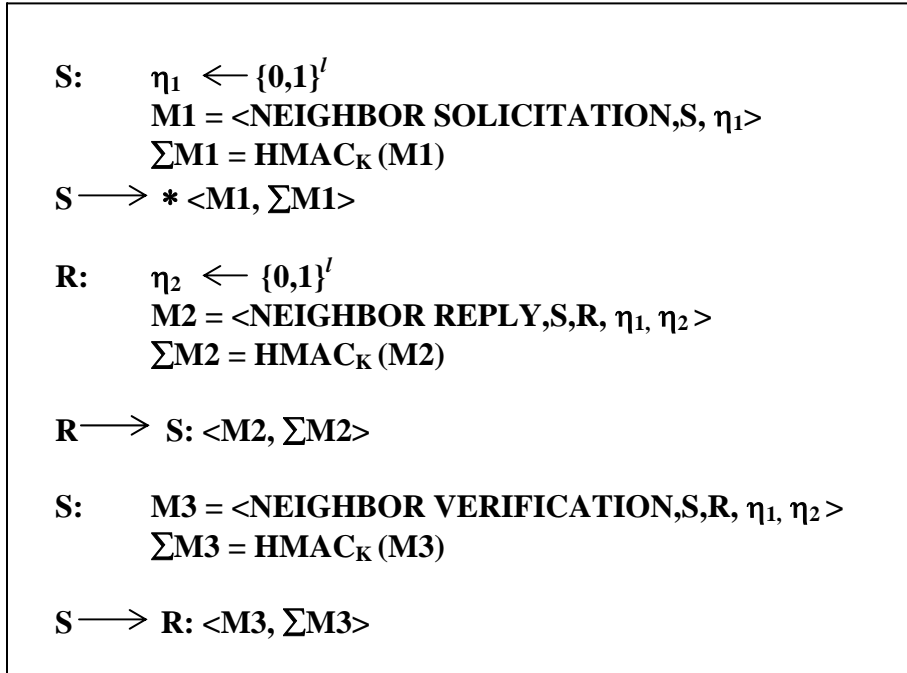
#### 6.3.1 Secure Neighbor Detection

We have borrowed the concept of secure neighbor detection of [23] with minor modification.



**Figure 6.5** Maximum distance between nodes

This protocol allows both the initiator and target to verify that both are within their maximum transmission range by tight time delay mechanism. Here ignore negligible MAC protocol delay. Basically it is a simple three-round mutual authentication protocol. In the first round, the initiator sends a Neighbor Solicitation packet by unicasting to a specific node or by broadcasting. After receiving the packet the target sends the reply by a Neighbor Solicitation packet. In final round the initiator sends a Neighbor Verification, which includes broadcast authentication of a timestamp and the link from the source to the destination. This is shown in the figure 6.6. In order to detect multiple neighbors the initiator must perform separate detection process for each of them.



**Figure 6.6** Three-way handshake for neighbor detection [23]

To prevent replay attack and to ensure the freshness of reply message the protocol uses nonces  $\eta_1$  and  $\eta_2$ . The initiator randomly selects  $\eta_1$  (with sufficiently enough length that makes it harder to guess). The initiator can verify the freshness of reply message by comparing  $\eta_1$  with contained in the reply message (M2). In the same way the target can also verify the freshness of reply message using  $\eta_2$ . The initiator records the sending time of M1 at  $t_0$ , and also records the receiving time of message M2 at  $t_1$ . Now it can find the total delay between these two subsequent message by  $\delta = t_1 - t_0$ . The distance between them (with respect to initiator) is bounded by:

$$d \leq \delta/2 \times c , \text{ where } c \text{ is the speed of light in vacuum.}$$

Thus the initiator can check that the other party is within its maximum transmission range.

It is worth mentioning that the process of secure neighbor detection is performed off line, i.e. when the node does not have packets to process. And it must be done periodically as there is every possibility of frequent node migration or transfer.



### 6.3.2 Trust Factor Assignment

Every nod’s trust factor is measured by its neighbor. We use the following notation:

$$T[A,B,t] = x$$

It implies that the trust factor of node A at time t is x which is measured by node B. Therefore it is a relative measure as  $T[A,B,t] \neq T[B,A,t]$  is not a necessary condition.

Here we have extended the concept of watchdog and pathrater [8] which has been discussed in chapter 2. In [8] the authors devised an algorithm to detect misbehaving node only. After certain threshold the node is declared as ‘malicious’. The scheme works well under the assumption that every packet (data and control) will be unencrypted. Furthermore, it only attempts to detect misbehaving nodes, and not concern about the enhancement of node status (from poorly trusted to highly trusted and vice-versa).

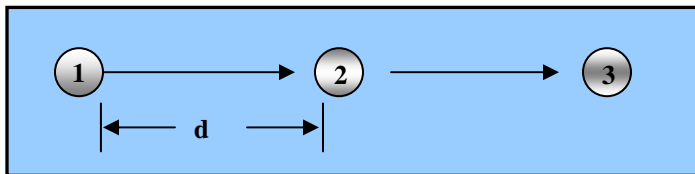


Figure 6.7 Packet Monitoring

Each node maintains a local database with the following format (with possible values):

Target Node	Packet ID <IP, BroadcastID>	Forwarded (Y/N)	Unaltered (Y/N)
X	X,102	1	0

Figure 6.8 Format of the local database of each initiator node

In the above figure node 1 sends a RREQ packet to node 2. Node 1 can easily overhear the packet to verify the fact that node 2 also forwards the packet to node 3. This works fine as long as there is no encryption used. When encryption is used we follow the following time-delay mechanism to detect any alteration of sent packet.

#### 6.3.2.1 Time Delay Mechanism

**Assumption:** We assume that there is an efficient mechanism for neighbor clock synchronization. In ad hoc network ensuring global clock is often very hard to ensure, but it is feasible to maintain clocking between two neighbors.

**Working Principal:** In this case the hop\_count and destination sequence number fields are encrypted by a shared key between the neighboring nodes. The intermediate receiving

node (node 2 in the figure) needs only to decrypt to hop\_count value and increments it and again encrypts with the next hop shared key. So, it is easy to measure the upper bound of internal node processing delay  $\delta_{imax}$ . Again from the previous section it is easy to show that the time to traverse the path (d in the figure) from sending node to the intermediate node (in the figure from node 1 to node 2) can be estimated. Let it be  $\delta_{traverse}$ . When a node sends RREQ packet to its neighbor it records the sending time  $t_0$  and waits to overhear from the next neighbor. As soon as the neighbor sends the packet to the next neighbor (at node 3) the initiator (node 1) records the time  $t_1$ . Now it can be shown that:

$$\text{Total delay } \delta = t_1 - t_0 = 2 \times \delta_{traverse} + \delta_{imax}$$

Both the terms of left side of the equation are well estimated.

$$\text{Now if } t_1 - t_0 > 2 \times \delta_{traverse} + \delta_{imax}$$

It implies that the intermediate node experiences much time than expected due to unexpected modification of the hop\_count / destination sequence number fields in the RREQ packet.

### 6.3.2.2 Node Status Update Mechanism

For determining the node status the local database as shown in figure 6.8 is used as follows.

**Degrade Mechanism:** To reduce the value of trust factor a predefined time period  $t_{mal}$  is set up. After the expiration of this period all entry of the local table is deleted. Also a threshold value of  $P_{mal}$  is predefined. This value is the determination parameter to detect a node as malicious. The main objective is to count the successful forwards by the target node. It can be easily computed by simply performing logical AND operation of the last two fields of the figure 6.9. Then summing up the total number of 1s generates the desired successful packet forwards. An example can explain the procedure. For instance, suppose we get the following snapshot for a period of  $t_{ml}$

Target Node	Packet ID <IP, BroadcastID>	Forwarded (Y/N)	Unaltered (Y/N)
X	X, 101	0	0
X	X,102	1	0
X	X,103	1	1
X	X,104	1	1
X	X,105	1	1
X	X,106	1	1

**Table 6.9** Local database with assumed values

Now by AND operation of the last two fields the resultant **transfer string** becomes **001111**

So, the number of successful packet forwards  $P_{\text{success1}} = 4$  and the success factor ratio is  $SFR1 = P_{\text{success1}}/P_{\text{total}} = 4/6 = 66.67\%$ . Now if the SRF goes below the value  $P_{\text{ml}}$  the target node is detected as malicious. As stated earlier each node maintains a local database of its neighbors with corresponding trust factor as mentioned in figure 6.2. So, the initiator updates the local database setting the trust factor of target node  $-1$ .

It is to be noted that the use of SFR1 instead of absolute number of successful packet forwards  $P_{\text{success}}$  has been carefully chosen. The rationale is that a node may be highly congested with outside packets. On the other hand another node may be idle for most of the period. Absolute numeric measurement will give a wrong direction of reasoning.

**Upgrade Mechanism:** Upgrade of trust factor mechanism is slightly different from the degrade mechanism. But it uses the same algorithm for building the **transfer string** as explained in the previous paragraph. In degrade mechanism it also predefines two values,  $t_{\text{trust}}$  and  $P_{\text{trust}}$ . It is to be mentioned that the values of  $t_{\text{trust}}$  and  $P_{\text{trust}}$  are not necessarily equal to values of  $t_{\text{mal}}$  and  $P_{\text{mal}}$  respectively. The computation of the value  $P_{\text{success2}}$  is performed by summing up the number of consecutive 1s from the LSB (Least Significant Bit). The SRF2 computation is similar to SRF1. If the SRF2 exceeds  $P_{\text{trust}}$  the trust factor of the node is incremented by 1.

### 6.3.3 Workload of Node

To monitor network traffic pattern we use two statistical measures

- i. Packet count over a certain period (PC)
- ii. Standard deviation of inter-packet intervals (SD)

Now traffic pattern can be characterized by vector  $\langle PC, SD \rangle$ . To enhance the traffic characterization following parameters can be used:

- Packet type (RREQ, RREP, REER, HELLO)
- Flow Direction (received, sent, forwarded)

The whole process is performed by node itself and it attaches (appends) Work Load (WL) field of the RREQ packet.

### 6.3.4 Clustered PKI and hop-to-hop Encryption

The protocol uses end-to-end encryption of the payload data and hop-to-hop encryption for routing information. For end-to-end encryption we employ symmetric encryption for efficiency. But the problem lies with the distribution of **globally shared secret key**. Because global secret key in ad hoc network poses severe risk as node mobility and dynamic participation are very common. It has the vulnerabilities

- Single point of failure

- The overhead of periodically change of secret key for node departure from the network

Hence, our proposal goes for **clustered PKI**. The basic concept has been borrowed from [24]. Although the details of this key exchange phase are beyond the scope of our research, here a very general outline is given in the following section.

### 6.3.4.1 Clustered PKI

The generic architecture of clustered PKI can be demonstrated as shown in figure 6.4.

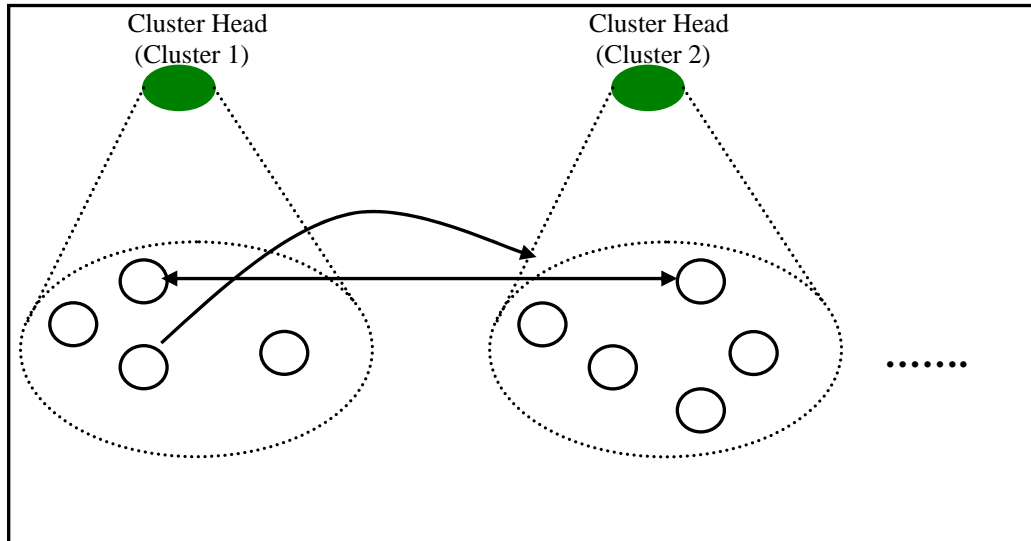


Figure 6.10 Clustered PKI for MANET

This approach is useful for a large size network (in terms of number of total node). Related nodes (geographically) are grouped into different clusters as shown in figure 6.4. In each group one node is selected as **cluster head** which works as the authority of distributing public key of the nodes within its cluster. Besides each cluster head performs all other co-ordination functions as explained in next sections.

For well functioning of the whole network in highly dynamic environment following issues must be carefully considered:

#### i. Intra Cluster Communication

To communicate within the nodes of the same cluster no additional steps are required. Before starting the actual data transfer both the source and the destination ask for their session key. After they are given the session key data transfer begins with symmetric encryption using the session key. Encryption is done over the payload data only.

**ii. Inter Cluster Communication**

In order to communicate with another node within another cluster (as shown in figure 6.4 with bi-directional arrow) the source node first communicates with its own cluster head. The cluster head then communicate with the foreign cluster and securely exchange the session-oriented shared secret. Any party can generate the key.

**iii. Node Migration Across Clusters**

Because of the high dynamic nature of the nodes it is often required for a node to migrate to another cluster (shown in figure 6.4 with uni-directional arrow). Before moving to a new cluster (in figure cluster 2) the node issues an RERR error to the active nodes indicating that the node is no longer valid.

**6.3.4.2 Hop-to-hop Encryption**

From the previous chapter it is evident that most of the attacks on routing protocol are due to absence of encryption for some classified fields in the routing packets, for example hop count and destination sequence number. By unauthorized modification of such fields could cause serious security threats.

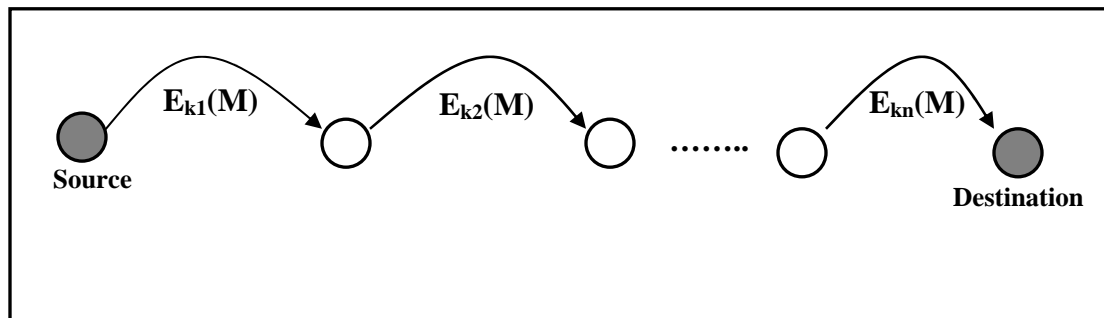


Figure 6.11 Hop-to-hop encryption

Hop-to-hop encryption is done with a shared key between two neighbors. It is quite feasible to manage a session-oriented shared key of neighboring nodes. We use DES for encryption mechanism.

**6.4 Modified Routing Protocol**

Now the modified routing algorithm works as follows.

It is to be noted that the modification requires additional routing fields in both RREQ and RREP packets.

In RREQ field a 3-bit **mode selector** is added. Although a 2-bit field is capable to handle our current configuration (i.e. 4 modes), but for future extensibility a 3-bit field is selected.

In RREP two additional fields are needed: **Trust level (TL)** and **Work Load (WL)**

### 6.4.1 Operational Mode

The modified routing protocol is highly flexible. It is dynamically adjusted with the security requirement of the application. So, before the routing begins, the initiator must carefully select the security requirement of the next session or simply for the application.

In general the protocol consists of four operational modes:

□ **Mode 0: No Encryption**

In this mode the protocol works as a simple AODV protocol. It does not require any additional step. This mode is applicable when the application does not require any security functionality. Additional fields of RREP message are ignored in this mode.

□ **Mode 1: Trusted Path Only**

When the application security requirement is set such that it demands the conventional trusted path based on the past history. Only **TL** field is updated accordingly.

□ **Mode 2: Bandwidth-efficient Path Only**

Unlike other secure routing protocol it truly emphasizes on every dimension of security (CIA) [15]. So, it also permits to use bandwidth-efficient nodes (i.e. less congested nodes). Only **WL** field is updated in this mode.

□ **Mode 3: Both Trusted and Bandwidth-efficient Path**

In this mode an optimal combination between the best possibilities of mode 1 and mode 2 is selected. It assures very high degree of security at the cost of processing over-head. Both **WL** and **TL** fields are processed in this combined mode.

One thing should be noted here that in mode 1,2,3 the hop-to-hop routing encryption and end-to-end data encryption are made by default. The protocol has the flexibility to set off any or both of them in any mode.

**Security of TL and WL strings:** The purposes of introducing TL and WL fields in RREP reply packet are to enhance security. But an attacker can modify them to create harm on the entire protocol. So, hop-by-hop encryption also covers the fields WL and TL.

### 6.4.2 Route Discovery

The first half of the route discovery protocol is similar to original AODV protocol. A node wishing to communicate with destination first broadcasts RREQ packet to its neighbors, neighbor can reply RREQ if it has the route information to the destination with greater or equal sequence number. Else it rebroadcasts the RREQ to its neighbors. The process continues until it reaches the destination. The destination replies by

unicasting a RREQ packet to the neighbor from which it got the RREQ packet. The neighbor also repeats the process. An additional step is performed here. The receiving node attaches the trust level of its neighbor (towards destination) and/or attaches the workload factor of itself depending on the application security requirement. The process repeats to all intermediate nodes and the initiator/source. The source thus gets a trust level string termed as **TL string** and / or a workload string called **WL string**. Now it is quite simple to adjust these two strings to reflect correspondence.

### 6.4.3 Route Selection

The process of route discovery is a function of the security requirement of application. But once it has been decided, it is trivial to get multiple paths from different nodes. Because the actual data transfer begins after a pre-defined amount of time. During this period some other routes are likely to be discovered. Obviously the best route is selected. Path selection criteria must be established in advance. The selection must include the following considerations:

- ❑ The number of hops to be tolerated for higher bandwidth or trust level.
- ❑ The minimum threshold for TL and WL i.e. below this value route will be discarded
- ❑ Average weight of the trust level or workload metrics.

The following example can explain the operation of the above protocol clearly.

**An Example:** The above protocol can be best explained with the help of a simple example. Suppose that, a network is consisting of 16 nodes labeled source, destination and from alphabet A to O as shown in figure 6.8.

The source wishes to communicate with the destination. Prior to the start of route discovery the source selects the security level of the application such that it matches with the mode 1 operation of the protocol. Also assume that the selection criteria include one hop to be tolerated for better trust level.

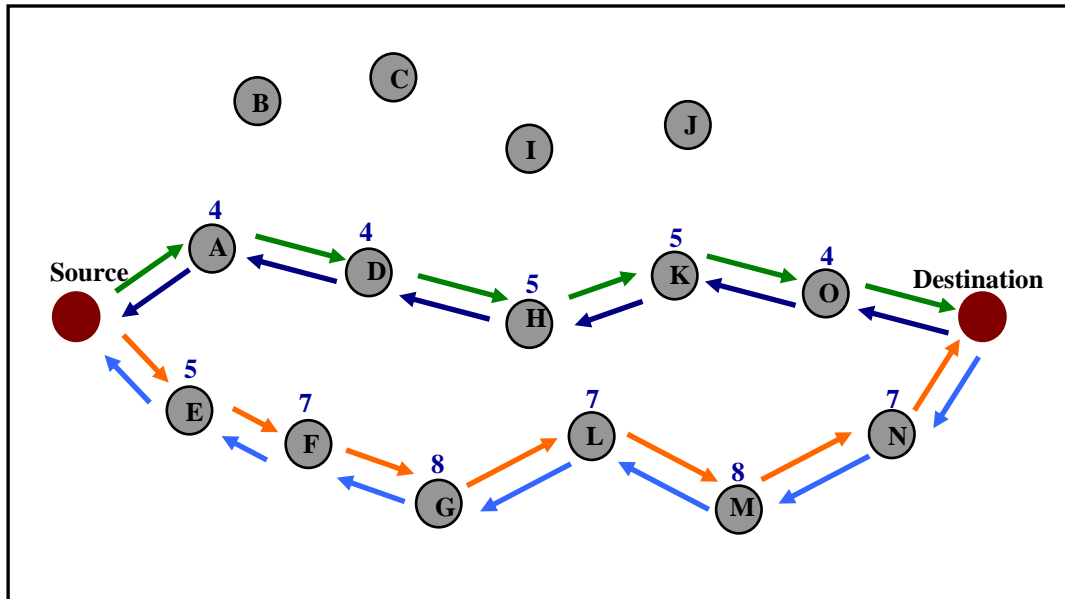


Figure 6.12 Modified routing protocol

The numeric numbers shown closer to each node indicate their corresponding trust level. The protocol begins with the broadcast packet of RREQ to the neighbors of source: A and E. They in turn broadcasts the packet on their neighbors. For our discussion only the relevant packet forwards are shown. As a result node A forwards to D, D forwards to H, H sends to K, K forwards to O and finally O forwards to the destination. The destination sends back the reply RREP by unicasting to O. Node O sends the RREP packet to K. Node K receives the RREP packet and sends to H. But before sending it to H it attaches the trust level of the node from which it just got the RREP packet i.e. node O. So the additional field TL now contains the value 4. Node H repeats the process and appends the trust level of its neighbor from which it just got the RREP i.e. node K. So the TL string now contains the value **45**. The process continues until it reaches the source node. So the source node finally got the TL string **45544**.

Now the source must wait for a pre-defined period of time to select the best route. In the meantime another RREP packet the source got from another neighbor E. In this case the source gets a different TL string **787875**.

Although the route through **A D H K O** contains 5 hops and another route (**E F G L M N**) contains 6 hops. But the application requires trusted path at the cost of maximum one more hop. Furthermore, average trust weight of **ADHKO** and **EFGLMN** are **4.4** and **7** respectively. Hence the route **EFGLMN** is selected.



It is to be noted that the suitable route selection may not be so obvious in most cases as in this example. Especially when the protocol operates in combined mode i.e. mode 3. Therefore, suitable algorithm must be devised to handle such critical selection criteria.

## Chapter 7

### Discussion of Result and Future Work

---

---

#### 7.1 Results

Securing routing protocol in ad hoc network is a daunting task. In this work the known vulnerabilities of existing routing protocol have been extensively studied. The solution has been carried out based on AODV protocol, although it is well suited for any standard routing protocol in MANETs. The following conclusions are drawn from this research:

- ❑ It prevents most of the denial-of-service attack by hop-by-hop encryption of the routing information. In order to reduce overhead only the sensitive fields of the routing packets (i.e. hop count and sequence number) are encrypted. Besides we employ symmetric encryption such as DES for better performance in light-weight mobile devices commonly found in MANETs.
- ❑ One of the most severe attacks on MANETs is wormhole attack. The major cause of this attack is the absence of any neighbor detection mechanism. The presented solution also counters this attack by an efficient secure neighbor detection mechanism.
- ❑ The use of two different metrics (trust level and workload) for routing selection is a probabilistic approach to enhance security of the discovered path.
- ❑ To remove a node from a route it uses the mechanism to detect malicious node which does not depend on global clock synchronization but on its local timing only.
- ❑ In order to prevent replay attacks it employs session-key for data transfer. Even if it is stolen or hijacked its consequence is limited to only the concern session as it expires after a certain period.

#### 7.2 Direction for Future Works

Due to shortage of time a complete simulation by any suitable simulator such as ns2 has not been possible. So it will be good direction for future work. Here the solution has been presented as protocol specific. It requires some explicit moderation to make it a general framework. We keep it as a future work. Apart from this, the key management of ad hoc network is an important issue. It is also kept as a future research direction.

## References

- [1] Charles E. Perkins. Ad hoc Networking and AODV. Nokia Research Center Mountain View, CA USA, 2000
- [2] L. Zhou and Z. J. Hass. Securing ad hoc networks. *IEEE Network*, 13(6):24-30,1999
- [3] M. Weiser. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM* 36(7), July 1993
- [4] Yih-Chun Hu and A. Perrig. A Survey of Secure Wireless Ad Hoc Routing. *IEEE Security and Privacy*, May/June 2004
- [5] Y.C. Hu, D.B. Hohnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks. *Proceedings of 4<sup>th</sup> IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 02)*, IEEE Press, 2002, pp.3-13
- [6] Y.-C. Hu, A. Perrig, and D.B. Johnson. Packet Leashes:A Defense against Wormhole Attacks in Wireless Ad Hoc Networks. *Proc. 22nd Ann. Joint Conf. IEEE Computer and Communications Societies (INFOCOM 2003)*, IEEE Press, 2003, pp. 1976–1986.

- [7] Y.-C. Hu, A. Perrig, and D.B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. *Proc. 8th Ann. Int'l Conf. Mobile Computing and Networking (MobiCom 2002)*, ACM Press, 2002, pp.12–23.
  
- [8] S. Marti et al. Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks. *Proc. 6th Ann. Int'l Conf. Mobile Computing and Networking (MobiCom 2000)*, ACM Press, 2000, pp. 255–265.
  
- [9] M. Guerrero Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. *Proc. ACM Workshop on Wireless Security (WiSe)*, ACM Press, 2002, pp. 1–10.
  
- [10] P. Papadimitratos and Z.J. Haas. Secure Link State Routing for Mobile Ad Hoc Networks. *Proc. IEEE Workshop on Security and Assurance in Ad Hoc Networks*, IEEE Press, 2003, pp. 27–31.
  
- [11] S. Buchegger and J.-Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes - Fairness In Dynamic Ad-hoc NeTworks). *Proc. 3rd Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 2002)*, ACM Press, 2002, pp. 226–236.
  
- [12] B. Awerbuch et al. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. *ACM Workshop on Wireless Security (WiSe)*, ACM Press, 2002, pp. 21–30.
  
- [13] J. Zhen and Sampalli Srinivas. Preventing Replay Attacks for Secure Routing in Ad Hoc Network. *ADHOC-NOW 2003, LNCS 2865*, pp. 140-150, 2003

- [14] A. Menezes, P.C. van Oorschot, S. A. Vanstone. Handbook of Applied Cryptography. 1997, CRC Press LLC, Florida 33431
- [15] Charles P. Pfleeger, Shari L. Pfleeger. Security in Computing. Third Edition, 2003. Pearson Education (Singapore) Pte. Ltd.
- [16] B. Schneier. Applied Cryptography: Protocols, Algorithms and Source code in C. John Wiley & Sons, Inc. New York, 1996
- [17] Tutorial on Cryptographic Primitives. Available at:  
[http://www.opengroup.org/messaging/G260/pki\\_tutorial.htm](http://www.opengroup.org/messaging/G260/pki_tutorial.htm)
- [18] Tutorial on PKI. Available at  
[http://www.opengroup.org/messaging/G260/pki\\_tutorial.htm](http://www.opengroup.org/messaging/G260/pki_tutorial.htm)
- [19] Charles E. Perkins. AD Hoc Networking. Addison-Wesley. 2001
- [20] L.R. Ford Jr. and D.R. Fulkerson, Flows in Network, Princeton University Press, 1962
- [21] E.M. Belding-Royer and C.K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*, pages 46-55, April 1999
- [22] K. Sanzgiri, B.Dahill, B.N.Levine, C.Shields and E. M.Belding-Royer. A Secure Routing Protocol for Ad Hoc Networks. *Proceedings of 10<sup>th</sup> IEEE International Conference on Network Protocols (ICNP'02)* 2002

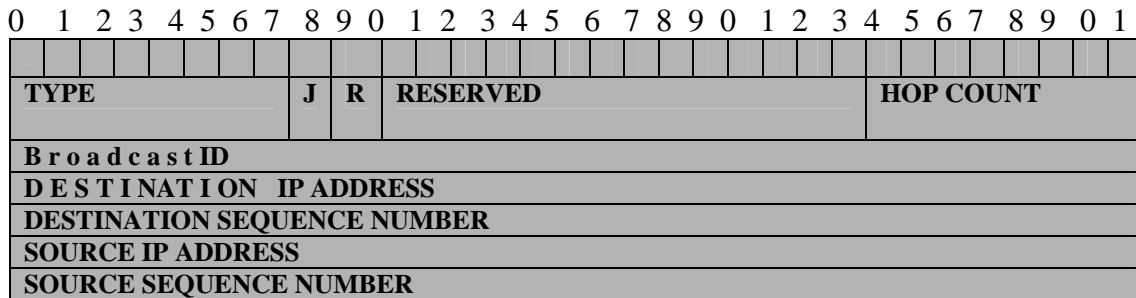
- [23] Y.C.Hu, A. Perrig and David B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols. *ACM Conference on Wireless Security (WiSe)* September 2003.
  
- [24] C.Zouridaki, B.L.Mark, K.Gaj and R.K.Thomas. Distributed CA-based PKI for Mobile Ad Hoc Networks Using Elliptic Curve Cryptography. EuroPKI 2004, LNCS 3093, pp. 232-245, 2004

## Appendix

### AODV Data types

#### 1. RREQ (Route Request)

Route Request (RREQ) message format is shown in the following figure.



**Figure a** RREQ message format

The fields are explained as follows:

**TYPE**      the value is set to 1

**J**    Join flag, used to join in a multicast group

**R**    Repair flag, set when a node initiates to repair two previously disconnected portion of the multicast tree

**RESERVED**      For future use, now set 0, receiver ignores it

**HOP COUNT**      Number of hops from the source node to the intermediate node (or the destination) handling the packet

**BROADCAST ID**      A monotonically increasing counter that uniquely identifies a RREQ packet when combined with the IP address of the source

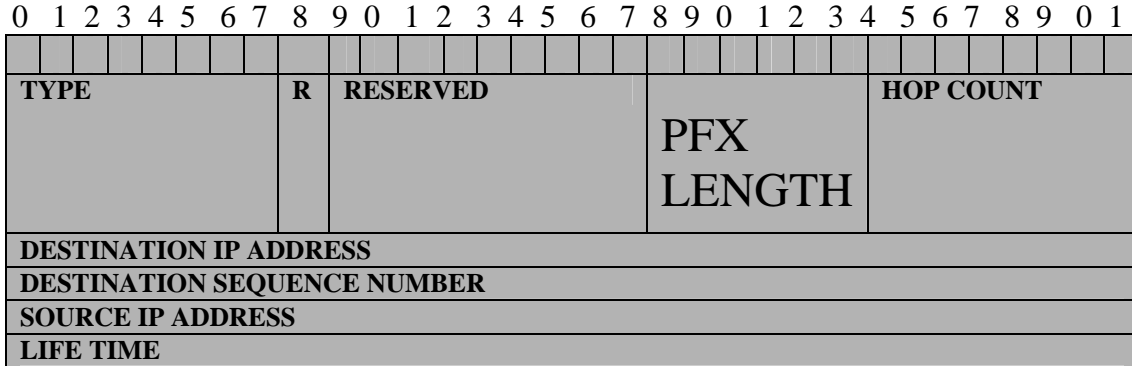
**DESTINATION IP ADDRESS**      IP address of the destination

**DESTINATION SEQUENCE NUMBER**      The latest sequence number known by the source for any route towards the destination

**SOURCE IP ADDRESS**      IP address of the source node

**SOURCE SEQUENCE NUMBER** The present sequence number to be used for route entries pointing to the source of the request

**2. RREP (Route Reply)**



**Figure b** RREP message format

Fields:

**TYPE** 2

**R** Repair flag as in RREQ header

**RESERVED** same as in RREQ header

**PREFIX SIZE** If nonzero, it indicates that the next hop may be used for any nodes with the same routing prefix as that of the requested destination

**HOP COUNT** The number of hops from the source IP address to the destination IP address. For multicast route requests it indicates the number of hops to the multicast tree member sending the request

**DESTINATION IP ADDRESS** IP address of the destination

**DESTINATION SEQUENCE NUMBER** The destination sequence number associated with the route

**SOURCE IP ADDRESS** IP address of the source node

**LIFETIME** The time for which nodes receiving the RREP consider the route to be valid



### 3. RERR (Route Error)

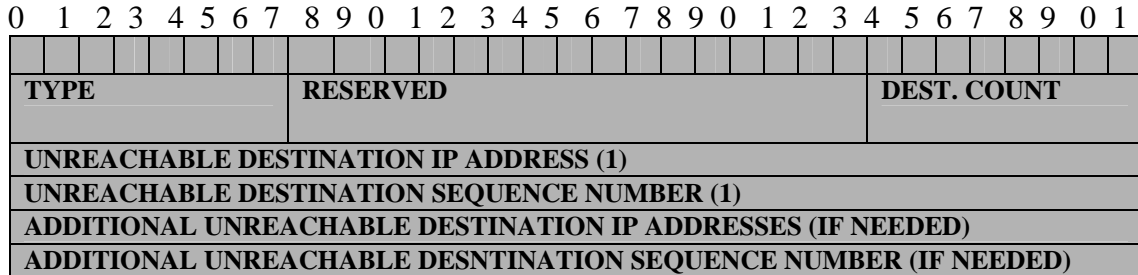


Figure c RERR message format

Fields:

**TYPE 3**

**RESERVED** 0, ignored by receiver

**DESTINATION COUNT** The number of unreachable destinations included in the message, it must be set at least 1

**UNREACHABLE DEST. IP ADDRESS** The IP address of the destination that has become unreachable due to link failure

**UNREACHABLE DEST. SEQUENCE NUMBER** The last known sequence number (incremented by 1) of the destination listed in the previous field (UNREACHABLE DEST. IP ADDRESS)