SECURED VECTOR ROUTING PROTOCOL FOR MANET'S IN PRESENCE OF MALICIOUS NODES

A THESIS SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF

Master of Technology
in
Computer Science and Engineering
(Specialization: Computer Science)

By

P. LAKSHMIRAMANA

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Orissa-769 008, India
May 2009

SECURED VECTOR ROUTING PROTOCOL FOR MANET'S IN PRESENCE OF MALICIOUS NODES

A THESIS SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF

Master of Technology

in

Computer Science and Engineering
(Specialization: Computer Science)

By

P. LAKSHMIRAMANA

Roll No-207CS109

Under the Supervision of

Prof. Pabitra Mohan Khilar

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Orissa-769 008, India
May 2009

Dedicated to my Parents

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

Certificate

This is to certify that the work in the thesis entitled **"Secured VRP for MANETs in presence of Malicious Nodes"** submitted by **Mr. P. LakshmiRamana** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere

Prof. Pabitra Mohan Khilar
Professor
Dept. of Computer Science & Engineering
National Institute of Technology
Rourkela-769008 Orissa (India)

Place: NIT Rourkela
Date: 26 May 2009

ACKNOWLEDGEMENT

I would like to express my gratitude to my advisor, Professor Pabitra Mohan Khilar for his invaluable support and guidance.

I thank Professor Banshidhar Majhi, Head of Computer Science & Engineering Department, for providing excellent study environment.

I thank Faculty of Computer Science & Engineering Dept., for cooperation in completion of this Thesis.

I also thank PhD Scholars Pushpendra Chandra, Niranjan Ray and classmates who motivated me towards my Thesis work.

Finally I would like to thank my parents, brother and sister for their constant love, understanding and encouragement. My parents have been a source of inspiration and motivation for me. I am very grateful to them for their sacrifices and efforts that made this Thesis possible.

26th May 2009 P. LAKSHMIRAMANA

LIST OF DISSEMINATION

[1] P. LakshmiRamana and Pabitra Mohan Khilar, " Secured VRP for MANETs in presence of Malicious nodes", in Proceedings of National Conference on Computing Techniques and Systems (NCCTS 09), 24th April, Muthayammal Enginnering College, Rasipuram, Tamil Nadu. Pg 220-222.

CONTENTS

ABSTRACT

Due to their inherent properties Mobile Ad hoc Networks (MANETs) are finding their way in diversity of applications. As these are infrastructureless networks, normal traditional routing algorithms are not applicable to MANETs. Many routing algorithms have been proposed for MANETs till now. Vector Routing Protocol (VRP) is proposed for efficient utilization of limited bandwidth in MANETs.

Providing security to the network layer in MANETs is gaining importance nowadays. The two main functions of network layer are *ad hoc routing* and *data forwarding*. Due to the inherent properties of MANETs there is huge scope to attack on functions of network layer. Attacks on network layer can be broadly classified in two categories namely *routing attacks* and *packet forwarding* attacks. The attacks which try to modify the routing data and mislead the routing protocols are treated as routing attacks. These attacks are specific to the routing protocol used by the MANETs. Packet forwarding attacks on other hand do not disrupt the routing protocol, instead they cause the data packets intentionally inconsistent with routing states. Examples of these attacks are an intermediate node on the path, dropping the data packet, modifying the contents of the packet etc.

In this Thesis we consider only on packet dropping attack on VRP algorithm. We proposed a solution to mitigate the effects of malicious nodes on VRP. Our proposed solution 'Secured VRP' secures the VRP algorithm from malicious nodes. Our Secured VRP (SVRP) works far better than VRP in presence of malicious nodes and works exactly as VRP in absence of malicious nodes.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Chapter 1 INTRODUCTION

1.1 INTRODUCTION

A Mobile Ad-hoc Network (MANET) is self configuring network formed by Mobile hosts. The hosts are connected by wireless links. It is one form of Wireless network. Due to arbitrary topologies and mobility of nodes in the network, these networks are called Mobile Ad-hoc Networks.

MANETs have striking differences with Cellular Networks (another kind wireless network). The basic difference is Cellular networks have pre constructed infrastructure made of fixed and wired nodes. These fixed and wired nodes are called as base stations. The base stations act as access points, and communication between two nodes completely rely on wired backbone and fixed base stations. In a MANET no structure exists, hence these are called Infrastructureless networks.

In MANETs a network is formed dynamically through the cooperation of an arbitrary set of independent nodes. There is no prearrangement regarding the specific role each node should assume. Instead, each node makes its decision independently, based on the network situation, without using a preexisting network infrastructure.

As per IETF MANET working group the following are salient characteristics of MANETs:

1. Dynamic Topologies

   All nodes in the network are free to move arbitrarily making the network topology to change randomly and rapidly at unpredictable times. The topology may consist of both bidirectional and unidirectional links.

2. Bandwidth constrained, Variable capacity links

   Wireless links will continue to have significantly lower capacity when compared to traditional hardwired links. The realized throughput of wireless communications after accounting for the effects of multiple access, fading, noise and interface conditions, is often much less than a radio's maximum transmission rate.

3. Energy constrained operation

   Some or all of the nodes in a MANET may relay on batteries or other exhaustible means for their energy. For these nodes, the most important system design criteria for optimization may be energy conservation.

4. Limited Physical Security

Mobile wireless networks are generally more prone physical security attacks than are fixed ones. Existing link security techniques are often applied to reduce security threats.

Due to the above properties, normal traditional algorithms developed for wired networks are not applicable to MANET's. All routing algorithms should consider all the above properties.

1.2 MOTIVATION

As mentioned in previous section routing algorithms should consider all the properties. Many proposed algorithms tried to concentrate on few properties only. For this reason only we are having different categories of algorithms like Bandwidth efficient algorithms, Energy efficient algorithms and Secured Routing algorithms.

In our study we came across Vector Routing Protocol (VRP). This algorithm is proven to be bandwidth efficient algorithm. Like other algorithms success of VRP depends on cooperative nodes. This property (believing other nodes) makes VRP vulnerable to security attacks. The concept of malicious nodes is discussed by Marti. In this paper Marti studied the effect of malicious nodes on DSR algorithm and proposed a solution to mitigate the effects of malicious nodes. This work motivated us to study the effects of malicious nodes on VRP algorithm.

1.3 THESIS OUTLINE

The organization of rest of the thesis is as follows. In Chapter 2, we discuss some Basic Cryptography preliminaries. In Chapter 3, we have described some routing algorithms like DSDV, AODV, DSR and VRP. In Chapter 4, we discussed two related exiting works which paved us way for our proposal. Chapter 5 discusses our solution and simulation results. We have concluded our thesis with future scopes followed by a number of references.

CHAPTER 2

CRYPTOGRAPHY PRELIMINARIES

Chapter 2 CRYPTOGRAPHY PRELIMINARIES

2.1 INTRODUCTION

Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient.

While cryptography is the science of securing data, *cryptanalysis* is the science of analyzing and breaking secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Although in the past *cryptography* referred only to the *encryption* and *decryption* of messages using secret keys, today it is defined as involving four distinct mechanisms: Symmetric and asymmetric Encryption, Hashing, Digital signatures and Public Key Infrastructure. The following sections explain these mechanisms.

2.2 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 **cipher text**.

2.2.1 SYMMETRIC ENCRYPTION

It is the simplest but very efficient form of encryption. Here one secret is shared between the communicating parties (say 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 figure 2.1 [14]:

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

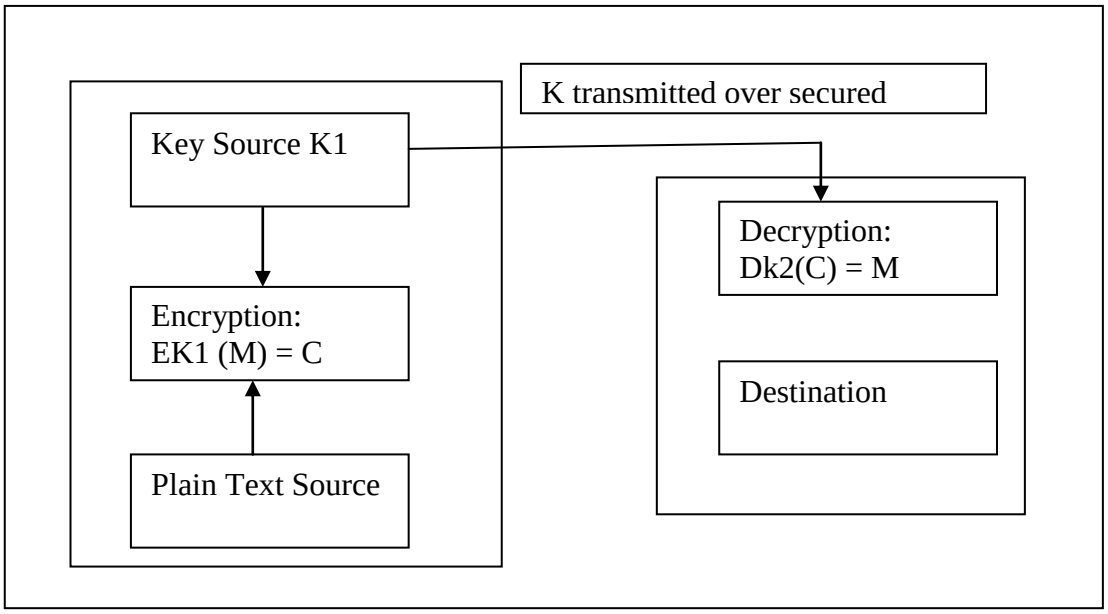

Figure 2.1 Symmetric Encryption (Adopted from [14])

2.2.2 ASYMMETRIC 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 is 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.

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]

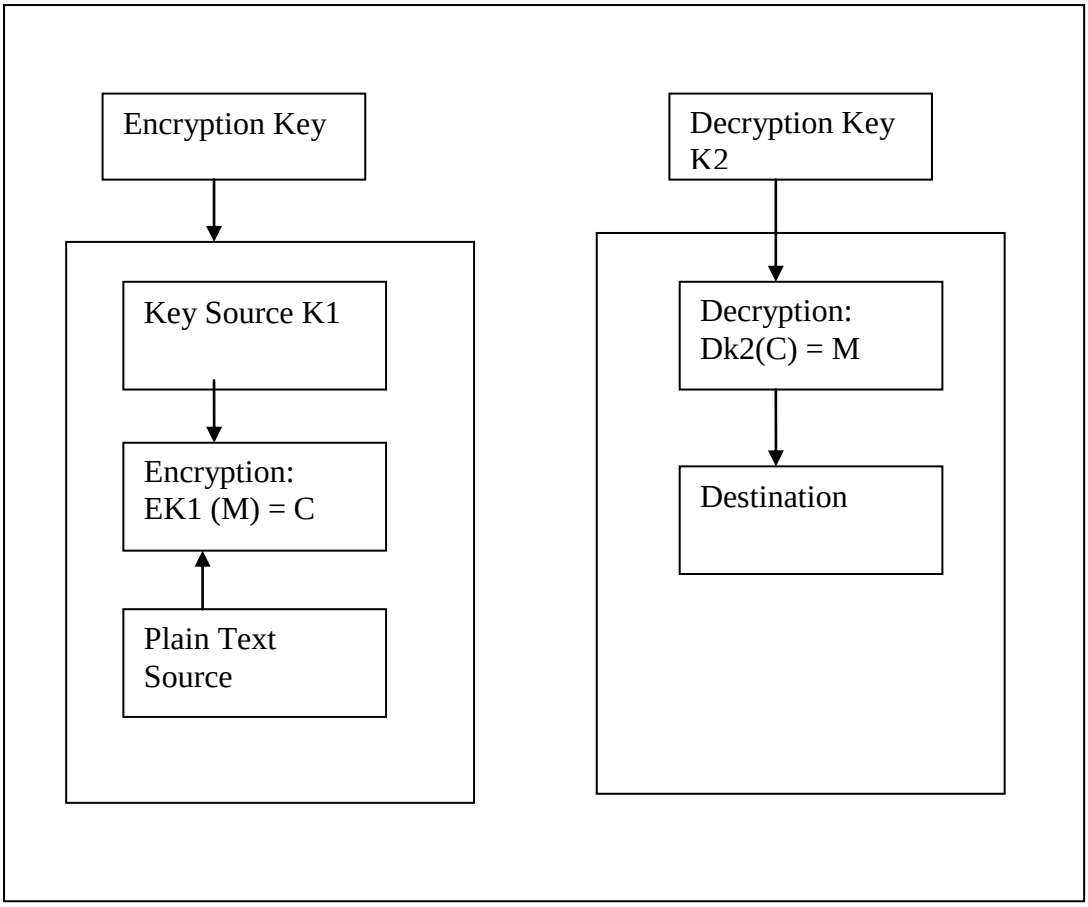

Figure 2.2 Asymmetric Encryption (Adopted form [14])

2.3 CRYPTOGRAPHY HASH FUNCTION

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=x3 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).

2.4 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 2.3. 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 singing 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 2.3.

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

- It must be unforgeable. If an entity sings a document M with signature S(M), it is not possible for other entity to produce the same pair **<M, S(M)>**
- 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.

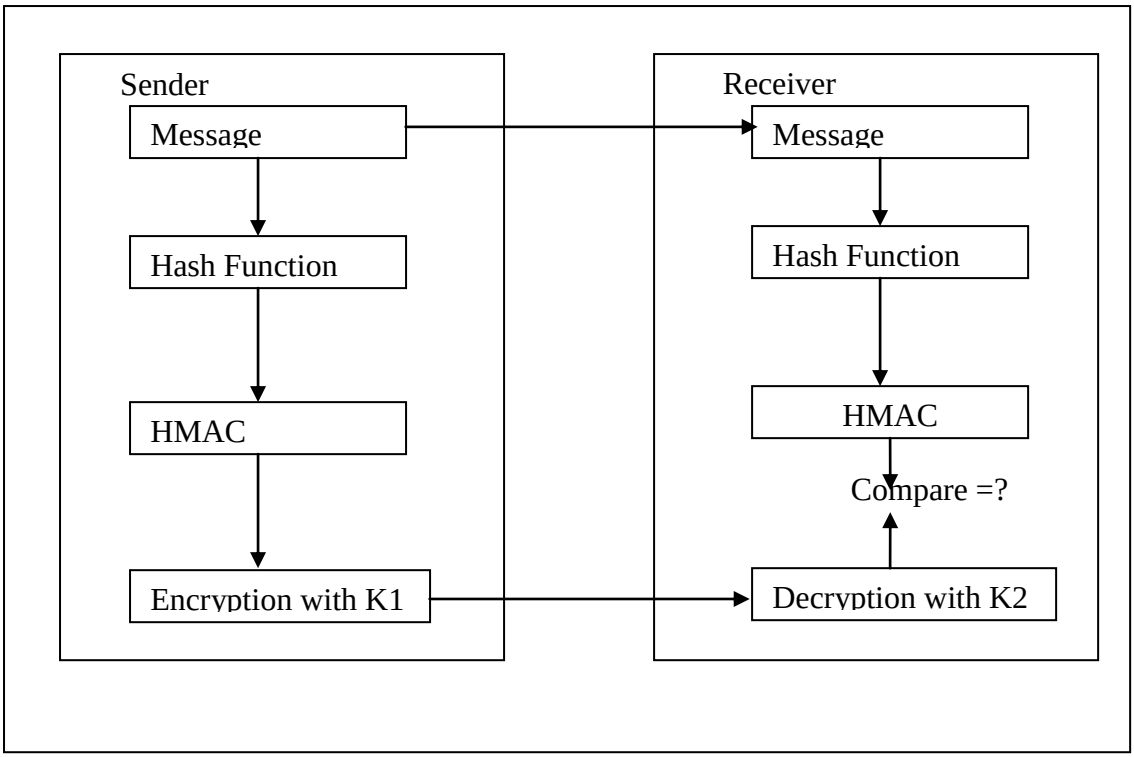

Figure 2.3 Digital Signature

On the reception end the receiver uses the same hash algorithm to compute the hash code of the message and decrypts the encrypted message using the corresponding public key and compares the hash value. The process is illustrated in the figure 2.3

2.5 PUBLIC KEY INFRASTRUCTURE

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.

The primary purpose of PKI is to distribute public key and certificates with security and integrity [17]. 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 3

ROUTING ALGORITHMS

Chapter 3 ROUTING ALGORITHMS

3.1 INTRODUCTION

MANET routing has received huge attention from research community. Due to inherent properties as specified in previous chapter, traditional routing techniques cannot be applied for MANETs. The problem of routing in network has two components: route discovery and route maintenance. Based on route maintenance routing can be broadly classified into two categories namely proactive and reactive algorithms.

In proactive algorithms routes are maintained up-to-date. Topology updates are propagated throughout the network to maintain a consistent view of the network. Keeping routes for all destinations has the advantage that communication with arbitrary destinations experiences minimal initial delay. The disadvantage of these algorithms is in dynamic environment more control packets are involved to maintain consistent topology.

Reactive routing protocols like Dynamic Source Routing (DSR) and Ad Hoc On-demand Distance Vector (AODV) only react when a route is needed between a source and a destination node, and they do not need to try and maintain routes to destinations that they are not communicating with.

3.2 DESTINATION SEQUEENCED DISTANT VECTOR (DSDV) ROUTING

DSDV [2] is a distance vector routing protocol that builds and maintains routing tables at each network node. This routing table contains the next hop for, and the total number of hops to, all reachable destinations. As previously stated, DSDV tries to maintain and keep all routing tables completely updated for all connections at all times. It achieves this by periodically sending out updates to all nodes, a network flood–type situation. DSDV also uses a hop count and sequence number strategy for routes along its network. This is a combination method that shows how fresh and short a route is. For instance, consider a route A. This route will be considered more favorable than another route, say route B, if either A had a higher sequence number, indicating that it is a fresher route, or if they both had the same sequence number, indicating that route A has a lower hop count. Essentially, the use of sequence numbers helps prevent routing loops when broken links occur. Consider a situation where a route is found to be redundant, and it can no longer be reached. As DSDV broadcasts any change within a network, this redundant information will not remain for very long and potentially waste time and bandwidth. A node along the route would detect the redundant route. Information

comprising an infinite hop count would be relayed to the destination node, to enable the destination to increase the sequence number to valid routes by one. Johansson et al[3] investigated this and the following two main leading routing protocols and came to the conclusion that DSDV is not suitable for wireless ad hoc routing environments, due to a number of failings in the testing carried out. DSDV performs poorly in situations with increased mobility, a metric that helps simulate moving nodes. Because of the way that the DSDV protocol works, it struggles in maintaining valid routes to every node, and lost packets will result. Also, as the network in the simulations grew larger, the protocol had substantial difficulties handling the increased network load as a result of increased updates.

3.3 DYNAMIC SOURCE ROUTING

Dynamic source routing [4] is based on source routing, where the source specifies the complete path to the destination in the packet and each node along this path simply forwards the packet to the next hop indicated in the path. It utilizes a route cache where routes it has learned so far are cached. Therefore, a source first checks its route cache to determine the route to the destination. If a route is found, the source uses this route. Otherwise, the source uses a route discovery protocol to discover a route.

In route discovery, the source floods a query packet through the ad hoc network, and the reply is returned by either the destination or another host that can complete the query from its route cache. Each query packet has a unique ID and an initially empty list. On receiving a query packet, if a node has already seen this ID (i.e., duplicate) or it finds its own address already recorded in the list, it discards the copy and stops flooding; otherwise, it appends its own address in the list and broadcasts the query to its neighbors. If a node can complete the query from its route cache, it may send a reply packet to the source without propagating the query packet further. Furthermore, any node participating in route discovery can learn routes from passing data packets and gather this routing information into its route cache.

A route failure can be detected by the link-level protocol (i.e., hop by hop acknowledgments) or it may be inferred when no broadcasts have been received for a while from a former neighbor. When a route failure is detected, the node detecting the failure sends an error packet to the source, which then initiates route discovery protocol

again to discover a new route. In DSR, no periodic control messages are used for route maintenance.

The major advantage of DSR is that there is little or no routing overhead when a single source or a few sources communicate with infrequently accessed destinations. In such a situation, it does not make sense to maintain routes from all sources to such destinations. Furthermore, because communication is assumed to be infrequent, a lot of topological changes may occur without triggering new route discoveries.

Even though DSR is suitable for the environment where only a few sources communicate with infrequently accessed destinations, it may result in large delay and large communication overhead in highly dynamic environments with frequent communication requirements [5]. Moreover, DSR may have a scalability problem [6]. As the network becomes larger, control packets and message packets also become larger because they need to carry addresses for every node in the path. This may be a problem as ad hoc networks have limited available bandwidth.

3.4 AD HOC ON-DEMAND DISTANCE VECTOR (AODV) ROUTING

The Ad Hoc On-demand Distance Vector [7] routing protocol shares DSR's on-demand characteristics in that it also discovers routes on an "as needed" basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, with one entry per destination. This is a departure from DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate a route reply back to the source and, subsequently, to route data packets to the destination. AODV uses sequence numbers maintained at each destination to determine freshness of routing information and to prevent routing loops. These sequence numbers are carried by all routing packets.

When a route is needed, a node broadcasts a route request message. The response message is then echoed back once the request message reaches the destination or an intermediate node finds a fresh route to the destination. For each route, a node also maintains a list of those neighbors actively using the route. A link breakage causes immediate link failure notifications to be sent to the affected neighbors. Similar to DSDV, each route table entry is tagged with a destination sequence number to avoid loop

formation. Moreover, nodes are not required to maintain routes that are not active. Thus, wireless resources can be effectively utilized. However, because flooding is used for route search, communication overhead for route search is not scalable for large networks. As route maintenance considers only the link breakage and ignores the link creation, the route may become nonoptimal when network topology changes. Subsequent global route search is needed when the route is broken.

An important feature of AODV is maintenance of timer-based states in each node, regarding utilization of individual routing table entries. A routing table entry expires if not used recently. A set of predecessor nodes is maintained for each routing table entry, indicating the set of neighboring nodes that use that entry to route data packets. These nodes are notified with route error packets when the next hop link breaks. Each predecessor node, in turn, forwards the route error to its own set of predecessors, thus effectively erasing all routes using the broken link.

The specification of AODV in Parkins et al[8], includes an optimization technique to control the route request flood in the route discovery process. It uses an expanding ring search initially to discover routes to an unknown destination. In the expanding ring search, increasingly larger neighborhoods are searched to find the destination. The search is controlled by the TTL field in the IP header of the route request packets. If the route to a previously known destination is needed, the prior hop-wise distance is used to optimize the search.

3.5 VECTOR ROUTING PROTOCOL (VRP)

Vector Routing Protocol [1] is proved to be bandwidth efficient algorithm. It works in two phases namely start up phase and maintenance phase. Each node in the network has three vectors namely neighborhood, access and routing vectors. Neighborhood vector of a node stores the information of its neighbor nodes, while access vector stores accessibility information. Routing vector stores the routing information. In the network of n nodes, each node is given unique identity in the range *0* to *n-1*. The three vectors of a node are of length *n*. The access, neighborhood and routing vectors of node *u* are denoted as follows

$$a^u = (a_0^u, a_1^u, a_2^u, ..., a_{n-1}^u)$$

$$v^u = (v_0^u, v_1^u, v_2^u, ..., v_{n-1}^u)$$

$$r^u = (r_0^u, r_1^u, r_2^u, ..., r_{n-1}^u)$$

The neighborhood vector of node *u* denoted by $\mathbf{v^u}$ has n bits. If node *i* is neighbor of node *u* then the i^{th} bit denoted as $\mathbf{v_i^u}$ is set to 1 else its value is 0. Initially all bits in this vector is set 0, except $\mathbf{v_u^u}$ =1.

The access vector of node *u* denoted by $\mathbf{a^u}$ has n bits. If node *i* is having route from node *u* then the i^{th} bit denoted as $\mathbf{a_i^u}$ is set to 1 else its value is 0. Initially all bits in this vector is set 0, except $\mathbf{a_i^u}$ =1.

The routing vector of node *u* denoted by $\mathbf{r^u}$ has n numbers. If node *i* is having route from node *u* then the i^{th} bit denoted as $\mathbf{r_i^u}$ has the id of the next node along the path to destination node *i*. Initially $\mathbf{r_i^u} = \mathbf{u}$ for all i and u in between 0 and 1.

Routing information is computed in two phases. The first phase called Start Up phase is performed only once. It is performed when the network is activated for the first time. In first stage of this phase, each node broadcasts 'hello' messages to detect its neighbor nodes. When node u receives a 'hello' message from neighboring node *i* it sets $\mathbf{v_i^u}$ =1. After detecting all neighbor nodes, the two vectors $\mathbf{a^u}$ and $\mathbf{r^u}$ are updated using (1) and (2) respectively.

$$a^u = v^u \text{.....(1)}$$

$$r_i^u = \begin{cases} i, & if \quad v_i^u = 1 \\ u & if \quad v_i^u = 0 \end{cases} \text{.....(2)}$$

As a next stage, each node *u* sends its $\mathbf{a^u}$ to every neighboring node *w*, and waits to receive $\mathbf{a^w}$ from them. When *u* receives the vector $\mathbf{a^w}$, it updates the two vectors $\mathbf{a^u}$ and $\mathbf{r^u}$ using (3) and (4). After updating a new stage starts, that is node *u* sends the new $\mathbf{a^u}$ to neighboring node *w* and waits to receive $\mathbf{a^w}$. When node receives the vector $\mathbf{a^w}$, it updates the two vectors $\mathbf{a^u}$ and $\mathbf{r^u}$. This process continues until each bit in vector $\mathbf{a^u}$ is 1 if network is not partitioned or the value of $\mathbf{a^u}$ remains unchanged if network is partitioned.

$$a_i^u = \begin{cases} 1, & if \quad a_i^u = 0 \quad and \quad a_i^w = 1 \\ 0, & if \quad a_i^w = 0 \quad and \quad r_i^u = w \end{cases} \qquad \text{.....(3)}$$

$$r_i^u = \begin{cases} w, & if \quad a_i^u = 0 \quad and \quad a_i^w = 1 \\ u, & if \quad a_i^u = 0 \end{cases} \qquad \text{.....(4)}$$

The following figure 3.1 describes the Start-up phase

Algorithm VRP (Start-up Phase)
// executed by every node u in the network
Initialize vectors: $\mathbf{a^u}$, $\mathbf{v^u}$ and $\mathbf{r^u}$
Each node u broadcasts hello messages
When receiving a hello message from node i {
Set $\mathbf{v_i^u} = 1$, $\mathbf{a_i^u} = 1$ and $\mathbf{r_i^u} = i$ }
After detecting all neighbors send au to every node w
such that ($\mathbf{v_w^u} = 1$)
Set DONE = false
Repeat the following steps until DONE
or ($\mathbf{a_i^u} = 1$ for all $0 \leq i \leq n-1$) {
Set DONE = true
Receive $\mathbf{a^w}$ from every node w such that $\mathbf{v_w^u} = 1$
For every pair of nodes w and i such that ($\mathbf{a_i^u} = 0$)
and ($\mathbf{a_i^w} = 1$) {
Set DONE = false, $\mathbf{a_i^u} = 1$, and $\mathbf{r_i^u} = w$ }
Send $\mathbf{a^u}$ to node w such that $\mathbf{v_w^u} = 1$ }
End VRP (Start-up Phase)

Figure 3.1 Start-Up Phase (Adopted from [1])

Maintenance phase is triggered when node *u* detects a change in the set of neighboring nodes. When a node *u* detects that the node *y*, which is one of its neighboring nodes, is out of its transmission range, it immediately updates the three elements $\mathbf{a}_y^u$ =0, $\mathbf{v}_y^u$ =0 and $\mathbf{r}_y^u$ = u, and then it sends a^u to its neighboring nodes. On the other hand, when the node *u* detects the existence of a new neighboring node *y*, it updates the vectors $\mathbf{a}_y^u$ =1, $\mathbf{v}_y^u$ =1 and $\mathbf{r}_y^u$ = y then it sends a^u to the neighboring nodes if $\mathbf{a}_y^u$ was zero before update.

Each time node *u* receives $\mathbf{a}^w$ from a neighboring node *w* it updates the two vectors $\mathbf{a}^u$ and $\mathbf{r}^u$ using (3) and (5). Then it sends $\mathbf{a}^u$ to the neighboring nodes only if the received vector $\mathbf{a}^w$ is different from $\mathbf{a}^u$ before update.

$$r_i^u = \begin{cases} w, & if \quad a_i^u = 0 \quad and \quad a_i^w = 1 \\ u, & if \quad a_i^w = 0 \quad and \quad r_i^u = w \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

The following figure 3.2 describes the working of Maintenance phase.

Algorithm VRP (maintenance phase)
// maintenance phase is executed by every node u in the network
if node u detects node y as a new neighbor {
    if $\mathbf{a}_y^u$ = 1 {
        set $\mathbf{v}_y^u$ = 1 and $\mathbf{r}_y^u$ = y; }
    else {
        Set $\mathbf{a}_y^u$ = 1, $\mathbf{v}_y^u$ = 1and $\mathbf{r}_y^u$ = y
        Send $\mathbf{a}^u$ to node w and receive $\mathbf{a}^w$ from node w
        such that $\mathbf{v}_w^u$ = 1 }}
    If node u detects that node y is not in its neighborhood {
        Set $\mathbf{a}_y^u$ = 0, $\mathbf{v}_y^u$ = 0and $\mathbf{r}_y^u$ = u
        Send $\mathbf{a}^u$ to node w and receive $\mathbf{a}^w$ from node w
        such that $\mathbf{v}_w^u$ = 1 }

Continue……..

If node u receives $\mathbf{a}^{w}$ from neighboring node w

If $\mathbf{a}^{u} \neq \mathbf{a}^{w}$ {

For every pair of nodes w and i

such that ($\mathbf{a}_i^{u} = 1$) and ($\mathbf{a}_i^{w} = 0$) and ($\mathbf{r}_i^{w} = w$) {

Set $\mathbf{a}_i^{u} = 0$ and $\mathbf{r}_i^{u} = u$ }

For every pair of nodes w and i such that ($\mathbf{a}_i^{u} = 0$) and ($\mathbf{a}_i^{w} = 1$) {

Set $\mathbf{a}_i^{u} = 1$ and $\mathbf{r}_i^{u} = w$ }

Send $\mathbf{a}^{u}$ to node w and receive $\mathbf{a}^{w}$ from node w such that $\mathbf{v}_w^{u} = 1$ }

End VRP (maintenance phase)

Figure 3.2: Maintenance Phase (Adopted from[1])

CHAPTER 4

MANETS SECURITY ISSUES AND SOLUTIONS

Chapter 4 MANETS SECURITY ISSUES AND SOLUTIONS

4.1 INTRODUCTION

Security in a MANET is an essential component for basic network functions like packet forwarding and routing: network operation can be easily jeopardized if countermeasures are not embedded into basic network functions at the early stages of their design. Unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks those functions are carried out by all available nodes. This very difference is at the core of the security problems that are specific to ad hoc networks. As opposed to dedicated nodes of a classical network, the nodes of an ad hoc network cannot be trusted for the correct execution of critical network functions.

If an *a priori trust relationship* exists between the nodes of an ad hoc network, entity authentication can be sufficient to assure the correct execution of critical network functions. A priori trust can only exist in a few special scenarios like military networks and corporate networks, where a common, trusted authority manages the network, and it requires tamper-proof hardware for the implementation of critical functions. Entity authentication in a large network, on the other hand, raises key management requirements. An environment where a common, trusted authority exists is called a *managed environment.*

When tamper-proof hardware and strong authentication infrastructure are not available, for example, in an *open environment* where a common authority that regulates the network does not exist, any node of an ad hoc network can endanger the reliability of basic functions like routing. The correct operation of the network requires not only the correct execution of critical network functions by each participating node but it also requires that each node performs a fair share of the functions. The latter requirement seems to be a strong limitation for wireless mobile nodes in which power saving is a major concern. The threats considered in the MANET scenario are thus not limited to maliciousness; a new type of misbehavior called selfishness should also be taken into account to eliminate nodes that simply do not cooperate.

With *lack of a priori trust,* classical network security mechanisms based on authentication and access control cannot cope with selfishness, and cooperative security schemes seem to offer the only reasonable solution. In a cooperative security scheme,

node misbehavior can be detected through the collaboration between a number of nodes, assuming that a majority of nodes do not misbehave.

The rest of the chapter is organized as follows. Section 4.2 presents the various types of possible on existing protocols. Section 4.3 discusses the various secured routing protocols briefly. In section 4.4 we discuss the packet dropping attack and counter measures taken for that attack. In this section we discuss two solutions which paved way for our solution for VRP algorithm.

4.2 ATTACKS ALLOWED BY EXISTING ROUTING PROTOCOLS

Current ad hoc routing protocols are basically exposed to two different types of attacks: *active* attacks and *passive* attacks. An attack is considered to be active when the misbehaving node has to bear some energy costs in order to perform the threat, whereas passive attacks are mainly due to lack of cooperation, with the purpose of saving energy selfishly. Nodes that perform active attacks with the aim of damaging other nodes by causing network outages are considered to be *malicious* whereas nodes that perform passive attacks with the aim of saving battery life for their own communications are considered to be *selfish.*

Malicious nodes can disrupt the correct functioning of a routing protocol by *modifying* routing information, by *fabricating* false routing information, and by *impersonating* other nodes. On the other side, selfish nodes can severely degrade network performance and eventually partition the network (X) by simply not participating to the network operation. Below is the brief description of various types of attacks on MANETs as specified in [18].

Threats Using Impersonation : Current ad hoc routing protocols do not *authenticate* routing packets, a malicious node can launch many attacks in a network by masquerading as another node (*spoofing*). Spoofing occurs when a malicious node misrepresents its identity in order to alter the vision of the network topology that a benign node can gather. As an example, a spoofing attack allows one to create loops in routing information collected by a node with the result of partitioning the network.

Threats Using Fabrication : The notation “fabrication” is used when referring to attacks performed by generating false routing messages. Such kinds of attacks can be difficult to identify as they come as valid routing constructs, especially in the case of fabricated routing error messages claiming that a neighbor can no longer be contacted.

Lack of Cooperation: A selfish node that wants to save battery life for its own communication can endanger the correct network operation by simply not participating in the routing protocol or by not executing the packet forwarding (this attack is also known as the black hole attack) . Current ad hoc routing protocols cannot cope with the selfishness problem and network performances severely degrade as a result.

The attacks on network layer in MANETs can be broadly classified into two categories namely routing attacks and packet forwarding attacks. The attacks which target the function of routing are called routing attacks. Routing attacks are specific to different routing protocols. Packet dropping attacks the packet forwarding operation.

4.3 ROUTING ATTACKS AND SOLUTIONS

For securing from routing attacks, the secure ad hoc routing protocols take the proactive approach and enhance the existing ad hoc routing protocols, such as DSR and AODV, with security extensions. In these protocols, each mobile node proactively signs its routing messages using the cryptographic authentication primitives described in chapter 2. By this way, collaborative nodes can efficiently authenticate the legitimate traffic and differentiate the unauthenticated packets from outsider attackers. However, an authenticated node may have been compromised and controlled by the attacker. Therefore, we have to further ensure proper compliance with the routing protocols even for an authenticated node. In the following, we describe how different types of routing protocols are secured from routing attacks.

4.3.1 *ARIADNE:*

Ariadne [11,18], developed by Hu, Perrig, and Johnson is an *on-demand* secure ad hoc routing protocol securing DSR algorithm from routing attacks. It achieves it by applying highly efficient *symmetric* cryptography. In this algorithm is able to authenticate the source who initiated route discovery process, source node can authenticate each intermediate node on the path to the destination present in the RREP message. This algorithm also guarantees that no intermediate node can remove a previous node in the node list in the RREQ or RREP messages.

Though Ariadne provides point-to-point *authentication* of a routing message using a message authentication code (MAC) and a shared key between the two parties, it uses the TESLA broadcast authentication protocol for broadcasting messages like RREQ and RREP messages. Ariadne copes with routing attacks performed by *malicious* nodes but cannot deal packet dropping attack performed by selfish nodes.

In Ariadne, the basic RREQ mechanism is enriched with eight fields used to provide authentication and integrity to the routing protocol:

<ROUTE REQUEST, initiator, target, id, time interval, hash chain, node list, MAC list>

The source node sets the values of fields of initiator, target and id. The time interval is the TESLA time interval at the pessimistic expected arrival time of the request at the destination. The hash chain is initialized to $MAC_{KS,\ D}$ (initiator, target, ID, time interval). The node list and MAC list are initialized to empty lists.

Any intermediate node *A* after receiving RREQ packet checks the values of <initiator, id> with that of its stored values, to determine if it has already seen a request from this same route discovery. If it has, the node discards the packet, as in DSR. The node can also discard the packet if its corresponding key is not disclosed yet. If everything goes correct, the node updates the request by appending its own address (*A*) to the node list in the request, replaces the hash chain field with H [A, *hash chain*] and appends a MAC of the entire REQUEST to the MAC list. After updating, node A rebroadcasts the modified RREQ, as in DSR.

Destination node authenticates the received RREQ by comparing the hash chain field with the calculated hash as given below

H [η_n, H [η_{n-1}, H [. . . , H [η_1, MACKSD (initiator, target, id, time interval)] . . .]]]

where η_i is the node address at position i of the node list in the request, and where n is the number of nodes in the node list. If there is mismatch it discards the received RREQ. If the destination node determines that the request is valid, it returns a RREP to the source, containing eight fields:

<ROUTE REPLY, target, initiator, time interval, node list, MAC list, target MAC, key list>

The fields starting from target to MAC list are set to the corresponding values from the RREQ message. The target MAC is set to a MAC computed on the preceding fields in the reply with the key KDS, and the key list is initialized to the empty list. The RREP is then returned to the initiator of the request along the source route obtained by reversing the sequence of hops in the node list of the request.

An intermediate node after receiving the RREP waits until its key is disclosed. Once it gets the key in specified time interval it appends its key from that time interval to

the key list field in the reply and forwards the packet along to the source route indicated in the packet.

Sender after receiving the RREP verifies the validity of each key in the key list, target MAC and that each MAC in the MAC list. If all of these tests succeed, the node accepts the RREP; otherwise, it discards it.

Ariadne avoids false RERR messages by making each node on the broken route to authenticate the error. Each node that encounters broken link keeps the error in buffer and waits for authentication. The node that encountered the broken link discloses the key and sends it over the return path, which enables nodes on that path to authenticate the buffered error messages. With this arrangement a malicious node can't generate false RERR messages.

4.3.2 *ARAN:*

The Authenticated Routing Ad hoc Networks (ARAN) [13, 18] is a secure routing protocol proposed by Dahill, Levine, Royer, and Shields. It is an on-demand routing protocol that detects and protects against malicious actions carried out by third parties and peers in the ad hoc environment. ARAN introduces *authentication*, message *integrity*, and *nonrepudiation* as part of a minimal security policy for the ad hoc environment and consists of a preliminary certification process, a mandatory end-to-end authentication stage, and an optional second stage that provides secure shortest paths.

Each node before entering into the ad hoc network should possess a certificate signed by common trusted certificate server T. The certificate contains the IP address of the node, its public key, a timestamp of when the certificate was created, and a time at which the certificate expires, along with the signature by T. All nodes are supposed to maintain fresh certificates with the trusted server and must know T's public key.

As a first stage, source node A initiates the route discovery process to reach the destination *X* by broadcasting a route discovery packet (RDP) to its neighbors:

$$[\text{RDP}; \text{IP}_X; cert_A; \text{N}_A; \text{t}]\text{K}_{A-}$$

The RDP includes a packet type identifier ("RDP"), the IP address of the destination (IPX), *A*'s certificate (*certA*), a nonce NA, and the current time t, all signed with A's private key. For each route discovery, node A monotonically increases the nonce.

Each intermediate node records the neighbor from which it receives RDP message. After recording as neighbor, it checks the fields of <NA; IPA>. If it had already seen the tuples it discards the packet else it forwards the message to each of its neighbors after signing the contents of the message. This signature prevents spoofing attacks that may alter the route or form loops. Let *A*'s neighbor be *B*. It will broadcast the following message:

$$[[\text{RDP}; \text{IP}_X; cert_A; \text{N}_A; \text{t}]\ \text{K}_{A-}]\text{K}_{B-}; cert_B$$

Upon receiving the broadcast, *B*'s neighbor *C* validates the signature with the given certificate. *C* then rebroadcasts the RDP to its neighbors, first removing B's signature:

$$[[\text{RDP}; \text{IP}_X; cert_A; \text{N}_A; \text{t}]\ \text{K}_{A-}]\text{K}_{C-}; cert_C$$

Eventually, the message is received by the destination, *X*, which replies to the first RDP that it receives for a source and a given nonce. There is no guarantee that the first RDP received traveled along the shortest path from the source. The destination unicasts a Reply (REP) packet back along the reverse path to the source. Let the first node that receives the RDP sent by *X* be node *D*. *X* will send to *D* the following message:

$$[\text{REP}; \text{IP}_A; cert_X; \text{N}_A; \text{t}]\text{K}_{X-}$$

The REP includes a packet-type identifier ("REP"), the IP address of A, the certificate belonging to *X*, and the nonce and associated timestamp sent by *A*. Nodes that receive the REP forward the packet back to the predecessor from which they received the original RDP. All REPs are signed by the sender. Let *D*'s next hop to the source be node *C*. *D* will send to *C* the following message:

$$[[\text{REP}; \text{IP}_A; cert_X; \text{N}_A; \text{t}]\ \text{K}_{X-}]\text{K}_{D-}; cert_D$$

C validates *D*'s signature, removes the signature, and then signs the contents of the message before unicasting the following RDP message to *B*:

$$[[REP; IP_A; cert_X; N_A; t] K_{X-}]K_{C-}; cert_C$$

A node checks the signature of the previous hop as the REP is returned to the source. This avoids attacks in which malicious nodes instantiate routes by impersonation and replay of *X*'s message. When the source receives the REP, it verifies that the correct nonce was returned by the destination as well as the destination's signature. Only the destination can answer an RDP packet. Other nodes that already have paths to the destination cannot reply for the destination. Although other protocols allow this networking optimization, ARAN removes several possible exploits and cuts down on the reply traffic received by the source by disabling this option.

The second stage of the ARAN protocol guarantees in a secure way that the path received by a source initiating a route discovery process is the shortest. Similarly to the first stage of the protocol, the source broadcasts a *Shortest Path Confirmation* (SPC) message to its neighbors. The SPC message is different from the RDP message only in two additional fields that provide the destination *X* certificate and the encryption of the entire message with *X*'s public key (which is a costly operation). The onion-like signing of messages combined with the encryption of the data prevents nodes in the middle from changing the path length because doing so would break the integrity of the SPC of the packet.

Also, the route maintenance phase of the ARAN protocol is secured by digitally signing the route error packets. However, it is extremely difficult to detect when error messages are *fabricated* for links that are truly active and not broken. Nevertheless, because messages are signed, malicious nodes cannot generate error messages for other nodes. The non repudiation provided by the signed error message allows a node to be verified as the source of each error message that it sends.

As with any secure system based on cryptographic certificates, the key revocation issue has to be addressed in order to make sure that expired or revoked certificates do not allow the holder to access the network. In ARAN, when a certificate needs to be revoked, the trusted certificate server T sends a broadcast message to the ad hoc group that announces the revocation. Any node receiving this message rebroadcasts it to its neighbors. Revocation notices need to be stored until the revoked certificate would have expired normally. Any neighbor of the node with the revoked certificate needs to reform routing as necessary to avoid transmission through the now untrusted node. This method is not failsafe. In some cases, the untrusted node that is having its certificate revoked

may be the sole connection between two parts of the ad hoc network. In this case, the untrusted node may not forward the notice of revocation for its certificate, resulting in a partition of the network, as nodes that have received the revocation notice will no longer forward messages through the untrusted node, whereas all other nodes depend on it to reach the rest of the network. This only lasts as long as the untrusted node's certificate would have otherwise been valid, or until the untrusted node is no longer the sole connection between the two partitions. At the time that the revoked certificate should have expired, the untrusted node is unable to renew the certificate, and routing across that node ceases. Additionally, to detect this situation and to hasten the propagation of revocation notices, when a node meets a new neighbor, it can exchange a summary of its revocation notices with that neighbor; if these summaries do not match, the actual signed notices can be forwarded and rebroadcast to restart propagation of the notice.

The ARAN protocol protects against exploits using *modification*, *fabrication,* and *impersonation*, but the use of asymmetric cryptography makes it a very costly protocol to use in terms of CPU and energy usage.

4.4 PACKET FORWARDING ATTACK AND SOLUTIONS

The protection of routing message exchange is only part of the network-layer security solution for MANET. It is possible for a malicious node to correctly participate in the route discovery phase but fail to correctly forward data packets. The security solution should ensure that each node indeed forwards packets according to its routing table. This is typically achieved by the reactive approach because attacks on packet forwarding cannot be prevented: an attacker may simply drop all packets passing through it, even though the packets are carefully signed. In the following sub sections we study two solutions which mitigate the effects of packet forwarding attack.

4.4.1 SECURED DSR

To mitigate the effects of misbehaving nodes Marti [9] proposed a solution. This solution proposed two techniques namely watchdog and pathrater. The authors first evaluated DSR algorithm in presence of malicious nodes. They found huge degradation in performance.

Watch Dog: This method detects misbehaving nodes. For operation of this method all nodes in the network need to be in *promiscuous* mode. In promiscuous mode node A, which in range of node B, can overhear the communications to and from B even if those

communications do not directly involve A. The source node S relies on intermediate nodes A, B and C to detect malicious nodes. Each node on path will be whether its successor is forwarding data or not.

Each intermediate node stores recently forwarded packets in a buffer. It compares the buffered packet with the overheard one. If there is match it removes the buffered packet. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for forwarding on the packet. If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node.

It seems that watchdog technique works well, but this is not the case always. This technique is suffered from problems like ambiguous collision, receiver's collision, false misbehavior, limited transmission, collusion and partial dropping.

Ambiguous collision occurs when A receives packet from S at the time of overhearing to B. The following figure 4.1 explains this situation. Here A looses both the packets in collision.

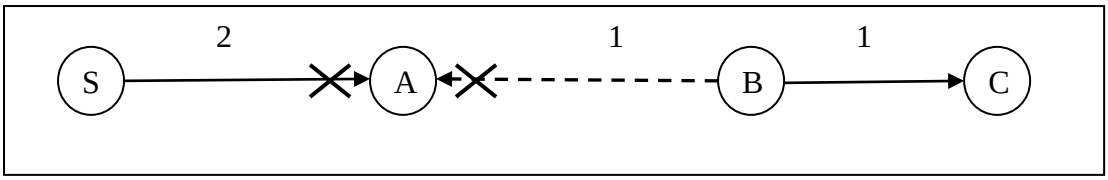

Figure 4.1: Ambiguous Collision Problem (Adopted from [9])

In receiver's problem, node C gets the packets from both B and D at the same time. Due to collision C cannot receive the packet from B. Node A can only determine whether B has sent. Even though B knows about collision at C it will not retransmit the packet because of its selfishness. If B is malicious then it intentionally sends packets to C only when C becomes ready to receive packets from its neighbors. The following figure4.2 explains this situation.

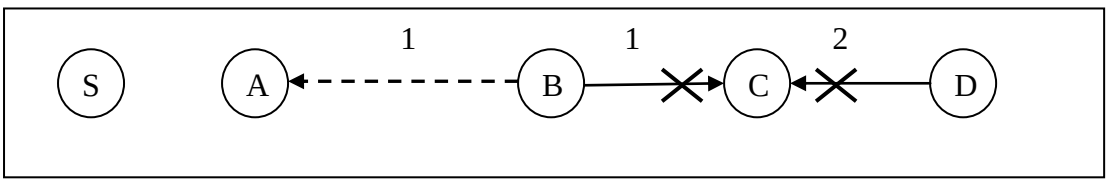

Figure 4.2: Receiver's Collision Problem (Adopted from [9])

In false misbehavior, A may intentionally complain to S that B is misbehaving. By getting false information S may treat B as misbehavior node. But this situation can be detected easily as S receives ACK from D, node S cannot believe A. If A drops ACKs from D, node B informs misbehavior of A to D.

Node B can set the transmission power in such a way that whenever it transmits A can overhear it, but C is not able to receive the packet. In this case A believes B. this situation is called limited transmission problem.

In collusion problem, two nodes collude to drop the packets. In this case, node B forwards a packet to C and do not report to A when C drops the packet.

As mentioned earlier, node A complains the misbehavior of B after exceeding threshold limit. But B just before exceeding the threshold limit forwards the packet sent by A. In this way B will be dropping the packets partially now and then.

The other limitation of watchdog is, it is applicable to source routing algorithms only, this because whenever A wants to complain misbehavior it needs the source.

Path Rater: As the name suggests, this method rates the different paths. To rate a path it combines the misbehaving knowledge with link reliability.

Each node maintains the rating for every other node it knows about in the network. When the network is configured first time each node assigns 1.0 to itself and 0.5 for other nodes. Using these values of nodes in path, path rater calculates rate of path.

Rating of nodes on all actively used paths is incremented by 0.01 at periodic intervals. Path rater decrements node's rating by 0.05 when it detects a link failure during packet forwarding. It assigns -100 to the misbehaving nodes. A path with negative value indicates the presence of misbehaving node. By seeing negative value of the path the source node avoids that path for transferring the data.

The disadvantage of this technique is instead of punishing the malicious node we reducing the overhead to it by turning the traffic to other path.

4.4.2 ON-DEMAND SECURE ROUTING PROTOCOL RESILIENT TO BYZANTINE FAILURES

This algorithm is proposed by Awerbuch [10]. It works in 3 phases namely Route Discovery, Byzantine Fault detection and Link weight management. The first phase route discovery phase discovers the routes from source to destination nodes. The second phase discovers the faults in the links. The third phase Link management phase maintains and

manages the links. As we are interested in packet dropping attack we discuss only fault detection phase.

FAULT DETECTION PHASE: The fault detection algorithm is based on acknowledgements (*acks*) of data packets. The source node after sending data waits for *ack* from destination. If valid *ack* is not received from destination within a timeout, source assumes that packet is lost. After a threshold number of packet loss, source node conducts a binary search on the path to identify the faulty link.

The source node uses *probe list* for identifying the faulty link. Probe list is a set of intermediate nodes which are required to send the *acks* to source in addition to destination. The nodes to list are added iteratively, until the faulty link is identified. Figure 4.3 illustrates the simple example of identifying the faults.

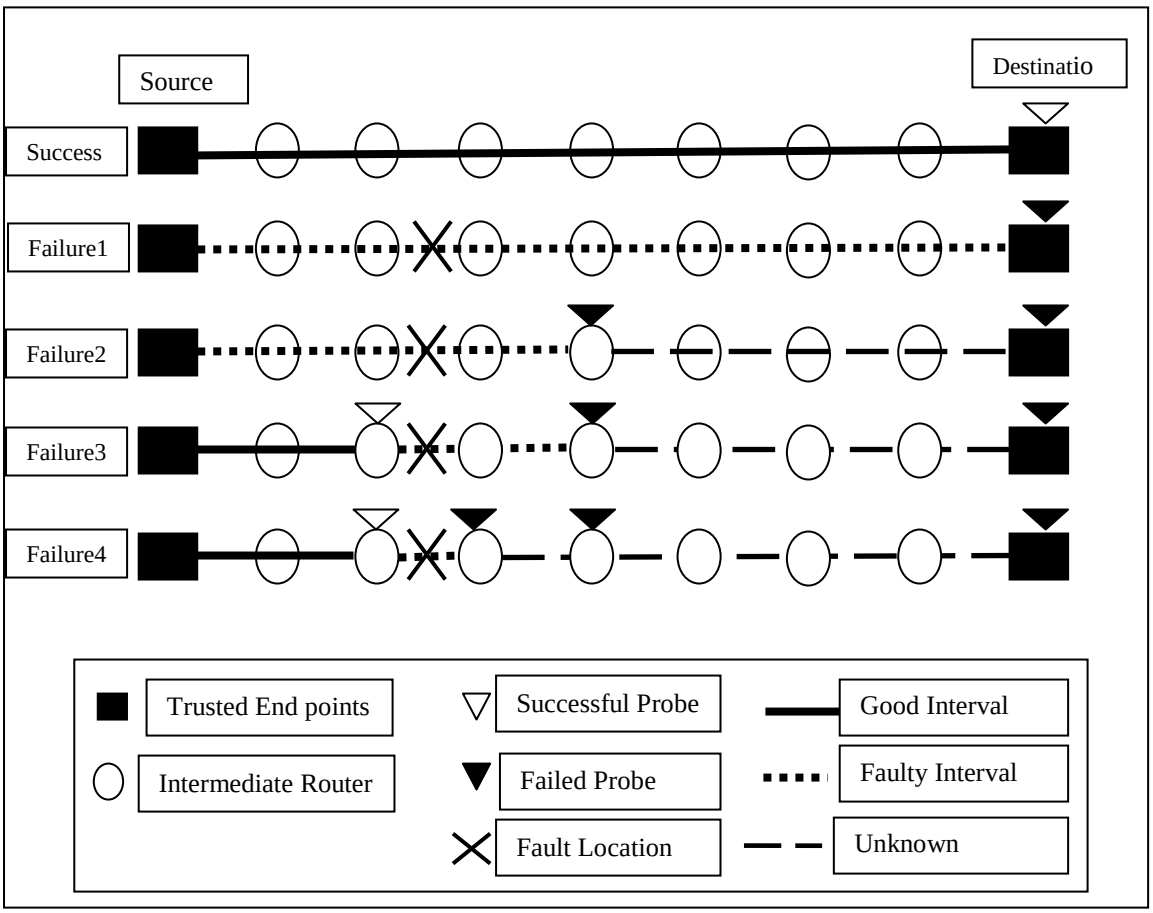

Figure 4.3: Fault Detection Process (Adopted from [10])

CHAPTER 5

SECURED VRP ALGORITHM

Chapter 5 SECURED VRP ALGORITHM

5.1 INTRODUCTION

The success of VRP depends on cooperative nodes. It assumes each node in the network is cooperative and each node forwards the data packets whatever they received. But this is not true always; there can be selfish nodes which do not forward the data packets given to them. These selfish nodes do not harm the network but they decrease the performance of the algorithm.

After studying the effects of Packet dropping attack on different algorithms, we got interest in knowing the effect malicious nodes on VRP algorithm. Section 2 explains our proposal and analysis is given in Section 3.

5.2 SECURED VRP (SVRP)

To detect and eliminate the malicious nodes in VRP we propose new algorithm called Secured VRP [12]. To mitigate the effects of misbehaving nodes we use the concept fault detection algorithm discussed in chapter 4. The fault detection algorithm is able to find the faulty link, but it cannot detect the misbehaving nodes. To detect the misbehaving nodes we modified the fault detection algorithm. For this we used the concept of promiscuous mode.

The following figure explains the modified fault detection algorithm to identify malicious nodes. Assume nodes A, B and C are intermediate nodes. Assume that node B is malicious node. With fault detection algorithm we can only identify the link between B and C is faulty. Here node B can be misbehaving by dropping packets or node C can be dropping packets. In our modified algorithm node S requests node A to observe B in promiscuous mode. Node A accepts request from S and observes whether B is forwarding and informs to S.

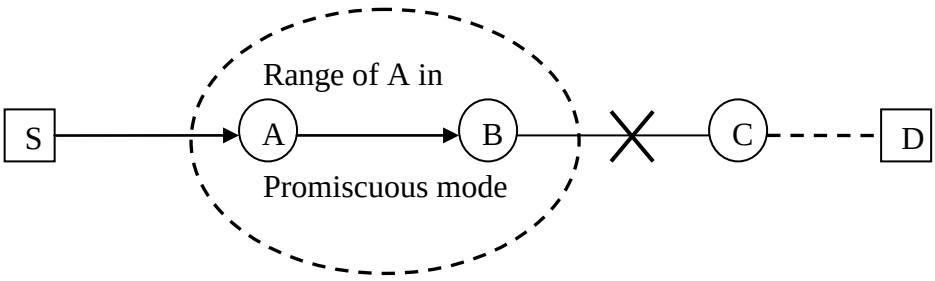

Figure 5.1: Promiscuous mode operation of A

Based on the feedback from node A, node S knows which node is misbehaving and informs to all its neighbors about the misbehaving node.

We propose few modifications to the existing VRP protocol to incorporate the fault detection algorithm. The first modification is each node in the network constructs and maintain entire path to all other nodes in the network. The second modification is that when a fault node is found the neighbors of fault node update their access vector with '-1', so that they permanently eliminate that node from the network.

5.3 ANALYSIS OF SECURED VRP

This section gives analytical analysis of SVRP. Let assume the scenario as shown in the following figure. In this scenario node 0 is having a path to node 4 through nodes 1, 2 and 3. Even though there are multiple paths to node4, we store only one path from node 0 to node 4. Let calculated routing vectors of nodes 0, 1 and 2 be (0, 1, 1, 1, 1, 5, 5, 5, 8, 8, 8), (0, 1, 2, 2, 2, 0, 2, 2, 0, 0, 2) and (1, 1, 2, 3, 3, 1, 3, 3, 1, 1, 3) respectively. We know that topology updates are made by means of access vectors. If a node finds any change in connectivity it updates its access vector and sends the updated access vector to its neighbors. The receiving node updates its access vector if the receiving access vector is different with its access vector.

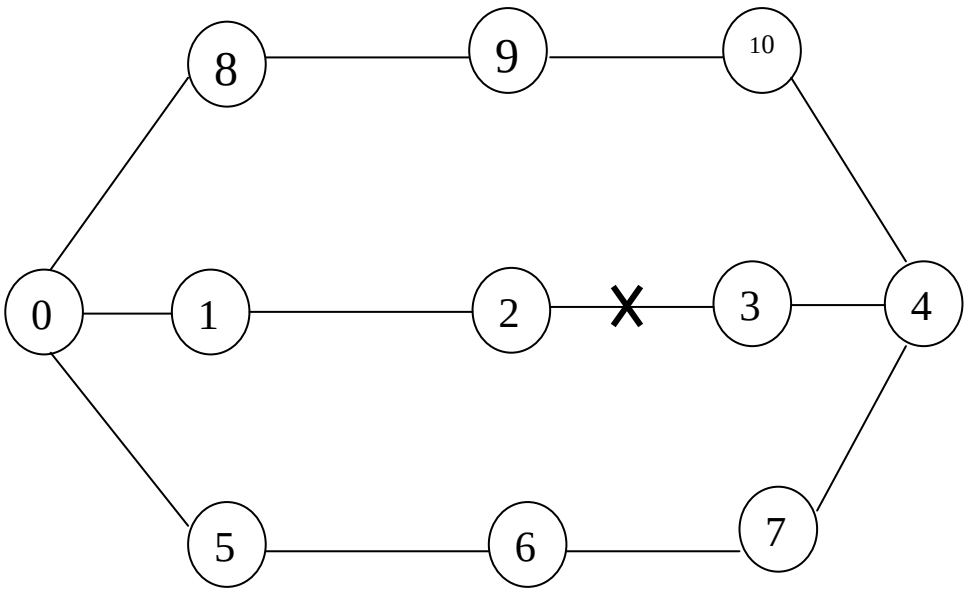

Figure 5.2: Packet Dropping Attack on VRP

VRP algorithm is able to detect the link failures. It cannot detect the packet dropping attack as shown in Figure 5.1. Here even if node 2 continues to drop the packets, no mechanism is there to detect this attack. As node 0 does not receive ACK from 4, it continues to retransmit the packets to 4. Hence due to these retransmissions the objective of VRP (bandwidth efficient) is no more achieved. Our SVRP detects this packet dropping attack as explained in previous section and eliminates node 2 from the network and saves packet retransmissions to some extent. In this way our SVRP tries to achieve the goal to some extent. Considering the overhead, SVRP uses some extra ACK packets from intermediate nodes in the process of finding the misbehaving node. Even though we spend extra packets we achieve huge gain in performance by eliminating the misbehaving nodes. In normal conditions (without misbehaving nodes) SVRP works exactly as VRP.

5.4 SIMULATION RESULTS

We have evaluated the performance of VRP algorithm using ns2 simulator. The evaluation is based on the simulation of 50 wireless nodes forming a MANET over the flat space of size (1500m × 300m) for 900s of simulated time. We used constant bit rate (CBR) traffic sources with packet sizes of 64 bytes. The sending rate is 4 packets per second. We used 30 CBR sources. We varied number of malicious nodes from 5 to 20 (10% to 40%) in the network.

We used packet delivery ratio as the metric. Packet delivery ratio is defined as the total number of packets received at the final destinations to the number of packets sent from traffic sources. Figure 5.3 shows the performance of VRP in both presence and absence of malicious nodes. The number of malicious nodes is varied from 10% to 40%. Results show that performance decreases with increase in number of malicious nodes.

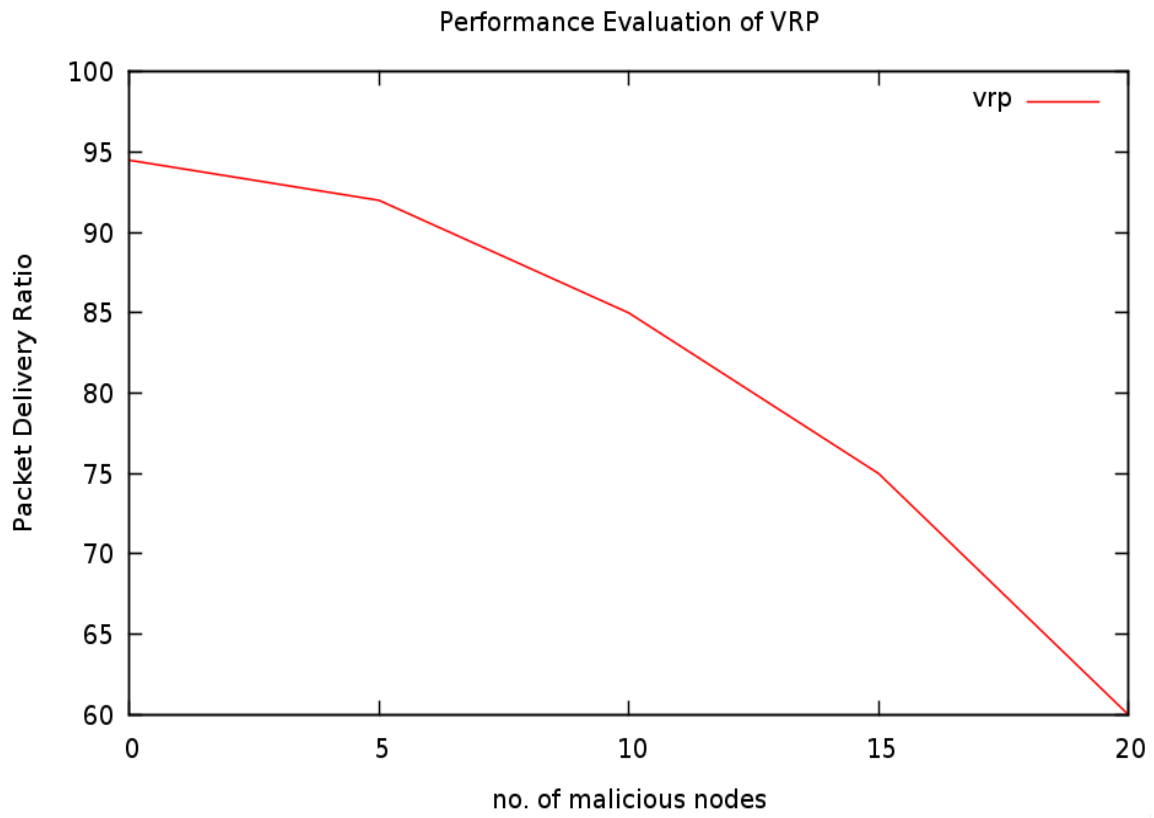

Figure 5.3 Performance of VRP

Chapter 6 CONCLUSION

6.1 CONCLUSION

We have studied MANETs, its properties and challenges in the Routing. We also studied the different types of attacks and solutions to avoid those attacks. We studied VRP algorithm which is proved to be bandwidth efficient. Using ns2 simulator we evaluated the effect of packet dropping attack on VRP and proposed a new algorithm SVRP to overcome the attack.

We analyzed our new proposed SVRP both in presence and absence of malicious nodes. We concluded that SVRP works far better than VRP in presence of malicious nodes and works exactly as VRP in absence of malicious nodes.

BIBLIOGRAPHY

[1] R. S. Al-Qassas, A. Al-Ayyoub, and M. Ould-Khaoua, "Bandwidth-efficient routing protocol for mobile ad hoc networks," IEE Proceedings, Software., vol. 150, No. 4, pp. 230–234, August 2003.

[2] Perkins, E. and Bhagwat, P., Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers, in *Proc. ACM Comp. Comm. Rev.*, Vol. 24, No. 4 (ACMSICOMM '94), October 1994, p. 234.

[3] Johansson, P. et al., Scenario-based Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks, in *Proc. of ACM/IEEE MOBICOM' 99*, Seattle, Washington, August 1999, p. 195.

[4] Johnson, B. and Maltz, D.A., Dynamic Source Routing in Ad Hoc Wireless Networks, in *Proc. Mobile Computing*, T. Imielinski and H. Korth, Eds., Kluwer, 1996.

[5] Das, S.R., Perkins, C., and Royer, E., Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks, in *Proc. IEEE INFOCOM*, Tel Aviv, March 26–30, 2000.

[6] Park, V.D. and Corson, M.S., A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks, in *Proc. IEEE INFOCOM*, Kobe, Japan, April 1997.

[7] Parkins, C.E. and Royer, E.M., Ad Hoc On Demand Distance Vector Routing, in *Proc. 2nd IEEE Workshop on Mobile Comp. Sys. and Apps.*, February 1999, p. 90.

[8] Parkins, C.E., Royer, E.M., and Das, S.R., Ad Hoc On Demand Distance Vector (AODV) Routing, IETF Internet draft, http://www.ietf.org/internetdrafts/ draft-ietf-manet-aodv-03.txt, June 1999.

[9] S. Marti *et al.*, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *ACM MOBICOM*, 2000.

[10] B. Awerbuch *et al*, "An On-Demand Secure Routing Protocol Resilient to Byzantine Failures," *ACM WiSe*, 2002.

[11] Y-C Hu, A. Perrig and D. B. Johnson, "Ariadne: A secure On-Demand Routing Protocol for Ad Hoc Networks," in *Proceedings of MOBICOM 2002.*

[12] P. LakshmiRamana, P. M. Khilar, "Secured VRP for MANETs in presence of Malicious nodes",in NCCTS 24th April, 2009, p.220 – 222.

[13] B. Dahill, B. N. Levine, E. Royer, and C. Shields, "ARAN: A secure Routing Protocol for Ad Hoc Networks," UMass Tech Report 02-32, 2002.

[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) Pvt. 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

[18] S Basagni, M Conti, S Giordano and I Stojmenovic, "Mobile Ad Hoc Networking", 2004. John Wiley Publication.