

SECURE HIERARCHICAL GROUP  
COMMUNICATION ON MOBILE  
AD HOC NETWORKS

By

SABRI ER

Bachelor of Science  
Ankara University  
Ankara, Turkey  
1986

Master of Science  
Oklahoma State University  
Stillwater, Oklahoma  
2003

Submitted to the Faculty of the  
Graduate College of the  
Oklahoma State University  
in partial fulfillment of  
the requirements for  
the Degree of  
MASTER OF SCIENCE  
August, 2003

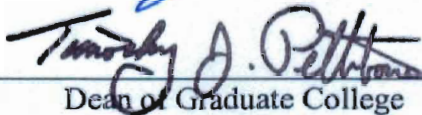
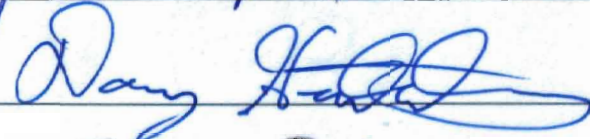
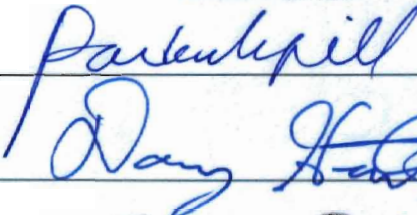
Oklahoma State University Library

SECURE HIERARCHIAL GROUP  
COMMUNICATION ON MOBILE  
AD HOC NETWORKS

Thesis Approved:



Thesis Advisor



Dean of Graduate College

## ACKNOWLEDGEMENTS

I sincerely thank my adviser Dr. Johnson Thomas, for his guidance, help, encouragement and continues support in finishing this thesis. Special thanks are extended to my committee members Dr. Nohpill Park and Douglas Heisterkamp for their advice, cooperation and suggestions for the completion of this thesis.

Several people have helped me during the completion of this thesis. It is impossible to acknowledge them all personally. I extend my special thanks to all.

I thank God for giving me intelligence, courage, and patience. My special gratitude is extended to my parents: my Mom, brother and sisters for their continues support for my education.

Finally, I would like to dedicate this thesis to my nice and patient wife Meltem for her priceless sacrifices for last four years of study at Oklahoma State University.

## TABLE OF CONTENTS

Chapter	Page
1. INTRODUCTION.....	1
1.1 Brief History and Motivation.....	2
2. LITERATURE REVIEW.....	7
2.1 Routing Protocols.....	7
2.1.1 Distance-Sequenced Distance Vector Protocol(DSDV).....	8
2.1.2 The Ad Hoc On-Demand Distance Vector Protocol(AOVD).....	8
2.1.3 Location-Aided Routing(LAR).....	9
2.1.4 The Dynamic Source Routing Protocol(DSR).....	9
2.2 Security in Mobile ad Hoc Networks.....	12
2.3 The Attributes of the security of Ad Hoc Networks .....	15
2.4 Threats, Attacks and vulnerabilities in Ad Hoc Networks.....	16
2.5 Different Security Mechanisms on Ad Hoc Networks.....	17
3. THESIS OBJECTIVES.....	19
4. DESIGN APPROACH.....	21
4.1 Assumptions.....	21
4.2 Main Structure of the design.....	22
4.2.1 Hierarchical Security.....	27
4.2.2 Group Communications.....	34



Chapter	Page
4.3 Key Revocation and Route maintenance.....	48
4.4 Conclusion.....	50
5. IMPLEMENTATION APPROACH.....	51
6. SIMULATION RESULTS AND ANALYSIS.....	53
6.1 Simulation Results.....	53
6.2 Simulating Regular DSR versus Secure DSR.....	58
6.2.1 Route Discovery Process.....	58
6.2.2 Sending Actual message process.....	63
6.3 Simulating Secure Group Communication.....	65
6.3.1 Overhead for re-keying.....	68
6.3.2 Efficiency for re-keying.....	69
REFERENCES.....	72

## LIST OF FIGURES

Figure	Page
1. Topology Changing in ad hoc Network.....	1
2. Path Discovery.....	10
3. Route Discovery.....	11
4. Route Maintenance.....	12
5. Hierarchical Horizontal Communication.....	13
6. Star key graphs before and after a leave.....	38
7. Example MANET.....	39
8. Example MANET after group key change.....	48
9. Discrete Distribution for population size of 10.....	56
10. Number of Nodes vs. Number of bits transmitted for actual message.....	64
11. Number of Nodes vs. Number of messages read for different re-keying mechanism.....	67
12. Overhead for different BG values.....	69
13. Efficiency for re-keying.....	71

## LIST OF TABLES

Table	Page
1. Frequencies for different population sizes with number of hops.....	55
2. Overhead and Efficiency values for different population sizes.....	70

# CHAPTER 1

## INTRODUCTION

Mobile Ad Hoc Networks(MANET) are a new wireless networking paradigm for mobile hosts[7]. An ad hoc network is a collection of wireless computers (nodes) communicating over possible multi hop paths. Unlike traditional mobile wireless networks, ad hoc networks do not rely on any fixed infrastructure. Because nodes are always mobile, links are continuously being set up and broken. In other words, topology is always changing as shown on the Figure 1, below.

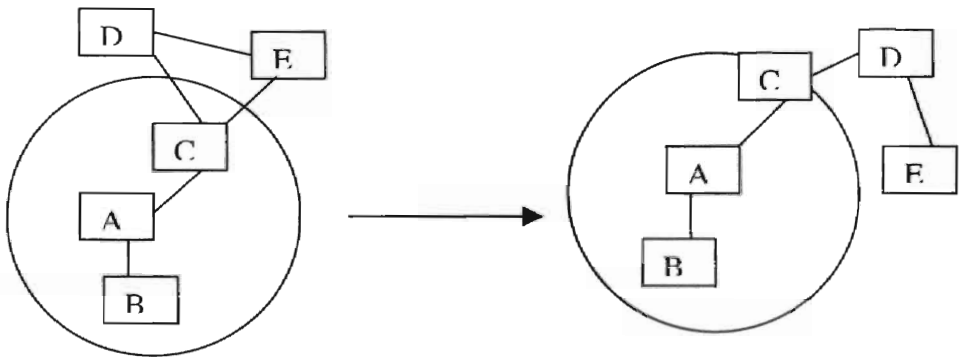


Figure 1. Topology changing in ad hoc Network: The circle represents the radio range of node A. Source: This Figure has been taken from[7] .

Due to the continuously changing topology of Ad Hoc networks, the routing protocols must provide dynamic routing. The nodes that are within each other's radio range can communicate directly via wireless links. However, the nodes that are far apart from each other rely on other nodes in order to transmit messages. That is, nodes are required to relay packets on behalf of other nodes to deliver data over the network.

As mentioned, MANET does not have any fixed infrastructure and thus topology is frequently changing. This situation causes several security problems such as jamming, eavesdropping, distortion[14], and so on. Likewise, relaying messages in MANET is based on implicit trust among the nodes in the network. However, this naive trust allows malicious nodes to paralyze an ad hoc network by inserting erroneous routing, replaying old routing information, and so on. Furthermore, mobile ad hoc networking may operate in unpredictable and dynamic environments. This also brings about several other security issues such as impersonation, modification and fabrication attacks. It is difficult to design ad hoc routing protocols due to

- (a) the highly dynamic nature of the MANET and
- (b) the need to operate efficiently with limited resources such as bandwidth, CPU capacity, memory, and battery [12].

Therefore, it is extremely challenging to provide security in ad hoc networks. However, security is an important issue for this kind of networks especially for those security sensitive applications. The security mechanism is proposed for the MANET should eliminate the problems mentioned above. However, very little progress has been achieved on security in mobile ad hoc networks. In this thesis, we are going to propose and implement security mechanisms in the ad hoc network.

## **1.1 BRIEF HISTORY AND MOTIVATION**

MANETs started in 1972 when Department of Defense(DoD) sponsored the project called Packet Radio Network (PRNET) which later evolved into Survivable Adaptive

Radio Networks(SURAN) in 1980s[10]. The aim for these programs was to provide packet-switched networking to mobile battlefield elements in an infrastructureless hostile environment. In the 1990s, notebook computers became popular and two conference papers were submitted which first used the term “ad hoc networks”. First paper is titled “Highly Dynamic Destination Sequenced Distance Vector Routing for Mobile Computers” by C.E. Perkins and P. Bhagwat published in IEEE Pers. Communication in 1999. Second paper is titled “Routing in Ad Hoc Networks of Mobile Hosts” by D.B. Johnson submitted in ACM Mobicom’94. DoD also continued funding different programs for ad hoc networks. Meanwhile, non-military ad hoc networking applications were proposed and interest for ad hoc networks rapidly grew. Today, ad hoc networking is an area of very active research.

As mentioned above, the first motivation for MANET came from military need for battlefield survivability. Under battlefield conditions, soldiers and their platforms will always be mobile and they should be able to move freely without any restrictions imposed by wired communication devices. Therefore, battlefield survivability requires mobile wireless communication among entities(soldiers) which is not based on centralized control stations. Furthermore, military actions cannot rely on fixed, pre-placed and defined communication infrastructure in battlefield environments. There are also many places such as desert and jungle environments where there is no communication infrastructure at all. Meanwhile, there are other applications for ad hoc networks. Some of them are rescue operations after fire or earthquakes where there is no infrastructure or the existing one has been damaged. New applications such as homeland

security based on sensors scattered throughout the city for biological detection, an infrastructureless network of notebook computers in a conference or campus setting for communication purposes and so on are being proposed. Currently the main applications of MANETs lie in the military domain. Security is critical as any loss of confidentiality of the sensitive information will cause the mission to fail.

We now mention some severe security problems that are present in the battlefield. These are:

**1- Authentication:** When any entity(soldier) is sending a message to his counter partner, it is highly possible that some malicious nodes can redirect the message to them by misrepresenting their IP addresses. For example: a node A wishes to communicate with a node X and sends a message to X. However, a node C through which the message to X passes en-route, may change the source address on the message from A to C. When X receives the message, X believes the message has come from C. X starts communicating with C and sensitive information could be given away to C. This is called IP spoofing.

**2- Confidentiality Attacks:** Another potential security breach may occur when Hackers trying to illegally listen to the network and steal the messages for the enemy. The military has a hierarchical command structure where a general may command one of more officers who in turn supervise a number of privates (although in reality the military has many more ranks, our simple command structure serves to illustrate the potential security problems). This security problem occurs when the military personnel(General,

Officer, Private) want to communicate within a group. General should read all messages coming from Officer or Private. By contrast, Officer or Private should not read the message between Generals. However, without security, it is highly possible that anyone can read any message regardless of hierarchical order. Assume that a General wants to communicate with another General about war plans. Obviously, they do not want anybody to read the messages during the communication. However, any military person in the same group can easily obtain the messages if their signals are within range. This simply causes the mission fail.

**3- Security problem for group communication:** The military also operates in groups. Each group may have one or more generals with a number of officers and privates. Such a hierarchical and group structure introduces a severe security problem. Some soldiers from one group may want to leave their group and join into another one. In this case, this entity or soldier will be able to communicate with the members in two different groups. This is undesirable as the soldier will be privy to conversations in both groups causing totally insecure communication between and within the groups. To solve these problems, we introduce security mechanisms for each problem identified above. In this thesis we propose to:

- 1- Develop a private/public key mechanism against IP spoofing to make sure that message being transmitted is authenticated.
- 2- Introduce Threshold Cryptography to prevent hacker attacks.
- 3- Design a re-keying mechanism to provide security for group communication.



In chapter 2, we present the Literature Review. It contains detail information about the routing protocols; Dynamic Source Routing protocol in particular, Security in Mobile Ad Hoc networks, the attributes of security in ad hoc networks, and possible threats and attacks in the ad hoc networks. In chapter 3, we present our basic thesis objectives. In Chapter 4, we introduce our design approach. This chapter is the main chapter of the thesis. It includes the basic theory of our study with an example. In Chapter 5, we propose our implementation goals as well as implementation tools that will be used to produce our outputs for our study. Chapter 6 shows Simulation Analysis and Results.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 ROUTING PROTOCOLS

A mobile ad hoc network consists of wireless mobile hosts that communicate each other without fixed infrastructure. Routes between nodes may consist of hops through other nodes in the network due to the limited range of each mobile host's wireless transmission. Node mobility can easily cause unpredictable topology changes. That is why, finding and maintaining route in MANET is a nontrivial task to accomplish. Many routing protocols have been proposed for ad hoc networks in order to achieve efficient routing. Some of them are Position-base routing protocol, Dynamic Source Routing protocol, Location-aided routing protocol, Ad Hoc On-demand Distance-Vector routing protocol, and The Zone routing protocol, and so on. Each protocol has its own pros and cons in terms of delay, overhead, and resource use. There is no single protocol developed for efficient routing being used in the MANET. Nor is there any single protocol developed providing full security for the MANET.

We will briefly describe some routing protocols. We will look at Dynamic Source Routing protocol (DSR) closely here, since we are going to use this routing algorithm to apply our security mechanism.

### **2.1.1 Distance-Sequenced Distance Vector Protocol(DSDV)[16]**

Each node implementing DSDV has a route table containing list of the entries. These entries in the list obviously may change dynamically over time. This protocol requires each mobile node to broadcast to its neighbors. In addition, each mobile node agrees to relay data packets to other nodes upon request. Whenever a data is broadcast, the route information consisting of sequence number, destination address, the number of hops required will be updated. Also, nodes have to wait for a certain of time called settling time to communicate their neighboring nodes. These operations cause high delay in this protocol.

### **2.1.2 The Ad Hoc On-Demand Distance Vector Protocol (AOVD)[2]**

The AOVD Protocol provides quick and efficient establishment among the mobile nodes that are trying to communicate. The goal of this protocol is to reduce the need for system-wide broadcasts to the furthest extend possible. In DSDV, whenever two nodes enter communication range of each other, they become neighbors and change the network topology. This triggers a broadcast of new connectivity information for the rest of the network. This is no longer required in AOVD. In fact, if the link status does not affect current communication, there is no need for the broadcast for connectivity information. Route Discovery process is achieved with a broadcast message called a route request generated by source node. When the route request message reaches to destination, a route reply message is sent back to source node.

### **2.1.3 Location-Aided Routing (LAR)[9]**

The major goal of this protocol is to decrease overhead of route discovery by utilizing location information for mobile hosts. Routing discovery can be improved by the technique called flooding. In flooding, the route request packet is broadcast to its neighbor from one node to another node to reach the destination. The average speed and position of a particular node is known and hence the protocol utilizes the Expected zone and Request zone to find a destination node and reduce the overhead. Simply, the idea of expected zone is that if source node does not know the destination, the whole region occupied by the MANET is considered as expected zone. If source node(S) knows destination node(D) at location L at time  $t_0$  and the current time is  $t_1$ . Then, the expected zone of D is the region that the node S expects to contain D at time  $t_1$ . The request zone, on the other hand, includes location of source S, and the expected zone. The size of the request zone is the proportional to average speed of movement, and the time elapsed. A node forwards a route request only if it belongs to the requested zone.

### **2.1.4 The Dynamic Source Routing protocol(DSR)[4]**

DSR was first proposed by David B. Johnson, David A. Maltz and Josh Broch in 1988. The basic operation in the DSR is that the source node is trying to discover a path reaching to destination over network.

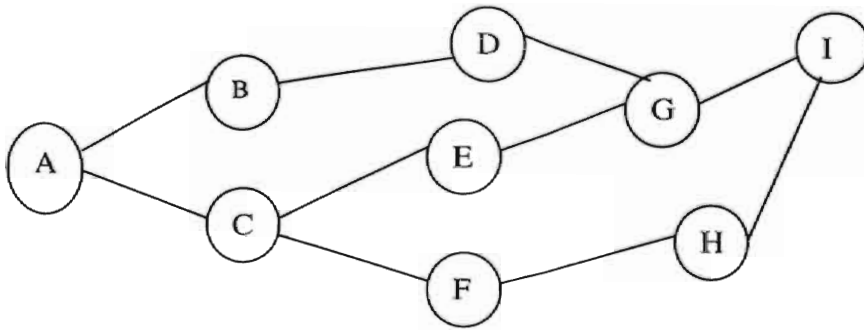


Figure 2. Path Discovery

As shown in the Figure 2, the source node A is trying to find a path to destination node I. The packet sent by the source node is traveling toward a destination has the path list called source route in which it holds the addresses of the intermediate nodes through which it will travel. Each node in the network maintains a dynamic route cache in which it stores routes to other nodes.

When a node sends a packet to next node, the sending node first checks its route cache if the route cache stores a route to the destination. If the sender has an entry for the destination, it inserts the source route into the packet header, listing the addresses of the nodes through which packet will travel to the destination. For example: If the source node has entry for the destination on Figure2, it will list all the addresses of intermediate nodes, say, addresses of the nodes B, D, G, and I. The sending node first transmits the packet to the first node in the list. Upon receiving the packet, each intermediate node forwards the packet to the next node till the packet reaches the destination.

If, on the other hand, the sending node does not have a source route to the destination, it initiates the DSR route discovery process. Figure 3 shows an example of the route discovery in which the source node A is attempting to discover the path to the destination node E.

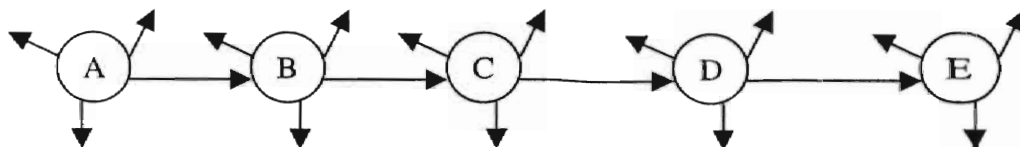


Figure 3. Route Discovery. A is the source node. E is the destination node

The source node A initiating a route discovery broadcasts a route request packet which can be received by the all nodes within wireless transmission range of the node A. The route request packet identifies the initiator as well as target of the route discovery process and it contains a unique request ID determined by the initiator of the route request packet. In addition, each route request packet contains a route record in which it stores the list of addresses of each intermediate node through which the route request packet will be transmitted to the destination.

When any host receives a route discovery packet, it processes the request packet according to the following steps[4]:

- 1- if the node receiving the route discovery packet finds another request packet from the same initiator, the node discards the packet and does not process further.

2- Otherwise, if this is the target node of the route discovery, it returns a route reply message having route record from the route request to the route request initiator. When the initiator receives the route reply, it caches this route in its own route cache in order to send the packets to the destination. Or if it is not target node, the receiving node forwards the packet to the next node on the address list.

Since wireless networks is inherently less reliable than wired networks, route discovery should be coupled with route maintenance in order to check the link connectivity. The route maintenance procedure monitors the operation of the route and informs the sender of any routing errors. As shown Figure 4, below, if the node B is unable to forward the packet to the next node C, B returns a route error to the initiator node A, stating that the link from B to C is broken.



Figure 4. Route Maintenance.

Then, the node A removes this broken link from its cache. If it needs to send a message to the destination node E, it will try another path available to the E. if it does not have any route to the E, it will initiate new route discovery for the target.

## 2.2 SECURITY IN MOBILE AD HOC NETWORKS

In this thesis, we propose an approach to ensuring Security against two attacks, IP spoofing and hacker attack for the nodes in the same logical group of ad hoc network. To

ensure this type of security, trust hierarchy, cryptographic techniques such as encryption and decryption, public key certification, group id, and threshold cryptography to reconstruct the secret key will be used.

If any node wants to join or leave into/from any group, secure group communication will be taken into account. The trust hierarchy will be based on privilege(priority) that nodes will have. There are three kinds of nodes in wireless ad hoc network each of which has different privilege in terms of position as mentioned in[12]. These are General, Officer and Private nodes. The general nodes have the highest privilege(or priority) whereas Private nodes have the lowest priority. The officer nodes have medium level of priority.

General(G) is able to read the messages from all other nodes. However, Officer(O) can read only the messages from another Officer or Private(P) nodes. Private node can read messages only from other Private nodes. The Figure 5 below, shows Horizontal communication where O (Officer) is communication with another O.

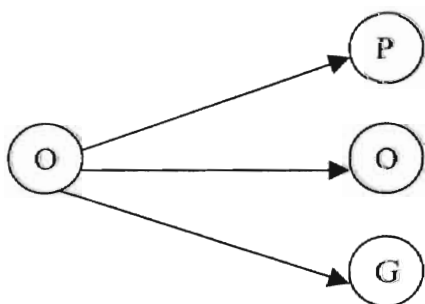


Figure 5. Hierarchical Horizontal Communication



*Cryptography* is the name of the science which hides messages so that only the sender and receiver can read them. *Encryption* is the technique that converts text to string that are visible but they do not seem to have any meaning. A string of these unintelligible characters is made up combination of bits that correspond to alphabetic or numeric values. *Decryption* converts the string that has been created by the process of encryption back to original text on the receiver side. *The RSA(Rivest,Shamir,Adleman) Public Key Cryptosystem*[11] developed at MIT consists of a pair of key called public key and private key. As understood from its name, *the public key* is freely distributed to the public at large and thus known by all other parties. It is used to encrypt messages. *The private key*, on the other hand, belongs to the key owner who keeps it very carefully and secretly. The owner uses the private key to decrypt messages sent to him or her. A *Certification Authority(CA)* issues a digital certificate to an organization or individual. In our study, a trusted server will perform the duty of CA. That is, the trusted server(key server) will be in charge of certifying each node. *Threshold Cryptography*[8] is a branch of cryptography in which the secret key is split into a number of pieces. Each piece is stored securely and carefully. The process of reconstructing a secret key is based on availability of certain number of secret keys at the same time. This certain number of secret keys represents the threshold value. In other words, we need at least the number of secret keys that equals to threshold value to decrypt a message. For example; think of a case where our secret key consists of 2 numbers, say, 1 and -3. Now if these two shares(numbers) are available, our secret key can be reconstructed. If any attacker finds one of the shares ,say, -3 , he will not still have sufficient information to reconstruct the secret key .

## 2.3 THE ATTRIBUTES OF THE SECURITY OF AD-HOC NETWORKS

To secure ad hoc networks, the following attributes can be considered. These attributes are availability, confidentiality, integrity, and authentication and non-repudiation[7].

**Availability:** This is the major issue in the MANET due to the dynamic and unpredictable conditions in the MANET systems. Because nodes may not be available for communication all the time. Availability ensures that network services survive despite of denial service attacks that may be launched by any layer of the network.

**Confidentiality:** Confidentiality ensures that information, particularly sensitive information such as strategic military information, is never disclosed to the unauthorized nodes. This is one of the most important attributes that should be ensured by the MANET.

**Integrity:** This guarantees that the message being transmitted is never corrupted. The message could be corrupted due to radio propagation or malicious attacks on the network.

**Authentication:** Whenever a message being transmitted over the network reaches a node, it must be checked and ensured that it is originated from the correct sender and coming from certain node. Without authentication, malicious nodes could gain the authority and access to sensitive information.

**Non-repudiation:** This ensures that the origin of the message cannot deny having sent the message. This is very useful to detect compromised nodes.

## **2.4 THREATS, ATTACKS AND VULNERABILITIES IN AD-HOC NETWORKS**

In general terms, security involves the potential of threats, attacks and vulnerabilities of a certain system [14]. The most severe vulnerabilities in ad hoc networks are caused by the lack of physical infrastructure among mobile nodes. Therefore, there is no any physical security between the nodes communicating with each other and no trust on any centralized resource.

Major potential attacks against routing can be divided into two groups[14]; passive attacks and active attacks. The passive attacks basically involve only eavesdropping messages on the network. These attacks are against privacy of the communication, rather than attempting to disrupt functioning of the network or its routing protocol. The active attacks, on the other hand, involve actions performed by adversaries such as message replication and deletion. An active attacker injects packets into the network and also generally eavesdrops. Also, it is possible to see attacks on an ad hoc network's routing protocols. These attacks generally fall into one of the two categories[14]; routing disruption attacks and resource consumption attacks.

In the former one, the attacker attempts to cause data packets to be routed incorrectly. In the latter category, the attacker injects the packets into the network to attempt to consume valuable network resources such as bandwidth, memory or computation power.

Threats can be divided into three groups[14]. First group is Denial of Service(DoS) threats which involve the exhaustion of the arbitrarily chosen nodes. The more severe threat of DoS today is the Distributed DoS(DDoS), which involve large amount of distributed nodes attacking the system simultaneously. The second group threat is Integrity threats. Integrity is one of the attributes of the MANET. Integrity threats involve damaging the message being transmitted. If the routing algorithm is not robust enough against this kind of threats, the messages on the way will be corrupted. The third group of threat is the disclosure threat that basically involves eavesdropper(s) trying to break the confidentiality of the information being sent.

## **2.5 DIFFERENT SECURITY MECHANISMS ON AD HOC NETWORKS**

Several different security mechanisms are under active research[1, 2, 7, 10, 12] to develop a security system(s) for different issues that MANETs' are facing. Each security mechanism addresses for a particular security problem. Some of the security mechanisms are:

- 1- **Key Management Service:** Public key infrastructure is used in order to achieve integrity and non-repudiation. In a public key infrastructure, each node has public/private key pair. Public keys can be distributed to other nodes while private keys should be kept confidential by each node. There is a trusted entity called a Certification Authority(CA) for key management. The CA is in charge of assigning the public/private key pair to each node. In addition, It is responsible to revoke the public key of a node if a node is not trusted anymore thereby providing security to the network.

- 2- **Threshold Cryptography:** As mentioned above, it basically a scheme which allows  $n$  parties to share the ability to perform a cryptographic operation so that any  $t+1$  parties can perform this operation jointly. However, it is infeasible for at most  $t$  parties to do so[8].
- 3- **Password-Based Key Agreement:** This is particularly used in a group of people who want to set up a secure session in a meeting room without any fixed infrastructure. Only those entities that know initial password can learn the session key essential for the communication among group of people. The session key is formed by the contributions of the all members in the group. So that any person who is not within the group is unable to learn the session key[1].
- 4- **Resurrecting Duckling Security:** This is mainly implemented for a secure transient association between two devices establishing a master-slave relationship. It is secure in the sense that both master and slave share a common secret. Also it is transient because the master can only solve the association. In addition, a master can identify the slave in a set of devices. The duckling is here slave device while the mother duck is the master controller. The duckling will always obey its mothers who tells to it whom it can talk through an access list[1].

## CHAPTER 3

### THESIS OBJECTIVES

Route discovery, route reply processes and upon discovering a route, the sending of messages over the discovered path on DSR has been heavily studied particularly by academia. Furthermore, our work considers different hierarchical levels for people involved in the battlefield namely General, Officer, and Private. Moreover, each military person belongs to a designated group. While a message is being transmitted, security consideration should be taken into account within a group as well as between communicating groups. However, small amount of work has been done on group communication. Also, little work has been carried out for providing security in both route discovery process and transmitting messages over ad hoc networks. Therefore, our major objectives in this thesis are:

- 1- Security and hierarchical group communication: Communication originating at lower levels of hierarchy is accessible to those in the higher levels. We use threshold cryptography and group key in order to ensure secure communication between nodes.
- 2- Security and communication within a group: We use only group key. Everyone within a group can have the group key to communicate with other members of the group.

- 3- IP Spoofing: Some malicious nodes within the network try to misrepresent the IP addresses of the nodes so as to read the message. We use Private/Public key description against IP spoofing.
- 4- Hacker Attacks: Hackers will try to attack the network to eavesdrop the communication. We use threshold cryptography against Hackers.
- 5- Evaluation: We will evaluate the effectiveness of these security mechanisms.

## **CHAPTER 4**

### **DESIGN APPROACH**

Our goal is to provide secure routing in a group-based environment using the Dynamic Source Routing Protocol(DSR) in Mobile Ad Hoc Wireless Networks(MANET). This chapter consists of four main parts:

- 1- Assumptions
- 2- Main Structure of the design
- 3- Key Revocation and Route maintenance
- 4- Conclusion

#### **4.1 ASSUMPTIONS**

The basic assumptions for security mechanism over DSR are as follows;

- 1- We have a battlefield scenario. Before the process begins, trusted server for each group assigns certificate, IP address, group-id, and individual key to each node
- 2- The ad hoc network consists of logically defined groups
- 3- There is trust among different groups
- 4- There is a trusted server for each group which certifies each node and assigns group id
- 5- Nodes representing soldiers are mobile and broadcasting messages continuously
- 6- The source node is trying to find the destination node through route discovery and route reply mechanism to send packets
- 7- Each node has a static IP address assigned by the server
- 8- Each node has cache memory to hold the table structure



9- Hierarchical level to each node is assigned. General (G) has the highest priority to read the messages while Private (P) has the lowest priority. Although we consider a 3-level hierarchical system, our approach is applicable to a hierarchical system with  $n$  levels.

#### **4.2 MAIN STRUCTURE OF THE DESIGN**

As mentioned above, we have three different kinds of nodes in our wireless ad hoc network system. These are General(G), Officer(O), and Private(P). At the same time, we presume that we have logically defined groups in our ad hoc network. Each node(G,O, or P) belongs to a certain group. A node can belong to only one group at a time. There is a trusted server for each group which distributes a group key to be shared by all group members. The trusted server also distributes private keys(individual keys). The private key will be used for two different purposes. First, it will be used to encrypt the message for route discovery and sending message processes. Secondly, it will be used for confidential communication between a node and the key distribution server during group communication. Members in the same group can decrypt(read) their messages by using their group key. If they are from different groups, however, they cannot read their messages. When a node wants to send a message within a group, it encrypts the message and sends it. The nodes which are from the same group as the sender have the key(Group key) to decrypt the message. The nodes that do not have the same group key as the sender cannot decrypt the message. For instance, a General in group 1 cannot communicate with a General in group 2 or with other nodes(O,P) in other groups. In other words, our approach ensures that a node from one group cannot decrypt a message belonging to a

node from another group even if the hierarchy of the former node is equal to or higher than that of the latter one.

However, when routing a message, the message can be routed through nodes that belong to other groups (even though the message cannot be read by nodes that belong to a different group). In other words, the path discovered by the source node can contain the nodes belonging to different logically defined groups.

The source node attempts to find a route to the destination node through route discovery. We will be considering two major security requirements for our study. We first consider the authentication of IP addresses of the nodes involved in route discovery. This protects against IP spoofing. When a message arrives at a node, the message has to be authenticated, that is, it has to be confirmed that the message is coming from the node that claims to be the sender of the message. One or more malicious nodes may try to misrepresent a node's identity in the network by changing its IP address. As described in chapter 1, the malicious node can direct the message to itself. The second security threat is a Hacker attack. Hackers attempt to decrypt the message. Threshold cryptography based on private key reconstruction will be used against hacker attacks. A hacker has to have certain number of keys(threshold value) to construct the secret key and hence to decrypt the message. If he has less than threshold value of keys, he will not be able to decrypt the message.

When a node sends a packet to another node in the network, first it checks if it has a full list of addresses of the nodes including destination address. If the sender has an entry for the destination node in its route cache, it inserts the source route into the packet's header, listing all the addresses through which the packet will travel to the destination. The source route is the path discovered to send packets from source to destination. In our study, the source node does not have entry for the destination and thus it does not have list of addresses of the nodes to reach destination. Therefore, the source node will initiate the route discovery based on the DSR route discovery process. We will be checking for spoofing attacks during the route discovery process as this is the most likely kind of attack during route discovery. Each route request packet initiated by the source node is uniquely identified by the packet type identifier("RDP"), source address(IP address of source), destination address(IP address of destination), Request-id, and certificate of source node (which consists of source address, public key of the source node, Node hierarchy, and time stamp that issued the certification). The time stamp will be needed to decide when a certificate of particular node should be revoked. Private keys( $K_s^-$ , and  $K_t^-$ ) seen on the route request packet and certificate are not part of the packet and certificate being transmitted to the destination. These are just operations. The reason to illustrate them here is to show that route request packet and certificate are encrypted by using these private keys.

$S \longrightarrow$  broadcast:  $[RDP, IP_s, IP_d, R-id, cert_S]_{K_s^-}$

$T \longrightarrow$  S:  $cert_S = [IP_s, K_s^+, H, t]_{K_t^-}$

S: Source node

T: Server issuing certification for the nodes

RDP: Route Discovery Packet

IPd: IP address of destination

IPs: IP address of source node

Ks-: Private key of the source node

Ks+: Public key of the source node

R-id: Request ID

Kt-: Private key of the server

H: Hierarchy of the source node; P, O, or G

t: A timestamp issued when a certificate is created

certS: certification of the source

The key server gives each node a certificate. The certificate contains the public key of the node, the timestamp, the source IP address and the hierarchical level of the node. The certificate is encrypted with the private key of the key server. The purpose of the certificate is to authenticate the sender of the message. When a node receives a packet, it decrypts the message with the public key of the source. To authenticate the packet, the receiving node decrypts the certificate contained in the message with the public key of the key server. Authentication is achieved as the contents of the certificate confirm to the receiving node that the packet is from the node that claims to be the sender. The timestamp issued by the server is used to decide the time at which certificate will be revoked. One of the most important reasons for using a certificate is that each node has

its own values or elements(Public key, time stamped, etc.). Moreover, the certificate is distributed by a trusted server(key server).

Once a route is discovered, the source node will immediately begin sending data packets (actual messages) to the destination over intermediate nodes. Again, we will have to check security in each node during message transfer to make sure that our security mechanism prevents any IP spoofing, Hacker attack, and any problem that will jeopardize secure group communication. The actual message format will be as follows:

$$[[[M]_{xx}, ]y, IPs, IPd, certS]_{Ks-}$$

M: Message

xx: Threshold Cryptography for node hierarchy

y: Group key

Other notations have already been notified above.

As you see from the message format, three different keys are being used in the message format. These keys are just for operations. They are not part of the message being sent to the destination. One of them is  $Ks-$ , private key of the source node. This key is used to encrypt the entire message format, before a message is transferred from source node to any other intermediate node. The node receiving the whole message can decrypt the message by applying the public key of the source node (this is known publicly). While the message is being transferred towards the destination, each intermediate node will simply transmit the original encrypted message. The decrypted packet authenticates the source and also identifies the final destination. The Second key being used is the group

key nominated by the letter  $y$ . Upon decrypting the whole message packet by using the public key of the node that has sent the message, the receiving node will check if it is within the same group with the source node by trying to decrypt the partial message,  $[[M]xx]y$  by using its group key. If they (source and receiving node) are from the same group, it will be able to decrypt the partial message. The group key is not like private/public key. All nodes from the same group have the same group key. The third key used is the one represented by  $xx$ . This shows the hierarchical level within a group. If the receiving node is from the same group as the source node, it will be able to decrypt the partial message. However, to read the actual message nominated by  $M$ , the receiving node should be able to construct the secret key described by threshold cryptography, as explained below. It is worth mentioning one more time that these keys are not part of the message sent. These are used for encryption purpose at the beginning. Then this original message is sent to destination through intermediate nodes.

#### 4.2.1 Hierarchical Security

Threshold Cryptography will be applied for vertical communication among the nodes within the same group. That is, threshold cryptography will be applicable from lower level to higher level communications and from higher level to lower level communications. In order to accomplish this goal, we have some assumptions:

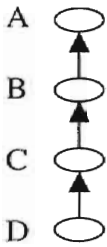
- 1- The highest level (General) can read any message
- 2- Message sent from a lower level to a higher level can be read by all higher level nodes

- 3- Message sent from higher level X to lower level Y can be read by all levels between and including Y

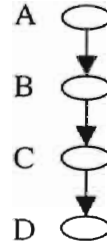
Our approach can be extended to a  $n$ -level hierarchy. We assume that each node knows the hierarchy key of the highest level, its own hierarchy key and the hierarchy keys of all levels below. The security criteria is that lower levels do not read messages between higher levels. For example, a message between levels 1 and 3 should not be read by levels 4 and below.

Given  $n$  levels, the secret key to encrypt and decrypt a message at a level  $n-i$  is composed of  $2 + i$  keys. However, as we are using threshold cryptography, the secret key can be reconstructed if two of the  $2 + i$  keys are known. The secret key composed of  $2 + i$  keys consists of: the hierarchy key of the highest level, the hierarchy key of level  $n-i$  and the hierarchy keys of the  $i$  levels below level  $n-i$ . As the highest level can read all messages, we assume it knows the keys of all the levels below it and can therefore construct the secret key easily. For example, assume a 4-level hierarchy. At level 4, ( $n = 4, i = 0$ ), the secret key is composed of 2 keys. These will be the key of the highest level and the key of level  $n$ . At level 2 ( $n = 4, i = 2$ ), the secret key is composed of 4 keys. These will be the hierarchy key of the highest level, the hierarchy key of level 2 (level  $n-i$ ) and the hierarchy keys of the  $i$  levels (2 levels) below level  $n-i$  (level 2). Therefore the secret key is composed of the keys at levels 1, 2, 3, 4. Level 2 can get the secret key if it can obtain two of these keys. It knows its own level key (level 2) and it knows the key of the highest level. Level two can therefore obtain the secret key. As each level knows its own hierarchy key and the key of the highest level, a message is encrypted and decrypted using these two keys. Remember, from the definition of threshold cryptography, we need

$t$ (threshold) number of keys out of  $m$  keys ( $t \leq m$ ) in order to construct secret key essential to decrypt and encrypt the message.



(From lowest to highest hierarchy level)



(From highest to lowest hierarchy level)

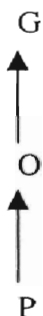
For example: Assume we have 4 level hierarchy ( $n=4$ ), say, from A(highest level) to D(lowest level) as shown above. Suppose the lowest level node, D, wants to communicate with the highest level node, A. In other words, D is the source and A is the destination. The node C's secret key is composed of the keys A, D, and C. When node C receives a message from D, it can decrypt the message by constructing the key DC, (2 out of 3 keys for threshold cryptography). Now when node B receives the message from C, it can decrypt the message by constructing DB, (2 out of 4 keys for threshold cryptography). The node B's secret key is composed of A,B,C, and D. Similarly, the highest node A can decrypt the message by just using any key, 1 out of 4 keys. Let us see another scenario from C to A. In this level, D will not be able to read the message. We know that B's hierarchy key includes A,B,C, and D. When the node B receives a message from C, the node B will decrypt the message by constructing the key AB, (2 out of 4 keys for threshold cryptography). Similarly, the node A will decrypt the message by just using any key. If we look at another scenario from B to A, the node A will decrypt the message by using any key, 1 out of 4 keys.



To transmit a message lower down the hierarchy, on the other hand, we have the same assumptions made above. Now assume the node A(highest level) wants to send a message to D(lowest level). When node B receives the message from the node A, it can construct the secret key AB, 2 out of 4 keys for threshold cryptography. B's hierarchy key is composed of A,B,C, and D. Likewise, when node C whose hierarchy key is composed of A,C,D receives the message from the node B, it can decrypt the message by constructing secret key AC, 2 out of 3 keys for threshold cryptography. Now when node D receives the message from the node C, it can construct secret key AD, 2 out of 2 keys for threshold cryptography. D's hierarchy key is composed of A, and D. Let us see second scenario from B to D. The node B will encrypt the message and send to node C. The important point over here is that since every node has lower and highest level keys, the message can be decrypted by using those keys(highest and lower level keys). When the node C whose hierarchy key is composed of A,C, and D receives the message , it will decrypt the message by using AD key description. Likewise, when the node D receives the message, it will also use AD key description to decrypt the message. If we look at another scenario where the node C is sending message to D, the node C will encrypt the message by just implementing AD key description. Upon receiving the message, the node D will decrypt the message by AD, 2 out of 2 keys for threshold cryptography. As shown above particularly in the examples, the idea of threshold cryptography can be applied to n-level hierarchy.

In our approach which is also based on threshold cryptography, at least two keys are needed to obtain the secret key. For communications from a lower level to a higher level, the key is composed of the key of the source node and the key of the highest level.

For example: In our study, we may have the following scenarios:



(lowest to highest level)



(highest to lowest level)

In the first scenario (lowest to highest), the message will be encrypted by P. The nodes O and G can construct the secret key consisting of 2 keys out of 3 keys (P,O,G) by using threshold cryptography. These 2 keys are PO and PG. That is, encrypted message can be read by O thereby using PO and also it can be read by G by using PG. If message is just going from O to G, G will use the OG secret key (2 out of 2 keys, O and G) to decrypt the message. In the second scenario, the message will be encrypted by the G. This message will be decrypted by O and P thereby constructing GO and GP secret keys (2 out of 3 keys, G,O, and P). If message is going from O to P, P will use the GP secret key(2 out of 2 keys, O and P).

For horizontal communication (the same level communication), on the other hand, we will just use level key for encryption and decryption of the message. We may have source and destination have the same level of hierarchy. In this case our message format will be as follows:

[[[M]]y, IP<sub>s</sub>, IP<sub>d</sub>, cert<sub>S</sub>]K<sub>s</sub>-                      **l: level key**

The assumption we have over here is that each node (P, O, G) has level key given by the trusted server in addition to their group and individual keys. As mentioned earlier, these keys including level key are not part of the message. They are used just for encryption purpose.

The major assumption is that the level key of the lower level is known by the higher level. However, the level key of the higher level is not known by the lower level.

For example: Assume that we have the following scenarios;



A message is transferred from a private to another private through other intermediate nodes that can be G, O, or P. The message will simply be encrypted by the source node using level key and will be sent. Since P is the lowest level, all other intermediate nodes within the same group know the level key of the P. Hence, they can easily decrypt the message by using level key of the P. However, if the message is going from G to another G,



We do not want other intermediate nodes (O, P) to read the message. According to our assumption, lower level does not know the level key of the higher level. Since G has the highest hierarchy, P and O do not know the level key of the G. So, the source G will encrypt the message by using its level key and send it. Any node except other G(s) cannot decrypt the message.

Similarly, if a message going from O to another O,



The intermediate nodes O and G (G knows the level key of O) only can read the message. P cannot because it has lower level than O thus it does not have the level key of O.

Also, we will have group key description for the group communication. *This description will be applied for y in the message format.* Actually, this description is showing that nodes must be in the same group to communicate each other. We may have  $n$  numbers of group: G1, G2, G3, ....., Gn.

As seen from the message format, the actual message the source node is sending to destination sits into most inner part of the message format. In order to reach the message, in other words, to decrypt and read the message, each intermediate node first should decrypt the entire message by using public key of the source node ( $K_s^+$ ). Then, it has to have one of the group key descriptions for the group (G1, G2, G3, ....., Gn). Lastly, it has to have one of the threshold cryptography descriptions for the nodes P, O, and G. As

explained above, it might be 2 out of 3 or 2 out of 2 key description. For horizontal communication, there will only be level key to decrypt the message.

#### 4.2.2 Group Communications

The two major security checking will be enforced as long as all the nodes stay in the same group. Nevertheless, nodes may change their present group by moving from one to another. In this case, we have to take into account secure group communication as well. In other words, if any node leaves a group and joins another group, the group keys of these two groups have to be changed (re-keyed). So that the joining node will be unable to access previous communications in the new group and a leaving node will have no access to future communications of the group it is leaving. This is essential to achieve a high level of security for group communication. For this reason, based on [3] we define a *secure group* which is a triple  $(U, K, R)$ .  $U$  is the set of users,  $K$  is the set of keys and  $R$  is a relation between  $U$  and  $K$ . As mentioned above, each secure group has a *trusted key server* responsible for generating and securely distributing keys to the users in the group. Every user has the key set consisting of two keys, individual key(private key) and group key. The individual key is shared with the server and used for confidential communication with the server within a group. The group key, on the other hand, is shared by the server and all other users in the group and is used by each user to send messages confidentially to other members of the group.

There are several algorithms and methods that have been developed for re-keying strategies. Baseline re-keying, immediate re-keying, delayed re-keying[6] are some

examples of re-keying algorithms. Similarly, the key graph method[3] is one of the methods being used as re-keying strategy. We are going to use one of the special classes of key graphs in our study named *star key graph*. Using the star key graph, the complexity of secure group is much less than that of other methods such as a tree structure particularly in terms of structure of the graph and re-keying strategies.

A key graph is a directed acyclic graph  $G$  with two types of nodes: *u-nodes* representing users and *k-nodes* representing keys. Each *u-node* has one or more outgoing edges but no incoming edge. Each *k-node* has one or more incoming edges. If a *k-node* has incoming edges only and no outgoing edge, then this *k-node* is called a root. The star key graph is a special class of a secure group  $(U, K, R)$  where each user has only two keys: individual key and group key. Whenever a user  $u$  wants to join or leave a group, it sends a join(or leave) request to the server  $s$ . The server  $s$  has to grant the request to perform join or leave into/from a group. The individual key is used for this communication between a node and the server. After each join or leave, a new secure group has to be formed. The server  $s$  has to update the group's key via two steps. First it will generate new group keys for the nodes in the group. Secondly, it will securely distribute the new group keys to the nodes. The server constructs and sends re-key messages to the users in order to securely distribute the new keys.

**Leaving a star key graph:** After granting a leave request from user  $u$ , the server  $s$  updates the key graph by deleting the *u-node* for user  $u$  and the *k-node* for its individual key  $ku$  (individual key for a node  $u$ ) from the key graph. This key is the individual key  $(K_n^*)$  assigned by the trusted server when the group was formed. Then the server  $s$  generates a

new group key for the new secure group without  $u$ , encrypts it and sends to the users in the group. Then users receive a new group key by decrypting the new group key by using their individual keys. Suppose  $u4$  wants to leave from group1(G1), Figure 6 (a). and join group4(G4), Figure 6 (b). So that there will remain only three nodes in G1. Likewise, the number of nodes in G4 will increase from three to four. Also, assume that the server  $s1$  changes the group key from G1 to a new key G2, server  $s1$  needs to send out the following re-key message:

$$s1 \longrightarrow \{u1, u2, u3\}:\{G2\} G1$$

$s1$ : The trusted server for G1.

That means that the server sending new encrypted group key to the users  $u1$ ,  $u2$ , and  $u3$  is stating that the group key they have will be changed from G1 to a new one G2. Remaining nodes will decrypt the message by using their private keys ( $k1$ -,  $k2$ -, and  $k3$ -) and own their new group key G2.

**Joining a star key graph:** After granting a join request from user  $u$ , the server  $s$  updates the key graph by creating a new  $u$ -node for  $u$  and a new  $k$ -node for  $ku$  (individual key for a node  $u$ ,  $K_n$ ) and attaching them to the root node. Then server generates a new group key, encrypts it against any attacker and sends the encrypted new group key to every user in the group including joining node. Each user receives the new group key thereby decrypting the message(new group key)by using its own individual keys( $k1$ -,  $k2$ -,  $k3$ -, and  $k4$ -). Now we know that  $u4$  has left G1, and joining into new group G4. After the server for G4 grants the join for  $u4$ , the server will send the following re-keying messages

for the nodes in G4 as well as new joining node,  $u4$ . Assume also that the server( $s4$ ) will assign new group key, G5.

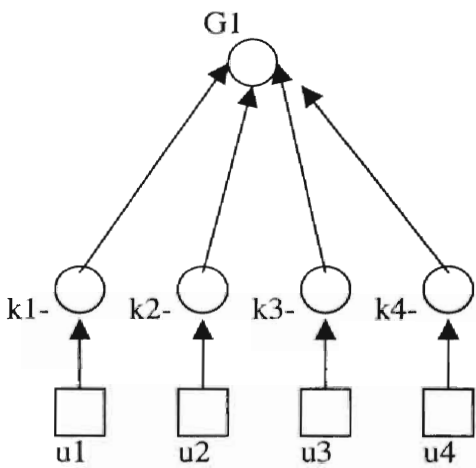
$s4 \longrightarrow \{u1, u2, u3\} : \{G5\} G4$

$s4 \longrightarrow \{u4\} : \{G5\} G1$

$s4$ : The trusted server for the G4.

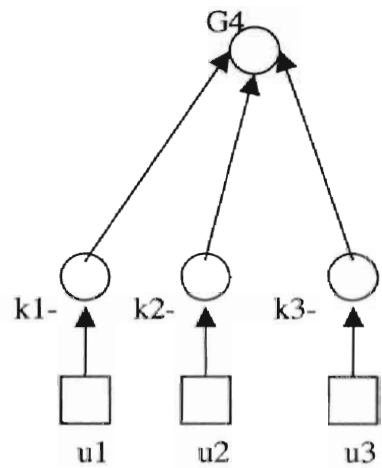
That means that the group key current nodes have will be changed from G4 to G5.

Likewise, joining node will have new group key G5.



(Before the node  $u4$  leaves)

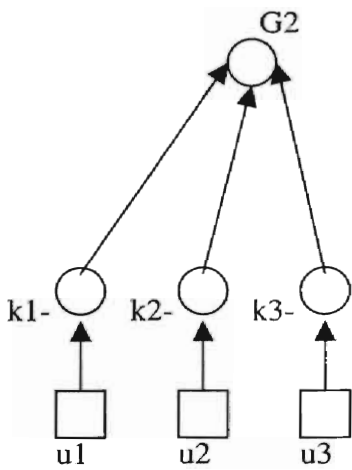
(a)



(Before the node  $u4$  joins a new group)

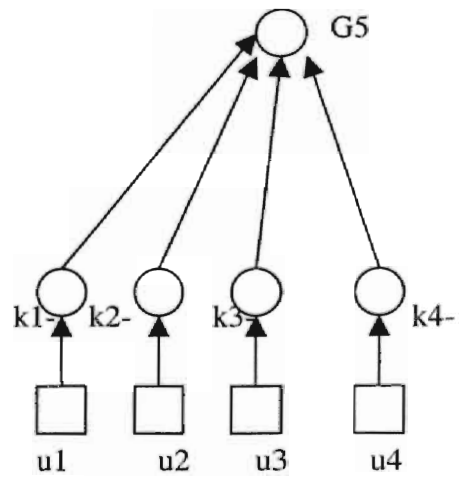
(b)





(After the node u4 leaves)

(c)



(After the node u4 joins a new group)

(d)

Figure 6. Star key graphs before and after a leave. Source:[3]

We can specify the secure group for the example above as follows assuming that u4 is leaving the G1, (a).

$$U = \{u1, u2, u3, u4\}$$

$$K = \{k1-, k2-, k3-, k4-, G1\}$$

$$R = \{(u1, k1-), (u2, k2-), (u3, k3-), (u4, k4-), (u1, G1), (u2, G1), (u3, G1), (u4, G1)\}$$

**Example :**

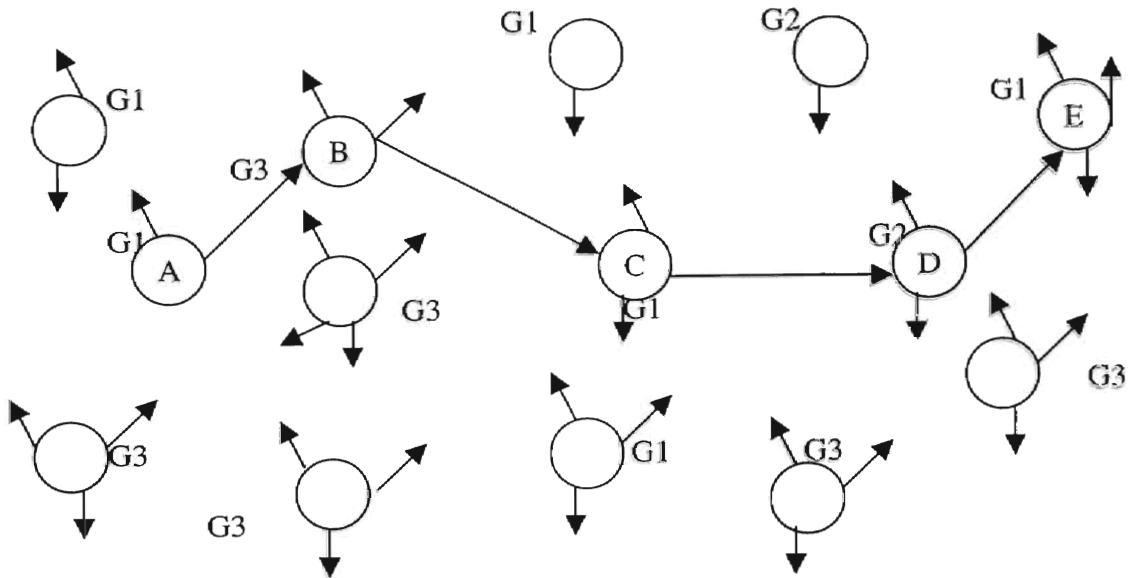


Figure 7. Example MANET

Now we will show implementation of security mechanisms we have mentioned above. We are going to have three steps to illustrate our security mechanisms in our ad-hoc network example.

- 1- Ensuring security for the route discovery packet
- 2- Ensuring security for actual messages
- 3- Ensuring security for group communication

Let us assume that we have a wireless ad hoc network system as shown in figure 7. Each node has its own IP address and certification issued by the trusted server. Likewise, the trusted servers for each group also issue the group ids' (group keys) for each node in their

groups from group1 to group3 (G1, G2, G3). Each node has its own radius range (not shown in Figure 7). The node A is the source node trying to reach the destination node, E through route discovery mechanism. When the node A is broadcasting, it is looking for the node whose cache memory has the address of the destination. If there is more than one node which includes the address of the destination in the radius range of the node, the node will choose the shortest path. As seen from the figure, the route from node A (group1) to node E (group 1) is through nodes B (group 2) . . . etc.

Also assume that the nodes from A – E are assigned as follows;

A = Private

B = Officer

C = General

D = Officer

E = General

*1- Ensuring security for route discovery packet:* We are only discovering a route and thus sending only route request packets. The route discovery packet does not contain actual message. Consequently, we are basically ensuring authentication against IP spoofing.

The source node, A will set up its route table and initiates route discovery packet as follows;

Destination	Next	Metric
E	B	NA (4)

The Metric shows the number of hops that the packet has to go through from the specified node to the destination. NA: Not Available. At the beginning the node A does not know how many hops it will go through to reach the destination. After it receives a route reply packet, it will learn the number of hops the messages has to go through to reach the destination. The number of nodes(hops) will be 4 in our example.

A  $\longrightarrow$  broadcast: [RDP, IPa, IPe, certA]K<sub>a-</sub>

T  $\longrightarrow$  A: certA = [IPa, K<sub>a+</sub>, ta, P]K<sub>t-</sub>

When the source node A broadcasts the packet, the node B in group3 receives the message. Keep in your mind that private keys K<sub>a-</sub>, and K<sub>t-</sub> are not part of the route request packet and certificate. They are just used to encrypt the packet and certificate at the beginning. The first encrypted packet format will be forwarded through intermediate nodes till destination is reached. Assume that node B has the address of the destination. When B receives the message, it decrypts the packet by using the public key of the node A known by everyone. Then it verifies the certificate of A and decrypts the certificate by using the public key of the server T. Finally, it checks and sees the public key of the node A which ensures that this packet is coming from the node A. This authenticates that the

packet belongs to node A. This protects against IP spoofing. Then node B will set up its route table and broadcast.

Destination	Next	Metric
E	C	NA (3)

B → broadcast: [RDP, IPa, IPe, certA], certB

When node C in group1 receives this packet, it will authenticate by decrypting the certificate B by using public key of the server. The authentication will ensure that the packet is coming from B. Once node C has made authentication check, it will set up route table and broadcast as follows:

Destination	Next	Metric
E	D	NA (2)

C → broadcast: [RDP, IPa, IPe, certA], certC

When the node D receives this packet, it will repeat the same procedure performed by the previous nodes. It will perform an authentication check. It must make sure that it is from the node C. Then it will set up its own route table and broadcast.

Destination	Next	Metric
E	E	NA (1)

D → broadcast: [RDP, IPa, IPe, certA], certD

When the node E receives this message, it will repeat the same process by checking that the packet is coming from node D. When the destination E receives the packet, it will replace RDP with REP(Replay Packet). Similarly, it will replace IPe with IPa(Source node address) and encrypt the reply packet by using private key of E( $K_{E-}$ ). However, this private key is not part of the reply packet. This encrypted reply packet will be forwarded back to the source through intermediate nodes.

E → D: [REP, IPa, certE] $K_{E-}$

When route replay packet is sent to the source, it will follow the same steps and procedures applied for the request packet.

**2- Ensuring security for the actual messages:** Once the reply packet reaches the source, the source node starts sending messages to the destination over the path discovered. Each intermediate node has now the route table set up during the route discovery process. The source node A is now sending data packets. The message is being sent from Private to General, from lower level to higher level. The message format being sent by the node A will be as described earlier:

[[[M]xx]y, IPa, IPe, certA] $K_{A-}$

All keys( $K_a$ ,  $y$ ) and description for threshold cryptography( $xx$ ) are not part of the message being sent. They are just operation to encrypt the first message. When the node B receives this data packet, the first thing it will do is to make sure that it is coming from the node A. In other words, it will authenticate against IP spoofing. It will encrypt the entire packet by using public key of the node A. Then, it will encrypt the certificate of A by using public key of the server. It will see the public key of the node A ensuring authentication. Next step is to check if these two nodes (A and B) are from the same logically defined group. This will be performed by looking at the group key set up for group communication corresponding  $y$  in the message format.

To decrypt the message, node should have the same group key. We know, however, the node B is from  $G_3$ . Therefore, node B cannot decrypt this part of the data packet. That is, these two nodes (A and B) are not from the same group. Thus, they cannot communicate each other. As a result, the node B cannot read the message. So node B will replace its own certification on the data packet and send it to the node C by looking at its own route table set up previously. Now the data packet looks like as follows:

$[[[M]], IP_a, IP_e, cert_A], cert_B$

When node C receives data packet, it has to make sure that it is coming from the node B. So the node C encrypts the certification of B and see the public key of B ensuring authentication. Now node C will decrypt the entire packet by using public key of the A.

Then it will check if it is possible to decrypt the next part of the packet by looking at the group key.

Since the nodes A and C are from the same group, node C is able to decrypt this part of the message. When the node C opens certificate of A, it will see the letter P that corresponds to Private. That is, the message is coming from Private node. However, we do not know yet whether the node C can read the actual message due to the hierarchical ordering within a group. In other words, we have threshold cryptography description for the node hierarchy. Now node C is going to check the node threshold cryptography description to determine whether it can decrypt the actual message. Now the node C that is General, highest level can decrypt the message by just using any key, 1 out of 3 keys. Now the node C will drop certification of the node B, then replace them with its own certificate and send the message to the next hop, D, in its route table. So message format looks like:

[[[M]], IPa, IPe, certA], certC

When node D receives this data packet, it is going to repeat the same procedures performed by the previous nodes. It will make sure that the message is coming from the node C by encrypting the certification of the node C. Then, it will decrypt the entire message packet and check group key.



Since the node D is from Group 2, the node D cannot decrypt this partial message format. As a result, the node D will drop the certification of the node C and replace with its own certificate and send the data packet to the next hop, E.

[[[M]], IPa, IPe, certA], certD

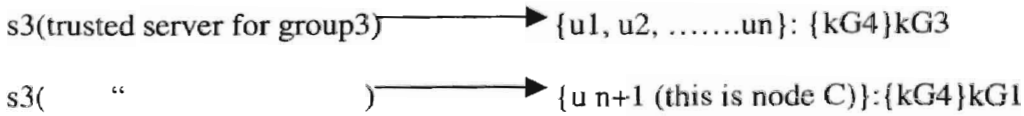
Upon node E receiving this data packet, it will repeat the same procedure one more time. Node E will authenticate the packet, decrypt the entire message and check for group key.

The destination node E, is from Group 1. the node E is therefore able to decrypt this part of the data packet and it sees P when it decrypts the certificate of A showing that the message originally came from a Private node. Then, it will check whether it can read the actual message by verifying the node threshold cryptography. The node E is also General, highest level can decrypt the message by just implementing 1 out of 3 keys. Whenever we reach the new node, the node checks whether it is a destination node. Since E is the destination node and it is able to read the actual message, the message transmission is completed.

### ***3- Ensuring security for group communication:***

As long as nodes stay in the same group, the system basically follow what has been described so far. However, it is always possible that highly mobile nodes might move from one group to another one. In this case, we should consider group communication security as mentioned before. Let us assume that the node C in group1(G1) wants to join group3(G3). Now, the server for group3 will change the group key from G3 to a new

group key, say, G4. Then it will encrypt new group key and send re-key message to all nodes in group3 including to the new joining node, node C. The re-key message will be as follows:



All nodes in group3 as well as new node (node C) will decrypt the message by using their individual key and owning new group key, G4. The server for group1, on the other hand, will change group key from G1 to, say, G5 in order to prevent leaving node accessing to previous communication. Then, it will encrypt the group key and send re-key messages to all the nodes in group1 excluding leaving node (node C).

The re-key message will be like:



All the nodes in group1 will receive the encrypted message, decrypt by using their individual keys and hence owning a new group key, G5. So that our new ad hoc network will look like as follows:

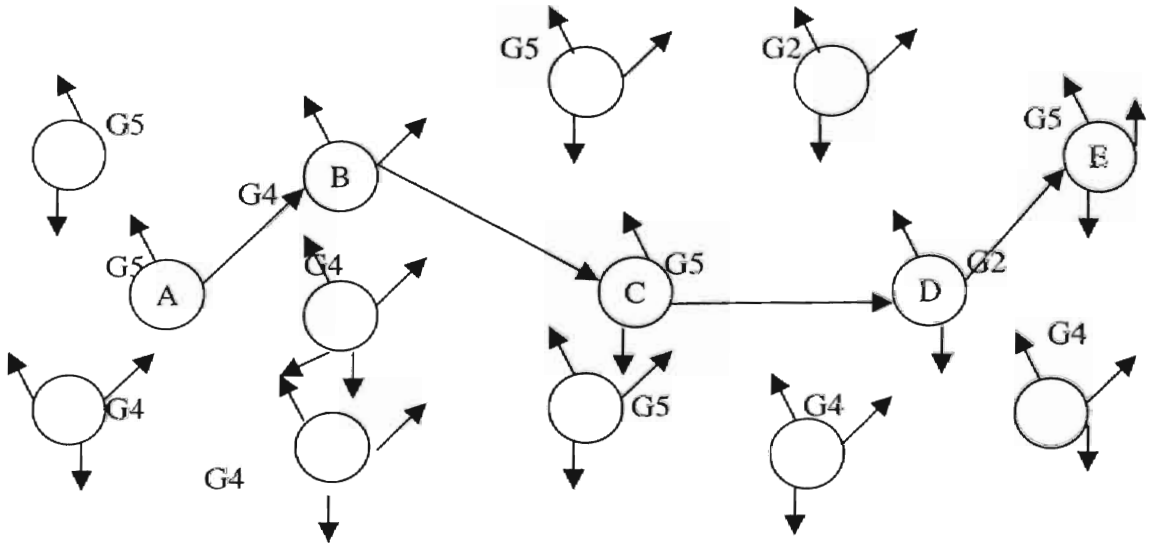


Figure 8. Example MANET after group key change

### 4.3 KEY REVOCATION AND ROUTE MAINTENANCE

We have to keep in our minds that the discovered route cannot be available forever. Each node has a time stamp within its own certificate issued by the server before entering the network. The trusted server has a table storing the time stamps it has issued for each node. When the time stamp is expired, the certificate should be revoked. In the event of certificate revocation, the trusted server, T, sends a broadcast message to the ad hoc network that announces the revocation.

To call the revoked certificate, say,  $\text{certX}$ , the transmission appears as:

$T \longrightarrow \text{broadcast: } [\text{revoke, certX}]_{K_t}$

Any node receiving this message re-transmits it to its neighbors. The nodes store this notice until the revoked certificate expires normally. Now the node for which certificate has been revoked is called an untrusted node. Simply, this node is out of order. Therefore, the neighboring nodes will avoid transmitting the message through the untrusted node. When a node's certificate is revoked among nodes that compose the route, this route now cannot be used to transmit the message. As soon as there is an untrusted node on the route, it has to be known by the source node sending messages to the destination. This process is called route maintenance. In other words, the node which is trying to avoid using the untrusted node should send "ERR" message back to the source node indicating that the route is broken and unable to forward the message. For example; let us assume that the certificate of the node C in our example has expired and hence its certificate has been revoked by the server. The server will broadcast the revocation message to the whole network. Thus, every node in the network knows that C is an untrusted node.

Now the neighboring node B will be unable to forward the message to the node C and it will generate the ERR message for the node C as follows:

$B \longrightarrow C : [ERR, IPa, IPd, certC]Kb-$

This message is forwarded back to the source node, A, without any modification. Private key of B,  $Kb-$ , is not part of the message, just operation. Once the source node receives the ERR message, it stops sending messages over the route discovered previously. The sender either uses another path if it is available or tries to discover a new route, if it is not available. Obviously, certificate revocation is not only one reason to generate ERR

message. Another reason might be node movement, since we have very mobile nodes in an ad hoc network.

#### **4.4 CONCLUSION**

There are basically two kinds of security problems in this study being considered. These are IP spoofing and hacker attack. In addition, group security has to be considered whenever any node wants to join or leave into/from a group. This is basically accomplished through re-keying mechanism. Route discovery mechanism over DSR has been used in order to find a path to send messages. Three types of nodes are involved in the network; General, Officer, and Private. General has the highest priority whereas Private has the lowest priority to read the messages by looking at the key description based on the idea of threshold cryptography for node and group key. Moreover, our approach is directly applicable to a  $n$ -level hierarchy. If the time stamp expires in a node's certificate, the server revokes the certificate of that particular node making it untrusted node. Hence, ERR message is generated back to the source node indicating that the route being used is not available any more.

## CHAPTER 5

### IMPLEMENTATION APPROACH

We will use the software BRITE in order to generate a network with nodes and edges at the first time. The BRITE very simply receives the number of nodes and generates the network and puts into input file. We will write our own algorithm to read that input file, form mobile network, and produce our outputs from which we produce our graphs.

- 1- We will first find distribution for each population size starting from 10 nodes and incremented by 10 up to 100 nodes. Then we will calculate number of hops from source to destination for each population size based on the distribution of each population.
- 2- We will simulate regular DSR being used for route discovery versus secure DSR. Our goal is to see overhead with regard to packet size (or number of bits transmitted). Obviously, the packet size used on secure DSR will have bigger compared to one on regular DSR, since it will have additional field(s) to implement security mechanisms. We will investigate not only the route discovery process, but also the overheads for sending actual messages.
- 3- Re-keying: If any node wants to join/leave a certain group, there are going to be message readings from the group the node has left and from the group it has joined without security. That is, it will be possible for the node to read communications from both groups. So the criteria we will use might be the number of messages read by joining/leaving node with no re-keying mechanism, with half re-keying, and with full re-king. Apparently, with no re-keying

mechanism, the number of messages being read from both groups will be low or non-existent, but not secure at all. By implementing different level of re-keying (half or full), we want to see how many messages the node is going to read from both groups. However, our major interest for re-keying mechanism is to find overhead resulting from extra messages to perform re-keying. We will show how overhead varies for different population sizes while we fix group size. Then we will look at the efficiency which will basically reflect overhead.

## CHAPTER 6

### SIMULATION RESULTS AND ANALYSIS

#### 6.1 SIMULATION RESULTS

We have used different population sizes starting from 10 nodes up to 100 nodes incrementing by 10. In other words, population sizes we used are 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. We have used the BRITE network topology generator software to produce initial topology (x and y coordinates) of each node in each population. We run our simulation 1000 times and obtained frequencies for each number of hops for each population size. These frequency values for each number of hops and population size are given on Table 1 below. Then we obtained discrete distribution for each population size based on the data on Table 1. Discrete distribution for population size of 10 is shown on Figure 9 as an example. Also, we calculated weighted mean value that is actually the average number of hops from source to destination for each population size from 10 to 100 as shown below. We have used 200 units for radius of each node. Why 200 units?. We conducted a number of experiments to determine the signal range where a reasonable number of nodes (at least 1-2) are within the signal range, and route breaks are not very frequent or regular; however, route breaks do occur. Too small a signal range means that very few nodes will be within range, causing frequent route breaks, whereas too large a range will put a large number of nodes within range causing little or no route breaks. We appreciate that this value of 200 is not a fixed value and can be changed. In the real world, signal ranges vary from individual nodes to individual nodes. There is therefore no optimum range. We have calculated average distance from source to other nodes



including destination node for each population size by using x and y coordinate values. 200 units is approximate average distance for all population sizes. In other words, this is reasonable radius size based on information (x and y coordinate values) we obtained from the BRITE. Likewise, we have used 1500 units square shape for size of spatial world (or window size) Each node may move eight directions (North, South, East, West, Northeast, Northwest, Southeast, and Southwest). We have decided these boundaries because after we run our simulation for each population size, we have observed that x and y coordinates for each node fall mainly within these boundaries. If any node has x and/or y values larger than these boundaries, maximum boundary value is assigned for that node as x and/or y value.

Number of Hops	FREQUENCIES for different population sizes									
	10	20	30	40	50	60	70	80	90	100
3	165	37	22							
4	360	290	370		160	134	405	120	450	110
5	298	295	280	137	388	410	200	285	180	235
6	150	330	148	227	140	300	175	322	150	330
7	26	28	113	305	147	86	99	125	115	240
8	1	20	41	285	125	22	55	110	45	57
9		1	15	36	27	47	40	5	22	18
10			10	10	10	1	15	8	18	6
11			1		2		5	10	10	4
12					1		6	10	8	
13								5	2	
<b>Total</b>	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

Table 1. Frequencies for different population sizes with number of hops

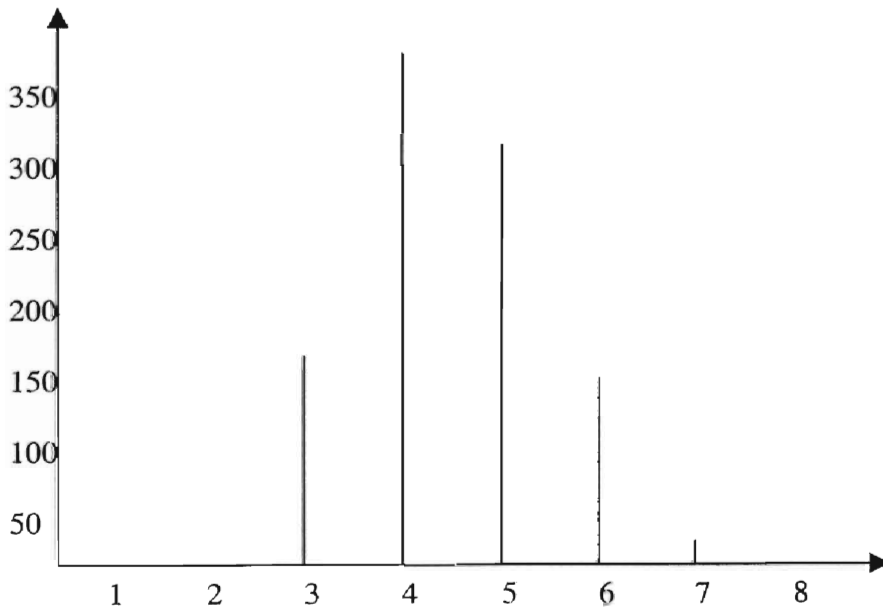


Figure 9. Discrete distribution for population size of 10

<u>Population size (n):</u>	<u>Weighted mean (number of hops):</u>
10	4.5
20	5
30	5
40	6.9
50	5.8
60	5.5
70	5.6
80	5.8
90	5.5
100	5.8

Once we got all distributions for each population size, we have calculated variance value for each distribution in addition to weighted mean. The following basic formulas have been used for these calculations.

$$\text{Weighted mean}(\bar{Y}) = \frac{\sum f_i \cdot y_i}{\sum f_i}$$

$$\text{Variance}(\hat{s}) = \frac{\sum f_i \cdot y_i^2 - (\sum f_i \cdot y_i)^2 / \sum f_i}{\sum f_i - 1}$$

From our distributions for each population size, our distribution looks like the Negative Binomial Distribution, one of the discrete distributions. However,  $X^2$  (ki-square) goodness of fit analysis should be made to justify this claim.

$X^2$  (ki-square) goodness of fit:

$$X^2 = \sum_{i=0}^k (y_i - 1000 * p(y))^2 / 1000 * p(y)$$

k: number of different hop size

$y_i$ : number of observations for a particular hop size

$p(y)$ : probability of success of a node to be destination

1000: we have 1000 observations for each population size

To calculate  $p(y)$ , we have used mean and variance formulas from the negative binomial distribution and solved for particular success (x) and for probability of success (p) from the mean and variance formulas below. These mean and variance formulas are:

$$\check{y} = x \cdot (1 - p) / p \quad x = \check{y} \cdot p / (1-p)$$

$$\hat{s} = p \cdot \check{y} \quad p = \hat{s} / \check{y}$$

Then we have used probability function for different number of hops and obtained different probabilities for different number of hops. The probability function is:

$$p(y) = \binom{x + y - 1}{y} p^x \cdot (1-p)^y$$

Once we got  $p(y)$  values for each population size and for each number of hops within the population, we have calculated  $X^2$  values for each population size. When we look at the  $X^2$  table with certain degree of freedom ( $df = k-1$ ) with 5% confidence interval,  $X^2$  value calculated for each population size by using the formula above should be less than certain value on the  $X^2$  table. We calculated each  $X^2$  value for each population size. We have taken 5% confidence interval and checked  $X^2$  table for certain degree of freedom. We have observed that our  $X^2$  values are greater than table values. As a result, we have concluded that our distribution is not the negative binomial distribution. It actually does not match with any discrete distribution.

## 6.2 SIMULATING REGULAR DSR VERSUS SECURE DSR

### 6.2.1 Route Discovery Process

We will simulate regular DSR vs. secure DSR in terms of packet size overhead for both route discovery and sending actual message processes. As far as the regular DSR is concerned, the route request and route reply packet formats will be as follows:

Route Request packet format for regular DSR:

Source Address	Destination Add.	Request-ID	Message Type
32 bits	32 bits	8 bits	3 bits

Route Reply packet format for regular DSR:

Source Address	Destination Address	Request-ID	Next Hop	Metric	Message
32 bits	32 bits	8 bits	32 bits	8 bits	3 bits

In our study, we have added security mechanism (basically IP spoofing for route discovery process) on regular DSR and thus we got secure DSR. For secure DSR, route request and route reply packet formats will be as follows:

Regular DSR packet format (total 75 bits)	Cert S			
	IPs	Ks+	H	t
	<b>32 bits</b>	128 bits	3 bits	8 bits

From the packet format description in Chapter 4 for route request packet, we have added the certificate of the node. This contains four fields. Now route request packet has total of

246 bits. The request reply packet for the secure DSR will be the same as the request reply packet of regular DSR.

Again from Chapter 4, upon discovering the path from source to destination, source node will begin sending actual messages to the destination. We assume that each packet being transmitted is going to have 512-bit size. For regular DSR, the 512-bit size packets will be sent over network. However, for secure DSR, we will add more extra fields for security in addition to a 512-bit size packet. Actual message formats for both regular and secure DSRs will be as follows:

Actual message format for regular DSR:

512 bits for actual message	IPs 32 bits	IPd 32 bits
-----------------------------	----------------	----------------

Actual message format for secure DSR

512 bits for actual message	IPs 32 bits	IPd 32 bits	Cert S 171 bits
-----------------------------	----------------	----------------	--------------------

The way we define overhead for route discovery process will be based on the analytical formula derived by in [4]. As mentioned in the previous chapters particularly in chapters 2 and 4, source node will initiate route discovery process in order to reach the destination. The source node, however, continuously broadcasts route discovery packets till it finds a

node that has communicated with the destination previously. Therefore, it will be misleading to calculate overhead simply based on number of bits transmitted.

The analytical formula developed by the authors of [4] is as follows:

The cost of a single route discovery is defined as:

$$1 + FwReq + OgRep + FwRep$$

where 1 represents the transmission of the original request, FwReq is the number of route requests forwarded, OgRep is the number of route reply originations, and FwRep is the number of route replies forwarded. For a single route discovery process, this metric measures the number of routing packets (requests and replies) that were transmitted to complete the discovery.

The cost equation above measures the cost in terms of messages transmitted. We extend to include the number of bits transmitted to measure overhead incurred. We assume that both request and reply messages are of the same size. This, however, is not strictly true in practice. Assuming each message carries  $n$  bits.

$$n.(1 + FwReq + OgRep + FwRep)$$

This formula actually can be considered for regular DSR on which no security mechanism is implemented and thus no extra bits resulting from security application. We can also modify this formula for secure DSR on which security is implemented and hence there will be extra bits due to implementing certificate and private key of the source node.



If  $x$  represents extra bits added in virtue of security mechanism, the formula defined above can be modified for secure DSR as:

$$(n + x) (1 + FwReq + OgRep + FwRep)$$

Now we can define overhead with percentage term as follows:

$$\frac{(n + x) (1 + FwReq + OgRep + FwRep) - n.(1 + FwReq + OgRep + FwRep)}{n.(1 + FwReq + OgRep + FwRep)} * 100$$

$$\frac{(1 + FwReq + OgRep + FwRep) (n + x - n)}{n.(1 + FwReq + OgRep + FwRep)} * 100$$

If we solve the formula, we will have:

$$\text{Overhead} = (x / n) * 100$$

If we use actual values from our work, extra bits being used for security mechanism is 171. Total number of bits for both route discovery and route reply packets is 190. If we replace these numbers on the formula, overhead value will be 90%. That is, we are sending 90% more in terms of bits transmitted by applying security mechanism on regular DSR. In other words, we are incurring 90% more cost with respect to bits being transmitted by virtue of implementing security mechanism on regular DSR for route discovery process.

### 6.2.2 Sending Actual message process

Once route is discovered, source node will be sending actual messages. We have assumed that each packet has a size of 512 bits. This time the path is known and the message will be sent on discovered route. The formula we have used for route discovery is:

$$(1 + FwReq + OgRep + FwRep)$$

Since we do not have any reply messages for actual message transmission,  $OgRep = FwRep = 0$ . That leaves the formula  $(1 + FwReq)$ .  $FwReq$  is the number of forwarded requests. Because we do not have requests, let us call it forwarded messages ( $FwMes$ ). The equation is therefore  $(1 + FwMes)$ . For example: Assume that there are five nodes (hops) in the discovered path. The first message is from node 1 to node 2, node 1 is the source node. From then on messages are forwarded from node 2 to 3, from 3 to 4, from 4 to 5. Therefore,  $FwMes = 3$ .  $1 + FwMes = 1 + 3 = 4$

The following formulas will be used for a packet being transmitted over network for regular and secure DSRs.

#### For Regular DSR

$$\text{Total} (Total \# \text{ of bits transmitted}) = (h-1) * tp = (1 + FwMes) * tp$$

$h$ : number of hops from source to destination taken from distribution

$tp$ : total packet size

## For Secure DSR

$$\text{Total2 (Total \# of bits transmitted)} = (1 + FwMes) * tp$$

Upon implementing security over actual message, in addition to actual message size, the whole packet will have 171 extra bits due to introducing security. As seen from the Figure 10, the number of bits transmitted swells up to 40 nodes. Then, however, we see drop in 50 nodes then almost no change (only slight fluctuation) for the number of bits transmitted as number of nodes is going up to 100 nodes.

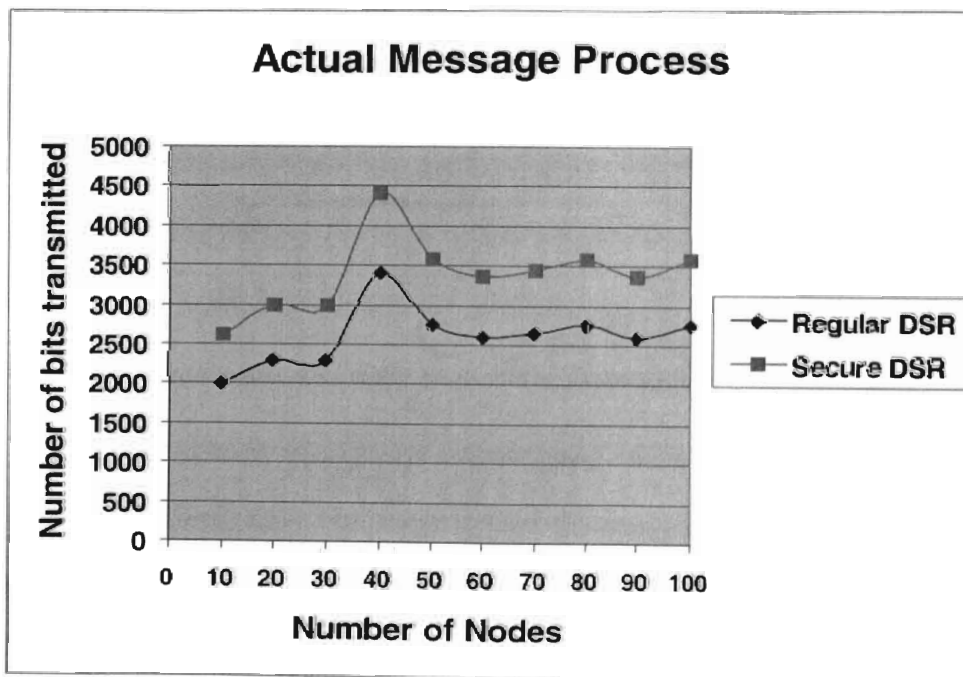


Figure 10. Number of Nodes vs. Number of bits transmitted for actual message

If we define overhead with percentage term here, overhead will be fixed regardless number of hops.

Overhead = (# of bits transmitted with regular DSR / # of bits transmitted with secure DSR) \* 100 or

Overhead = (x / n) \* 100

Apparently, percentage overhead value will be fixed number regardless number of hops. For our work, the extra bits used for security mechanism is 171 coming from certificate. Total number of bits used for actual message is 576. If we replace these values on the formula, overhead will be around 29.6%. That is, we are incurring extra 29.6% cost in term of transmitting bits because of implementing security mechanism on regular DSR for sending actual messages.

### 6.3 SIMULATING SECURE GROUP COMMUNICATION

Upon discovering a path, we randomly choose any node(except source and destination nodes) among the nodes involved into discovered path. If the chosen node changes its own group, and there is no re-keying mechanism established, it will be able to read the messages between the nodes in both groups: the group it is leaving and the new group it is joining. If we fix group size while population size is increasing, the number of messages the node can read without re-keying is going to be fixed number regardless of population size.

We can consider basically three scenarios: without re-keying, with half re-keying, and with full re-keying. Theoretically, without re-keying, the leaving/joining node will read all the messages within two groups, say, x number of messages. Likewise, with full re-

keying, the node will be unable to read any message within both groups. However, we should keep in our mind that there is going to take sometime to implement re-keying mechanism. Thus, a node may be able to read messages while it is in transit between two groups. Apparently, with half re-keying, the message read by the node basically will be  $x/2$ . For half re-keying, as soon as a node wants to leave a certain group, the trusted server for that group the group it is leaving will immediately apply re-keying mechanism. In other words, the leaving node is unable to read any message from the group from which it is leaving. Thus, it can only read the messages from the new group it is joining. That means that there is no re-keying in the new group it is joining apart from the one message to inform the joining key. The new group server basically assigns the current group key for the new coming node. Hence, the number of messages it can read basically half of the messages ( $x/2$ ) it used to read without re-keying. Similarly, when we implement full re-keying mechanism, the node in theory is not supposed to read any messages from both groups. Because; we know that when a node wants to leave one group and wants to join another one, trusted servers for each group will change the group key immediately. However, in reality, assigning new group key does not happen at the same time for both groups. There is always time lag when a node leaves a group and joins into new group. Therefore, the number of messages the node can read with full re-keying will not be zero, though we have shown zero number of messages read for full re-keying on Figure 11. So that it will be able to read a few messages depending on the speed of re-keying. As shown on the figure below, the number of messages that can be read by the leaving/joining node will be fixed for half and full re-keying as population size is going up. Our simulation has shown very close results what is shown below

theoretically for the fixed group size and different population sizes. We have used group size 10. The number of messages read will be the same as number of nodes. So that without re-keying, 10 messages will be read, 5 messages will be read with half re-keying. Apparently, if we have group size 20, there will be 20 messages read without re-keying starting from population size 20 and 10 messages will be read for half re-keying, and so on.

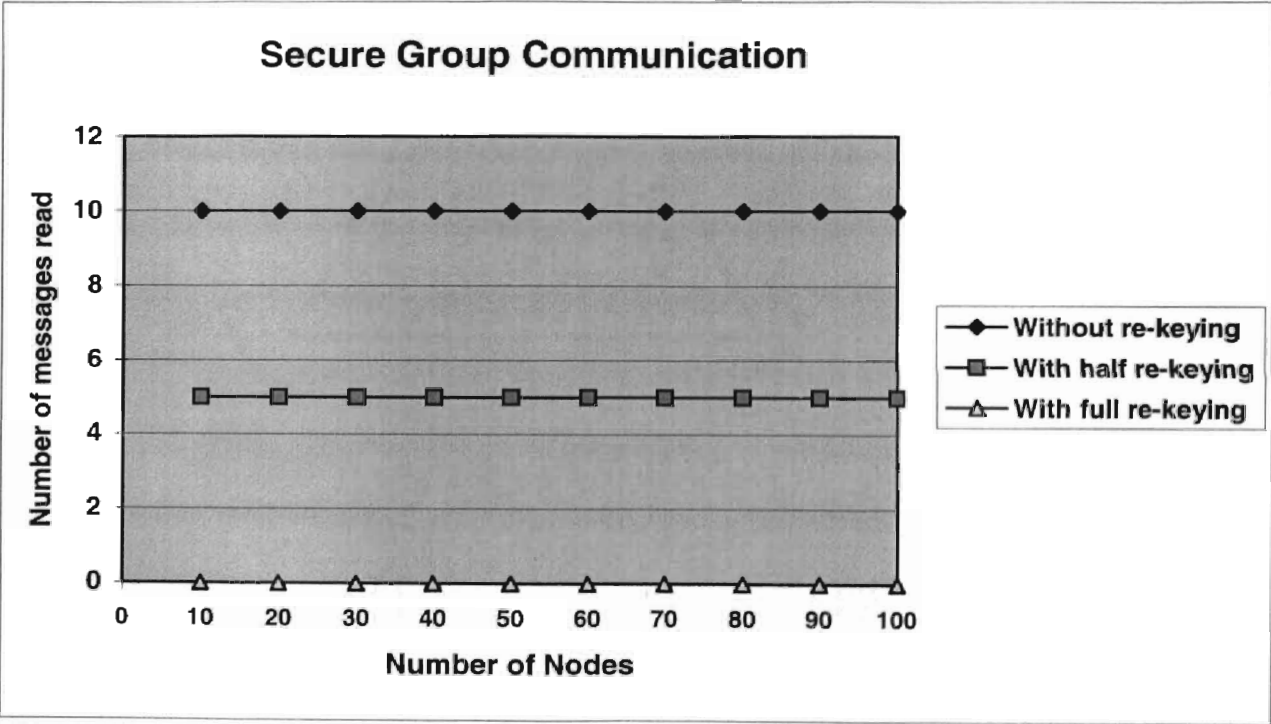


Figure 11. Number of Nodes vs. Number of messages read for different re-keying mechanisms.

### 6.3.1 Overhead for re-keying

Upon applying re-keying security mechanism over group communication, obviously we will have extra messages providing security. The way we calculate overhead is that we have to have at least two groups in each population. The number of group members in both groups will be fixed while population size is increasing. We know the number of hops for each population size from our distribution.

First of all, we will have four fixed messages for each re-keying. Once a node wants to leave a group, it will ask trusted server for permission to leave. The server will send a message back allowing the node to leave. Now, the node will ask to another trusted server to join its group. The new server will send a granting message back allowing the node to join into new group. Therefore, there will be fixed four messages. In addition, there will be re-keying messages for each population size depending upon number of hops. Thus, overhead value can be calculated as follows:

$$\text{Overhead} = \text{BG} \cdot h + 4 = \text{BG} \cdot (1 + \text{FwMes}) + 4$$

BG: Number of nodes in both groups (total nodes in both groups)

h: Number of hops taken from distribution

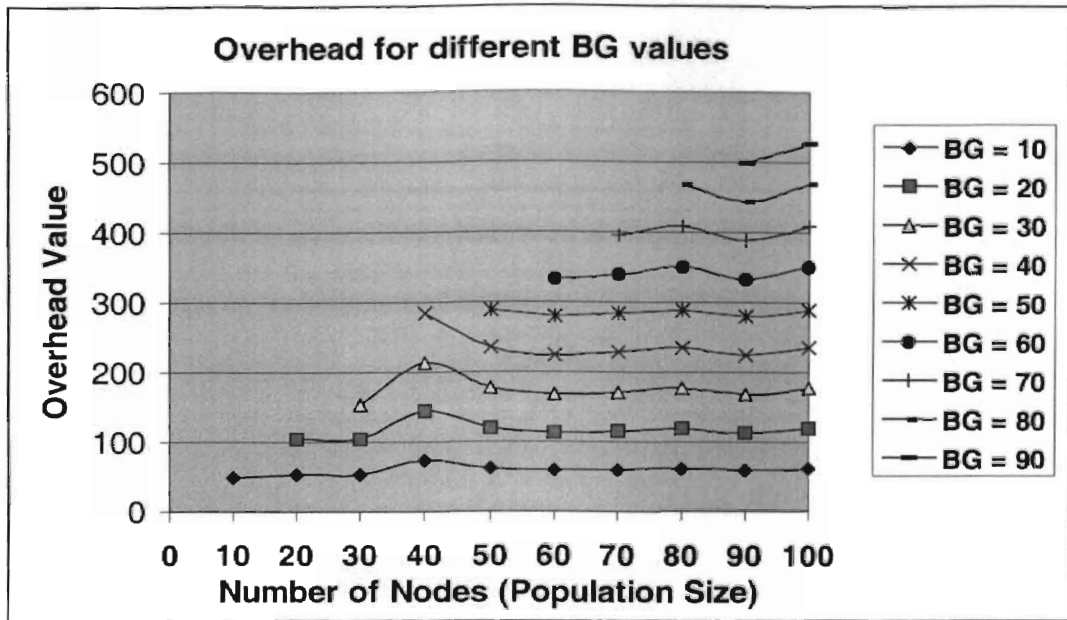


Figure 12. Overhead for different BG values

As seen on the figure above, we have fixed total number of nodes in both groups represented by BG from 10 to 90, though population size is increasing. We do not have linear relationship. Up to 40 nodes, overhead value is swelling then it is almost constant, though there exists some fluctuations. The basic question we are asking is that what is overhead value due to introducing re-keying mechanism. By doing this, however, we fix the number of nodes in both groups as increasing population size from 10 to 100.

### 6.3.2 Efficiency for re-keying

The efficiency of the network will be basically based on overhead for each population size with fixed number of group size (fixed number of BG). Our actual message format



for secure DSR consists of  $tp$  bits. If we use this message format, efficiency can be described as:

$$E(\text{Efficiency}) = tp / (tp + \text{Overhead})$$

If we take  $BG = 10$ , overhead values and efficiency values as percentage for each population size are given in Table 2. Since we will use actual message format for secure DSR,  $tp = 747$ .

Population Size	Overhead value for $BG = 10$	Efficiency(%)
10	49	93.8
20	54	93.2
30	54	93.2
40	74	90.9
50	62	92.3
60	59	92.5
70	60	92.4
80	62	92.3
90	59	92.5
100	62	92.3

Table 2. Overhead and Efficiency values for different population sizes

Efficiency value apparently is the highest when overhead is the smallest. Likewise, efficiency is the lowest, when overhead is the highest.

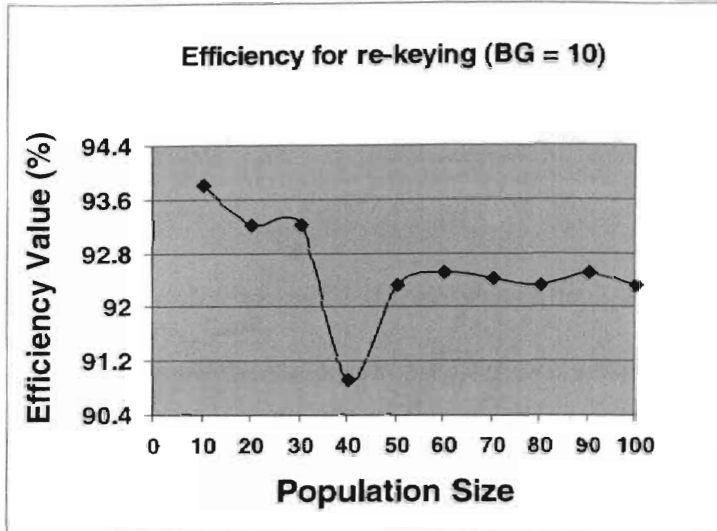


Figure 13. Efficiency for re-keying

## REFERENCES

- [1] Andre Weimerkskirch, Gilles Thonet. *A Distributed Light-Weight Authentication Model for Ad-Hoc networks*. The 4th International Conference on Information Security and Cryptology (ICISC 2001), December, 2001, Seoul. pp. 341-355
- [2] Bridget Dahill, Brain Neil Levine, Elizabeth Royer, Clay Shields. *A Secure Routing Protocol for Ad Hoc Networks*. Technical Report UM-CS-2001-037, Electrical Engineering and Computer Science, University of Michigan, August 2001.
- [3] Chung Kei Wong, Mohamed Gouda, Simon S. Lam. *Secure Group Communications Using Key Graphs*. February 2000. IEEE/ACM Transactions on Networking. Vol. 8 No.1. pp. 16-30.
- [4] David B. Johnson, and David A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pp. 153-181. Kluwer Academic Publishers, 1996.
- [5] David A. Maltz, Josh Broch, and David B. Johnson. *The Effects of On-Demand Behavior in Routing Protocols for Multi-hop Wireless Ad Hoc networks*. IEEE Journal on Selected Areas in Communications, 17 (8):1439-1453, August 1999.
- [6] DeCleene, L., Donte, T., Hardjono, D., Toesley, S. *Secure Group Communications for Wireless Networks*. February 2001. In *Proceedings of IEEE MILCOM 2001*, Mclean, VA, October 2001. pp. 113-117
- [7] Lidong Zhou, Zygmunt J. Haas. *Securing Ad Hoc Networks*. November/December 1999. IEEE Network, Special Issue on Network Security. Vol.13. No. 6. pp. 24 -30
- [8] Multiparty Computations – The Future in High-End Security. URL: <http://www>.

cryptomathic.com/company/multiparty.html. Ivan Damgaard, Chief Cryptographer, Cryptomathic Inc.

- [9] Panagiotis Papadimitratos, Zigmunt J. Haas. *Secure Routing for Mobile Ad Hoc Networks*. In SCS Communication Networks and Distributed Systems Modeling and Conference (CNDS), January 2002.
- [10] Ram Ramanathan, Jason Redi. *A Brief Overview of Ad Hoc Networks: Challenges and Directions*. IEEE Communications Magazine. 50<sup>th</sup> Anniversary Commemorative Issue/May 2002. pp. 20 - 22
- [11] Schneider, Gary P., Perry, James T. *Electronic Commerce*. 2001. Second Edition. Course Technology, Boston, MA.
- [12] Seung Yi, Prasad Naldurg, Robin Kravets. *Security-Aware Ad-Hoc Routing for Wireless Networks*. Technical Report UIUCDCS-R-2001-2241, Department of Computer Science, University of Illinois at Urbana-Champaign, August 2001.
- [13] Threshold Cryptography. URL: <http://www.cs.fsu.edu/~desdedt/topics-threshold.html>. Florida State University, Computer Science Department.
- [14] Vesa Karpijoki. *Signaling and Routing Security in Mobile and Ad-Hoc Networks*. May 2002. URL: [http://www.tcm.hut.fi/Opinnot/2000/papers/signaling\\_security](http://www.tcm.hut.fi/Opinnot/2000/papers/signaling_security)
- [15] Yih-Chun Hu, Adrian Perrig, David B. Johnson. *Aridna: A secure On-Demand Routing Protocol for Ad Hoc Networks*. Proceedings of the Mobicom'02 Conference. September 2002. Atlanta, GA, USA. pp. 12-23
- [16] Yih-Chun Hu, David B, Johnson, Adrian Perrig. *SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks*. June 2002. 4<sup>th</sup> IEEE Workshop on Mobile Computing Systems & Applications. pp. 3-13

2

VITA

Sabri Er

Candidate for the Degree of

Master of Science

Thesis: SECURE HIERARCHICAL GROUP COMMUNICATION ON MOBILE AD  
HOC NETWORKS

Major Field: Computer Science

Biographical:

Personal Data: Born in Ankara, Turkey, second oldest child of Mufide and Veyis.

Education: Received Bachelor of Science degree Agricultural Engineering from Ankara University, Ankara, Turkey in June 1986. Received Master of Science Degree in Agricultural Economics from Reading University, Reading, England in September 1994. Completed the requirements for the Master of Science degree in Computer Science at Oklahoma State University in August 2003.

Experience: Employed as an Engineer by the Department of Agriculture of Turkey from June 1987 to may 1997. Employed as Teaching Assistant by the Computer Science Department from January 2000 to present.