# DYNAMIC KEY MANAGEMENT

# FOR SECURE ROUTING IN LCMRMG

THESIS

Presented to the Graduate Council of
Texas State University-San Marcos
in Partial Fulfillment of
the Requirements

For the Degree of

Master of SCIENCE

By

YALIN WANG, B.S., M.S.

San Marcos, Texas
December 2003

## ACKNOWLEDGMENTS

First and foremost, my grateful thanks go to Dr. Wuxu Peng, the supervisor of my master's thesis. His invaluable and patient guidance, enthusiastic encouragements and support, accompanied me in every stage of my thesis research. What I learned from him will provide me with lifetime of benefits. I consider myself lucky to have accessed his supervision.

I would like to thank Dr. Carol Hazelwood, for her valuable suggestions on my thesis work.

Finally, I want to express my deepest feeling to my family. My special thanks go to my parents. Without their love and encouragement, I would not pursue new goals again and again.

# TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

# TABLE OF ACRONYMS

| | |
|---|---|
| ***AODV*** | Ad Hoc on Demand Distance Vector routing |
| ***CA*** | Certificate Authority |
| ***DSDV*** | Distance Vector Routing protocol |
| ***DSR*** | Dynamic Source Routing |
| ***GPS*** | Global Positioning System |
| ***LCMRMG*** | Locality Caching Multi-Root Multi-Generation Routing |
| ***MANET*** | Mobile Ad Hoc Network |
| ***MD5*** | Message Digest 5 algorithm |
| ***OLSR*** | Optimized Link State Routing |
| ***PGP*** | Pretty Good Privacy |
| ***PKI*** | Public Key Infrastructure |
| ***RSA*** | A public key cipher algorithm |
| ***TBRPF*** | Topology Broadcast based on Reverse-Path Forwarding |
| ***WLAN*** | Wide Area Network |

# ABSTRACT

Ad hoc mobile network (MANET) is characterized by its lack of fixed physical and administrative infrastructures (routers, server and stable communication links). As such MANET presents unparallel challenges in security. This thesis studies the threats that an ad hoc network faces and their security goals. In particular it proposes a security algorithm that is integrated into a tree-structured MANET routing protocol. Specifically the thesis presents a fully distributed multi-root key management scheme for a locality caching multi-root multi-generation (LCMRMG) MANET routing algorithm.

The major problem in providing security services in such infrastructure-less networks is how to manage the needed cryptographic keys. In order to design practical and efficient key management systems, we take advantage of the multi-root nature of LCMRMG and successfully integrate public key, certification and threshold technologies in our algorithm. Our simulation has proven that the proposed security algorithm is feasible for LCMRMG: it is secure, effective and scalable.

# Chapter 1 INTRODUCTION

## 1.1 Background

During the past decade, with the fast growth of wireless network, mobile ad-hoc network (MANET) has become a focal point of wireless computing. MANET began its applications mainly in military related networks. However with advances of computer network research and the availability of wireless technologies such as Bluetooth [1] and 802.11[2], MANET has become increasingly important in mobile/wireless communications.

Security is an important issue for ad hoc networks, especially for security-sensitive applications. For example, a military mobile ad hoc network certainly will need to secure the network to achieve confidentiality and to resist various types of service attacks.

Ad hoc network routing protocols are challenging to design, and providing viable security mechanisms are even more so [3]. However, researchers in ad hoc networking have generally studied the routing problem in a non-adversarial network setting, assuming a trusted environment. Relatively little research has been done in a more realistic setting in which an adversary may attempt to disrupt the communication [3] [7].

Several proposals exist regarding security issues in MANET. Generally, security goals in the research are achieved through cryptographic mechanisms such as public key encryption or digital signature [14][28][20]. However, most of them are still ongoing research topics and there exist many unsettled issues.

## 1.2 Motivation

How to manage the cryptographic keys in a MANET is one of the major challenges. In order to design an efficient and practical MANET routing protocol, we need carefully consider and redesign the traditional key management system.

The goal of this thesis is to study security issue in MANET that is applicable for a particular MANET routing algorithm – the LCMRMG routing protocol. It proposes an efficient key management scheme for LCMRMG. Specific questions that the thesis has answered include:

(1) To integrate the multi-root threshold public key management into LCMRMG routing protocol, how can we provide the security service while maintaining the network's availability and efficiency?

(2) Can we reduce handoff latency without triggering high ad hoc routing overhead?

(3) How much efficiency gain is obtained in the use of multi-roots?

### 1.3 Thesis Contribution

In this thesis, we present a novel scheme to integrate key management with the LCMRMG routing protocol. The main contributions of this thesis are:

- Survey of what has been done in MANET security, especially related to the issue of key management.
- Design of a novel scheme to integrate key management with the LCMRMG.
- Implementation of the proposed scheme based on the RSA implementation [], Threshold implementation [] and digital certificate [].
- Design and construction of a test-bed to verify the functionality of the scheme.
- Evaluation of performance aspect of the scheme. A set of performance benchmarks are designed and explored to assess the effects of various factors on the integration scheme.
- Investigation of efficiency gains obtained in applying the proposed scheme to LCMRMG over the corresponding single-root MANET routing algorithm.

### 1.4 Road Map

This thesis is structured as follows.

Chapter 1 briefly introduces the background of the thesis and presents motivations. Since the proposed scheme is being used in the existing LCMRMG protocol, Chapter 2 provides background concerning MANET and its routing protocol, including LCMRMG routing protocol. Chapter 3 describes the basic security algorithms we are using in securing LCMRMG. In Chapter 4, security in MANET is characterized and current works are reviewed and discussed. Chapter 5 is the core of the thesis and it presents our key management scheme for LCMRMG in detail. Chapter 6 briefly describes the simulation implementation of the proposed key management scheme. Chapter 7 analyzes performance aspects of the proposed scheme through simulation. Finally chapter 8 provides concluding remarks and also discusses possible future work.

# Chapter 2 MANET NETWORK ROUTING

## 2.1 MANET Introduction

There are currently two variations of mobile wireless networks. The first type of mobile wireless network is known as an infrastructure network, i.e., a network with fixed and wired gateways. A mobile unit within these networks connects to the nearest base station that is within its communication radius. As the mobile device travels out of range of one base station and into the range of another, a handoff occurs from the previous base station to the new one, and the mobile device can continue communication seamlessly throughout the network. Typical applications of this type of network include wireless local area networks (WLANs).

The second type of wireless network, Mobile ad hoc network (MANET) [27] has been proposed to support dynamic scenarios where no wired infrastructure exists. MANET is characterized by:

(1) Wireless links with broadcasting. A MANET host broadcast its messages to all neighbors;

(2) Infrastructure-less. Each node in a MANET has to function both as a host and a router, and the network is vulnerable without physical protection;

(3) Dynamically changing topology. Every node can be on the move at any moment. Connections among MANET nodes can be arbitrary at any particular time instance. Also, nodes could join or leave the network at any time. These present additional difficulties for trust relation to build up security mechanism and detect intrusion;

(4) Fully distributed. This is the key property of MANET, since it lacks of centralized control to provide routing and key management;

(5) Limited resources [28]. MANET nodes usually have low-power microprocessors, limited battery power, small memory, and limited bandwidth. This requires application-specific trade-off between the needed security level and available resources. Such a trade-off can often make MANET more subject to DoS (denial of service) attack as well.

Some examples of the possible applications of MANET include participants sharing information in a meeting, soldiers relaying information on battlefields, emergency disaster relief personnel coordinating efforts after a hurricane or earthquake, team member communications in a infrastructure-less remote area.

## 2.2 Important MANET Routing Protocols and LCMRMG

MANET routing protocols can generally be categorized as *on-demand routing protocols* or *table-driven routing protocols* [27]. To better understand LCMRMG routing, firstly, in the following sub-sections, we briefly review some

popular MANET routing protocols and categorize them according to their characteristics.

### 2.2.1 Table-Driven Routing Protocols

The well-known table-driven routing protocols are Destination Sequenced Distance Vector Routing protocol (DSDV) [29], Optimized Link State Routing (OLSR) [30] and Topology Broadcast based on Reverse-Path Forwarding (TBRPF) [31].

These table-driven routing protocols build routes in a proactive way between nodes in a MANET. Similar to the routing in infrastructure networks, route tables are created and maintained by each node to store consistent, up-to-date routing information for all other nodes. Nodes respond to the changes in the network topology by propagating update packets throughout the network. In each node, the routing information in route tables are updated and maintained according to the view of the whole network structure. The main characteristics that differentiate a table-driven routing protocol from another are the number of necessary route tables and the way network topology changes.

### 2.2.2 On-Demand Routing Protocols

Two well-known routing protocols, Ad Hoc on Demand Distance Vector (AODV) Routing [32] and Dynamic Source Routing (DSR)[33] are falling into

on-demand ad hoc routing protocols.

On-demand routing is the most recent entry in the class of scalable wireless routing schemes. This type of routing creates routes only when a route to the destination is needed. In a MANET running on-demand routing protocol, when a node wants to send a packet to a destination, it initiates a route discovery process to find a route to the destination. After a route is built, it is maintained through a route maintenance process. If any link on the route is broken or the route is no longer desired, the route is deleted. Compared with table-driven routing protocols, on-demand routing protocols may have lower computation costs and lower packet overhead since they do not need to exchange routing information periodically and maintain route tables. However, when a node using an on-demand routing protocol desires to send a packet to a destination, it has to wait until a route to the destination is discovered on-demand. This feature of on-demand routing protocols results in longer packet transfer delay than with table-driven routing protocols.

### 2.2.3 LCMRMG Routing Protocol

The LCMRMG routing protocol [34] is extended from the single spanning tree routing algorithm proposed by Chen and Jia in [35]. The fundamental idea of LCMRMG routing is the following. A MANET first builds a spanning tree and a generation table for each station. During the operation of the network, multi roots are generated on demand. Therefore the whole

MANET can consist of multi spanning trees with multi generation tables. Figure 1 shows the "mesh" of this kind of topology, note that in general some nodes may be in more than one spinning tree.

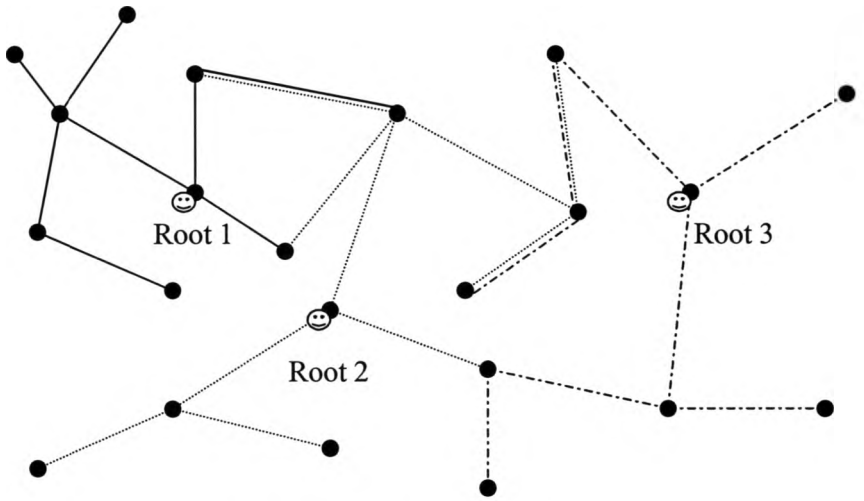


Figure1 LCMRMG Routing

Each spanning tree consists of a single root, which could be any node within the network. Hosts in a LCMRMG MANET are required to cache the locality of network routing traffic through them. Based on the cached locality statistics, a host can calculate the estimated reduction of network traffic if it becomes a new root router and can elect to do so if the reduction exceeds a predefined threshold value. With these multiple roots, the whole network topology information is distributed to multiple generation trees.

Because each node only keeps a small generation table instead of the global routing table, and the sign-on procedure guarantee a node to sign on to

the lowest generation possible, the LCMRMG protocol guarantees an optimal route from any source to any destination [34] while keeping network scalable and effective.

# Chapter 3 CRYPTOGRAPHIC PRIMITIVE

The security mechanisms studied in this thesis are based on the following well-known techniques: *symmetric encryption, public key encryption, Message Digest, digital signature, digital certificate and secret share*. They provide the basic "bricks" for our proposed key management scheme.

To better present our scheme, the five classic security services (*authentication, integrity, non-repudiation, availability and Confidentiality*)[36] are briefly discussed in Section 4.1.

## 3.1 Symmetric Encryption

In symmetric encryption the message can be encrypted and decrypted using the same key. The key must be kept secret, and is shared by the message sender and recipient.

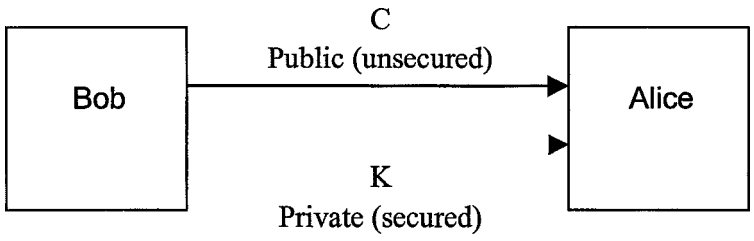


Figure2 Symmetric Encryption

Assume that Bob want to send to Alice something (Plain text, m) private

that she doesn't want to disclose to anyone else (Figure2).

(1) Bob will select a secret key k and encrypt E (k) the plaintext to get the cipher text(c). By any secret means bob will let Alice know the secret key.

(2) In an unsecured channel, Bob send the encrypted cipher text to Alice.

(3) Alice then uses the secret key to decrypt the cipher text to the plaintext.

Compared to asymmetric encryption (to be introduced shortly), the symmetric encryption is faster. But it also has two serious disadvantages: (a) both the sender and the recipient must have access to the same encryption key. In other words, secure distribution of the (encryption) key between the parties is required, and (b) simple symmetric encryption doesn't provide the other aspects of data security, including authentication and integrity.

### 3.2 Asymmetric Public Key Encryption

In symmetric encryption, a new key is used for each encryption, thus there's one key per encryption, the number of keys required to provide secure communications among those users' increases rapidly. For example, a network of 100 users would require almost 5000 keys if it used only symmetric cryptography. Doubling such a network to 200 users increases the number of keys to almost 20,000.

Thus, if we only use symmetric cryptography, key management quickly becomes unwieldy even for relatively small-scale networks. That is also one reason that we choose public key encryption for our system.

In asymmetric public key encryption, there are *two keys for per person.* Each person using public key cryptography, PKC, has a pair of different but mathematically related keys, called a key pair. Anything that's encrypted with one of the keys must be decrypted with the other. This is public key cryptography's major innovation. This scheme is illustrated in Figure3.

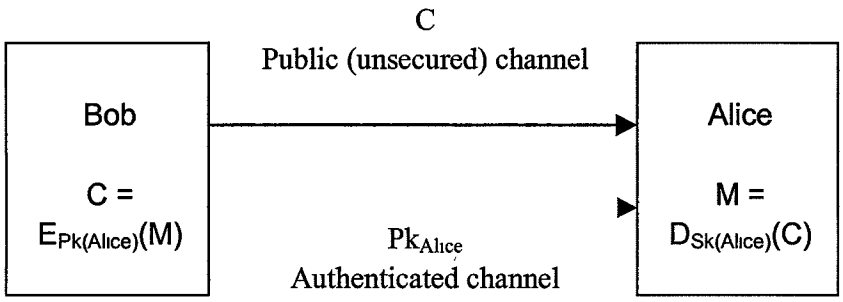


Figure3 Public Key Encryption Scheme

(1) Alice is using public key, so she has Pk/Sk pair. The Pk is distributed publicly so that everyone including Bob knows this Pk; the Sk is kept by Alice to herself so that no one else knows it.

(2) Bob want to send message M to Alice; he will encrypt the plaintext M into cipher text C using Alice's Pk, he will then just send out C to Alice.

(3) Upon receiving C, Alice can use her own Sk to decrypt it back to plaintext M. Since only she knows the Sk, nobody else could decrypt the cipher text.

(4) The process also works backwards. Alice could encrypt a plaintext with her private key and send the resulting cipher text to Bob. Decrypting the cipher text with Alice's public key proves that the cipher text has to come from Alice. This provides authenticity.

Public key encryption can also provide non-repudiation along with confidentiality, integrity and authentication. However, public key encryption requires much more computational resources than symmetric encryption and therefore could degrade the network performance. Therefore public key encryption is typically only used to encrypt small amounts of data, e.g. symmetric encryption keys and digital signatures.

### 3.3 RSA

This method of cryptography was developed in 1977 by three mathematicians: Ron Rivest, Adi Shamir, and Len Adleman[37], hence the acronym; RSA. It is a public key encryption algorithm that can be used to provide confidentiality, integrity, authentication and non-repudiation services.

The RSA key generator G produces two large random primes $p_1$ and $p_2$, and computes $n = p_1 * p_2$ and the Euler toitent function $\varphi(n) = (p_1-1)(p_2-1)$. To

compute the secret key d, the generator chooses a random d such that *gcd (d, φ (n)) = 1*. The public key is n and e where $ed \equiv 1 \bmod \varphi(n)$. The one way (hard) direction of RSA is $S_m \equiv m^d \bmod n$, whereas the public direction is $(S_m)^e \bmod n$ which gives m.

To encrypt a message *m* or decrypt a cipher text *c*, the following calculations are performed:

(1) If the algorithm is intended to be used to provide confidentiality the values *n* and *e* are made publicly known while *d* is kept secret. Therefore the public key $Pk = \{e, n\}$ and the private key $Sk = \{d, n\}$.

(2) For Alice to encrypt a message intended for Bob, B's public key $Pk_{Bob}$ is used for the encryption: $c = E_{Pk\text{-}Bob}(m) = m^e \bmod n$.

(3) Since only Bob has knowledge of the secret key $Sk_{Bob}$, it alone can decrypt the cipher text and recorver the plain text by decryption: $m = D_{Sk\text{-}bob}(C) = C^d \bmod n = m^{ed} \bmod n$.

### 3.4 Digital Signature

In Section 3.2, Step (4) provides authentication at the price of with no privacy. This is because that Alice's public key is public, so anyone could decrypt this ciphertext, not just Bob. In public key cryptography, Digital signatures will provide authentication (that Alice is actually the one who sends the original text), and integrity (that the message has not been altered in transit or in storage), without sacrificing privacy.

A digital signature is a data structure that provides proof of origin, i.e. authentication and integrity, and depending on how it is used, it can also provide non-repudiation. Figure 4 illustrates how a digital signature is used.

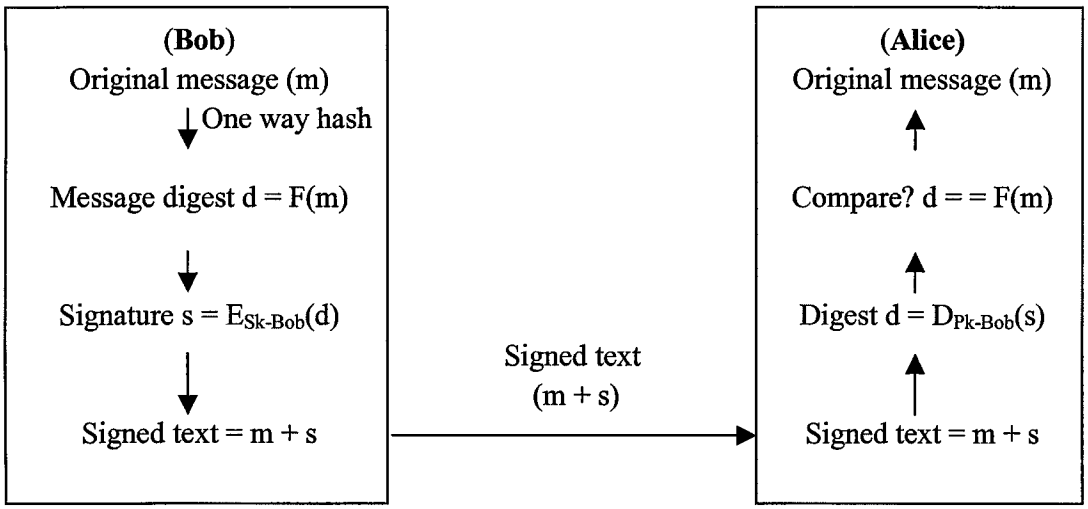


Figure4 Digital Signature

(1)Bob uses his private key to digitally sign the original text to:

(a) Calculate a message digest, a short, one-way digest of m

(b) Encrypt the message digest with Bob's private key.

The calculated and encrypted message digest is the digital signature; it is attached to or stored with the original text, forming the signed text.

(c) Send the signed text to Alice.

(2) Alice uses Bob's public key to verify the signature. She:

(1) Decrypts the digital signature, the message digest that was encrypted with the signer's private key; She uses Bob's public key to do this.

(2) Calculates a message digest from the text of the message as you received it.

(3) Compares the two message digests.

If the message digest that Bob sent out and the one that Alice calculates match, it proves two things:

- That the original text hasn't been changed since it was signed
- That the message digest Alice received was really produced by Bob, whose private key was used to sign it. (Otherwise, Bob's public key would not have successfully decrypted the message digest he sent.)

The digital signature also provides non-repudiation -- the signer, in this case Bob, cannot deny being the source of the original text and the receiver, further more, Alice, can also prove that she didn't modify the text after she received it.

**3.5 MD5**

In the above digital signature example, Bob uses something called a "hashing algorithm" to transform a message of any size into a string of numbers of a fixed (and usually smaller) size. If the hashing algorithm is designed properly (and there are a number in current use), different messages should not produce the same hash. In our proposal, we are using MD5 algorithm.

MD5 is a secure one way hash function which was developed by Ronald L. Rivest (http://theory.lcs.mit.edu/~rivest/homepage.html). It takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA. In essence, MD5 is a way to verify data integrity, and is much more reliable than checksum and many other commonly used methods.

### 3.6 Digital Certificate

Digital certificates are used to prevent the type of attack described above. Basically a digital certificate is a statement issued by some trusted party saying that it verifies that the public key *pkA* in fact belongs to the user A. The trusted party digitally signs this statement and therefore anyone with the authentic public key of the trusted party can verify the certificate

and thereafter use *pkA* and be sufficiently sure that it actually belongs to node A.

Table 1 shows the normal fields in an X.509 certificate [38].

| Serial Number |
|---|
| Issuer Name |
| Validity Period |
| Subject Name |
| Public Key Information |
| Key Usage |
| Extensions |
| Certificate Authority Digital Signature |

Table 1 Typical X.509 Certificate Format

### 3.7 Threshold Secret Sharing

This scheme was proposed by Adi Shamir[26] based on polynomial interpolation. This (k,n) threshold secret share scheme allows a secret(in our case, the network private key, Sk), to be shared among k users in such a way that no single user, or users less than k, can deduce the secret from his(their) shares alone. Only by combining at least k number of shares can the secret be reconstructed. The following steps show how this works:

1. A prime p is chosen such that $p > \max(S, n)$; (S means the secret)
2. A polynomial $f(x) = a_0 + a_1 x + ... + a_{k-1} x^{k-1}$ is generated where $a_0 = S$ and

$a_i$, i = 1,...,k-1 are chosen randomly from $Z_p$

3. The additive secret shares $S_i, i = 1,...,k-1$ are generated as $Si = f(id_i)(\mathrm{mod}\ p)$
4. The shares are securely distributed to the respective shareholders.

To reconstruct the secret, Lagrange interpolation is used. With the knowledge of a minimum of *k* shares the polynomial *f(x)* can be reconstructed and the secret recovered by calculating *f(0)*. The Lagrange interpolation is described below:

$$f(x) = \sum_{i=1}^{k} S_i \square_{id_i}(x)(\mathrm{mod}\ p)\ where\ l_{id_i}(x) = \prod_{j=1, j\neq i}^{k} \frac{x - id_j}{id_i - id_j}$$

### 3.8 Proactive Secret Sharing

As mentioned in [20], adversaries can be characterized as at least two models: Model-1, adversaries cannot comprise k or more nodes during the entire time of whole network; Model-2, adversaries cannot comprise k or more nodes during time interval T. Regular secret sharing can defend model-1 adversaries, but can not defend model-2 adversaries. As described in detail in Chapter 5, that is why we are using proactive secret sharing in our security scheme along with regular secret sharing.

The proactive secret sharing scheme updates the shares in a regular basis. Since only the shares belonging to same update period can reconstruct

the secret, model-2 attackers must then compromise at least k shares between the update intervals.

By Hertzberg's proposal [39], a round of share updates is achieved by adding a random update polynomial $f_{update}(x)$ to the original sharing polynomial f(x) as follows:

$$f(x) = a_0 + a_1 x + ... + a_{k-1} x^{k-1} (\mathrm{mod}\ p)$$

$$f_{update}(x) = b_1 x + b_2 x_2 + ... b_{k-1} x^{k-1} (\mathrm{mod}\ p)$$

$$f_{new}(x) = f(x) + f_{update}(x)$$
$$= a_0 + (a_1 + b_1)x + ... + (a_{k-1} + b_{k-1}) x^{k-1} (\mathrm{mod}\ p)$$

In our key management system, the shares of update polynomial ($S_i'$) are calculated and then distributed to respective share holders. Each shareholder then adds it to old share to obtain the updated share, $S_{i\text{-}new} = (S_i' + S_{i\text{-}old})$ mod p.

### 3.9 PKI

PKI stands for public key infrastructure. Each public key must be, in some way, labeled authoritatively with its owner's name and there has to be some way of reliably getting the proper public keys to everyone who needs them. The labeling and distribution of public keys in public key systems is the primary function of PKI. PKI is an important part of any public key cryptosystem, and it is also the most scaleable form of key management [37].

The most important component of PKI is the CA (Certificate Authority), the trusted entity in the system which is responsible for validity of digital

certificates. Success of PKI depends on the availability of the CA to the principals in the system or the nodes in the network since a principal must correspond with the CA to get a certificate, to check the status of another principal's certificate, to acquire another principal's digital certificate, and so on.

In Ad hoc network, how to set up/manage CA is much different and difficult than in normal networks. Various forms of techniques have been proposed for use in MANET. We will review and discuss them in next chapter.

# Chapter 4 MANET ROUTING SECURITY THREATS AND SOLUTIONS

## 4.1 MANET Security Services

Security issues in MANET are different and much more complicated than those within fixed networks. The usual requirements for fixed networks are availability, confidentiality, integrity authentication and non-repudiation [36]. In MANET, each of these requirements will be more or less different. This is due to the nature of the ad hoc network we discussed previously. To secure MANET, the following attributes need to be considered:

1. *Availability*, which ensures the survivability of network services despite DoS. On the routing level, an attacker could disrupt the routing protocol and jam or partition the network. And since each node of the network need act as a router, even one malicious node might heavily affect the packet routing.
2. *Confidentiality*. This property ensures that certain information is never disclosed to unauthorized entities. In a critical environment such as military or E-commerce, the transmission of sensitive information will require confidentiality. Furthermore, routing information may also require confidentiality in certain case.

3. *Integrity*. This guarantees that a message being transferred is never corrupted. Because of wireless links, both the routing information and the message body could be easily corrupted by an adversary.
4. *Authentication*. It enables a node to ensure the peer node it is communicating with is authentic. Without authentication, an attacker could impersonate as a legitimate node, thus gaining unauthorized access to resource and sensitive information, and further, interfering with the operation of other nodes. And due to the high frequent changes in both the topology and membership, trust relation also changes dynamically.
5. *Non-repudiation*. The attribute ensures that the origin of a message cannot deny having sent the message. This is useful in intruder detection and reaction for misbehavior or malicious nodes.
6. *Cooperation fairness*. It ensures that all the nodes are willing to forward messages as routers. This is a special feature of MANET. Due to limited resources, some nodes could be too selfish to forward others packets in order to save battery power and CPU cycles.

### 4.2 General Security Attacks In MANET Routing

We can divide network attacks into two categories based either activities or sources. In terms of activities there are two classes: *passive* and *active attacks*.

*Passive attacks* do not send messages -- they only eavesdrop on the network. Passive attackers are mainly threats against the confidentiality of the network. For example, routing headers may contain information regarding node locations which are confidential in a battle field. Normally, this kind of attacks can be defended by encrypting the routing information. For example, a network-wide shared secret key could limit an attacker's interpretation of eavesdropped routing messages.

*Active attacks* inject packets into network sand generally also eavesdrop. *Active attacks* can be characterized by the number of nodes in a network and the number of nodes an attacker has compromised: we have *active-0-1, active-0-x, active-1-x, and active-y-x.* Normally an active attack on MANET routing protocols falls into one of two categories [3]: *routing disruption attacks* and *resource consumption attack*. From perspective of the application layer, both attacks are instances of DoS attacks.

In a *routing disruption* attack, the attacker attempts to cause legitimate data packets to be routed in dysfunctional ways. Routing disruption includes routing loop, black hole, detours, partition, wormhole, and rushing attack. To prevent these attacks, each node that interprets routing information must verify the origin and integrity of that packet. Normally this kind of authentication mechanism does not cause too much computation and communication overhead.

An attacker can cause a *routing loop* by sending forged routing packets that cause packets to traverse nodes in a cycle without reaching their destinations. *Black hole* happens when an attacker routes all the packets for some destination to itself and then discard them, or the attacker could route all the packets for some destination to an area where the destination doesn't exist at all. *Detours* means routing the packet in a non-optimal path. *Partition* means preventing one set of nodes from reaching another.

In a *wormhole attack* [40] an attacker receives packets at one point in the network, "tunnels" them to another point in the network, and then replays them into the network from that point. This kind of attacks can still be performed even if the network communication provides confidentiality and authenticity, and even if the attacker has no cryptographic Keys. Packet leashes is a way to defend against this type of attacks [40].

*Rushing attac*k is a particular attack against on-demand routing protocols. An attacker disseminates route requests quickly throughout the network, suppressing any later legitimate route requests when nodes drop them due to the duplicate suppression mechanism.

*Routing resource consumption* attack is like the attack of injecting extra data packets or injecting extra control packets. By doing this the attacker could consume a lot of bandwidth or computational resources so that the performance of the whole network suffers dearly.

*Sleep deprivation torture attack* was named especially for battery exhaustion [24]. A malicious user may interact with a node in a way for no other purpose than to consume its battery energy. Many portable devices will try to spend most of the time in a sleep mode in which they only listen for radio signals once in a while. In this environment, power exhaustion attacks are a real threat, and are much more powerful than many other better known DoS attacks such as CPU exhaustion.

In terms of resources, there are also two classes: *external and internal* [19]. External attacks by its name come from external attackers. By injecting erroneous routing information, replaying old routing information, or distorting routing information, an attacker could successfully partition a network or introduce excessive traffic load into the network to cause repeated retransmissions and inefficient routing.

Internal attacks are related to external attacks and are more serious. Comprised routers/computers may advertise incorrect routing information to other nodes. Detection of such incorrect information is difficult, because compromised nodes are able to generate valid signatures using their private keys.

## 4.3 Current MANET Routing Security Proposal

### 4.3.1 MANET Routing Prevention Mechanism

*TESLA* adds a single message authentication code (MAC) to a message for broadcast authentication [40]. Different from asymmetric protocols such as RSA, TESLA archives the asymmetry from loosed clock synchronization and delayed key disclosure, using one-way hash chain. The scheme is efficient but needs the support of loosed synchronization technique.

*SRP* (Secure Routing Protocol) guarantees correct route discovery, so that fabricated, compromised, or replayed route replies are rejected or never reach the route requester [16]. SRP assumes a security association between end-points of path only. Intermediate nodes do not have to be trusted for the route discovery. This is achieved by requiring that the request along with a unique random query identifier to reach the destination, where a route reply is constructed and a message authentication code is computed over the path and returned to the source. The correctness of the protocol is proven analytically.

*ARIADNE* (secured base on on-demand routing protocol) prevents attackers from tampering with uncompromised routes consisting of uncompromised nodes [41]. It is based on Dynamic Source Routing (DSR) and relies on symmetric cryptography only (using TESLA). The protocol uses the key management protocol TELSA that relies on synchronized clocks. On

route discovery, however, ARIADNE also presented the possibility of three different techniques: TESLA, Digital signatures and MAC. The protocol described so far is vulnerable to an attacker that happens to be along the discovered route. Simulations have shown that the performance is close to DSR without optimizations.

*SEAD* (secure efficient Ad hoc Distance vector routing protocol) is a secure ad hoc routing protocol based on the design of DSDV (Destination-Sequenced Distance-Vector routing protocol) [15]. For routing update message, it uses efficient one-way hash functions rather than relying on expensive asymmetric cryptographic operations. By the metric and sequence number authenticators, SEAD is robust against multiple uncoordinated attackers from creating incorrect routing state in another node. Also, by using the destination sequence numbers, it provides replay protection of routing update messages. To authenticate neighbors, two approaches were proposed. One is using TESLA, HORS, or TLK, which needs synchronized clock and incurs either an authentication delay or relatively high communication overhead. Another one uses the respective key in conjunction with a Messages Authentication Code by assuming a shared secret key among each pair of nodes. Performance evaluation has shown that SEAD outperforms DSDV-SQ in terms of packet delivery ratio, but SEAD adds overhead and latency to the network.

### 4.3.2 Intrusion Detection and Reaction in AD HOC Routing

According to Schneier [42] a prevention-only strategy will only work if the prevention mechanisms are perfect; otherwise, someone will find out how to get around it. Most of the attacks and vulnerabilities have been the result of bypassing prevention mechanisms. To apply this on ad hoc security, combining intrusion detection and prevention together would lead better mechanisms.

*Intrusion detection* is one of the key techniques behind protecting a network against intruders. An intrusion detection system tries to detect and alert on attempted intrusions into a system or network. An intrusion is considered to be any unauthorized or unwanted activity on the system or network [9]. The traditional IDS systems in wired network normally have a centralized decision making entity, which ad hoc networks lack of.

Detecting several kinds of misbehaving is very hard because it is difficult to distinguish misbehaving from transmission failures and other kind of failures [7].

In ad hoc networks, routing protocols can keep track of perceived malicious nodes in a "blacklist" at each node as proposed in watchdog and pathrater [11]. However, an attacker may blackmail a good node, causing other good nodes to add that node to their blacklists, thus avoiding that node in route. A better way is to let a node only trust itself for acquiring information

about which nodes in the network are malicious, or further by authentication decide whether the destination it is talking with can also be trusted.

In [8], Intrusion detection for wireless ad-hoc networks has been proposed. The author presented a distributed IDS system with a cooperative decision algorithm. Each mobile host has IDS client installed that runs a local detection engine to analyze local data for anomalies. The system uses a majority voting mechanism to classify behaviors by consensus. Responses include re-authentication or isolation of compromised nodes. The authors argue that an architecture for intrusion detection should be distributed and cooperative, using statistical anomaly-detection approaches and integrating intrusion-detection information from several networking layers. However, Oleg [9] points out that anomaly detection has proven to cause poor performance and high false alarm rate. Also, since clients are structured around several layers and are self contained entities, they are subjects to attacks themselves.

*Watchdog and pathrater* proposed in [11] introduced two extensions to the dynamic routing algorithm (DSR) to mitigate the effects of routing misbehavior: a watchdog for detection of denied packet forwarding and a pathrater for trust management and routing policy rating every path used, which enable nodes to avoid malicious nodes in their routes as a reaction. Using the well-known NS network simulator, they observed a throughput increase. Although this reaction does not punish malicious nodes that do not cooperate and actually relieves them of the burden of forwarding for others

while having their messages forwarded, it allows nodes to use better paths and thus to increase their throughput.

Agent-based ad hoc network IDS uses a clustered network monitoring node selection algorithm to dynamically assign a few key nodes [9]. These nodes host sensors that monitor network packets and works as agents that make decisions as well. Each agent contains a state machine for all nodes within the cluster it resides in. As intrusion or anomalous activity evidence gathers for each node, the agent can decide with a certain confidence that a node has been compromised by looking at reports from the node's own local monitoring information. This IDS makes a total network load smaller by distributing the workload of whole IDS among a few of key nodes to minimize the power consumption and IDS related processing time by all nodes.

*Packet leashes* [40] was proposed against the wormhole attacks in MANET. The author presented two types of leashes: geographic leashes and temporal leashes. The key intuition is that by authenticating either an extremely precise timestamp or location information combined with a loose timestamp, a receiver can determine if a packet has traversed a distance that is unrealistic for the specific network technology used. Also, an authentication protocol called TIK was introduced to implement leashes, which is based on symmetric cryptography. The temporal leashes require tight time synchronization, and the geographic leashes require loose time synchronization.

### 4.3.3 Enforcing (Stimulation) Cooperation in Ad Hoc

Unlike networks using dedicated nodes to support basic functions like packet forwarding, routing and network management, those functions in MANET are carried out by all the nodes. The network needs all nodes to cooperate and provide service to each other, while there is no good reason or guarantee that all nodes will cooperate, since these services will cost the limited resource at each node. If every node (or many enough) follows the selfish strategy, the MANET will be a non-functional or entirely absent network [N01]. So ensuring cooperation is a special security issue in ad hoc networks.

Incentives to cooperate have been proposed by Butty¡äan and Hubaux [6] in the form of so-called nuglets that serve as a per-hop payment in every packet or in the form of nuglets counters [21] to encourage forwarding. Both nuglets and counters reside in a secure module in each node and are incremented when nodes forward for others and decremented when they send packets for themselves. One of their findings is that, given such a module, increased cooperation is beneficial not only for the entire network but also for individual nodes. However, the application of this scheme is limited by the assumption: the existence of an overlaid geographic routing infrastructure and a public key infrastructure which requires an on-line certification authority. Also the technique is limited by the high computational overhead (hop by hop public key cryptography for each packet).

In CORE (a collaborative reputation mechanism) proposed in [5], node operations are stimulated by a collaborative monitoring technique and a sophisticated reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task-specific behavior), which are weighted for a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. Reputation values are obtained by regarding nodes as requesters and providers, and comparing the expected result to the actually obtained result of a request. A performance analysis by simulation is stated for future work there.

CONFIDANT (Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks) was also proposed in [5] and it detects malicious nodes by means of observation or reports about several types of attacks and thus allows nodes to route around misbehaved nodes and to isolate them from the network. Nodes have a monitor for observations, reputation records for first-hand and trusted second-hand observations, trust records to control trust given to received warnings, and a path manager for nodes to adapt their behavior according to reputation. Simulations for "no forwarding" have shown that CONFIDANT can cope well even with half of the network population acting maliciously.

# Chapter 5 KEY MANAGEMENT IN LCMRMG

## 5.1 Overview

The PKI technique has already been proposed to be used to implement secure MANET [14][18][19][20]. However there are a number of problems in the current work:

(1) Some of them are not using threshold secret sharing. A consequence is that all existing MANET security schemes are vulnerable and have low degree of availability because only one single CA exists and unavailability of this single CA would defeat the whole security scheme. In *SEAD [15]* and *ARIADNE [41]*, although they proposed to use symmetric cryptographic algorithm, they still need a third party, i.e. a CA, to provide the key distribution.

(2) Some are using threshold secret sharing, but the SK shares are distributed evenly to the entire network nodes, which introduces a lot of computation overhead. The work named *Fully distributed certificate authority* in *[20]* is a good example, where all the nodes in network are equals and each holds a share of the signing key, it has share update. However, distributing the SK shares all over a MANET makes it possible to compromise enough k

nodes within a share update period, and also introduces more communication and computation overload.

(3) Most of the works do not update contents of the shared secrecy. Given enough time, a model-2 attacker still can compromise k nodes to expose the system secret SK.

Further, the share distribution algorithms proposed by [20][14] are separated from routing algorithm, i.e., an extra algorithm (computation) to build key distribution topology is needed.

Our security scheme extends the work in [20] and dynamically distributes the CA function into LCMRMG's multiple roots, which are the leaders of each spanning tree. These roots will provide unified certificate service, and there is no extra combiner as in [18]. Whoever requests the certificate has to be responsible for combining the partial certificates. Our scheme uses a *(k,n)* secret threshold to distribute the RSA certificate signing key, and it extends Herzberg's periodical secret share update mechanism with scalable algorithms [39] to further improve the security robustness and keep the certificate service available even during update periods.

### 5.2 System Model

Our security scheme has been implemented in LCMRMG routing protocol. It provides PKI on a LCMRMG MANET and handles both model-1

and model-2 adversaries. The PKI can provide support to other mechanisms that can be used to handle other attacks as well.

### 5.2.1 Assumptions

It assumes that a MANET contains n mobile nodes; each node may dynamically join, leave, move or fail independently. The reliability of multi-hop packet forwarding is provided by LCMRMG. It also assumes that:

(1) Each node has a unique nid, which is known publicly; also each has its own unique id(s), which is used for secret shares.

(2) A node can discover it one-hop neighbors.

(3) A node has some local diction mechanism to monitor its one-hop neighbors.

(4) A node has some ability to do cryptographic computations such as RSA.

(5) Through some means, a node can get an initial certificate before it join in the network. This could be achieved before the network bootstrapping, or the node can prove its authority to k-coalition roots.

It is not assumed that a node has at least k neighbors to get certificate service.

### 5.2.2 Architecture

**RSA key**:

There is a network wide PK/SK pair exists. The PK is network wide known, while the SK is distributed to multi-roots using secret threshold sharing mechanism. Any coalition of roots with k polynomial shares can potentially recover SK by Lagrange interpolation, while no coalition of roots that contain up to k-1 polynomial shares could generate the correct SK.

Each node has its own pk/sk pair. The sk is kept secret by itself and pk is known within the spanning tree. Since the root of this tree knows this node's sk and nodes from other trees can contact the root, each node's pk is network wide known as well.

**Certificate with threshold secret share**:

Each node carries a certificate signed by SK, since PK is network wide known, so any node can verify the validity of the certificate. Without a valid certificate, the node will be excluded by the network, which means that it is denied access to any network resources such as routing.

The network wide secret, i.e. the private certificate signing key SK, is shared among all possible roots. To further defend adversaries, we extend Herzberg 's periodical secret update technique with scalable algorithms to dynamically refresh secrets.

Within the certificate valid period, a node can use its certificate as "passport" for trust proof in normal operations such as packet forwarding, and use its own pk/sk pair to meet the other security requirements such as message encryption/decryption.

Before the certificate expires, a node must contact certificate servers (roots) to obtain new certificate. Since the SK is distributed among multiple roots, the node must contact enough roots that form at least a coalition of k threshold shares. Upon request, a root will check the legibility of the requesting node. If its record shows that the node is a well behaved legitimate node, it will generate partial certificate by applying its share(s) of SK. Whenever a requesting node collected enough these partial certificates signed by SK share, it then combines them to form a new valid certificate with new expiration time.

**Proactive share update**:

At regular time intervals, a share update phase will be initialized. During each update phase, certificate service will continue as usual using the old secret shares. After the share update phase is completed, new shares are used and coalition of old shares will not be able to reconstruct the secret SK any more.

## 5.3 Protocol and Algorithm

This section describes the algorithms to setup and maintain the distributed keys. Figure 5 illustrates the basic scenarios.

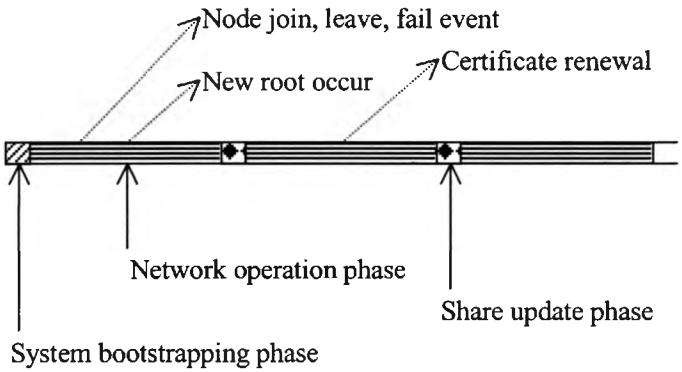


Figure5 Network Time Deviation

**System bootstrapping phase:**

At the very first time instance, a LCMRMG MANET starts with one single node. This node works as a root and we assume this first starting node is trusted. This node will generate network PK/SK and its own pk/sk pairs, then it will use secret share algorithm described in Section 3.7, and use SK as $a_0$. Then the SK itself is destroyed and k parts of secret shares are stored by the first node.

When another root was created by LCMRMG, it will follow "new root occurs" algorithm to dynamically redistribute secret shares.

**Normal operation phase**:

**(1) Node join, leave, fail event**

When a new node wants to join the existing MANET, it must prove its authority by its initial certificate, if the initial certificate expires or invalid, the node is denied to certificate service, therefore denied from the whole network. Once its authority is proved, by following the "*certificate renewal*" algorithm, it will get a singed certificate with stamped expiration time.

When a node leaves or fails, if it is non-root, it won't affect the certificate service. If a root leaves and does nothing, it may affect the certificate service: the total secret shares may be less than the needed k threshold, and thus the whole certificate will fail. In our algorithm, before a root leaves, it will inform other roots and a informed root will initialize a "threshold recover" (the k coalition may include the leaving root). The threshold recover algorithm is almost the same as the "new root occurs" algorithm. The only difference is that the request node is an existing root and this root is requesting more shares. When a root fails, if the failure occurs at the beginning stage of MANET operation and there is no much secret shares redundancy (in "new root occurs" algorithm), this failure may affect the certificate service. But when there is enough secret redundancy, the certificate service will still work fine.

**(2) Certificate renewal event**

If node p noticed its own certificate is expiring soon, it needs renew its certificate to extend its certificate validity. The work in [20] requires node p to broadcast its request to collect shares, assuming node p has at lease k neighbors. To avoid inefficient broadcasting and improve availability, in LCRMG, node p will only contact its root(s) directly.

If the coalition of its root(s) has enough (k) secret shares, node p could directly obtain enough partial certificate $cert_i$ and it certificate renewed locally by it root(s). If its roots do not have enough secret shares, p has two choices:

(1) If it could or is willing to move, it will collect as much (m) partial certificates ($m < k$) from it's current roots, and then move to another location, find new roots to get the rest for its partial certificate. Of cause, not every time can a node move freely. In our simulation implementation, we have experimented with various numbers of moves allowed during the entire time interval in which a node is attempting to renew its certificate. Our method minimizes the commutations overhead but has the risk of that a node will never get enough shares before its current certificate expires. More about the relationship between the number of allowed moves and the renewal success ratio will be discussed in Chapter 7.

(2) If it cannot or is unwilling to move, its roots will attempt to locate other roots in MANET and then collect partial certificate for node p. With this choice normally a node can almost guarantee its certificate renewed at the risk of

incurring certain overhead, as a lot of multi hop communications may be necessary.

Figure 6 shows the partial certificate algorithm by luo[20].

$$to\ get\ cert_{updated} = (cert)^{SK}$$

$$need\ collect\ coalition\ of\ k\ \ cert_i = (cert)^{S_i} \bmod N$$

$$then\ cert_{updated'} = \prod_{i \in Coalition} (cert_i)^{l_{id_i}(0)}$$

$$= (cert)^{\sum_{i \in Coalition} S_i \cdot l_{id_i}(0)}$$

$$= (cert)^{t \cdot N + SK}$$

$$then\ use\ Luo's\ "k-bounded\ offsetting\ algorithm":$$

$$Begin:$$

$$Y_0 = cert_{updated'}$$

$$Z = cert^{-N} \bmod N$$

$$j = 0, w = 1$$

$$while\ j \le k\ do$$

$$Y = Y_0 \cdot w \bmod N$$

$$w = w \cdot Z \bmod N$$

$$if\ (cert \equiv Y^{PK} (\bmod N))\ then$$

$$return\ Y$$

$$end\ if$$

$$j+ = 1$$

$$end\ while$$

$$End$$

Figure6 K-Bounded Coalition Offsetting Algorithm

**(3) New root occur event**

In LCMRMG, nodes cache the locality of network routing traffic passing through them. Based on these cached locality statistics, a node can calculate the estimated reduction on network traffic if it becomes a new root. When the reduction calculated exceeds the given threshold value, the node becomes a new root node.

Despite the routing topology change, this new root now can claim to join the certificate service. Since new roots are elected in LCMRMG MANET in an optimistic way with regard to communication traffic, the communication traffic of certificate service is optimized as well.

The following is the "new root occur event" algorithm, which describes how a new root p gets secret share(s) to become of a part of CA:

(1) Before p actually becomes a new root, it is still a sibling of some roots. With same procedure as "certificate renewal", p locates a coalition of roots that contain k shares.

(2) Each root in the coalition verifies the certificate of node p. If the verification fails, the request is denied.

(3) After the verification succeeds, each existing root$_t$ in the coalition generates the partial share $S_p^{\,t} = sj \cdot l_{id_t}(id_p)$ for node p and securely sends it to p.

(4) By applying the formula:

$$Sp = \sum_{t=1}^{k} S_p^{\,t} \bmod N = \sum_{t=1}^{k} S_t \cdot l_{id_t}(id_p) \bmod N = f(id_p) \bmod N,$$

Node p gets its secret share.

(5) The roots who belong to the coalition dynamically reallocate their secret shares using "secret share management" algorithm which will be discussed in Section 7.1 security analysis.

**Share update phase**

During share update phase, the following steps are performed:

(1) At the beginning, each root initializes the update process with the probability 1/n (n is current root number). A root that decides to initialize the update process will flush a notification to announce that it will initialize the update, to say, from 00:00:00 to 00:15:00 is update time.

(2) The root will need to generate the update polynomial

$$f_{update}(x) = b_1 x + b_2 x^2 + ... + b_{k-1} x^{k-1} \bmod N$$

(3) The coefficients are then encrypted and broadcast to other roots

(4) Each root generates its update share $S_{i_{update}} = f_{update}(id_i)$ and then the new share $S_{i_{new}} = (S_{i_{old}} + S_{i_{update}}) \bmod N$

(5) During update time, normal communication continues, every root will still use old shares to performance certificate service. Since the network secret remains the same, the public key remain the same too, and the network authentication will still work during this periods. When the update time is over, say, 00:15:00, the network will use the new shares to provide certificate service.

### 5.4 K value, Security, and Availability

Obviously, the k value is essential for our whole security scheme to work efficiently and reliably.

(1) First, the value *k* determines the robustness of the security. Between any two share update periods, an adversary must compromise at least *k* shares to gain the SK. So the smaller the *k* is, the less robust the network security is.

(2) On the other hand, the value *k* also determines the availability of our services. Since a node must be able to communicate with at least *k* share roots to gain the certificate service, if the *k* value is too large, the node may have difficulty to gain the service due to the unavailability of enough share roots.

(3) Traditionally, the value ***k*** remains unchanged no matter the size of whole ad hoc network. Fixed value *k* makes security schemes not scalable and can weaken the security in large networks. In large networks, the probability that a *k* coalition can be found to compromise is higher.

In traditional secret share schemes, every certificate server has the same security level in the form of having the same share number (normally 1).

This is not suitable for real world applications. For example, in a real battle field, different security levels must exist between the commander and his solders. More likely the commander may have more physical security mechanism and may have more computation power. He may desire to have the ability to provide certificate service by himself while requiring at least five soldiers to sign a certificate. In this case, it is unreasonable to assume that the commander has the same secret shares as the solders.

Our research has tried to integrate these three aspects (k, security, availability) dynamically to maximize efficiency without compromising security. In our securing LCMRMG, the secrets will be shared among all the roots, and the k value will be preset as a value which can provide enough security despite the network size(i.e., say it is 16). In contrast to other existing schemes in which each node holds one share, however, each root in our scheme may hold more than one share. Based on their importance, different root nodes will have different number of shares. For example, in a (3,n) threshold scheme, a node having two $S_i$ shares only need another extra one share to sign a certificate, while a node having only one $S_i$ share must obtain another two shares to sign the certificate. This is illustrated in Figure 7.

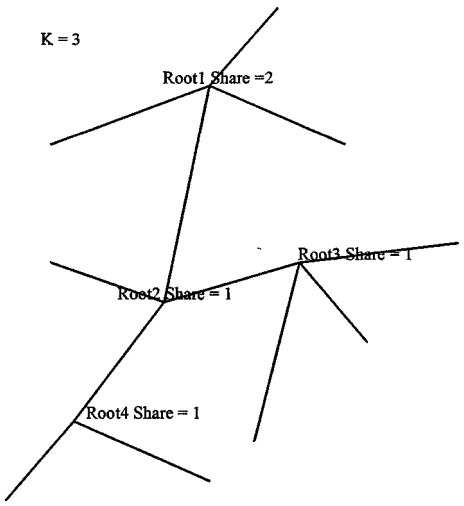


Figure7 Different Security Level Root Have Different Number of Shares

For example, if the k value is 12, and we have two roots, each roots may have 6 share. So these two roots will be enough to provide certificate service -- the actual working k value is 2. If another root is generated, then the two existing roots may each "reallocate" the shares so that each roots has 4 shares now. The consequence is that these three roots must work together to provide certificate service and the actual working k value changes to 3, as shown in Figure 8.

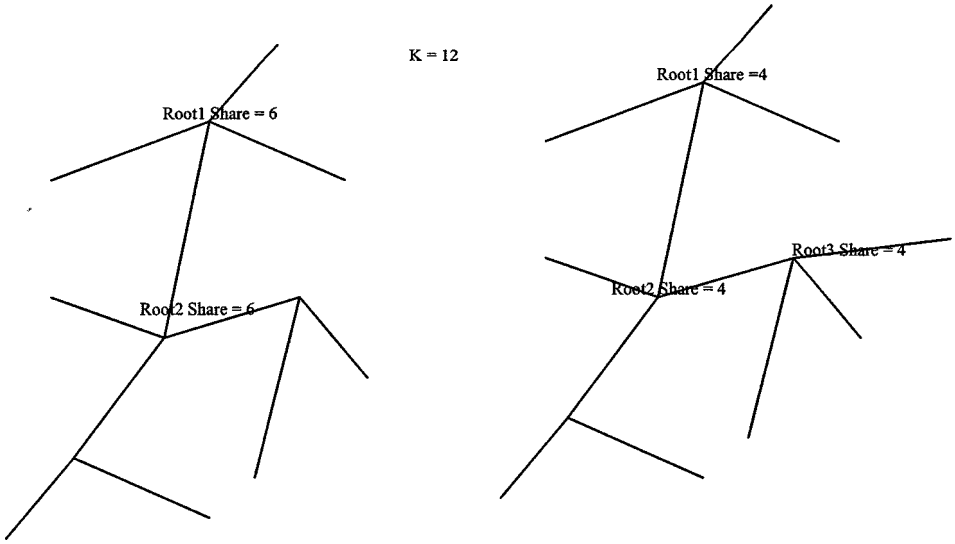


Figure8 Dynamically Change the K Value

With careful control, we can combine the above schemes to allow roots at different hierarchies to have different amount of shares. The main issue is how to dynamically allocate the secret shares while keeping balance between security and service availability. In our research this is achieved by our share management algorithm.

### 5.5 Share Management

As discussed previous, if the number of shares at each root node remains stable, adversaries may have higher probability to invalidate or weaken the k-threshold mechanism. For example, assume k=16, the first root certainly owns 16 shares if there is no other roots. When there exists more than one root node while all the 16 shares remain at the first root, attackers could compromise the network by only compromising that root node, because the 16 shares are enough to reconstruct the SK. Therefore our scheme allows each root to own more than one secret share.

Spreading the shares has its own risks. For example, suppose k = 16 and there are 16 roots, each of which owns only one share. If one of the roots fails, there will be never enough shares to provide the certificate service. That is why we need some redundancy of the secret shares. Our share management scheme is simple but effective:

Whenever a new root is generated, the existing roots and the new root will work together to rebalance the share distribution. This means, without extra communication, the old roots will decide if they need delete some shares based on their knowledge of the total number of shares.

To make sure the total share number is bigger than k but not too big, our security scheme will make sure that there at least minimal k share exists by simply making sure that each of the old roots in coalition will keep at least k/r + 1 shares, where r is the number of roots, including the new root. The new root' will be assigned least k/r+ 1 shares too.

This scheme provides a minimal level of redundancy that can defend against the problem of single root failure bringing down the whole security mechanism. More redundancy can be provided similarly to meet the need of higher level availability.

# Chapter 6 IMPLEMENTATION

In the previous chapters, we introduced the design of our approach to securing LCMRMG routing protocol. To investigate the efficiency of our approach for supporting key management with dynamic threshold share, we design and build a testbed to conduct experiments with our implementation. Different test scenarios are designed to demonstrate the functionality of our implementation and evaluate the performance of our approach.

Based on the LCMRMG's testbed, which was written in c language to simulate the MANET routing, I modified them into c++ language to support my security algorithm in class. The RSA, MD5, Threshold secret share algorithms were successfully integrated into the LCMRMG routing. However, I didn't implement the X509 certificate; instead, I only simulate a certificate which contains necessary fields for our evaluation: Node id, cert_begin_time, cert_end_time, and RSA signed MD5 message.

Since the RSA Public-Key Cryptography needs large integers for reasonable security. The 32-bit or 64-bit integers available on most machines just aren't big enough. Therefore, the RSA Public-Key Cryptography package

uses another package, called the Multiple-Precision Unsigned Integer Arithmetic, to do its arithmetic.

In this package, the number of bits can be any multiple of 16. It is written by Philip J. Erdelsky [http://www.alumni.caltech.edu/~pje/]. Some modification was made for it to be used by our testbed.

MD5 algorithm was written followed by Dr R. Rivest's memo on "The MD5 Message-Digest Algorithm" [http://rfc.sunsite.dk/rfc/rfc1321.html].

The realization of Secret Threshold algorithm is from Stefan Karrmann's sharesecret-0.4.0[http://www.mathematik.uni-ulm.de/ftp/pub/soft/crypto/]

# Chapter 7 EXPERIMENTS AND EVALUATION

## 7.1 Security Analysis

Our proposed security scheme has been focused on how to secure the LCMRMG routing by applying secret shares to build a public key security mechanism. The most innovative aspects of our security scheme that separate it from those traditional PKI based schemes are: (1) the CA is dynamically distributed to the multiple roots of LCMRMG without a third party; (2) a hierarchical scheme is proposed to support different roots that may have different number of shares. The hierarchical scheme that we adopted from LCMRMG, coupled with the dynamic distribution of CA and proactive updates, makes our security scheme more robust and efficient.

The LCMRMG routing's security scheme possesses the following desirable properties:

(1) *Authenticity*: Routing updates must originate from authenticated nodes and users. Each entity carries a public key certificate, signed by k-coalition CA to claim its authenticity. The receiving node can verify routing claim by examining the certificate. Use of distributed secret share makes it unnecessary to have a centralized authority to issues and validate certificates in LCMRMG.

(2) *Authorization*: Our security mechanism allows an authenticated node to issue an un-forgeable credential by the distributed certificate authority. These credentials specify the privileges and permissions associated by the nodes. Due to time limitation, use of credentials has not been simulated in our implementation. We assume any trusted packet can trigger update propagations and modifications to the routing table.

(3) *Integrity*: The information carried in the routing updates can cause the routing table to change and alter the flow of packets in the LCMRMG. Therefore, the integrity of the content of these messages must be guaranteed. This is accomplished by using message digests and digital signatures.

(4) *Non-repudiation*: Routers cannot repudiate ownership of routing protocol messages they send. Ad-hoc nodes obtain information from their neighbors and forward it to their other neighbors. These neighbors in turn may forward it to their own neighbors and so on. In the original LCMRMG, nodes cannot be sure of the authenticity of updates that are not generated by their immediate neighbors. Our solution is forming a chain of routers and authenticating every node in the routing path (chain), following the path to the source. Although such chaining finding is not cheap, it is still necessary to enforce non-repudiation.

(5) *Confidentiality*: In addition to integrity, sometimes it may be necessary to prevent intermediate or non-trusted nodes from understanding the contents of packets as they are exchanged between routers. Encrypting the routing protocol packets themselves can prevent unauthorized users from reading it. Only nodes that have the decryption key can decrypt these messages while the nodes without the decryption key can only participate in the routing.

### 7.2 Performances Result and Analysis

In our simulations, node movement is simulated as follows. Each node is initially placed at a random location and pauses for a period of time; it then chooses a new location at random and moves there with a velocity randomly chosen between 1 and a predefined maximum speed. Without going through the details, the simulation environment is shown in Table 2 where the value k is the secret share threshold.

| | |
|---|---|
| CPU | Intel P4 3.06 G Hz |
| Memory | 1GB |
| Number of Nodes | 1000 |
| Maximum Velocity | 10 m/s |
| Dimensions of Space | 1000* 1000m |
| Radio Range | 50m |

Table2 Simulation Environment

To evaluate the performance of the key management on LCMRMG, our performance tests focus on the following three metrics:

1) *Certificate Latency*: it compares the average time of certificate renewal, secret share update in different k value, RSA key length and different number of roots.
2) *Communication overhead*: the number of extra control packets for

certificate service with different k value and different number of roots.

3) *Certificate availability*: the ratio of the number of successful certification service over the total number of certificate requests

### 7.2.1 Certificate latency

As mentioned in Section 2.1, normally a node in MANET has limited resources such as computation power. Using of RSA algorithm is expensive in computation cost [23] and hence will cause certificate latency in our key management scheme. We tested the latency under various RSA key lengths and various numbers of roots.

| RSA key (bit) | Latency (s) |
|---|---|
| 32 | 0.237840 |
| 64 | 0.624264 |
| 128 | 0.842025 |
| 256 | 1.311491 |
| 512 | 3.765 |
| 1024 | 13.922 |
| 2048 | 56.311491 |

Table3 Certificate Latency vs. RSA Key Length (k=8, root number = 16)

As shown in Table 3, the RSA key length is critical to the performance of our certificate service. When the RSA key is 32 bit, certificate renewal only takes 0.238 seconds. In contrast, it takes more than 56 seconds when RSA key is set to 2048 bits. In fact, our simulation can hardly run when the RSA key was set more than 512 bits. This indicates that the low-end MANET devices will not be able to use large RSA keys.

Figure 9 shows that when we change the k value and root number while keeping RSA key length at 128, the certificate latency varies between 0.8 and 1.2. This tells us that the k value and root number are not as important as the RSA key length in causing certificate latency.

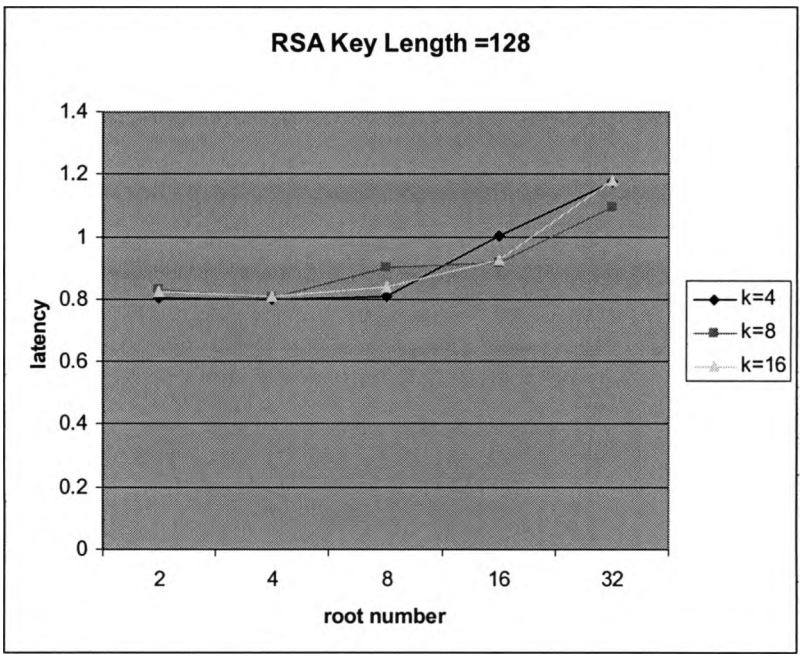


Figure.9 Latency Under Different K and Root Number

### 7.2.2 Communication overhead

Figure 10 shows the total communication overhead on certificate service in packets, versus the number of roots nodes and k value. To facilitate comparison, when the number of roots is less than k, we distribute the shares so that k is equal to the number of roots (i.e. a node must contact all roots to

get certificate renewal); after the number of roots exceeds k, each root will contain one share only.

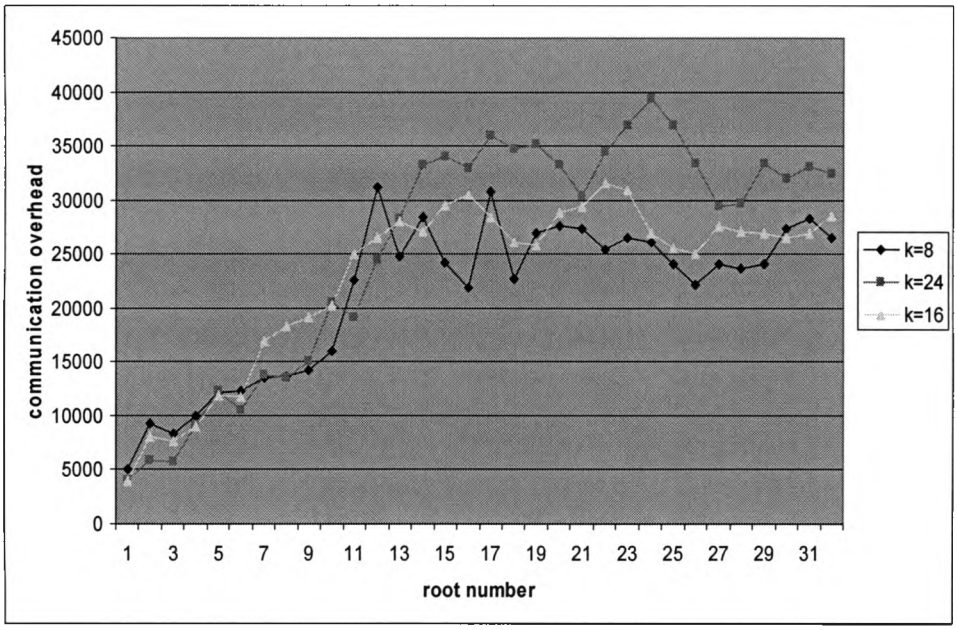


Figure10 Communication Overhead

Before root number exceeds 8, all three cases of k values (k =8, 16, 24) give us almost the same performance on communication overhead. This is because in all these cases the three k values are actually the same - equal to the number of roots. Nodes have to actually contact with the same number of roots regardless of the k value. In addition, when $k_{actual}$ grows lineally before it reaches value k, the overhead growth is lineally as expected.

By contrast, when the k value is larger than 8, it has direct impact on the overhead. For example, when the root number is 32, we can tell that the

bigger the k is, the more overhead the key management takes. This confirms with our previous discussion that k is the most important performance parameter in our architecture. An implementation with larger k values can tolerate more powerful adversaries at the cost of degraded performance. On the other hand systems with smaller k enjoy low overhead and robustness against ordinary DoS attacks, but are subject to more sophisticated hacking attempts. Therefore, we must carefully balance the performance and security aspects. We believe that the dynamically share distribution and hierarchical scheme proposed in this thesis provides us with more flexibility to balance between performance and security.

Recall in our previous discussion, in theory when the number of roots increases, overhead should in general decrease as nodes can more easily obtain the needed certificate service. To our surprise, this does not happen in our simulation. For example, in the k=16 line, after root number exceeds 16, we didn't observe a steady overhead decrease. In fact, the overhead line is almost flat. Almost the same happened in the k =24 line. The k=8 line is a little bit different, where the overload keeps increasing until the root number exceeds 12, then it decrease a little, and then it goes steadily as well. This can be explained as follows. Adding more roots in LCMRMG inevitably increase routing cost [34]. Whenever the topology changes in LCMRMG, the spanning trees change, and this change incurs more maintenance. This

explains the initial stead overhead increase. However, when the number of root reaches 12 and larger, certificate availability improves, which cancels out the effect of maintenance overhead increase. Therefore we saw the flat line after the 12-root point. More about this is discussed near the end of the next section.

### 7.2.3 Certificate Availability

Intuitively node movement should have certain impact on certificate availability. To understand this impact, our simulation controls the number of movements while a node is requesting a certificate. Our initial intuition is that more movements should allow a node to more easily obtain a certificate because it can reach more roots more quickly. However our simulation does confirm this intuition. Our simulation shows that having more than three movements while a node is requesting certificate renewal will not help the node to finish the renewal more quickly.

Figure 11 illustrates the relationship between the certificate renewal success ratio and number of roots. The certificate success ratio at the initial segment is lower than 20 percents. The certificate success ratio reaches over 80 percent only when the root number exceeds 17, and it reaches 90 percent only when the root number exceeds 23.

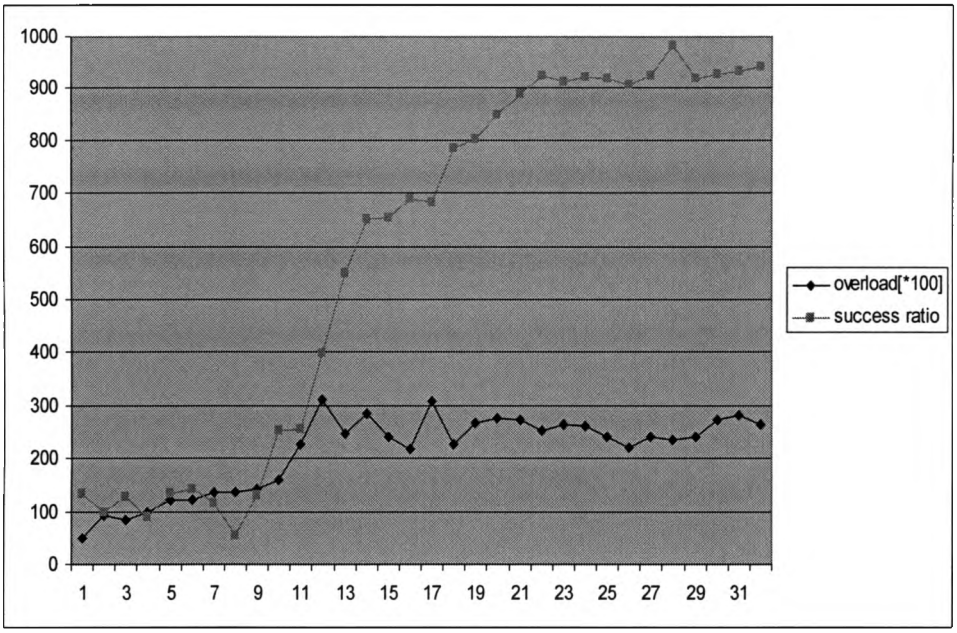


Figure11 Certificate Success Ratio vs. Communication Overload

When root number is small than k, a node must contact all the roots in order to get the needed certificate. However, the smaller root number means more likely that a node only belongs to a very smaller number of spanning trees, which may not provide enough k shares before the node must move to another location. In fact, this failure of obtaining enough shares most likely will happen before the three movements allowed during a certificate renewal operation. By contrast, when there are relatively more roots, a node may belong to several spanning trees in LCMRMG. Since the roots of these spanning trees are more likely to meet the k coalition requirement, the node has better chance to successfully get the needed certificate service.

The certificate availability also explains the last question of Section 7.2.2: why the certificate overhead remains flat, not decreasing, while the root

number increases. With k = 8, when the number of roots exceeds 12, the roots of the spanning trees the node belongs to may already have enough shares, it will need few (zero or one) moves to get the certificate no matter how many roots exist in LCMRMG. The consequence is that the overload keeps flat while roots number increase.

We also notice that the success ratio turns flat on 90 percent when the root number exceeds 23. This suggests that when the root number is large enough, increasing root number alone will not improve certificate availability.

# Chapter 8 CONCLUSION AND FUTURE WORK

The thesis studies an important class of wireless mobile networks – the MANET. It focuses on the security aspect of MANET. The research analyzes the security threats, identifies the security requirements for MANET, and systematically studies existing security solutions. The thesis proposes a dynamic key management scheme for the newly proposed LCMRMG protocol. The scheme takes advantage of the tree structure and multiple root nature of LCMRMG routing protocol and utilizes LCMRMG's routing mesh tables as our key management's "backbone".

We focus on how to secure MANET and how to establish a secure key management service in LCMRMG. To provide a highly available and highly secure key management service, we employ threshold cryptography to distribute trust among the roots of LCMRMG. Furthermore, our key management service uses proactive secret share update to guard against model-2 adversary. We have implemented our security scheme through simulation and evaluated its security and performance aspects. The implementation has proven that the proposed security scheme is both efficient and reliable and is a valuable addition to the standard LCMRMG protocol. The main characteristics of the proposed scheme are:

(1) With the multi-root key share, the whole network does not expose to any single point of compromise or single point of failure. Certificate service can be performed within minimal communication cost.

(2) When a new root is being dynamically generated in LCMRMG, the security shares can be dynamically re-balanced across the whole MANET. Hence proposed solution scales favorably to large networks. When the security share redundancy is controlled carefully, the scheme is robust against adversaries while providing highly available certificate services.

(3) Hierarchical security level has been realized by allowing different number of shares on nodes at different security levels.

(4) The scheme is flexible – it can be applied to small as well as large MANET. The performance can be controlled using the number of roots, the number of shares each root owns, and the value k.

Since LCMRMG assumed and utilized GPS in its routing algorithm, a future work should be optimizing the dynamic key management by taking advantages of that feature.

Our security scheme is based on unicast. In unicast, peer to peer communication only requires that each node has one pair pk/sk. However, in

multicast, an extra group key pair is required for different multicast groups. Although we can extend our scheme to multicast by distributing secret share to multiple roots of LCMRMG, we still need to find a way to dynamically create/allocate the group key for each multicast group member. One solution for setting up group key is: Before multicast, the multicast source node creates a pk/sk pair as group key, and then it sends to other group members by unicast. This solution is straight forward but not efficient, we are currently working on optimizing this process.

## REFERENCES


1. The Bluetooth Specification. http://www.bluetooth.org/specifications.htm

2. IEEE Std. 802-11, *"IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification* ," June 1997.

3. Yih-Chun Hu, Adrian Perrig, David B. Johnson. Ariadne: *A secure On-Demand Routing Protocol for Ad hoc Networks.* MobiCom 2002, September 23-28, 2002, Atlanta, Georgia, USA

4. Sonja Buchegger and Jean-Yves Le Boudec. *Cooperative Routing in Mobile Ad-hoc Networks: Current Efforts Against Malice and Selfishness.* In Lecture Notes on Informatics, Mobile Internet Workshop, Informatik 2002, Dortmund, Germany, October 2002. Springer.

5. Pietro Michiardi, Refik Molva Core: *A COllaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks.* Communication and Multimedia Security 2002 Conference

6. Levente Buttyan and Jean-Pierre Hubaux: *Enforcing Service Availability in Mobile Ad-Hoc WANs.* Proceedings of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Boston, MA, USA, August 2000

7. Manel Guerrero Zapata: *How to Design Wireless Security Mechanisms.*1st Workshop on Security in Ad-Hoc Networks,Ruhr University Bochum, Germany, December 2, 2002

8. Yongguang Zhang & Wenke Lee, *Intrusion Detection in Wireless Ad-Hoc Networks*. Proceedings of The Sixth International Conference on Mobile Computing and Networking (MobiCom 2000), Boston, MA, August 2000

9. Oleg Kachirski, Ratan Guha: *Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks*, IEEE Workshop on Knowledge Media Networking (KMN'02) July 10 - 12, 2002 p. 153 Kyoto, JAPAN

10. S. Yi and R. Kravets: *Key Management for Heterogeneous Ad Hoc Wireless Networks.* Technical Report UIUCDCS-R-2002-2290, 2002; Poster Presentation, 10th IEEE International Conference on Network Protocols (ICNP 2002)

11. Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. *Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks.* In *Proceedings of the Sixth Annual IEEE/ACM International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 255–265, August 2000.

12. Sonja Buchegger, Jean-Yves Le Boudec Nodes Bearing Grudges: *Towards Routing Security, Fairness, and Robustness in Mobile Ad Hoc Networks.* 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing Canary Islands, Spain, January 2002. IEEE Computer Society.

13. J. Kong, M. Gerla, R. Gadh, B. S. Prahbu *Providing Multi-Layer Security Support for Wireless Communications Across Multiple Domains.* ACM Workshop on Wireless Security (WiSe) September 28, 2002. Westin Peachtree Plaza, Atlanta, Georgia, U.S.A.

14. Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu and Lixia Zhang. *Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks.* IEEE 9th International Conference on Network Protocols (ICNP'01), 2001.

15. Yih-Chun Hu, David B. Johnson, and Adrian Perrig. SEAD: *Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks.* Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), pp. 3-13, IEEE, Calicoon, NY, June 2002

16. Panagiotis Papadimitratos and Zygmunt J. Haas: *Secure Routing for Mobile Ad hoc Networks SCS Communication Networks and Distributed Systems.* Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January 27-31, 2002

17. Pasi Eronen, Christian Gehrmann, Pekka Nikander: *Securing ad hoc Jini services.* In Proceedings of the 5th Nordic Workshop on Secure IT Systems (NordSec 2000)

18. Lidong Zhou, Zygmunt J. Haas. *Securing ad hoc networks.* IEEE Networks Special Issue on Network Security. November/December, 1999

19. Srdjan Capkuny, Levente Buttyan and Jean-Pierre Hubaux: *Self-Organized Public-Key Management for Mobile Ad Hoc Networks.* Swiss Federal Institute of Technology Lausanne (EPFL) Tech. Report (Jun 2002)

20. Haiyun Luo, Jiejun Kong, Petros Zerfos, Songwu Lu and Lixia Zhang, *Self-securing Ad Hoc Wireless Networks.* Seventh IEEE Symposium on Computers and Communications (ISCC'02).

21. L. Buttyán and J.-P. Hubaux, *Stimulating Cooperation in Self-Organizing.* Mobile Ad Hoc Networks Technical Report No. DSC/2001/046, Swiss Federal Institute of Technology, Lausanne, August 2001.

22. Dirk Balfanz, D. K. Smetters, Paul Stewart and H. Chi Wong Talking To Strangers: *Authentication in Ad-Hoc Wireless Networks In Symposium* . Network and Distributed Systems Security (NDSS '02), San Diego, California, February 2002

23. Jean-Pierre Hubaux, Levente Buttyan, Srdjan Capkun: *The Quest for Security in Mobile Ad Hoc Networks.* Proceedings of the 2001 ACM International Symposium on Mobile ad hoc networking & computing 2001, Long Beach, CA, USA

24. Frank Stajano and Ross Anderson The Resurrecting Duckling: *Security Issues for Ad-hoc Wireless Networks.* Proceedings of the 7th Internetional Workshop on Security Protocols (1999)

25. Haiyun Luo, Songwu Lu: *Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks,* Technical Report TR-200030, Dept. of Computer Science, UCLA, 2000

26. A. Shamir, "*How to share a secret,*" *Communications of ACM,* 1979

27. The mobile ad-hoc networks (MANET) working group, http://www.ietf.org/html.charters/manet-charter.html.

28. Konrad Wrona: Distributed Security: *Ad Hoc Network & Beyond,* Ad Hoc Networks Security Pampas Workshop, RHUL, Sep 16-17, 2002

29. C. E. Perkins and P. Bhagwat, "*Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers*", in Comp. Communication Rev., Oct. 1994.

30. Thomas Clausen, Philippe Jacquet, et al. *Optimized link state routing protocol.* IETF Draft, Jan. 2003 (http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-11.txt)

31. R. Ogier, M. Lewis and F. Templin. *Topology dissemination based on reverse-path forwarding (TBRPF).* IETF Draft, Oct. 2003. ( http://www.ietf.org/internet-drafts/draft-ietf-manet-tbrpf-11.txt)

32. Charles E. et al. *Ad Hoc On-Demand Distance Vector(AODV) Routing.* IETF Draft, Feb, 2003 (http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt)

33. David B. Johnson, David A. Maltz and Yih-Chun Hu. *The dynamic Source Routing Protocol for Mobile Ad Hoc Networks(DSR).* IETF Draft, Apr, 2003. (http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt)

34. Wuxu Peng, Xin Zhang and Kia Makki, *Locality Caching Multi-Root Multi-Generation Routing Algorithm in Mobile Ad Hoc Networks,*

35. X.Chen and X.D. Jia. *Package Routing Algorithms in Mobile Ad-Hoc Wireless Networks.* Proc. Of 2001 international Conference on Parallel Processing Workshops, Valencia, Spain, Sep 2001. pp.485

36. W. Stallings, Cryptography and Network Security: *Principles and Practice*, 2nd ed., Prentice-Hall 1999, ISBN 0138690170

37. Shafi Goldwasser and Mihir Bellare, *Lecture Note on Cryptography*, Aug, 2001. (www.cs.ucsd.edu/users/mihir/papers/gb.pdf)

38. R. Housley, et al. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, IETF, RFC, Jan, 1999. (http://www.ietf.org/rfc/rfc2459.txt)

39. A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "*Proactive secret sharing,*" *Extended abstract*, 1995

40. Y.-C. Hu, A. Perrig, and D. B. Johnson. Packet Leashes: *A Defense against Wormhole Attacks in Wireless Ad Hoc Networks.* In Proceedings of INFOCOM. IEEE, 2003.

41. Adrian Perrig, Ran Canetti, Dawn Song, and J. D. Tygar. *Efficient and Secure Source Authentication for Multicast.* In Proceedings of the 2001 Network and Distributed System Security Symposium, NDSS '01, pages 35.46, February 2001.

42. Bruce Schneier and John Kelsey. *Secure audit logs to support computer forensics*. ACM Transactions on It*formation aml System Security, 2(2):159 176. ACM, May 1999.

# VITA

Yalin Wang was born in QingDao, China, on March, 1974, the son of Hongping Wang and Xinhui Qiao. In 1992, he entered ShanDong University, China, and in 1996 received the degree of Bachelor of Science, majoring in Biology and Biotechnology. After graduation, he entered Institute of Genetics, Chinese Academia. In 1999, he received the degree of Master of Science in Genetics. In spring 2000, he entered the graduate school of Southwest Texas State University, San Marcos, which is now Texas State University-San Marcos.

Permanent Address:

Woluozi, PingDu

QingDao, ShanDong, 266800

P.R.China

This thesis was typed by Yalin Wang.