

# SECURED AD HOC ROUTING PROTOCOL

## A DISSERTATION

*Submitted in partial fulfillment of the  
requirements for the award of the degree*

*of*

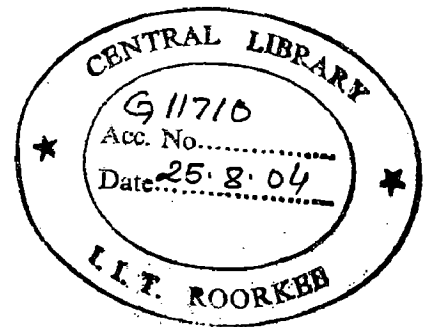
MASTER OF TECHNOLOGY

*in*

INFORMATION TECHNOLOGY

*By*

**M.B.BHAGYA LATHA**



**IIT Roorkee - CDAC, NOIDA,  
c-56/1, "Anusandhan Bhawan"  
Sector 62, Noida-201 307**

**JUNE, 2004**

## CANDIDATE'S DECLARATION

---

I hereby declare that the work which is being presented in this dissertation titled "SECURED AD HOC ROUTING PROTOCOL", in partial fulfillment of the requirement for the award of the degree of **MASTER OF TECHNOLOGY** with specialization in **INFORMATION TECHNOLOGY**, submitted in the Department of Electronics & Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, is an authentic record of my own original work carried out from August 2003 to June 2004, under the guidance and supervision of **Prof. Moinuddin**, Department of Electrical Engineering, Jamia Millia Islamia University, New Delhi.

I have not submitted the matter embodied in this dissertation for the award of any other degree.

Date: 26-6-2004

Place: Noida

  
(M.B. BHAGYA LATHA)


## CERTIFICATE

---

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 26.6.2004

Place: Noida

  
Prof. MOINUDDIN  
Dept. Of Electrical Engineering,  
Jamia Millia Islamia University,  
New Delhi.  
India.

## ACKNOWLEDGEMENT

---

First of all, I would like to thank my guide **Prof. Moinuddin**, Dept. of Electrical Engineering, Jamia Millia Islamia University, who has been having a tremendous influence on my professional development. On top of being a devoted advisor, and a dear friend, he is an excellent teacher far beyond the ordinary. His colorful and creative feedback on my writing style has transformed my dissertation report into excellent shape.

It give me pleasure to take this opportunity to thank and express my gratitude to **Prof. Prem Vrat**, Director, IIT Roorkee, **Shri R.K.Verma**, Executive Director, CDAC, Noida, **Prof. Sarje**, Head Of the Department, I.I.T.Roorkee, **Mr. V.N.Shukla**, Course Coordinator M.Tech(IT), CDAC, Noida and **Prof. R.P.Agarwal**, Course Coordinator M.Tech(IT), IIT Roorkee for providing me excellent academic environment and required lab facilities to undergo my dissertation.

I owe special thanks to my friends, all of my classmates and other friends who have helped me directly or indirectly in the process and contributed towards this dissertation work.

Thanks to my family members for their moral support. Last but not the least, I thank almighty for being on my side from the conception of this dissertation to this implementation.

*M.B. Batha*  
**(M.B.Bhagya Latha)**  
Enroll No:029010

## ABSTRACT

---

Ad hoc networks are a new wireless networking paradigm for mobile hosts. Unlike traditional mobile wireless networks, ad hoc networks do not rely on any fixed infrastructure. Instead, hosts rely on each other to keep the network connected. Although there is a trend to adopt ad hoc networks for commercial uses, the military tactical and other security-sensitive operations are still the main applications of ad hoc networks, due to their unique properties. One main challenge in design of these networks is their vulnerability to security attacks. Providing security support for large ad hoc wireless networks is challenging due to their unique characteristics, such as mobility, channel errors, dynamic node joins and leaves, and occasional node break-ins. In this work, these characteristics are exploited and presented a new design that supports ubiquitous security for mobile nodes, which scales to network size, and is robust against adversary break-ins to make any routing protocol on it secured. In this design, the functionality of conventional security servers, specifically the authentication services is distributed, so that each individual node can potentially provide other nodes certification services. Centralized management is minimized and the nodes in the network collaboratively self-secure themselves. A suit of fully distributed and localized protocols that facilitate practical deployment is proposed. The proposed protocols also feature communication efficiency to conserve the wireless channel bandwidth. It can be combined with any routing protocol to make it more secure. A novel aspect of the proposed mechanism is its independency from both the underlying transport layer protocols and the network layer routing protocols. At the end, a proposal, which describes a solution to the network-layer security in ad hoc networks in the context of AODV routing protocol is proposed. Further simulation and results show that the proposed solution exhibits less overhead compared to conventional mechanisms.

# CONTENTS

---

	Page No.
<b>Candidate's Declaration</b>	<b>ii</b>
<b>Acknowledgment</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>1 Introduction</b>	
1.1 Motivation	1
1.2 Challenges in providing security	1
1.3 Overview of the dissertation	2
1.4 Organization of the dissertation	5
<b>2 Literature Survey</b>	
2.1 Introduction	7
2.2 Mobile Ad hoc networks	7
2.3 Security in Ad hoc networks	8
2.4 Popular network authentication architectures	10
2.4.1 Centralized certification services	11
2.4.2 Hierarchical trust model	12
2.4.3 Pretty Good Privacy	12
2.4.4 Distributed public key model	13
2.5 Summary	14
<b>3 Design of new secured mechanism</b>	
3.1 Introduction	16
3.2 Trust model	16
3.3 System model	17
3.4 Adversary model	18
3.5 Design issues	19
3.6 Summary	20

<b>4</b>	<b>The architecture of the proposed system</b>	
4.1	Introduction	21
4.2	Primitives	21
4.3	Overview of the architecture	23
4.4	Distributed and Localized certification services	24
4.4.1	Certificate issuing / renewal	24
4.4.2	Certificate Revocation and Certificate Revocation List	29
4.4.3	Verifiable partial certificates	32
4.4.4	Important parameters	35
4.5	Distributed Self-initialization	35
4.5.1	Algorithms for Distributed Self-initialization	36
4.5.2	Communication protocol for Self-initialization	38
4.5.3	Verifiable partial share	39
4.6	Scalable share update	40
4.6.1	Algorithms for share update	43
4.6.2	Communication protocol for share update	45
4.7	Summary	46
<b>5</b>	<b>Cryptographic analysis</b>	
5.1	Introduction	48
5.2	Distributed and Localized services	48
5.3	Distributed Self-initialization	50
5.4	Scalable share update	51
<b>6</b>	<b>Self-Organized Solution for Network-Layer Security in Ad hoc Networks</b>	
6.1	Introduction	52
6.2	Overview of the security solution	52
6.3	Security issues in AODV routing protocol	53
6.4	Network-Layer vulnerabilities	54
6.5	Network-Layer security solution	55
6.5.1	Frame work	55
6.5.2	Neighbor verification	57

6.5.3	Security enhanced routing protocol	58
6.5.4	Neighbor monitoring	60
6.5.5	Intrusion reaction	63
6.6	Summary	63
<b>7</b>	<b>Simulation and Results</b>	
7.1	Introduction	64
7.2	Comparison of proposed mechanism with conventional approaches	65
7.2.1	Performance comparison with varying mobility	65
7.2.2	Performance comparison with varying network size	71
7.3	Analysis of proposed algorithms	73
7.3.1	Analysis of Self-initialization algorithm	73
7.3.2	Analysis of Proactive update algorithm	74
<b>8</b>	<b>Conclusion and Future work</b>	<b>76</b>
	<b>References</b>	<b>79</b>

## INTRODUCTION

---

### 1.1 MOTIVATION

In recent years, infrastructureless ad hoc networking technologies such as MANET [1] and Bluetooth [2] have received critical attention in both academic and industry. This emerging technology seeks to provide users "anytime, anywhere" networking services in a potentially large-scale ad hoc wireless network. Mobile users are expected to exchange secure data communications among one another and with the rest of the Internet, at any time, and at any place. Military tactical operations are still the main application of ad hoc networks today. For example, military units (e.g., soldiers, tanks, or planes), equipped with wireless communication devices, could form an ad hoc network when they roam in a battlefield.

This proliferation of Ad hoc networks in a hostile environment like military applications made security as a major concern. Although many solutions exist to provide vigorous security, none of them addressed network performance aspect. In an attempt to provide ubiquitous security for routing, a protocol, which is both security concerned and network performance-centric solution, is proposed and implemented.

### 1.2 CHALLENGES IN PROVIDING SECURITY

Ad hoc networks can also be used for emergency, law enforcement, and rescue missions. Since an ad hoc network can be deployed rapidly with relatively low cost, it becomes an attractive option for commercial uses such as sensor networks or virtual classrooms. The growing commercial and military deployments of these networks have made security design increasingly important.

Providing security support for ad hoc wireless networks is challenging for a number of reasons:

- Wireless networks are susceptible to security attacks ranging from passive eavesdropping to active interfering and denial-of-service (DoS) attacking.
- Occasional break-ins in a large-scale mobile network are inevitable over a large time interval.
- Ad hoc networks provide no infrastructure support.



- Mobile nodes may constantly leave or join the network.
- Mobility-induced wireless link breakage/reconnection and wireless channel errors make timely communications over multihop highly unreliable.
- A scalable solution is a must for a large-scale network.

Therefore, adequate security support for authentication, confidentiality, integrity, nonrepudiation, access control and availability is critical to deploying this wireless networking technology in commercial environments.

Several recent research [4, 1, 2] have started to address security issues in such networks. While these early proposals each have their own merit, they mainly focus on the security vigor of the design and leave the *network performance* aspect largely unaddressed. As a result, these solutions may be extremely secure from the cryptographic standpoint, but their real performance when deployed in the network is unclear. This concern is further aggravated by the unique characteristics of ad hoc networks, such as highly dynamic network topology, frequent node arrival/ departure, and bandwidth-constrained wireless links. In this work, I shift my main attention from the cryptography-centric design approach to a more network centric design scheme, and focus on the practical network performance aspect of the security design. The main goal is to develop *network performance-centric* security solutions that effectively balance security strength and network performance in practice. Node authentication is mainly focused on, which is very important for trusted communication of routing messages

### 1.3 OVERVIEW OF THE DISSERTATION

In this dissertation, It is proposed to provide ubiquitous, robust, and scalable security services for mobile ad hoc wireless networks for secured routing. The proposed design has been driven by the following four main goals:

- **Ubiquitous service availability:** Mobile users may freely roam inside the network. So security service must be available everywhere and be robust against potential DoS attacks.
- **Robustness against break-ins:** Complete intrusion-free systems are expected to be costly and unrealistic. The design seeks to work in the presence of break-ins. The overall system security should not be compromised if the break-ins are under a certain threshold.

- **Scalability:** A wireless mobile network may consist of a large number of networking nodes. The network size may constantly change as nodes leave and new nodes join. The design should scale to the network size.
- **Communication efficiency:** Wireless channel is bandwidth constrained and error-prone. Routing in infrastructureless ad hoc wireless networks is unreliable due to the node mobility and link breakage/ reconnection [8]. The design should be communication efficient to conserve the wireless channel bandwidth. The protocols should work without assumptions on the reliability of the underlying transport layer protocols or routing mechanisms.

In essence, the mobile user is ensured to provide anywhere, anytime, always -available services. The proposed design should be robust against Denial of Service (DoS) attacks and break-ins; it has to be scalable and works well in a highly dynamic mobile network.

Confidentiality, authentication, integrity, non-repudiation, access control and availability are considered as the main services of a security system [9]. Among these services authentication has been identified as the bottleneck for transmission of routing messages. The compromise of the authentication service breaks down the whole security system, and service provider cannot proceed to provide the other services without the valid identities of communicating nodes being successfully established [35]. In this dissertation, the authentication service is mainly focused on in ad hoc wireless networks. This work is based on asymmetric cryptographic techniques, specifically the *de facto* standard RSA [20] algorithms. Once the authenticated channels are established with proper access control between communicating parties, confidentiality, integrity and non-repudiation can be further realized by following the typical Diffie-Hellman key exchange protocols [29].

The security scheme has several techniques to achieve the above design goals:

- Ubiquitous authentication service availability is provided by taking a *certificate-based* approach. In the proposed design, any two communicating nodes establish a temporary trust relationship via globally verifiable certificates. With a *scalable threshold sharing of the certificate signing key*, further certification services is distributed, such as certificate issuing/renewal and revocation, to each node in the network. Not a single node holds the complete certificate signing key. Each node only possesses a share of it. While no single node has the power of providing full

certification services, multiple nodes in a network locality can collaboratively provide such services that are the same as if an authority with a complete certificate signing key would do.

- By the distributed certification services, together with the further enhancement of a *scalable proactive update mechanism*, it ensures service robustness in the presence of short-term computation bounded adversaries (as discussed in section 3.4).
- While the certification service distribution and periodical proactive update can be solved in theory using known cryptographic techniques such as threshold secret sharing [10], threshold multisignature [31], and proactive RSA [21], the approach taken in this work is to focus on *scalable and practical solutions in large-scale ad hoc networks with dynamic node membership*. In this dissertation, a suit of new algorithms is proposed that are *fully distributed and localized*, based on but different from existing works [10, 30, 25, 31, 34, 36, 21, 22] to achieve this goal.
- The proposed fully localized (typically within one-hop neighborhood) protocols further achieve communication efficiency and load-balancing over the network to avoid network congestions. Through the localized design, the proposed communication protocols makes the underlying transport layer protocols and routing mechanisms secured.

Furthermore, the design has two additional features:

- **Provable cryptographic security:** The proposed security algorithms are as secure as the underlying cryptographic primitives (e.g., RSA) by the simulatability arguments.
- **Self-defensive, built-in detection mechanisms.** While the proposed design works with any intrusion detection algorithms and mechanisms that are of each individual node's choice, verifiable techniques [12, 14] is applied as built-in mechanisms to detect adversaries that attack the security protocols.

The proposed design features the fully distributed and localized algorithms and protocols. These properties comply with the ad hoc nature of the infrastructureless wireless networks, which are critical to practical deployments. While the main focus in this dissertation has been on ad hoc networks, the application of design is not limited in the specific scenario. The general architecture can be potentially applied in

other systems where the cost of centralized management and maintenance outweighs the benefits.

The main contributions in this work are summarized as follows:

- A localized trust model that characterizes the localized nature of an individual node's security concerns in large ad hoc wireless networks. The new model and its realization provide another option for large systems that lack centralized security management, or the cost of centralized approaches outweighs the benefits.
- A certificate-based, distributed authentication architecture that supports ubiquitous service availability for mobile nodes and robustness against DoS attacks and break-ins. It scales to the network size and network dynamics. The protocols are communication efficient and the load is balanced over the network.
- A suite of fully distributed and localized algorithms and protocols that consist of certificate issuing/renewal and revocation, self-initialization, share updates, and verifiable mechanisms.

## 1.4 ORGANISATION OF THE DISSERTATION

The rest of the dissertation is organized as follows:

**Chapter 2** Provides an overview of the literature survey carried out for the dissertation. It discusses MANET, routing protocols, its security issues and popular network authentication architectures.

**Chapter 3** Defines the assumed trust model, system model, adversary models and discusses some design issues.

**Chapter 4** Outlines the overall proposed architecture that provides ubiquitous, scalable and robust authentication services. It discusses the proposed distributed and localized certification services, presents the new self-initialization algorithms and protocols, propose scalable share update enhancements.

**Chapter 5** Details the Cryptographic analysis of proposed algorithms.

**Chapter 6** Contains a proposal, which is a self-organized security solution for network layer in ad hoc networks.

**Chapter 7** Evaluates the proposed mechanism and compares with centralized and hierarchical centralized authority's in terms of overhead, success ratio, delay etc.

**Chapter 8** Contains conclusion and future work.

**LITERATURE SURVEY**

---

**2.1 INTRODUCTION**

This chapter provides an overview of the literature survey carried out for the dissertation. A Mobile Ad Hoc Network is an infrastructureless wireless network in which the nodes act as both hosts and routers. Due to mobility of the nodes the topology of the network will dynamically change making wired routing useless in this environment. Thus IETF has designed various routing protocols according to characteristics of MANET. Due to lack of infrastructural support, they are susceptible to wireless link attacks hence security in ad hoc network is a vital concern. But due to unique characteristics of ad hoc networks, providing security is challenging. Availability, confidentiality, integrity, authentication, and non-repudiation are considered to be parameters to secure an ad hoc network. Among these, authentication is identified as bottleneck in making any routing protocol secured. To provide these authentication services, we have centralized servers, hierarchical servers etc., but, these are not scalable with the size of the network and are exposed to single points of compromises and failures. The following sections elaborates the above points in detail.

**2.2 MOBILE AD HOC NETWORKS**

A Mobile Ad Hoc Network (MANET) is a group of wireless mobile nodes forming a dynamic network using a fully mobile infrastructure. The nodes communicate with each other without the help of a base station and serve as router and hosts to each other, exchanging data between them to form a network. Mobile nodes that are within each other's radio range communicate directly via wireless links, while those that are far apart rely on other nodes to relay messages as routers. Due to the mobility of the nodes, the topology of the network may rapidly be changing, making it impossible to use conventional routing tables maintained at fixed points (routers). Instead, each node is required to determine the best route to a given destination node by itself. Given their dynamic nature, route discovery in a MANET differs significantly from the more or less static routes in wired networks. Not all nodes in a MANET

necessarily have the same capabilities. Two nodes, even if they are direct neighbors, may differ with respect to signal strength, available power, reliability etc.

The characteristics of Ad-hoc networks are:

- Independent Mobile-users.
- No Centralized Connectivity
- Dynamic Topology
- Bandwidth Constrained
- Energy Constrained
- Limited Physical Connectivity
- Unidirectional Links

These salient characteristics of MANET make the traditional fixed-network routing protocols inadequate. The IETF MANET Working Group has researched and developed a number of protocols for mobile ad-hoc networks. These protocols can generally be categorized into two groups: *pro*-active and *reactive* protocols.

Pro-active or table-driven protocols, in order to maintain the constantly changing network graph due to new, moving or failing nodes, require continuous updates, which may consume large amounts of bandwidth – clearly a disadvantage in the wireless world, where bandwidth is often sparse. The family of Distance-Vector protocols, including Destination-Sequenced Distance-Vector Routing, falls into the category of pro-active protocols.

In contrast, reactive protocols determine the proper route only when required, that is, when a packet needs to be forwarded. In this instance, the node floods the network with a routerequest and builds the route on demand from the responses it receives. This technique does not require constant broadcasts and discovery, but on the other hand causes delays since the routes are not already available. These reactive (or on-demand) protocols include Dynamic Source Routing (DSR) [8] and Ad-hoc On demand Distance Vector Routing (AODV) [48], as well as the classical flooding algorithms.

## 2.3 SECURITY IN AD HOC NETWORKS

The build up of ad hoc network can be envisaged where support of wireless access or wired backbone is not feasible. Ad hoc wireless network does not have any predefined infrastructure and all network services are configured and created on the fly. Thus it is

obvious that with lack of infrastructural support and susceptible wireless link attacks, security in ad hoc network becomes inherent weakness. Achieving security within ad hoc networking is challenging due to following reasons [34]:

➤ *Dynamic Topologies and Membership*

A network topology of ad hoc network is very dynamic as mobility of nodes or membership of nodes is very random and rapid. This emphasizes the need for secure solutions to be dynamic

➤ *Vulnerable wireless link*

Passive/Active link attacks like eavesdropping, spoofing, denial of service, masquerading, impersonation are possible

➤ *Roaming in dangerous environment*

Any malicious node or misbehaving node can create hostile attack or deprive all other nodes from providing any service.

Nodes within nomadic environment with access to common radio link can easily participate to set up ad hoc infrastructure. But the secure communication among nodes requires the secure communication link to communicate. Before establishing secure communication link the node should be capable enough to identify another node. As a result node needs to provide his/her identity as well as associated credentials to another node. However delivered identity and credentials need to be authenticated and protected so that authenticity and integrity of delivered identity and credentials cannot be questioned by receiver node. Every node wants to be sure that delivered identity and credentials to recipient nodes are not compromised. **Therefore it is essential to provide security architecture to secure ad hoc routing.**

The above mentioned *identification* problem simultaneously leads to *privacy* problem. In general mobile node uses various types of identities and that varies from link level to user/application level. Also in mobile environment very frequent mobile node is not ready to reveal his/her identity or credentials to another mobile node from privacy point of view. Any compromised identity leads attacker to create privacy threat to user device. Unfortunately the current mobile standards do not provide any location privacy and in many cases revealing identity is inevitable to generate communication link. Hence a seamless privacy protection is required to harness the usage of ad hoc networking.

To secure an ad hoc network, we consider the following attributes: availability, confidentiality, integrity, authentication, and non-repudiation.



*Availability* ensures the survivability of network services despite denial of service attacks. A denial of service attack could be launched at any layer of an ad hoc network. On the physical and media access control layers, an adversary could employ jamming to interfere with communication on physical channels. On the network layer, an adversary could disrupt the routing protocol and disconnect the network. On the higher layers, an adversary could bring down high-level services. One such target is the key management service, an essential service for any security framework.

*Confidentiality* ensures that certain information is never disclosed to unauthorized entities. Network transmission of sensitive information, such as strategic or tactical military information, requires confidentiality. Leakage of such information to enemies could have devastating consequences. Routing information must also remain confidential in certain cases, because the information might be valuable for enemies to identify and to locate their targets in a battlefield.

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

*Authentication* enables a node to ensure the identity of the peer node it is communicating with. Without authentication, an adversary could masquerade a node, thus gaining unauthorized access to resource and sensitive information and interfering with the operation of other nodes like wrong route updates, packet dropping etc.,

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

Among these services 'authentication' has been identified as the bottleneck. The compromise of the authentication service breaks down the whole security system, and we cannot proceed to provide the other services without the valid identities of communicating nodes being successfully established[35]

## **2.4 POPULAR NETWORK AUTHENTICATION ARCHITECTURES**

There are many popular network authentication architectures in the literature. Some of them include Kerberos [3], standard X.509 [4], PKIX [5] and PGP[6, 7].

## 2.4.1 CENTRALIZED CERTIFICATION SERVICES

In Centralized Certification architectures like Kerberos, X.509, and PKIX standards, two communicating entities authenticate each other via a globally trusted centralized Certification Authority (CA). The CA has a public/private key pair, with its public key known to every node, and signs certificates binding public keys to nodes. The CA has to stay on-line to reflect the current bindings, because the bindings could change over time as the public key should be revoked if the owner node is no longer trusted or is out of the network; a node may refresh its key pair periodically to reduce the chance of a successful brute-force attack on its private key. While the architecture works fine in a wired network of manageable scale, it fails in a large-scale ad hoc wireless network for several reasons.

### Limitations of Centralized Authentication Services

- Centralized approaches are generally not scalable. The cost of maintaining such centralized servers for a large system without enough infrastructure support may outweigh the benefits.
- The CA servers are exposed to single points of compromises and failures. They are inviting targets of the DoS attacks: simply jamming the wireless channel around the locality of the authentication servers can effectively isolate these servers to make the service unavailable.
- Frequent host mobility leads to frequent mutual authentications, which aggravate the scalability issue. It may also cause severe channel congestions around the CA servers.
- High mobility causes frequent route changes, thus locating and contacting a CA server in a timely fashion is non-trivial [8].
- Multihop communication over the error-prone wireless channel exposes the data transmission to high loss rate that leads to significantly high average latency.
- If the CA is compromised and leaks its private key to an adversary, the adversary can then sign any erroneous certificate using this private key to impersonate any node or to revoke any certificate.

## 2.4.2 HIERARCHICAL TRUST MODEL

In the hierarchical model, the entire network is logically partitioned into domains where local CA's are deployed. At a first glance, this scales to network size and fits well in large wireless network. Several characteristics of mobile networks make this approach ineffective. In this model root certificate authority (CA) issues certificates to delegated CAs or end users. The CAs in turn issue certificates to end users or to other CAs. PKI X.509 (PKIX) [4] exemplifies this trust model.

### Limitations of Hierarchical Trust model

- High mobility causes frequent route changes, thus contacting the local CA in a timely fashion is non-trivial. Besides, in ad-hoc networks the local CA may be multi-hops away and may also move. This not only causes complicated dynamic repartitioning of the network, but also stretches the problem of locating and tracking a local CA server.
- Multi-hop communication over the error-prone wireless channel exposes data transmission to high loss rate. This reduces the success ratio and increases the average latency.
- Every local CA is exposed to single point of compromises or DoS attacks.

Hierarchical CA's cannot solve the problem of ubiquitous service availability and robustness. There are also several proposals to build logical infrastructure for ad hoc networks such as clustering [39] and virtual backbone [40, 41]. However, the overhead of these protocols themselves and how these virtual infrastructures can help instantiation of other services are not fully justified for deployment in mobile and highly dynamic environments.

## 2.4.3 PRETTY GOOD PRIVACY

PGP [6,7] follows a "web-of-trust" authentication service model that is similar to the proposed design. In the web-of-trust model, there is no distinction between a CA and an end user. End users are responsible for certificate management. End user tasks include: issuing and revocating of certificates, and vouching for the credibility of other users. In PGP each node signs other nodes' certificates based on certain policy. A node accepts a certificate if the certificate bears a threshold number of signatures from nodes that is already known to be trustworthy.

### **Limitations of Pretty Good Policy(PGP)**

- It does not scale beyond a relatively small community of trusted individuals.
- In a mobile system, any two nodes will potentially meet, communicate with, and route packets for each other. It would be difficult for each node to maintain a long list of trusted friends, potentially as large as the list contains all nodes in the whole network.

### **2.4.4 DISTRIBUTED PUBLIC KEY MODEL**

This model makes use of threshold cryptography to distribute the private key of the Certification Authority over a number of servers. Security function sharing has been a very active research area in cryptography research [12, 21, 22, 25,26, 27, 30, 31, 35]. By distributing the functionality of the centralized CA server among a group of servers, the availability of such services is improved. The single point of failure can also be avoided. Threshold secret sharing [10] serves as a basic primitive for function sharing. The concept of proactive secret sharing [11] is introduced to further improve the robustness of the threshold secret sharing by periodically updating the secret shares. However, the focus of these works is to maximize the security of the shared secret in the presence of possible compromises of the secret share holders. They typically target a group of a few servers with rich connectivity. Hence the proposed algorithms do not scale beyond a relatively small server group.

#### **Limitations of Distributed Public Key Model**

- The server group is still an inviting attack target since the compromises of the servers group effectively compromise the security services of the whole system. For the adversary that issues DoS attacks, it is more difficult to jam the channels around a group of servers than jamming the channel around a single server, but still not prohibitively impossible due to the limited size of such server group. This is true especially when several conspired adversaries collaboratively issue DoS attacks.
- Besides, in a large-scale network, relying on such a group of servers for services still suffer from all these communication issues as in the case of a single CA server.

These works motivated the new design, but extends the idea one step further for large-scale ad hoc wireless networks. The proposed algorithms and protocols are scalable to

distribute the security services, specifically certification services into each node. There is no differentiation of servers and clients in the new system any more: a threshold number of any nodes can collaboratively act as servers to provide certification services for other nodes. The service is ubiquitously available for any mobile node as long as it can locate a coalition of enough number of nodes, typically in its one-hop neighborhood, even including itself. No communication over multihop is involved. The effort and complexity of locating and contacting the service providers are minimized. The localized design also makes the underlying wireless transport layers and the routing protocols more secured. Furthermore, the impacts of DoS attacks and adversary break-ins are minimized and isolated. *Achilles' Heel* is completely eliminated from the system.

There are several recent researches on security in wireless networks. [18] proposes a Kerberos-based solution to authentication for mobile users in wireless cellular networks. [23] studies the problem of authentication in PCS. [34] directly applies the threshold secret sharing and proactive secret share updates mechanisms in a group of "special nodes" or servers in the ad hoc networks to increase service availability and robustness. It is assumed that communication links are reliable, which does not hold in typical mobile wireless networks. [15] studies the problem of intrusion detection in ad hoc networks. While intrusion detection is out of the scope of this dissertation, the proposed algorithms are self-defensive to detect potential attacks on this new security system. The proposed design works with any intrusion detection mechanisms. The detection mechanism are not specified but left it to each individual node's choice. In this work it is only assumed that nodes are equipped with some detection tools, such as the simple "passive acknowledgment" technique as proposed in [16], to detect their one-hop neighbors' abnormal behaviors for further isolation.

## 2.5 SUMMARY

Mobile ad hoc network is a dynamic infrastructureless wireless network which is known for providing 'any time, any where' services. The characteristics of ad hoc networks on one-side provide nomadic computing and on other side they became challenges for providing security and Quality of Service. The deployment of these ad hoc networks at various hostile environments made the security essential. Any security solution should have the attributes like Availability, Confidentiality,

Integrity, Authentication and Non-repudiation; Among these, authentication is identified as bottleneck. To provide these authentication services we have centralized, hierarchical approaches. But these are not scalable with the size of the network and are exposed to single points of compromises and failures.

---

**DESIGN OF NEW SECURED MECHANISM**

---

**3.1 INTRODUCTION**

This chapter views the proposed design from its underlying trust model, system model and adversary model. A localized trust model was proposed in which any entity is trusted if any neighboring  $k$  trusted entities claim so. Trust management and maintenance overhead is distributed in both space domain and time domain. The system model assumes that in a network the number of nodes is dynamic, they may join, leave, fail communicating through bandwidth-constrained, error-prone and insecure wireless channel. The adversary is modeled with different computational power as Long-term and short-term adversaries depending upon duration of their attack. The following sections elaborate the above model in detail.

**3.2 TRUST MODEL**

A localized trust model was proposed to characterize the localized nature of security concerns in large ad hoc wireless networks. The proposed trust model and its implementations provide another option for large systems that lack centralized security enforcement, or the cost of centralized management outweighs the benefits.

In the dominating TTP (Trusted Third Party) trust model [37], an entity is trusted if a trusted central authority (CA) claims so. These central authorities arbitrate the trust by signing certificates or tickets. While the implementations of the TTP model features the efficiency and flexibility as a centralized system, it also suffers from the scalability and robustness issues.

PGP(Pretty Good Privacy) “web-of-trust” model [6, 7] is different from the TTP model in the sense each entity manages its own trust based on direct recommendation. The authors of [38] seek to quantify the trust and recommendation to reason trustworthiness on line. The proposed localized trust model is similar to these models in the sense that they all distribute the centralized trust to realize a democratic trust environment.

In general, an entity is trusted if any  $k$  trusted entities claim so in the proposed localized trust model. These  $k$  trusted entities are typically the neighboring nodes of the entity. A locally trusted entity is globally accepted and a locally distrusted entity is

regarded untrustworthy anywhere.  $k$  is a system-wide parameter that sets the global acceptance criteria. It should be honored by each entity in the system. By holding trust on any  $k$  entities of the system, the scalability issue of previous distributed trust models [7, 37, 38] is avoided. In the proposed trust model, each entity contributes to the trust system by local efforts on its neighborhood. Trust management and maintenance overhead is distributed in both space domain (locally) and time domain (on-line). This property is particularly appropriate for a large dynamic ad hoc wireless network, where centralized trust management is difficult or expensive. Besides, an ad hoc-networking node typically cares the trustworthiness of their immediate neighbors most due to the broadcast nature and the inherent local interactions of wireless transmissions. The node has to rely on its neighboring nodes for packet forwarding, routing and other network resource access. In the mean time, the inherent local interactions also provide opportunities for nodes to locally contribute to the overall trust system by monitoring and certifying their neighbors, which is critical for practical realization of the trust model.

### 3.3 SYSTEM MODEL

A dynamic ad hoc wireless network with  $n$  networking hosts/nodes is considered. Nodes communicate with one another via the bandwidth-constrained, error-prone, and insecure wireless channel. Nodes may freely roam in the network. The number of networking nodes  $n$  may be dynamically changing because mobile hosts may join, leave, or fail over time. Besides,  $n$  is not constrained; there may be a large number of networking nodes. The network provides neither physical nor logical infrastructure supports. The nodes may be equipped with wireless transport layer protocols [42, 43, 45] and the network may adopt some ad hoc routing protocols [44] to enable multihop communication. However, the reliability of multihop packet forwarding that based on these supports is not assumed.

In the proposed architecture, the following six assumptions are made. (1) Each node  $i$  has a unique nonzero ID  $v_i$ , such as its MAC layer address. (2) Each node has some one-hop neighborhood discovery mechanism. (3) Communication between one-hop neighboring nodes is more reliable compared with multihop packet forwarding, since the exposure to wireless channel error and interference is limited and no ad hoc routing is involved. (4) Each node has at least  $k$  one-hop legitimate



neighboring nodes, or the network has a minimum density of well-behaving nodes. (5) The mobility is characterized by a maximum node moving speed  $S_{max}$ . (6) Each node is equipped with some detection mechanism to identify misbehaving nodes among its one-hop neighborhood.

The first two assumptions are based on the features of IEEE standard 802.11 MAC protocols [46] that is adopted in dominant commercial products such as WaveLAN and Airport interfaces. It is observed that nodes are connected with multiple routes in typical ad hoc networks. It is identified as the inherent connectivity redundancy of ad hoc networks [34]. The assumption on minimum node degree seeks to quantitatively characterize this advantage. Finally, each node employs some local detection mechanism to monitor its one-hop neighbors' behavior is assumed. The assumption is based on another observation that although intrusion detection in ad hoc networks is generally much more difficult than in wired networks [15], monitoring and detecting misbehaviors or attacks among one-hop neighboring nodes are readily easier and more practical [16], since each wireless transmission is a broadcast among the one-hop neighborhood and local interactions are inherent features of wireless networks.

### **3.4 ADVERSARY MODEL**

An adversary is a malicious node that uses every available means to break in (such as node compromises) or shut down (such as DoS attacks on servers) the enforced security system. In the proposed design, it is assumed that the underlying cryptographic primitives such as RSA are practically secure in term of the computation power of the adversary. However, occasional break-ins is allowed through factors such as insecure OS, software bugs and backdoors [17], etc.

When a networking node is compromised, all its information, public or private, is exposed to the adversary. These information includes the node's private key that corresponds to its certified public key, and its share of the certificate signing key (as defined in Section 4.4.1). The adversary may control or impersonate the compromised nodes with these information to further attack other nodes or the security system. The adversary may record all the information of its victims for later reference.

Several adversaries may also conspire into a group to combine their computation power and share their victims. For ease of presentation, such an adversary group is denoted by a single adversary. The following two models is employed as proposed in [11] to characterize adversaries of different power:

- **Model I: (Long-term constrained adversary):** During the entire lifetime of the network, the adversary cannot break or control  $k$  or more nodes.
- **Model II: (Short-term constrained adversary):** Assume time is divided into intervals of length  $T$ . During any time interval, the adversary cannot break or control  $k$  or more nodes.

Although at any time constant it cannot break or control  $k$  or more nodes, the adversary of model II can choose its victims at the beginning of each time interval. As time goes on each node in the network can be broken by the adversary during some time interval. It is easy to see that, model II defines more powerful adversaries than model I. In this work, the distributed authentication services that are defensive to model I adversaries are proposed, and then improved with scalable share update techniques to handle adversaries of model II. The parameter  $k$  is chosen to be consistent with my trust model as defined in Section 3.2.

### 3.5 DESIGN ISSUES

The issues that are addressed in this work are summarized as follows.

- ***Liability to node break-ins:*** Wireless transmissions are prone to security attacks. It is very likely that adversaries will eventually break into a limited number of nodes over a large time window.
- ***Mobility and channel errors:*** Mobile nodes incur dynamic topology changes. A mobile user may not be able to perform effective and timely communications with a remote node except with its local neighbors (e.g., DSR works for less than 10-hop scenarios [44]). Moreover, wireless channel errors also cause multihop communication highly unreliable.
- ***Network scale and dynamic membership:*** The number of networking nodes can be large and under constant change as nodes leave or fail and new nodes join in over time. The design has to be scalable for practical deployment.

- *Constrained communication bandwidth:* Wireless channel bandwidth is a scarce resource. The protocols should be communication efficient to conserve the limited wireless channel bandwidth.
- *No infrastructure support:* By the ad hoc nature of the networks of interest, there is no infrastructure support available, either physical or logical.
- *Ubiquitous service availability and robustness:* Mobile users demand ubiquitously available security services. The service should also survive certain degree of denial of service attacks.

The fundamental problem is that, to provide security services, specifically authentication services that support ubiquitous availability for mobile users in the presence of DoS attacks and occasional adversary break-ins, the protocols have to be communication efficient and work in a large-scale, high dynamic wireless network, which may not have any infrastructure support.

### 3.6 SUMMARY

The centralized authorities like dominating TTP(Trusted Third Party) arbitrate the trust by signing certificates. While the implementations of the TTP model features the efficiency and flexibility as a centralized system, it suffers from the scalability and flexibility issues. The proposed localized trust model characterizes the localized nature of security concerns in large ad hoc wireless networks. In this model each entity contributes to the trust system by local efforts on its neighborhood. By holding trust on any  $k$  entities of the system, the scalability issue of previous distributed trust models is avoided. In my system model the number of networking hosts/nodes is not constrained; there may be large number of networking nodes. The assumptions are based on the features of IEEE standard 802.11 MAC protocols that is adopted in dominant commercial products such as WaveLAN and Airport interfaces. The adversary model characterizes adversaries of different power into two models- Model I (long-term constrained adversary) and Model-II (short-term constrained adversary). The proposed distributed authentication services are defensive to model I adversaries and scalable share update techniques handle adversaries of model-II.

## THE ARCHITECTURE OF THE PROPOSED SYSTEM

---

### 4.1 INTRODUCTION

This chapter presents overall architecture of the proposed system that provides ubiquitous, scalable and robust authentication services. In the proposed architecture, each node carries a certificate signed by a secret key SK. The security of SK is protected by the k-threshold polynomial sharing mechanism. At boot strapping this SK is issued by a dealer and then the network progressively self-initializes using self-initialization algorithm. In order to realize this localized certification services many distributed algorithms are proposed for certificate issuing/renewal, certificate revocation and distributed CRLs. The following sections discusses above algorithms in detail along with the communication protocols.

### 4.2 PRIMITIVES

a) **RSA** In the proposed architecture, each networking node  $i$  with nonzero ID  $v_i$  is associated with a personal RSA key pair  $\langle \overline{sk}_i, \overline{pk}_i \rangle$ .  $\overline{sk}_i$  denotes  $v_i$ 's private key for decryption and signing.  $\overline{pk}_i$  denotes  $v_i$ 's public key for encryption and verification. The private key  $\overline{sk}_i$  is only possessed by node  $v_i$  for two typical usage. One is for  $v_i$  to decrypt messages that are encrypted by the corresponding public key  $\overline{pk}_i$ . The other is for node  $v_i$  to sign some messages or statements to generate a signature. The public key  $\overline{pk}_i$  is advertised to other nodes. It is used for other nodes to encrypt messages toward  $v_i$ , and to verify a signature that is supposedly signed by node  $v_i$  with its private key  $\overline{sk}_i$ .

b) **Authentication via certificates** Authentication in the RSA context relies on two factors. For node  $v_i$  to authenticate itself to another node, it has to prove its knowledge of the private key  $\overline{sk}_i$ , and the association between itself and the advertised public key  $\overline{pk}_i$ . A challenge/response protocol can be followed to prove the knowledge of the private key  $\overline{sk}_i$ . The association is proved by a certificate.

Generally a certificate  $CERT_i$  is a statement  $cert_i$  that is signed by some trusted authority's private key  $SK$ . The statement  $cert_i$  may read: "It is certified that the personal public key of node  $v_i$  until time  $t$  is  $\overline{pk}_i$ ." In RSA,  $CERT_i = (cert_i)_{SK}$ . The certificate does not need to be private. It is actually made public by some directory service or being carried by node  $v_i$ . The authority's public key  $PK$  is assumed to be well known. Each node can verify the validity of  $v_i$ 's certificate  $CERT_i$  by applying  $PK$  to check if  $cert_i = (CERT_i)_{PK}$ . A valid certificate proves the association between node  $v_i$  and its public key  $\overline{pk}_i$ .

c) **Certification services** As identified by [47], certification services include issuing/renewing certificates, revoking certificate, storing and retrieving certificates and certificate revocation lists (CRLs). Each certificate is stamped with an expiration time. Nodes have to renew and get a new certificate before expiration. Certificates may become invalid before expiration. The revocation service has to put such information on the revoked certificates into CRLs for queries from nodes. The focus of this dissertation is to provide ubiquitous certification services to mobile nodes in an ad hoc wireless network.

d) **Polynomial secret sharing** The proposed design makes extensive use of the polynomial secret sharing proposed by Shamir [10]. A secret, specifically the certificate signing key  $SK$ , is shared among all  $n$  nodes in the network. To distribute  $SK$ , a dealer chooses a polynomial of order  $k-1$ :  $f(x) = SK + f_1x + \dots + f_{k-1}x^{k-1}$ . Coefficients  $f_1, f_2, \dots, f_{k-1}$  are uniformly distributed over a finite field. Node  $v_i$  gets its share  $P_{v_i} = f(v_i)$  privately from the dealer. Any coalition of  $k$  nodes  $\{v_1, v_2, \dots, v_k\}$  can potentially recover  $SK$  by Lagrange

interpolation:  $SK = f(0) = \sum_{i=1}^k l_{v_i} P_{v_i}$  where  $l_{v_i} = \prod_{j=1, j \neq i}^k \frac{v_j}{v_j - v_i}$ . No coalition up to  $k-1$

nodes yields any information about  $SK$ . This mechanism is robust against the attacks of the adversaries of model I as defined in Section 3.3.

e) **Proactive secret share update** There are a lot of candidate polynomials that can be applied to share  $SK$  in the polynomial secret sharing context. To further defend the privacy of the shared  $SK$  against the adversaries of model II (as discussed in Section

3.4), Herzberg et al. proposed the mechanism that periodically updates the secret shares with different polynomials [11]. This technique is applied with scalable algorithms to further improve the robustness against model II adversaries.

### 4.3 OVERVIEW OF THE ARCHITECTURE

In the proposed architecture, each node carries a certificate signed by  $SK$ . The corresponding  $PK$  is assumed to be well known, so that certificates are globally verifiable. Nodes without valid certificates will be isolated, that is, their packets will not be forwarded by the network. Essentially these nodes without valid certificates are treated the same as adversaries. They are denied from access to any network resources. When a mobile node moves to a new location, it exchanges certificates with its new neighbors and goes through mutual authentication processes to build trust relationships. Neighboring nodes with such trust relationship help each other forward and route packets. They also monitor each other to detect possible attacks and break-ins. Specific monitoring algorithms and mechanisms are left to each individual node's choice.

Each certificate is stamped with an expiration time. Nodes have to renew and be issued a new certificate upon the expiration of its old certificate. In the centralized authentication architecture, nodes have to contact a CA server for this service. In the proposed architecture, the private key  $SK$  is distributed that is used to sign certificates into each node of the network by a polynomial of order  $k-1$ . A node  $v_i$  with its old certificate expiring requests a new certificate from any coalition of  $k$  nodes, typically among its onehop neighbors. A neighboring node checks its record on  $v_i$  that requests certification services. If its record shows  $v_i$  a well-behaving legitimate node, it returns a "partial" certificate by applying its share of  $SK$ . Otherwise the request is dropped. By collecting  $k$  partial certificates,  $v_i$  combines them together to generate the full new certificate as if it were from a CA server. A misbehaving or broken node that is detected by its neighbors will be unable to get a new certificate. It will be cut off from the network at the expiration of its current certificate. Moreover, Explicit Certificate Revocation and distributed CRL storage/retrieval mechanisms are proposed to deal with the scenario when a node is detected misbehaving or broken well before its current certificate's expiration time.

A valid certificate in the new system represents the trust of a coalition of  $k$  nodes. A node with a valid certificate is globally trusted. And a node without a valid certificate or with its certificate revoked will be cut off the network. Besides, each node contributes to the overall trust management and maintenance by certifying its neighboring nodes. By this means, the localized trust model is realized as defined in Section 3.2. A network of legitimate nodes only is enforced. Adversaries or compromised nodes will be effectively isolated once detected. Their impact on the overall network is localized and minimized.

The security of the certificate-signing key  $SK$  is protected by the  $k$ -threshold polynomial sharing mechanism. It is robust against adversaries of model I as defined in Section 3.4. The security of  $SK$  is further enhanced by employing a scalable share update algorithm. Each node's share of  $SK$  is periodically updated to defend against model II adversaries.

At the bootstrapping phase of the network, a dealer has to send each networking node privately its share of the  $SK$ , according to a polynomial of order  $k-1$ . This process is denoted as the "initialization" of nodes. A large ad hoc wireless network may contain hundreds or even thousands of networking nodes. Just relying on the dealer for the initialization, as most works in threshold secret sharing do [10, 11, 12, 21, 27, 31, 34, 36] is not scalable and may not be feasible. Moreover, new nodes may join over the lifetime of the network. Maintaining a dealer on line to handle future node joins would compromise the overall system robustness and security. The dealer would become the single point of failure and inviting target of DoS attacks. Since the dealer is the only entity that holds the complete  $SK$ , maintaining a dealer on line also increases the adversaries' chance to compromise the dealer and therefore  $SK$ , thus effectively turn down the whole services. These issues are addressed by a scalable initialization mechanism called "*self-initialization*".

In the proposed architecture, the dealer is only responsible to initialize the very first  $k$  nodes, no matter how large the network would be. The initialized nodes collaboratively initialize other nodes, typically their neighboring nodes. Repeating this procedure, the network progressively "self-initializes" itself. The same mechanism is applied when a new node joins. The node need only contact its neighbors. Each of its neighbor returns a "partial" share of  $SK$ . By collecting  $k$  such partial shares and combining them together, the node is initialized with its full share.

By distributing the certification services into each node's one-hop neighborhood, ubiquitous service availability realized for mobile nodes and robustness against DoS attacks. The communication is localized into one-hop range, which minimizes the exposure to wireless channel errors. The proposed protocols are immune from the unreliability of underlying transport layer protocols and ad hoc routing mechanisms, since no multihop communication is involved. The overhead is balanced over the network, hot spots of congestions are avoided. By the self-initialization technique, the proposed design is completely scalable to the network size. No on line dealer is necessary to handle dynamic node joins. Centralized management is minimized in the architecture.

#### **4.4 DISTRIBUTED AND LOCALIZED CERTIFICATION SERVICES**

In this section, the newly proposed algorithms and protocols that realize distributed and localized certification services, specifically the certificate issuing/renewal, certificate revocation and distributed CRLs are presented. Firstly the certificate issuing/renewal algorithms and protocols are presented. Based on the analysis of two different certification policies, the corresponding certificate revocation policies and the distributed CRL maintenance is presented. In Section 4.4.3 these algorithms and protocols are further enhanced to be self-defensive.

##### **4.4.1 CERTIFICATE ISSUING/RENEWAL**

Distributed and localized certificate issuing/renewal consists of two parts: the distribution mechanisms of the certificate signing key  $SK$ , and the multi-signature schemes with the shares of  $SK$ . The algorithms and protocols for the node that is requesting a new certificate, and the coalition of  $k$  nodes that are collaboratively serving the request are presented. The policies of how a node decides to serve or drop request is discussed in Section 4.4.2.

###### **4.4.1.1 EXISTING APPROACHES ON SECRET SHARING AND MULTI-SIGNATURE**

There are mainly two secret sharing mechanisms in the literature: polynomial secret sharing [10] and additive secret sharing [21]. Applications of additive secret sharing in multi-signature have appeared in [21, 22, 26]. However, additive secret sharing



cannot handle dynamic grouping effectively. Each node has to maintain multiple shares, which is not a scalable option for large systems.

Polynomial secret sharing features the simplicity that each node needs only one share.  $k$  polynomial shares of a coalition of  $k$  nodes can be converted into  $k$  additive shares by the technique of Lagrange interpolation. These  $k$  additive shares are then applied in the RSA multi-signature algorithms [31]. Let  $N$ , a product of two large random primes  $N = pq$ , denote the RSA modulo of the signing key  $SK$ . The polynomial shares and the corresponding additive shares are usually calculated over the ring  $Z_{\phi(N)}$  or  $Z_{\lambda(N)}$ <sup>1</sup>. However, the release of either  $\Phi(N)$  or  $\lambda(N)$  makes the factorization of the RSA modulo  $N$  trivial. We cannot reveal them for the conversion from  $k$  polynomial shares to  $k$  additive shares, specifically the calculation of the Lagrange interpolation. The solutions that are presented in [25, 31] involve the IDs of all the polynomial share holding nodes in the system, thus cannot be applied in this context due to the potential large number of networking nodes and dynamic node membership.

#### 4.4.1.2 CERTIFICATE ISSUING/RENEWAL ALGORITHMS

The polynomial secret sharing to share  $SK$  among the network is adopted. In the proposed algorithms, node  $v_i$ 's polynomial share  $P_{v_i}$  and its additive share  $SK_{v_i}$  in term of a specific coalition, are defined over the ring  $Z_N$  instead of  $Z_{\phi(N)}$  or  $Z_{\lambda(N)}$  as the previous works [25, 31] did. After the initialization of the network (as discussed in Section 4.5), each node with its ID  $v_i \neq 0$  holds a polynomial share  $P_{v_i} = f(v_i) \bmod N$ , where

$$f(x) = SK + f_1x + \dots + f_{k-1}x^{k-1} \quad (4.1)$$

is the secret polynomial. Since the certificate verification key  $PK$  is assumed to be well-known,  $N$  is also well-known as part of  $PK$ . By this choice the insecurity of releasing  $\Phi(N)$  or  $\lambda(N)$  is eliminated. Moreover, with the  $k$ -bounded coalition offsetting, the conversion from polynomial shares to additive shares scalable to the overall network size is made. Only the IDs and shares of the participating  $k$  nodes are involved.

---

<sup>1</sup>  $\Phi(N)$  denotes Euler Tortient Number and  $\lambda(N)$  denotes Carmichael number.

Detailed algorithms are as follows. Node  $v_i$  firstly chooses a coalition of  $k$  nodes from its neighborhood. Without loss of generality, let the coalition be  $B = \{v_1, v_2, \dots, v_k\}$ . Node  $v_i$  itself can be one of the coalition.  $v_i$  broadcasts the request, together with the IDs of these  $k$  nodes. Once a node  $v_j \in B$  receives the request and decides to serve the request, it firstly calculates its additive share  $SK_{v_j}$ :

$$SK_{v_j} = P_{v_j} l_{v_j}(0) = P_{v_j} \prod_{r=1, r \neq j}^k \frac{v_r}{v_r - v_j} \text{ mod } N \quad (4.2)$$

$SK_{v_j}$  is called  $v_j$ 's additive share because by Lagrange interpolation,

$$\sum_{j=1}^k SK_{v_j} = \sum_{j=1}^k P_{v_j} l_{v_j}(0) = SK \text{ mod } N. \quad (4.3)$$

$$\text{i.e., } \sum_{j=1}^k SK_{v_j} = t.N + SK$$

where  $t$  is an integer and  $0 \leq t < k$ . In the next step node  $v_j$  applies its additive share  $SK_{v_j}$  on the statement of the new certificate  $cert$  to generate  $CERT_{v_j}$ :

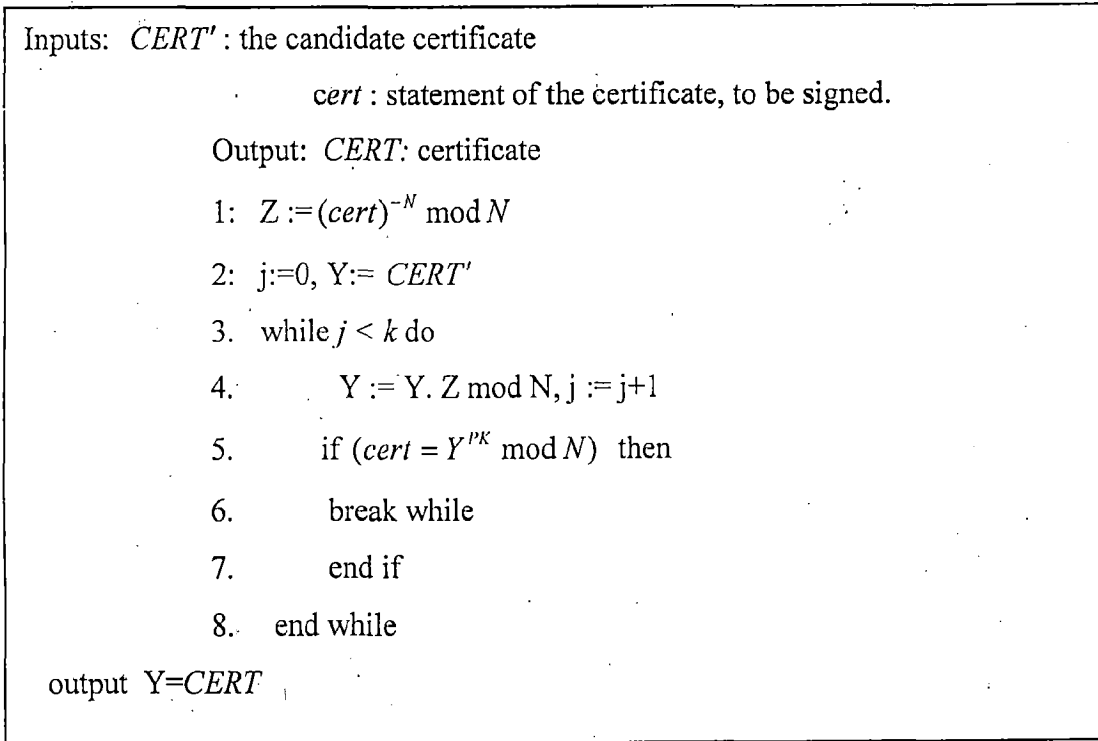
$$CERT_{v_j} = (cert)^{SK_{v_j}} \text{ mod } N \quad (4.4)$$

Node  $v_j$  finally sends  $CERT_{v_j}$  to the requesting node  $v_i$ . Upon receiving  $k$  partial certificates  $\{CERT_{v_1}, CERT_{v_2}, \dots, CERT_{v_k}\}$  from the coalition  $B$ , node  $v_i$  combines them together by multiplication to generate a "candidate certificate"  $CERT'$

$$CERT' = \prod_{j=1}^k CERT_{v_j} = (cert)^{\sum_{j=1}^k SK_{v_j}} = (cert)^{t.N + SK} = CERT \cdot (cert)^{t.N} \text{ mod } N \quad (4.5)$$

That is, the candidate certificate  $CERT'$  is different from the "real" certificate  $CERT$  by a constant. Finally  $v_i$  applies the following  $k$ -bounded coalition offsetting algorithm to recover its new certificate  $CERT'$ :

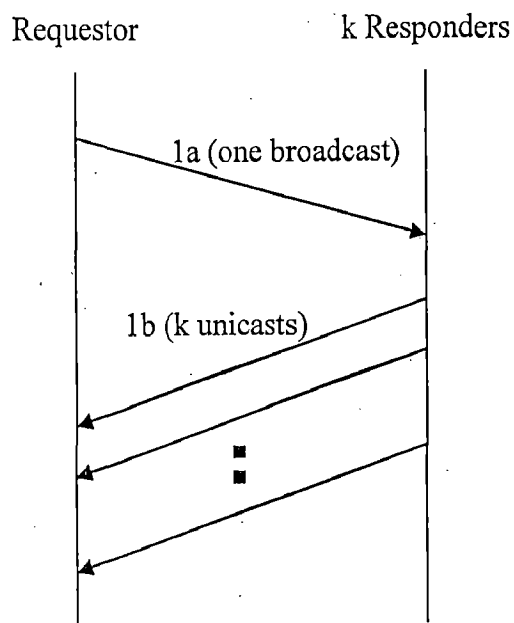
If we denote an RSA signing or verification as a unit computation, the computation complexity for each participating node is  $O(k)$ , independent of the network size  $n$ .



**Algorithm 4.1. k-bounded coalition offsetting algorithm**

#### 4.4.1.3 COMMUNICATION PROTOCOL FOR CERTIFICATE ISSUING / RENEWAL

The proposed communication protocol simply consists of a single round of localized communication, one request broadcast and k response unicasts. Figure 4.1 illustrates the communication protocol for Certificate Issuing/Renewal.



**Figure 4.1. Communication protocol for Certificate Issuing/Renewal**

(1a) Node  $v_i$  broadcasts a request in its one-hop neighborhood.

(1b) Node  $v_i$  receives  $k$  partial certificates from its neighbors. It picks  $k$  partial certificates and manipulates them to generate its new certificate *CERT*.

#### **4.4.2 CERTIFICATE REVOCATION AND CERTIFICATION REVOCATION LIST**

In the proposed architecture, certificates are carried by their owners and exchanged among communication parts for authentication. When a mobile node moves into a new location, it exchanges certificates with its neighbors. Trust relationships are established between the mobile node and its new neighborhood, so that they help each other on networking activities such as packet forwarding and routing. In the mean time, each node employs some detecting mechanism to monitor its one-hop neighbors' behavior. The specific mechanism is up to each individual node's choice. One example may be the "passive ACK" technique that takes advantage of the broadcast nature of wireless transmission [16]. When a node receives a certificate issuing/renewal request, it checks its records on the requesting node. If the records show the requesting node a well-behaving legitimate node, it will follow the protocols as presented in Section 4.4.1 to serve the request. Otherwise the request is dropped.

It should be noticed that, the certificate expiration time is already a free "implicit" certificate revocation mechanism. Misbehaving or broken nodes will be unable to get new certificates and will be cut off the network at the expiration of their current certificates. However, relying on this implicit revocation mechanism may not be enough if the validity period of certificates is long. For instance, the certificates or tickets in Kerberos are typically valid for several hours [3]. A broken node that is controlled by an adversary may be immediately isolated from network access by its neighbors. But the node can simply move to a new location to get network access from its new neighborhood. The implicit certificate revocation mechanism cannot stop this kind of "roaming" adversaries until the certificate expires hours later. Due to the ubiquitous availability of the distributed and localized certification services, we can potentially decrease the validity period of certificates into smaller time scale, e.g. tens of minutes, without overloading any centralized servers. The impact of break-ins is decreased proportionally.

There is still another problem in the above picture. When node *Alice* checks her records on node *Bob* that is requesting certification service, the records may not provide enough information for Alice to make a deterministic decision. It may simply be because the trust relationships between Alice and Bob does not last long enough to derive any meaningful insights about Bob. In the extreme case, Bob's records may not exist at all if they just meet. Alice has two choices in this scenario. She can decide to serve Bob's request, since no bad records are located. The risk is that Alice may be actually serving a roaming adversary who just moves in because he has no hope to get a new certificate from his previous location. Alice can also decide to drop the request, since no records can demonstrate Bob well behaving. But a legitimate node may not be able to get a new certificate just because he moves. If the second policy is picked to handle roaming adversaries, the cost will be the limitation on legitimate nodes' mobility: it is required that a mobile node Bob carry a certificate with long enough validity time, so that he can use this time period to build a good record with Alice.

To handle the roaming adversaries without any limitation on legitimate nodes' mobility, together with the motivation to immediately isolate a broken or misbehaving node, an explicit distributed certificate revocation mechanism is proposed. My solution exploits the characteristics of the roaming adversaries and the validity period of certificates. In the rest of this section the detailed components of the records that Alice maintains, and the policies and protocols that Alice should take is presented.

#### **4.4.2.1 MAINTAINING CERTIFICATE REVOCATION LISTS**

The records that Alice maintains consist of two parts. One is her direct monitoring record on neighboring nodes, and the other is a Certificate Revocation List (CRL). Each entry of the CRL is composed of a node ID and a list of the node's accusers. If in Alice's CRL a node's accuser list contains less than  $k$  legitimate accusers, the node is marked as "suspect". Otherwise, the node is determined by Alice to be broken or misbehaving and marked as "convicted". The threshold to convict a node is chosen as  $k$  to comply with the trust model and adversary models that the new architecture is designed to handle (as discussed in Chapter 3). The threshold ensures that a well-behaving legitimate node not be convicted by malicious accusations from an adversary.

There are two scenarios when a node is marked “convicted”. The first is when by direct monitoring records Alice determines one of her neighboring nodes to be broken or misbehaving. She puts the node into her CRL with herself as the accuser and directly marks the node “convicted”. In this scenario Alice also locally floods a signed accusation against the node. The range of the flooding is studied below. The second scenario is when Alice receives an accusation against some node. She firstly checks if the accuser is a convicted node in her CRL. If it is, the accusation is concluded to be malicious and dropped. If not, Alice updates her CRL entry of the accused node by adding the accuser into the node’s accuser list, and updates the accused node’s mark according to the number of accusers. A suspect node will be marked “convicted” if the number of accusers reaches  $k$ .

Since no convicted node can be a legitimate accuser, whenever there is a new convicted node, Alice updates her CRL by deleting the node from all accuser lists. The node marks are also updated accordingly. This is the only scenario that a convicted node’s mark changes to “suspect”, since its number of accusers may drop below  $k$ .

The range of the accusation flooding is an important design parameter. A large flooding range causes excessive communication overhead, while a small flooding range may not be enough to track a roaming adversary. The accusation should be propagated in a range to guarantee that roaming adversary cannot get a new certificate before its current certificate expires by roaming to another location. That is, before its current certificate expires, the roaming adversary cannot “escape” the area where it is convicted by  $k$  accusations.

The practical scheme for controlled flooding is by setting the  $TTL^2$  (*Time To Live*) field in the IP header of the accusation packet. One way to set  $TTL$  is based on the certificate validity period  $T_{cert}$ , the one-hop wireless transmission distance  $D$ , and the assumption on maximum node moving speed  $S_{max}$ . In a uniformly distributed network, to ensure a misbehaving node or a compromised node that is controlled by some adversary cannot escape the area of accusation before the expiration of its current certificate, the  $TTL$  of the accusation packet has to be set at least  $\frac{T_{cert} \cdot 2S_{max}}{D}$

That is

---

<sup>2</sup> TTL is defined as “time to live”: the maximal number of hops that a packet can traverse in the network.

$$TTL \geq \left\lceil \frac{T_{cert} \cdot 2S_{max}}{D} \right\rceil \quad (4.6)$$

If the  $TTL$  of the accusation messages is set to  $m$ , the nodes whose accusations reach Alice must be at most  $m$  hops away from her. Alice's CRL will contain nodes at most  $m + 1$  hops away. To further decrease the CRL length,  $T_{cert}$  after an entry's last update, Alice can remove it from her CRL. The reason is that after  $T_{cert}$  a convicted node should already be cut off the network by the expiration of its current certificate. Alice holds each CRL entry for  $T_{cert}$  to make sure that she will not serve a convicted node that carries still-valid certificate.

Alice's CRL is constrained in both space domain and time domain. It is built and maintained on demand, and stored locally. These properties comply with the overall scalability and robustness of the proposed architecture.

#### 4.4.2.2 CERTIFICATE ISSUING/RENEWAL POLICIES

With the explicit certificate revocation policies and mechanisms, the certificate issuing/renewal policies become clear. When Bob requests a new certificate from his neighbor Alice, Alice checks the validity of Bob's current certificate and her CRL. If Bob holds a still-valid certificate that is not expired, and Bob is not a convicted node in her CRL, Alice proceeds to serve the request. Otherwise the request is dropped. With this policy, a legitimate node can move around freely. It does not need to worry about being unable to renew its certificate in any location.

The assumption behind the policy is that there are always  $k$  legitimate nodes around to detect and convict a broken or misbehaving node. This assumption is clarified in the proposed system model (as discussed in section 3.3).

#### 4.4.3 VERIFIABLE PARTIAL CERTIFICATES

In Section 4.4.1.2, once node  $v_i$  receives  $k$  partial certificates from its neighbors, it generate a new certificate  $CERT$ . By the globally verifiable nature of the certificate itself,  $v_i$  can verify  $CERT$  by checking

$$cert = (CERT)^{PK} \bmod N \quad (4.7)$$

Where  $cert$  is the statement of the certificate and  $PK$  is the well-known certificate verification key. If the equation does not hold,  $v_i$  knows that  $CERT$  is an invalid certificate and should be discarded. In this scenario atleast one of these  $k$  partial certificates is faulty. It may be from a broken node that is controlled by the adversary to attack the protocols. It may also be a node that makes a mistake. However, so far  $v_i$  cannot identify the faulty partial certificates by themselves. What  $v_i$  can do is to try another combination of  $k$  partial certificate pool, until it gets a valid  $CERT$ . By comparing the coalition of partial certificates that generates the valid  $CERT$  with those coalitions that generate invalid certificates,  $v_i$  can get some clue or even identify who is/are cheating.

However, the above approach is computation expensive at node  $v_i$ . In the worst case, a single faulty partial certificate may cause  $k$  rounds of computation to be picked out. It is beneficial if every faulty partial certificate can be identified by itself, not through the try-and-error process. Moreover, above approach cannot guarantee the detection of all invalid partial certificates unless a comprehensive “scan” of each possible combination of  $k$  partial certificates is performed. Suppose the  $v_i$ 's partial certificate pool consists of  $m$  partial certificates, from  $m$  neighbors. In the best case a comprehensive scan takes  $m-k$  rounds of computation as presented in Section 4.4.1.2.

It takes  $C_m^k = \frac{m!}{k!(m-k)!}$  rounds in the worst case.

An alternative approach to detect faulty partial certificates is presented. The idea is to make every partial verifiable by itself. Any node that receives or overhears a partial certificate can verify the validity. This scheme can be integrated as part of the monitoring mechanism to help immediate detection of misbehaving or broken nodes.

The proposed approach is to apply the technique of publicly verifiable secret sharing as proposed in [13,14]. Assume node  $v_j$  is serving  $v_i$  with a partial certificate  $CERT_{v_j}$ . The challenge here is that although the statement  $cert$  is public, the share  $P_{v_j}$  that is supposed to be applied on  $cert$  is private to  $v_j$ . The problem can be formulated as given  $CERT_{v_j}$  and  $cert$ , how node  $v_j$  proves to an arbitrary node  $v$ , including  $v_i$ , that

$$CERT_{v_j} = cert^{P_{v_j}} \quad (4.8)$$



without exposing  $P_{v_j}$  to  $v$ .

From Section 4.5 we know that  $v_j$ 's share witness

$$W_{v_j} = g^{P_{v_j}} \quad (4.9)$$

is public, where  $g$  is a well-known constant. In polynomial secret sharing,  $v_j$ 's share is defined as

$$P_{v_j} = f(v_j) = SK + f_1 v_j + \dots + f_{k-1} v_j^{k-1} \quad (4.10)$$

With the public witnesses of the coefficients  $f = \{g^{SK}, g^{f_1}, \dots, g^{f_{k-1}}\}$ , any node can derive  $W_{v_j}$  as

$$W_{v_j} = g^{P_{v_j}} = g^{SK + f_1 v_j + \dots + f_{k-1} v_j^{k-1}} = g^{SK} \cdot (g^{f_1})^{v_j} \dots (g^{f_{k-1}})^{v_j^{k-1}} \quad (4.11)$$

The equivalent problem can be formulated as given  $CERT_{v_j}$ ,  $cert$ ,  $W_{v_j}$ ,  $g$ , how  $v_j$  proves to  $v$  that

$$\log_{cert} CERT_{v_j} = \log_g W_{v_j} \quad (4.12)$$

The equivalent problem can be solved by the protocol in [14] that was proposed for publicly verifiable secret sharing. Node  $v_j$  follows the following steps to prepare  $\{A_1, A_2, r\}$  as proofs to  $v$ . Firstly  $v_j$  randomly chooses a  $u$  and calculate  $A_1 = g^u$  and  $A_2 = cert^u$ .  $v_j$  then employs a one-way hash function  $HASH(x)$ , such as MD5, to calculate

$$c = HASH(W_{v_j}, CERT_{v_j}, A_1, A_2) \quad (4.13)$$

$$r = u - cP_{v_j} \quad (4.14)$$

$\{A_1, A_2, r\}$  is then signed by  $v_j$  and presented to  $v$ .

When  $v$  receives (in the case  $v=v_i$ ) or overhears (in the case that  $v$  is one of  $v_j$ 's neighbor and is monitoring  $v_j$ 's behavior) the signed  $\{A_1, A_2, r\}$ , it calculates  $c = HASH(W_{v_j}, CERT_{v_j}, A_1, A_2)$  and verifies if  $A_1 = g^r \cdot W_{v_j}^c$  and  $A_2 = cert^r \cdot CERT_{v_j}^c$ . Node  $v_j$  believes that the partial certificate  $CERT_{v_j}$  is valid if both equations hold. Otherwise  $v$  believes that  $v_j$  is misbehaving or broken. It may further mark  $v_j$  "convicted" in its CRL and send out an accusation.

#### 4.4.4 IMPORTANT PARAMETERS

a) **Coalition size  $k$ :** The most critical parameter for the proposed design is  $k$ . In the proposed architecture, each node has  $k$  well-behaving legitimate neighbors (Section 3.3) is assumed. Certification services are provided by any coalition of  $k$  nodes (Section 4.4.1). A suspect node is convicted as broken or misbehaving with  $k$  accusations (Section 4.4.2). So far the proposed algorithms and protocols are robust against the adversary of model I (Section 3.4) that is able to break in no more than  $k-1$  nodes. Implementation with large  $k$  can tolerate more powerful adversaries, but the service availability degrades. On the other hand systems with a small  $k$  feature high service availability, but is more vulnerable to malicious attacks. In general,  $k$  characterizes the ad hoc network density and represents the tradeoff between service availability and system robustness.

b) **Certificate validity period  $T_{cert}$ :** There are several factors for a proper choice of the certificate validity period  $T_{cert}$  for systems that rely on implicit certification revocation, i.e., the expiration of the certificates.  $T_{cert}$  represents the overhead of the overall certification, the timeliness that a detected adversary can be isolated from network access, and the level of tolerance that the system has for the potential impact of the adversaries. For systems that are equipped with explicit certificate revocation mechanisms (Section 4.4.2),  $T_{cert}$  also characterizes the overhead of accusation propagation and the complexity of the local CRLs.

#### 4.5 DISTRIBUTED SELF-INITIALIZATION

Section 4.4 presents the certification service instantiation by a coalition of  $k$  nodes with their polynomial shares of  $SK$ . This section, discusses the study of distribution of these shares. An initialized node is defined as the node that possesses a valid polynomial share of  $SK$ . In the bootstrapping phase of the network, all the networking nodes need to be initialized. During the lifetime of the network, a new joining node needs to be initialized also. Most works in threshold secret sharing rely on a dealer for the initialization [10, 11, 12, 21, 27, 31, 34, 36]. For ad hoc wireless networks that consist of hundreds of nodes, it is not scalable and may not be feasible. Moreover, maintaining a dealer on line to handle node joins compromises the overall system robustness and security. The dealer would become the single point of failure and

inviting target of DoS attacks. Maintaining a dealer on line also increases the adversaries' chance to compromise the dealer and therefore  $SK$ , since the dealer is the only node that possesses the full  $SK$ . In this section, a scalable initialization mechanism called "self-initialization" is proposed to address the above issues.

The dealer initializes a node  $v_i$  by the following procedure. It firstly picks a random polynomial  $f(x) = SK + f_1x + f_2x^2 + \dots + f_{k-1}x^{k-1}$  and sends  $v_i$  its share  $P_{v_i} = f(v_i)$  privately. The dealer also broadcasts the "witnesses" of the polynomial  $\{g^{SK}, g^{f_1}, \dots, g^{f_{k-1}}\}$ , as defined in verifiable secret sharing [12].  $g$  is assumed to be well-known as the certificate verification key  $PK$ . When node  $v_i$  receives its share  $P_{v_i}$  from the dealer, it verifies the validity of  $P_{v_i}$  by checking

$$g^{P_{v_i}} = g^{SK} \cdot (g^{f_1})^{v_i} \cdot (g^{f_2})^{v_i^2} \cdot \dots \cdot (g^{f_{k-1}})^{v_i^{k-1}} \quad (4.15)$$

In the proposed architecture, the dealer is only responsible to broadcast the witnesses of the polynomial, and initialize the first  $k$  nodes. After that, it destroys the polynomial and quits. The initialized nodes collaboratively initialize other nodes, typically their neighboring nodes. As more and more nodes are initialized, they further serve to initialize their neighboring nodes, thus generating a diffusion process. The same mechanism is applied when a new node joins. The node need only contact its neighbors. Each of its neighboring nodes returns a "partial" share. By collecting  $k$  such partial shares and combining them together, the node is initialized with its valid share. All proposed algorithms and protocols are fully distributed and localized for practical deployments in large ad hoc wireless networks.

#### 4.5.1 ALGORITHMS FOR DISTRIBUTED SELF-INITIALIZATION

In the context of proactive secret sharing [11], a lost secret share can be recovered by a coalition of  $k$  entities. Each entity generates a random polynomial to prevent the recovery of the polynomial that is used to share the secret. A more straightforward mechanism called *complete shuffling* is employed. Instead of shuffling the secret sharing polynomial, the partial shares are directly shuffled. Both algorithms have the same computation and communication complexities. In the rest of this section, the algorithm for the distributed initialization is presented.

When a node  $v_i$  joins the network, it exchanges certificates to build trust relationships with its new neighboring nodes. If  $v_i$  is not initialized yet, it locates a coalition of  $k$  nodes  $B = \{v_1, v_2, \dots, v_k\}$ .  $v_i$  broadcasts the request for initialization, together with the IDs of these  $k$  nodes. Once a node  $v_j \in B$  receives the request, it double checks  $v_i$ 's certificate and its CRL. If  $v_j$  decides to serve the request, it calculates a partial share for  $v_i$  as

$$P_j = P_{v_j} l_{v_j}(v_i) \bmod N \quad (4.16)$$

where  $P_{v_j}$  is  $v_j$ 's own share and  $l_{v_j}(v_i) = \prod_{r=1, r \neq j}^k \frac{v_i - v_r}{v_j - v_r}$ .

If  $v_j$  simply returns its partial share  $P_j$  to  $v_i$ ,  $v_i$  can construct its complete share by adding these  $k$  partial shares. By Lagrange interpolation, we have:

$$P_{v_i} = f(v_i) = P_{v_1} l_{v_1}(v_i) + P_{v_2} l_{v_2}(v_i) + \dots + P_{v_k} l_{v_k}(v_i) = \sum_{j=1}^k P_{v_j} l_{v_j}(v_i) = \sum_{j=1}^k P_j \bmod N \quad (4.17)$$

Unfortunately, it is insecure for node  $v_j$  to return  $P_j$  directly to  $v_i$ . As  $l_{v_j}(v_i)$  is only dependent on the IDs of the coalition, node  $v_i$  can easily recover  $v_j$ 's share  $P_{v_j}$  from the partial share  $P_j$ . If  $v_i$  receives  $k$  such partial share  $\{P_1, P_2, \dots, P_k\}$ , it can recover  $k$  polynomial shares of the coalition  $\{P_{v_1}, P_{v_2}, \dots, P_{v_k}\}$ . With these  $k$  polynomial shares, node  $v_i$  can interpolate the polynomial  $f(x)$  and retrieve the certificate signing key  $SK$ .

The problem is that we can only let  $v_i$  obtain the sum of all these  $k$  partial shares

$$\sum_{j=1}^k P_j, \text{ but not any individual } P_j. \text{ The proposed approach is that among coalition } B,$$

nodes completely shuffle their individual partial shares. The shuffled partial shares are then returned to  $v_i$ . Detailed procedure is as follows. Firstly, among the coalition  $B$  each node pair  $\{v_j, v_r\}$  securely exchanges a shuffling factor  $d_{i,j}$ . One of the pair adds  $d_{i,j}$  to its partial share. The other subtracts  $d_{i,j}$  from its partial share. For node  $v_j$ , there are totally  $k-1$  shuffling factors, and it must apply all of them, by either addition or subtraction, to its partial share  $P_j$ . The result is a completely-shuffled partial share

$$\bar{P}_j = P_j + \sum_{r=1, r \neq j}^k \text{sign}(v_r - v_j) d_{r,j} \quad (4.18)$$

where  $\text{sign}(x)=1$  if  $x>0$  and  $\text{sign}(x)=-1$  if  $x<0$ . Once node  $v_i$  receives  $k$  signed shuffled partial shares from the coalition, it recovers its partial share as:

$$\begin{aligned} \sum_{j=1}^k \bar{P}_j &= \sum_{j=1}^k (P_j + \sum_{r=1, r \neq j}^k \text{sign}(v_r - v_j) d_{r,j}) \quad (4.19) \\ &= \sum_{j=1}^k P_j + \sum_{j=1}^k \sum_{r=1, r \neq j}^k \text{sign}(v_r - v_j) d_{r,j} \\ &= \sum_{j=1}^k P_j + 0 = P_{v_i} \end{aligned}$$

Except the sum,  $v_i$  cannot derive any other information from these  $k$  shuffled shares without the shuffling factors. The computation complexity for each participating node is  $O(k)$ .

#### 4.5.2 COMMUNICATION PROTOCOL FOR SELF INITIALIZATION

The communication protocol for self-initialization consists of two rounds of communications among one hop neighborhood of node  $v_i$ . Figure 4.2 illustrates the communication protocol for self-initialization.

- (1a) Node  $v_i$  broadcasts/multicasts an initialization request among  $B$ .
- (1b) Each node  $v_j \in B$  generates shuffling factors according to the cryptographic algorithm and returns these encrypted shuffling factors with their signatures to  $v_i$ .
- (2a)  $v_i$  combines these encrypted shuffling factors and broadcasts/multicasts them among the coalition  $B$ . Here a broadcast/multicast channel is been simulated for the coalition  $B$  to exchange information.
- (2b) Each node  $v_i \in B$  returns a shuffled partial secret share to  $v_i$ . Upon receiving  $k$  such partial secret shares,  $v_i$  simply adds them up to recover its full secret share  $P_{v_i}$ .

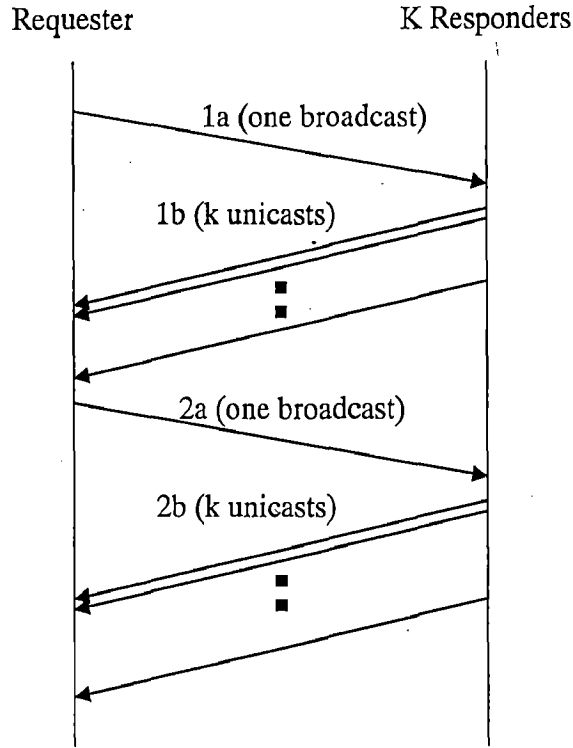


Figure 4.2 Communication Protocol for Self Initialization

#### 4.5.3 VERIFIABLE PARTIAL SHARE

Once node  $v_i$  receives  $k$  shuffled partial shares  $\{\bar{P}_1, \bar{P}_2, \dots, \bar{P}_k\}$ , it adds them to recover its polynomial share  $P_{v_i} = \sum_{j=1}^k \bar{P}_j$ . Node  $v_i$  verifies the validity of  $P_{v_i}$  by checking

$$g^{P_{v_i}} = g^{SK} \cdot (g^{f_1})^{v_i} \cdot (g^{f_2})^{v_i^2} \cdot \dots \cdot (g^{f_{k-1}})^{v_i^{k-1}} \quad (4.20)$$

where  $\{g^{SK}, g^{f_1}, \dots, g^{f_{k-1}}\}$  are  $k$  public witnesses of the polynomial  $f(x) = SK + f_1x + \dots + f_{k-1}x^{k-1}$ . If the equation does not hold,  $v_i$  knows that  $\sum_{j=1}^k \bar{P}_j$  is not its valid polynomial share. In this scenario, at least one of these  $k$  partial shares is faulty. It may be from a broken node that is controlled by the adversary or a node that makes a mistake. However, so far  $v_i$  cannot identify the faulty partial certificates by themselves.  $v_i$  can only try another  $k$  neighboring nodes until it gets its valid polynomial share.

Similar to the scenario of partial certificates (discussed in Section 4.4.3), the above approach is communication expensive for participating nodes. With the same motivations as in the design of verifiable partial certificates, to make each shuffled partial share verifiable by itself, an approach for a node that receives (node  $v_i$ ) or overhears (neighboring nodes) a shuffled partial share to be able to verify the validity I presented. Once  $v_i$  detects a faulty shuffled partial share, it can at least choose a coalition without the node from whom the faulty shuffled partial share is generated. Furthermore,  $v_i$  can propagate an accusation against the node with necessary proofs so that other nodes can be aware of the node's misbehaviors. This scheme can be integrated as part of the monitoring mechanism to help immediate detection of misbehaving or broken nodes. The implementation of the verifiable design can greatly discourage an adversary's potential attacks on the protocols.

The approach is a direct extension of the verifiable secret sharing as proposed in [12]. When nodes in coalition B exchange the shuffling factors (as discussed in Section 4.5.1 and step (1b) and (2a) in Section 4.5.2), they also advertise a signed witness  $g^{d_{r,j}}$  on each shuffling factor  $d_{r,j}$ . For each shuffled partial-share  $\bar{P}_j$ , node  $v_i$  verifies

$$g^{\bar{P}_j} = g^{P_j} \prod_{r=1, r \neq j}^k (g^{d_{r,j}})^{\text{sign}(v_r - v_j)} \quad (4.21)$$

where  $g^{P_j} = g^{SK} \cdot (g^{f_1})^{v_j} \cdot (g^{f_2})^{v_j^2} \cdot \dots \cdot (g^{f_{k-1}})^{v_j^{k-1}}$  is the public witness of  $v_j$ 's polynomial share. If the equation does not hold,  $v_i$  concludes that  $\bar{P}_j$  is faulty.  $v_i$  then marks  $v_j$  "convicted" in its CRL and sends out an accusation against node  $v_j$ , with  $\bar{P}_j$ ,  $v_j$ 's signature on  $\bar{P}_j$ , and all the witnesses as proofs.

#### 4.6 SCALABLE SHARE UPDATE

So far the system is robust against the adversaries of model I as defined in Section 3.4. In this section, mechanisms to further enhance my system to defend against the model II adversaries are presented. There are two options to achieve this goal. The first is to periodically update the shared certificate signing key  $SK$ . This can be achieved by applying self-initialization in Section 4.5 (i.e., resetting the system periodically). However, this option involves network-wide well-known change of  $PK$ ,

which may not be desirable from the application's perspective. The second option is the proactive secret sharing mechanism [11]. Instead of changing  $SK$  and  $PK$ , the secret share of each node is updated while keeping the shared  $SK$  intact. However, in existing proposals [11, 27] each node has to collect inputs from all other nodes to finalize its update. Their approaches are not applicable in the present scenario for several reasons:

- (a) The solution is not scalable. In an ad hoc network with dynamic membership and topology, a node cannot afford to maintain global knowledge and network topology;
- (b) The communication overhead is too high to apply these protocols in wireless networking scenarios;
- (c) These proposals typically require a global broadcast channel, which does not exist in typical ad hoc wireless networks. Applying Byzantine agreement protocol to simulate an authenticated broadcast channel incurs prohibitively high communication overhead [27] even for wired networks. It is not feasible for wireless networks.

Now, the proposed two approaches to achieve scalable and efficient share update in ad hoc wireless networks are presented. The first approach is a simple sequential process based on the self-initialization as presented in Section 4.5. Firstly a coalition of  $k$  nodes updates their shares by applying the existing protocols as proposed in [11, 36]. The self-initialization protocols then follow to update the shares of the rest of the network. The second approach features parallel share updates over the network for fast convergence. It is also scalable to the network size. The cost is higher computation overhead at each node. Parallel share update mechanism in the used in this work and explained as follows.

Similar to [11], the time is divided into periods. Each time period is composed of a share update phase and an operational phase. During the operational phases, nodes periodically renew their certificates (as discussed in Section 4.4). At the beginning of the share update phases, a chosen coalition of  $k$  nodes in the system collaboratively generate a random share update polynomial

$$f_u(x) = f_{u,1}x + \dots + f_{u,k-1}x^{k-1} \quad (4.22)$$

where  $f_u(0) = 0$ .



$f_u$  is then encrypted by  $PK$  for privacy against adversaries. The coalition then collaboratively apply their polynomial shares of  $SK$  to sign the encrypted  $f_u$ . This signature prevents an adversary from simulating a coalition of  $k$  nodes to fake share updates. The encrypted polynomial, together with its signature, is then propagated among the network by flooding. Once a node receives the encrypted update polynomial, it verifies the signature and requests share-update service from  $k$  neighboring nodes to evaluate its update

$$P_{u,v_i} = f_u(v_i) \quad (4.23)$$

These  $k$  nodes do not need to update their share before they can serve the request. Otherwise it becomes a “chicken-and-egg” problem. The requesting node’s share update can be evaluated as long as these  $k$  neighboring nodes apply the same “version” of shares.

The same as in the proactive secret sharing context, when node  $v_i$  gets its share update  $P_{u,v_i}$  from its  $k$  neighbors, it simply adds  $P_{u,v_i}$  to its current polynomial share  $P_{v_i}$  to generate a new share  $P_{new,v_i}$ . Old shares will be held for graceful transitions and destroyed at the end of the share update phases. If we define  $f_{new} = f + f_u$ , we can see that  $f_{new}(0) = f(0) + f_u(0) = SK + 0 = SK$  and  $v_i$ 's share is updated as

$$P_{new,v_i} = P_{v_i} + P_{u,v_i} = f(v_i) + f_u(v_i) = f_{new}(v_i) \quad (4.24)$$

This way, the polynomial is updated while  $SK$  is kept unchanged.

The process is composed of three steps:

1. ***Collaborative generation of the update polynomial  $f_u$***  At the beginning of each update phase, each node initiates updates with probability  $\frac{1}{\hat{n}}$ , where  $\hat{n}$  is an estimate on the total number of networking nodes. This ensures that statistically there is only one node to initiate the update process. Once a node  $v_i$  decides to initiate the update, it locates a coalition of  $k$  neighbors and collaboratively generate the encrypted update polynomial  $(f_u)_{PK}$  and a signature.
2. ***Robust propagation of the update polynomial*** Node  $v_i$  floods the encrypted update polynomial  $(f_u)_{PK}$  with the signature among the network. The

advantage of the robustness of the network flooding protocol is taken to ensure that each node will receive the update polynomial at least once.

3. *Distributed evaluation of share update*  $P_{u,v_i}$  Since the propagated  $f_u$  is encrypted by the system  $PK$ , each node solicits its  $k$  neighbors to collaboratively evaluate  $P_{u,v_i} = f_u(v_i)$  for it. Node  $v_i$  then updates its share as

$$P_{new,v_i} = P_{v_i} + P_{u,v_i}, \text{ and erases its old shares } P_{v_i} \text{ at the end of the update phase.}$$

Once node  $v_i$  gets its new share, it further updates the polynomial's witnesses [12] as defined in Section 4.5. Since  $f_{new} = f + f_u$ , each coefficient  $f_l$ 's witness  $g^{f_l}$  can be updated as

$$g^{f_{new,l}} = g^{f_l} \cdot g^{f_{u,l}} \quad (\text{For each } l=1 \dots k-1.) \quad (4.25)$$

The collaborative generation of  $f_u$  only happens once in a coalition of  $k$  nodes in the system during each period. The flooding protocol is robust with the inherent connectivity redundancy of ad hoc networks. Therefore, in order to achieve scalable parallel share updates, the distributed evaluation of share update  $P_{u,v_i}$  is the critical step. In the next sections, detailed algorithms and communication protocol for the third step are presented.

#### 4.6.1 ALGORITHMS FOR SHARE UPDATE

This discusses the cryptographic algorithms for distributed evaluation of  $v_i$ 's share update  $P_{u,v_i} = f_u(v_i)$ . Since the share polynomial is encrypted by  $PK$ , node  $v_i$  need collaborative efforts from a coalition of  $k$  nodes. Without loss of generality, assume that the coalition  $B$  includes  $\{v_1, \dots, v_k\}$  with their shares  $\{P_{v_1}, \dots, P_{v_k}\}$ . Node  $v_i$ 's goal is to evaluate the update polynomial

$$f_u(v_i) = f_{u,1}v_i + \dots + f_{u,k-1}v_i^{k-1} \quad (4.26)$$

The update polynomial  $f_u$  with all  $k-1$  coefficients  $\{f_{u,1}, \dots, f_{u,k-1}\}$  has to be kept confidential. Upon receiving the encrypted share update polynomial  $\{(f_{u,1})_{PK}, \dots, (f_{u,k-1})_{PK}\}$ , node  $v_i$  generates  $\{(f_{u,1} \cdot v_i)_{PK}, \dots, (f_{u,k-1} \cdot v_i^{k-1})_{PK}\}$  by multiplying  $(f_{u,l})_{PK}$  with  $(v_i^l)_{PK}$ . If we denote  $\Theta_l = f_{u,l} \cdot v_i^l$ , the problem can be formulated as given  $\{(\Theta_l)_{PK}\} (l=1, \dots, k-1)$ , how do we assure that only  $v_i$  gets its

update  $P_{u,v_i}$  as the sum of  $\{\Theta_1, \dots, \Theta_{k-1}\}$ , while any coalition of size less than  $k$  has no information on any single  $\Theta_l$ .

The algorithms are composed of two techniques:

1. **Multiplicative sharing of  $\Theta_l$  among  $B$ .** For each  $\Theta_l$ , node  $v_j \in B$  holds a multiplicative share  $\alpha_{l,j}$  such that

$$\Theta_l = \prod_{j=1}^k \alpha_{l,j}. \quad (4.27)$$

2. **Conversion from multiplicative sharing to additive sharing.** We convert the multiplicative shares of  $\Theta_l$  into additive shares. That is, node  $v_j$  generates an additive share  $\beta_{l,j}$  for  $\Theta_l$ , such that

$$\Theta_l = \prod_{j=1}^k \alpha_{l,j} = \sum_{j=1}^k \beta_{l,j} \quad (4.28)$$

If node  $v_j$  adds up its additive share  $\beta_{l,j}$  for each  $\Theta_l (l = 1, \dots, k-1)$ , the sum becomes a partial share update

$$P_{u,j} = \sum_{l=1}^{k-1} \beta_{l,j} \quad (4.29)$$

for  $v_i$ . Node  $v_j$  returns  $P_{u,j}$  to  $v_i$ . Upon receiving  $k$  such partial share updates,  $v_i$  adds them up to recover its final share update:

$$\sum_{j=1}^k P_{u,j} = \sum_{j=1}^k \left( \sum_{l=1}^{k-1} \beta_{l,j} \right) = \sum_{l=1}^{k-1} \left( \sum_{j=1}^k \beta_{l,j} \right) = \sum_{l=1}^{k-1} \Theta_l = P_{u,v_i} \quad (4.30)$$

The distributed certificate issuing/renewal algorithms are applied to generate multiplicative shares of each  $\Theta_l$  (as discussed in Section 4.4.1.2). A random polynomial  $p(x)$  of order  $k$  is then applied to re-distribute  $\Theta_l$  among  $B \cup \{v_i\}$  so

that  $p(0) = \Theta_l = \prod_{j=1}^k \alpha_{l,j}$ . Each node holds a polynomial share of  $\Theta_l$  as  $p(v_j)$ . The

additive shares  $\beta_{l,j}$  are then derived by interpolating  $p_{v_j}$ 's.

1. Each node  $v_j \in B$  generates a random first order polynomial  $h_j(x) = \alpha_{l,j} + c_j x$  where  $c_j$  is a random number. Node  $v_j$  sends  $h_j(v_r)$  to each  $v_r \in B \cup \{v_i\}$  securely.

2. Upon collecting  $k$  such  $h_j(v_r)$ , node  $v_r \in B \cup \{v_i\}$  multiplies them together. If

polynomial  $p(x) = \prod_{j=1}^k h_j(x)$ , is defined, then this product  $\prod_{j=1}^k h_j(v_r) = p(v_r)$ .

Since  $p(0) = \prod_{j=1}^k h_j(0) = \prod_{j=1}^k \alpha_{i,j} = \Theta_i$ ,  $\Theta_i$  is re-distributed among  $B \cup \{v_i\}$ .

3. Node  $v_r$  applies the Lagrange interpolation to calculate  $\beta_{i,r} = p(v_r)l_r(0)$ ,

where  $l_r(0) = \prod_{j=1, j \neq r}^k \frac{v_r}{v_r - v_j}$ . Because  $\sum_{j=1}^k \beta_{i,j} = \sum_{j=1}^k (p(v_j)l_j(0)) = p(0) = \Theta_i$ , we

finish the process of conversion from multiplicative shares to additive shares of  $\Theta_i$ .

Since  $h_j(x)$  is of order 1, it is not robust against model II adversaries if each node  $v_j$  simply sends  $h_j(v_r)$  to node  $v_r$  in step 1 above. This problem is solved by letting  $v_j$  further shuffle  $h_j(v_r)$  before transmitting it. This is similar to the shuffling in section 4.5.1. Coalition B finally sends the sum of shuffled shares of  $\Theta_i$ 's ( $l = 1, \dots, k-1$ ) to  $v_j$ .  $v_j$  recovers its final share update  $P_{u,v_j}$  by adds them with its own additive shares. The computational complexity of each participating node is  $O(k)$ .

#### 4.6.2 COMMUNICATION PROTOCOL FOR SHARE UPDATE

Share update evaluation protocol involves three rounds of communications between node  $v_j$  and coalition B (Figure 4.3 illustrates the protocol for share updates)

- (1a)  $v_j$  broadcasts/multicasts a "share update evaluation" request among coalition B.
- (1b) Each node  $v_j \in B$  generates shuffling factors according to my cryptographic algorithm of Section 4.4.3 and sends these encrypted and signed shuffling factors back to  $v_j$
- (2a)  $v_i$  aggregates them into a single packet and locally broadcasts/multicasts among B.
- (2b) Each node  $v_j$  starts converting the multiplicative sharing into an additive form.
- (3a)  $v_i$  broadcasts/multicasts among B to complete the conversion.

(3b) Each node  $v_j$  returns a shuffled partial share update  $P_{u,j}$  to  $v_i$ .  $v_i$  simply adds

$$\text{them up to reconstruct its final share update } P_{u,v_i} = \sum_{j=1}^k P_{u,j}.$$

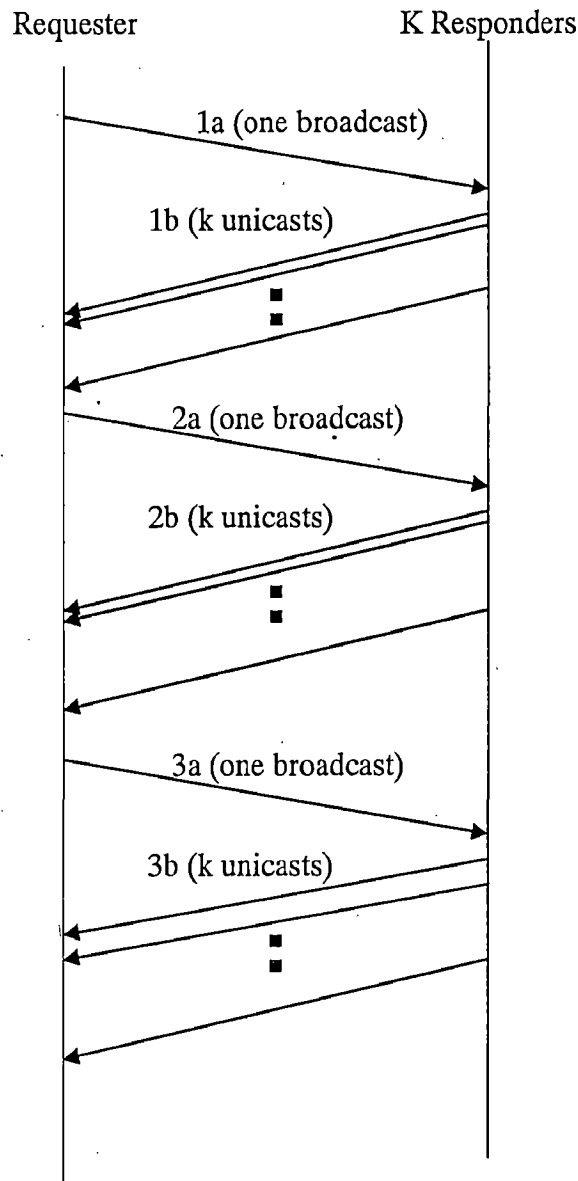


Figure 4.3 Communication protocol for Share updates

#### 4.7 SUMMARY

The proposed design makes extensive use of the polynomial secret sharing to share the certificate signing key  $SK$ , among all  $n$  nodes in the network. Each node carries a certificate signed by  $SK$  to authenticate itself to other nodes. At the bootstrapping phase of the network, a dealer initializes first  $k$  nodes by sending them their share of  $SK$ , according to a polynomial of order  $k-1$ . These initialized nodes collaboratively initialize other nodes. When a new node joins it needs to contact its one hop

neighbor's for initialization. By combining  $k$  partial shares returned by its neighbor's the node gets initialized with its full share. Nodes renew their certificates by broadcasting certificate renewal request to its one hop neighbors. By manipulating any  $k$  partial certificates from its neighbors the node generates its new certificate. Each node employs some detecting mechanism to monitor its one-hop neighbors behavior. When a node receives a certificate issuing/renewal request, it checks its records on the requesting node. If the record shows the requesting node a well-behaving legitimate node, it will follow the protocols to serve the request. Otherwise the request is dropped. The security of the certificate signing key  $SK$  is protected by the  $k$ -threshold polynomial sharing mechanism. It is robust against the adversaries of model I. Scalable share update algorithm further enhances the security of  $SK$  by periodically updating each node's share of  $SK$ .

## CRYPTOGRAPHIC ANALYSIS

## 5.1 INTRODUCTION

This chapter covers the cryptographical analysis of the proposed algorithms and prove that they are RSA(k,n) secure. By assuming that “An RSA function-sharing primitives is (k, n)-secure when for all possible subset  $\{i_1, i_2, \dots, i_j\}$  where  $0 \leq j < k \leq n$ , for all probabilistic polynomial time algorithms  $A$ , for any polynomial  $\text{poly}(\cdot)$ , for  $h$  large enough”; we prove that certificate issuing/renewal algorithm, self-initialization algorithm, share update algorithm are RSA (k,n)-secure in the following sections.

## 5.2 DISTRIBUTED AND LOCALIZED CERTIFICATION SERVICES

The security of the proposed algorithms and the robustness of model I as defined in Section 3.4.1 are now formally proved. First state the well-known RSA assumptions are based.

**Definition 5.2.1 (RSA Assumption [19]):** Let  $h$  be the security parameter. Let the key generation  $(e, d, m) \leftarrow G(1^h)$  be an RSA instance with security parameter  $h$ . For any probabilistic polynomial-time algorithm  $A$  and polynomial  $\text{poly}(\cdot)$ ,  $\Pr[u^e \equiv w \pmod{m} : (e, d, m) \leftarrow G(1^h); w \in_R \{0,1\}^h; u \leftarrow A(1^h, w, e, N)] < \frac{1}{\text{poly}(h)}$ .

Since  $n$  nodes in the system share the RSA certificate signing key SK, the proposed design is to be proven (k, n)-secure. The formal definition is as follows:

**Definition 5.2.2 [19]** An RSA function-sharing primitives is (k, n)-secure when for all possible subset  $\{i_1, i_2, \dots, i_j\}$  where  $0 \leq j < k \leq n$ , for all probabilistic polynomial time algorithms  $A$ , for any polynomial  $\text{poly}(\cdot)$ , for  $h$  large enough.

$\Pr[f_e(u) \equiv w \pmod{m} : (e, d, m) \leftarrow G(1^h); (P_1, \dots, P_n) \leftarrow \text{share}_{k,n}(e, d, m); w \in_R \{0,1\}^h; u \leftarrow A(1^h, w, H, (P_{i_1}, \dots, P_{i_j}))$

where  $H$  is a history record of length polynomial in  $h$ . It consists of a list  $L$  whose  $r$ th entry contains  $w_r \in_R \{0,1\}^h$ ,  $f_d(w_r)$ , and partial results generated by each member of the subset  $\Lambda$  ( $|\Lambda| \geq k$ ).

**Theorem 5.2.1 (Security)** *The proposed certificate issuing/renewal algorithm is RSA  $(k, n)$ -secure.*

**Proof** A simulatability argument is followed to prove the theorem. The proof is based on the following Lemma:

**Lemma 5.2.1 [19]** *Let the key generation  $(e, d, m) \leftarrow G(1^h)$  be an RSA instance with security parameter  $h$ . Let  $(P_{v_1}, \dots, P_{v_n}) \leftarrow \text{share}_{(k,n)}(e, d, m)$  and  $H$  is the history defined above. The function sharing primitives is  $(k, n)$  secure if*

1. *The function-sharing generation is simulatable. For any subset  $\Lambda$  with  $|\Lambda| = k-1 < n$ , there exists a probabilistic polynomial time algorithm  $SIM_{\text{share}_{(k,n)}}$  with given  $e$  and  $\Lambda$  as input generates  $k-1$  random partial functions with a distribution indistinguishable from the  $k-1$  random partial functions generated by  $\text{share}_{(k,n)}$ .*
2. *The function reconstruction is simulatable. There exists a probabilistic time simulator  $SIM_{\text{rec}}$  given  $e$ , partial functions  $\{P_{v_1}, P_{v_2}, \dots, P_{v_j}\}$  where  $0 < j < k$  and all the corresponding entries from the  $h$ , generates a simulated history record  $H'$  which is indistinguishable from  $H$  by any probabilistic polynomial time distinguisher.*

To apply Lemma 5.2.1, we need to construct  $SIM$  to simulate the view of the adversary with history  $H$  in the system. Assume  $L = \{CERT^1, \dots, CERT^L\}$  is a list of certificates from previous RSA operations. The  $SIM$  has up to  $k-1$  share functions  $P_{v_1}, \dots, P_{v_{k-1}}$  but it does not have the  $k$ th share function  $P_{v_k}$ .  $SIM$  needs to pass all these consistency checks. For each  $j \in \{1, 2, \dots, k-1\}$ ,  $SIM$  calculates  $CERT_j = cert_i^{P_{v_j}} \bmod N$ . For  $j = k$ ,  $SIM$  calculates



$$CERT_k = \frac{CERT \cdot cert^{t \cdot N}}{\prod_{j=1}^{k-1} CERT_j} \text{ mod } N.$$

Now the proof reduces to that the unencrypted parts of the real and simulated results are statistically indistinguishable. It can be done by following the same argument of [21]. This completes the proof of Theorem 5.2.1.

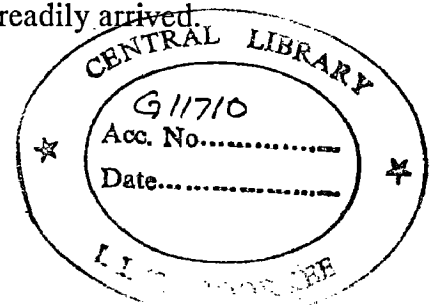
### 5.3 DISTRIBUTED SELF-INITIALIZATION

**Theorem 5.3.1 (security)** *The Proposed self-initialization algorithm is RSA (k,n)-secure.*

**Proof** The proof still follows a simulatability argument based on Lemma 5.2.1. The key is still to construct *SIM*. Here is a summary of how *SIM* is constructed. Assume node  $v_i$  is requesting initialization service from a coalition of  $\{v_1, v_2, \dots, v_k\}$ . Consider two scenarios for the subset of share functions  $\Lambda$  with size up to  $k-1$ , as input to the *SIM*:

1. If  $v_i \notin \Lambda$ , without loss of generality, assume  $\Lambda = \{v_1, \dots, v_{k-1}\}$ .  $k-1$  shares  $\{P_{v_1}, \dots, P_{v_{k-1}}\}$  and all the shuffling factors between  $\{v_1, \dots, v_k\}$  are available to *SIM*. *SIM* can calculate  $g^{\bar{P}_j}$  for  $1 \leq j \leq k-1$ . For  $j = k$ , *SIM* calculates  $g^{\bar{P}_k} = \frac{g^{P_{v_i}}}{\prod_{j=1}^{k-1} g^{\bar{P}_j}}$  where  $g^{P_{v_i}}$  is available from the history record  $H$ .
2. If  $v_x \in \Lambda$ , without loss of generality, assume  $\Lambda = \{v_i, v_1, \dots, v_{k-2}\}$ . Then  $k-1$  shares  $\{P_{v_i}, P_{v_1}, \dots, P_{v_{k-2}}\}$  and all the shuffling factors except  $d_{k-1,k}$  are available to *SIM*. *SIM* thus calculates  $g^{\bar{P}_j}$  for  $1 \leq j \leq k-2$  and  $j = i$ . *SIM* further calculates  $\bar{P}_{k-1} + \bar{P}_k = P_{v_i} - \sum_{j=1}^{k-1} \bar{P}_j$ . *SIM* then picks  $\bar{P}_{k-1}$  and  $\bar{P}_k = P_{v_i} - \sum_{j=1}^{k-1} \bar{P}_j - \bar{P}_{k-1}$  and calculates  $g^{\bar{P}_{k-1}}$  and  $g^{\bar{P}_k}$ .

With the arguments similar to Theorem 5.2.1, the conclusion is readily arrived.



## 5.4 SCALABLE SHARE UPDATE

**Theorem 5.4.1 (Security)** *The proposed share update algorithm is RSA  $(k, n)$ -secure.*

**Proof** The proof still follows the simulatability argument based on Lemma 5.2.1. The key is how to construct *SIM*. *SIM* construction is similar to the self-initialization scenario (as shown in the proof of Theorem 5.3.1). Since we distribute each individual coefficient among the coalition to generate a shuffled additive distribution, *SIM* simply repeats the same process for each individual coefficient to get the simulated results. Applying arguments similar to Theorem 5.2.1, the conclusion is readily arrived.

**SELF-ORGANIZED SOLUTION FOR NETWORK-LAYER  
SECURITY IN AD HOC NETWORKS**

---

---

**6.1 INTRODUCTION**

This chapter describes my proposal, which is a unified network-layer security solution in ad hoc networks. This solution protects both routing and packet forwarding functionalities in the context of the AODV protocol. To address the unique characteristics of ad hoc networks, this takes a self-organized approach by exploiting full-localized design, without assuming any a priori trust or secret association between nodes. In this design, each node has a token in order to participate in the network operations, and its local neighbors collaboratively monitor it to detect any misbehavior in routing or packet forwarding services. Upon expiration of the token, each node renews its token via its multiple neighbors. The period of the validity of a node's token is dependent on how long it has stayed and behaved well in the network. A well-behaving node accumulates its credit and renews its token less and less frequently as time evolves. In essence, this security solution exploits collaboration among local nodes to protect the network layer without completely trusting any individual node.

**6.2 OVERVIEW OF THE SECURITY SOLUTION**

Protecting the network layer in a mobile ad hoc network is an important research topic in wireless security. The core functionalities provided in the network layer are routing and packet forwarding. Malicious attacks on either of them will disrupt the normal network operations. This proposal is interested in devising a coherent, unified solution that protects both the routing and the data forwarding services in mobile ad hoc networks. Most existing security schemes proposed for mobile ad hoc networks either assume a priori trust or secret association between networking entities or assume that there is a centralized trusted server in the network. However, the self-organized nature of the ad hoc networks challenges this very basic assumption, and the existence of a centralized server may degrade the effectiveness of the security scheme. This proposal describes a solution to the network-layer security in ad hoc networks in the

context of AODV routing protocol. The self-organized feature of the solution is provided through fully localized design: each node shares a portion of a global secret, and each node is verified and monitored by its local neighbors collaboratively. Fundamentally, this security solution exploits the collaboration among local nodes to protect the network layer without completely trusting any individual node. In this design, each node is granted temporary admission into the network initially by obtaining a token that will expire soon. Once the token expires, the node has to renew it from its local neighbors, which are responsible for monitoring its behavior collaboratively. The node accumulates its credit as it stays and behaves well in the network. The period of validity of a node's token is proportional to its current credit. This way, a well-behaving node renews its token less and less frequently as time evolves. A malicious node will eventually be detected by its neighbors, its token will be revoked, and it will be denied network access.

### **6.3 SECURITY ISSUES IN AODV ROUTING PROTOCOL**

This section considers routing and data forwarding security issues in the context of the Ad hoc On-demand Distance Vector routing protocol (AODV). AODV has been one of the most popular on-demand routing protocols studied in the research community and IETF. For simplicity, we will focus on the basic version of AODV. In AODV, path discovery is entirely on-demand. When a source node needs to send packets to a destination to which it has no available route, it broadcasts a RREQ (Route Request) packet to its neighbors. Each node maintains a monotonically increasing sequence number to ensure loop-free routing and supersede stale route cache.

The source node includes the known sequence number of the destination in the RREQ packet. The intermediate node receiving a RREQ packet checks its route table entries. If it possesses a route toward the destination with greater sequence number than that in the RREQ packet, it unicasts a RREP (Route Reply) packet back to its neighbor from which it received the RREQ packet. Otherwise, it sets up the reverse path and then rebroadcasts the RREQ packet. Duplicate RREQ packets received by one node are silently dropped. This way, the RREQ packet is flooded in a controlled manner in the network, and it will eventually arrive at the destination itself or a node that can supply a fresh route to the destination, which will generate the RREP packet.

As the RREP packet is propagated along the reverse path to the source, the intermediate nodes update their routing tables using distributed Bellman-Ford algorithm with additional constraint on the sequence number, and set up the forward path. AODV also includes the path maintenance mechanism to handle the dynamics in the network topology. Link failures can be detected by either periodic beacons or link layer acknowledgments, such as those provided by 802.11 MAC protocol. Once a link is broken, an unsolicited RREP packet with a fresh sequence number and infinite hop count is propagated to all active source nodes that are currently using this link. When the source node receives the notification of a broken link, it may restart the path discovery process if it still needs a route to the destination.

#### 6.4 NETWORK-LAYER VULNERABILITIES

Ad hoc networks are vulnerable to a wide range of malicious attacks in the network layer due to the inherent peer-to-peer communication model. In these networks, each node functions as a router that maintains routes toward other nodes in the network, and each node relies on intermediate nodes to relay its packets to the destination. Malicious attacker may readily become a router and disrupt normal network operations. The core functionalities of the network layer are routing and packet forwarding. Routes from the source to the destination are established and maintained by the routing protocols, while data packets are forwarded by intermediate nodes along the established route to the destination.

Attacks on either functionality can disrupt the normal operations in the network layer. Although routing and packet forwarding functionalities are closely related to each other, we explicitly distinguish their vulnerabilities because the routing functionality is only responsible for establishing and maintaining the routes, and it can not enforce that the data packets are correctly forwarded along the routes by any means. Therefore, we describe the network-layer vulnerabilities by two categories of attacks: 'routing updates misbehavior' and 'packet forwarding misbehavior', *Routing updates misbehavior* means any action of advertising routing updates that does not follow the specifications of the routing protocol. Because ad hoc routing protocols typically assume that all nodes are cooperative, the attacker may exploit this vulnerability and inject malicious routing information into the network. In the context of AODV, the attacker may advertise a route with a smaller distance metric than its

actual distance to the destination; the attacker may advertise routing updates with a large sequence number and invalidate all the routing updates from other nodes; the attacker may also spoof its IP address and advertise that an operational link is broken. By exploiting routing updates misbehavior, the attacker can attract data traffic to itself, or cause the packets to be forwarded along a route that is not optimal, with poor quality, or even nonexistent.

The attackers can also intentionally introduce severe network congestion and channel contention in certain areas. If there are multiple attackers in the network, they may even collaborate to prevent a source node to find any route to the destination, partition the network, or create route loops and waste the network resource. Packet forwarding misbehavior means any malfunction of the data packet forwarding service as the consequence of an attack. For example, the attacker along an established route may drop the data packets, or duplicate the data packets that it has forwarded. Another type of packet forwarding misbehavior is the Denial of Service (DoS) attack of network layer packet jamming, in which the attacker injects large amount of packets into the network and wastes a significant portion of the network resource. Furthermore, the attacker may adopt more tricky strategies, such as dropping certain data packets or dropping the data packets with some probability, instead of blindly dropping all the packets. Attacks may be initiated toward each of these two dimensions of routing and packet forwarding, or both. Even though the attacker exactly follows the routing protocol, it can still generate various packet forwarding misbehaviors, such as the network-layer DoS attack.

## **6.5 NETWORK-LAYER SECURITY SOLUTION**

### **6.5.1 FRAMEWORK**

In order to protect the routing and packet forwarding functionalities in ad hoc networks, this network-layer security solution consists of both proactive and reactive mechanisms. Each legitimate node carries a token signed with the system secret key, which can be verified by its neighbors. Nodes without a valid token are isolated in the network in that all its legitimate neighbors will not interact with them in routing and forwarding services. The system secret is equally shared by all nodes in the network, but each node only knows a limited portion of it. The token has limited period of validity. Before its token expires, each node must renew the token from its neighbors,

which in turn collaboratively monitor it to detect any misbehavior. Once an attacker is detected, its token will be revoked, which deprives the attacker of the network access. This security solution is fully localized in that all the basic operations are performed in the local neighborhood. Each node monitors the behavior of its neighbors, verifies and issues tokens to its neighbors, and interacts only with its legitimate neighbors. When the attackers are detected in their local neighborhood, all the nodes in the network will be notified through the intrusion reaction mechanism, thus effectively isolating them and preventing them from further launching the attack. In essence, this security solution exploits collaboration among local nodes without completely trusting any individual node.

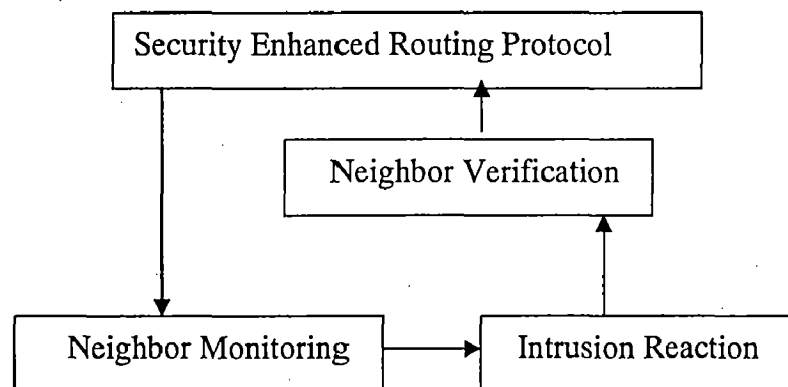
Figure 6.1 illustrates the composition of this security solution, which consists of four closely interacted components:

**Neighbor Verification:** Which describes how to verify whether each node in the network is a legitimate or malicious node.

**Security Enhanced Routing Protocol:** Which explicitly incorporates the security information into the ad hoc routing protocol.

**Neighbor Monitoring:** Which describes how to monitor the behavior of each node in the network and detect occasional attacks from malicious nodes.

**Intrusion Reaction :** Which describes how to alert the network and isolate the attackers.



**Figure 6.1 Framework of the network layer security solution**

In this framework, neighbor verification and security enhanced routing protocol proactively prevent the attackers from disrupting the network operations; neighbor monitoring detects any misbehavior in both routing and packet forwarding services; and intrusion reaction serves as the bridge between neighbor monitoring and neighbor

verification and isolates the detected attackers. In the following sections, the design of these four components in detail is described.

### 6.5.2 NEIGHBOR VERIFICATION

The neighbor verification mechanism is based on tokens and employs the asymmetric cryptographic primitives, specifically the de facto standard RSA. There is a global secret key pair SK/PK, and PK is known to all nodes when they join the network. Each legitimate node carries a token stamped with an expiration time and signed by SK. The token of a node contains the following three fields < identity, signing time, expiration time > each node periodically broadcasts the token in the hello message to its neighbors. Token verification is simple in that a token is valid if and only if

- 1) It is held by the node with the same identity as stated in the owner identity field
- 2) It has not expired and
- 3) It is signed by SK.

Any node without a valid token will be regarded by its neighbors as a malicious node, and all its packets, both routing updates and data packets, will be dropped. This is realized by employing the local trust model proposed in the dissertation. The next section briefs the localized token issuing process.

#### 6.5.2.1 LOCALIZED TOKEN ISSUING

Consider the case that a node in the network, which already possesses a token, needs to renew its current token. The message handshake in the localized token issuing process is illustrated in Figure 6.2. Before the expiration time of a node's current token, it broadcasts a TREQ (Token Request) packet to its neighbors, which contains its current token and a timestamp. Each node also keeps a Token Revocation List (TRL) learned from the intrusion reaction component. When a node receives a TREQ packet, the TRL will be used to decide whether to serve the request or not.

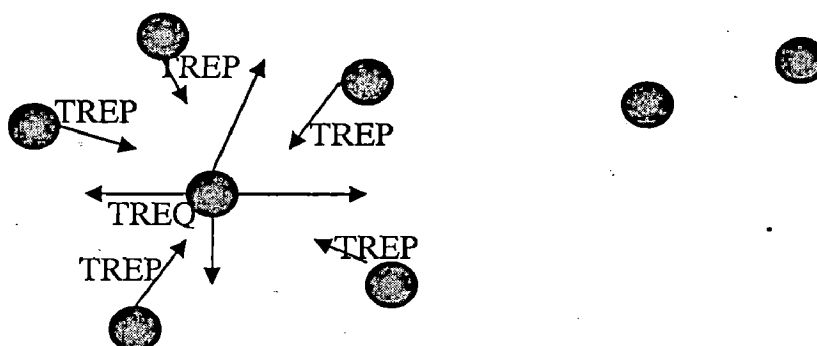


Figure 6.2 Message handshake in the localized token issuing process



Specifically, when a node receives a TREQ packet from its neighbor, it extracts the token from the packet. It checks whether the TREQ packet comes from the owner of the token therein, and whether the token has already been revoked by comparing it with the TRL. If the token is still valid and the source of the TREQ packet matches the owner of the token, it constructs a new token in which owner identity is equal to that in the old token, signing time is equal to the timestamp in the TREQ packet, and expiration time is determined by the additive increase algorithm described below. It then signs the newly constructed token using its own share of SK, encapsulates the partially signed token in a TREP (Token Reply) packet, and then unicasts the TREP packet back to the node from which it received the TREQ packet. TREQ packets from incorrect sources or containing revoked tokens will be silently dropped. When the node, which needs to renew its token receives  $k$  TREP packets from different neighbors, it can combine these partially signed tokens into a single token signed by SK.

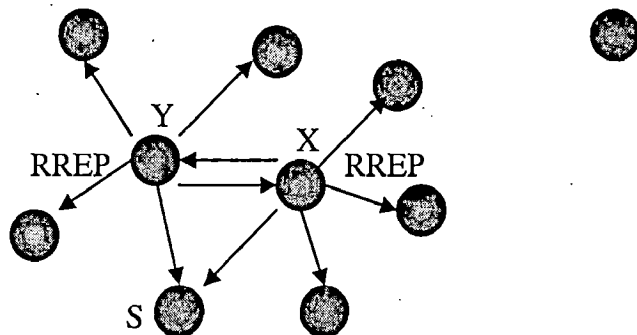
There is another case of token issuing: a newly joined node needs to obtain its first token. This is similar to the token renewing case from the message handshake perspective. In order to join the network, a node also broadcasts a TREQ (Token Request) packet, containing its identity and the current time, in its local neighborhood. Its neighbors apply the same rules as described above to determine whether to serve this request and, if they decide to issue the token, apply the same process to construct and send back the partially signed tokens. However, the expiration time field in the first token is different from that in a renewed token.

### **6.5.3 SECURITY ENHANCED ROUTING PROTOCOL**

This solution extends the AODV protocol and explicitly incorporate the security information in the security enhanced ad hoc routing protocol, which is called as AODV-S. AODV-S retains most of the AODV mechanisms, such as on-demand path discovery, reverse path setup, forward path setup, and softstate associated with the route entry, path maintenance, local connectivity management. In this section, we will mainly describe the difference between them. Each AODV-S node maintains the list of all its verified neighbors, which possess valid tokens. This can be easily achieved by taking advantage of the local connectivity management in AODV and the neighbor verification mechanism described earlier. Each AODV-S node only interacts with its

verified neighbors. All the routing updates received from a neighbor without a valid token will be dropped. One possible approach to prevent routing updates misbehavior is to encrypt or attach Message Authentication Code (MAC) to all routing updates. However, we do not take this approach due to several considerations. First, in distance vector routing protocols, the routing information is compressed into several routing metrics, such as hop count and destination sequence number in AODV. Each node disseminates routing updates on its own will, and each routing update is only directly visible to the neighbors of the sender, as opposed to source routing protocols. Neither encryption nor MAC based on one node own secret key can prevent compromised nodes to disseminate malicious routing updates. Second, encryption or MAC based on the source-destination pairwise secret key requires that each pair of nodes share a secret key (in the symmetric cryptography), or each node has the public keys of all the other nodes (in the asymmetric cryptography), which can be hardly achieved in the dynamic ad hoc networks without a centralized key management service. Third, encryption/decryption of the routing updates causes significant computation load, and may be utilized by the attackers to launch DoS attack. Instead, we rely on the redundancy of the routing information to prevent routing updates misbehavior.

The basic idea is that each node explicitly claims the next hop node when it disseminates a new routing update, and each node keeps track of the route entries previously announced by its neighbors. In this way, each node can maintain part of the routing tables of its neighbors. This redundancy of the routing information makes it possible for a node to examine the correctness of routing updates, because the execution of the distributed Bellman-Ford algorithm should be based on the route updates previously disseminated by some neighbors, which this node may also have received.



**Figure 6.3 Using redundant routing information to examine the correctness of routing updates**

Figure 6.3 illustrates how the routing updates are examined based on the redundant routing information. Node S is the neighbor of both node X and node Y. S has kept track of the route entries previously announced by Y. When S receives a new routing update from X and the next hop claimed by X is Y, it can examine the correctness of this routing update by comparing the new route entry with the corresponding route entry previously announced by Y. We can view this process as that S is reconstructing the execution of the distributed Bellman-Ford algorithm performed by X. Specifically, we add one more field, next hop in the RREP packet, which means that the RREP packet in AODV-S contains six fields < addr, dest addr, dest sequence #, hopcnt, next hop, lifetime >. We also modify the way of propagating RREP packets. Each AODV-S node broadcasts the RREP packets to its neighbors, as opposed to uncasting the RREP packets along the reverse path in AODV. Finally, in addition to its own routing table, each AODV-S node also maintains the route entries announced by its verified neighbors. An announced route entry contains the following fields < addr, dest addr, dest sequence #, hop cnt, next hop, lifetime >

In AODV-S, when a node receives a RREP packet, it first examines the correctness of the routing update, using the simple algorithm described in Section 6.5.4.1. Incorrect RREP packets will be dropped. If the routing update in the RREP packet is correct, it updates its own routing table in a way similar to AODV, and updates its cache of the announced route entries of the corresponding neighbor. A node receiving a RREP packet also checks whether it previously sent the corresponding RREQ packet. If so, it rebroadcasts the RREP using its updated route table entries. In this way, the RREP packets will be propagated back to the source.

#### **6.5.4 NEIGHBOR MONITORING**

In the neighbor monitoring mechanism, each node is responsible for monitoring the behavior of its neighbors and detecting any misbehavior in both routing and packet forwarding services. Those misbehavior will be regarded as indications of attacks. Furthermore, all the nodes in one neighborhood collaborate with each other to improve the accuracy of monitoring results and withstand sophisticated attacks.

#### **6.5.4.1 MONITORING ROUTING UPDATES MISBEHAVIOR**

The routing updates misbehavior is detected by examining the correctness of routing updates. When a node receives a RREP packet broadcasted by its verified neighbors, it first examines the correctness of the newly offered route. We consider the example scenario in Figure 6.3 again, in which S and X are the receiver and sender of the RREP packet, respectively, while D and Y are the destination and the next hop specified in the RREP packet, respectively. If S is also the neighbor of Y, it compares the new route entry offered by X with its cached route entry previously announced by Y and destined to D. The new route entry is correct if and only if the sequence number in the two route entries are the same, and the hop count in the new route entry is one larger than the hop count in the cached route entry announced by Y. If the routing update is not correct, the RREP packet is dropped and node S broadcasts a SID (Single Intrusion Detection) packet to its neighbors. Note that it is also possible for S to be out of the neighborhood of Y. In this case; S will skip this examination process, because S has no information about the next hop node in the offered route.

The routing updates examination algorithm has some weaknesses in that it might not work well in several situations:

- 1) Y only stayed in S's neighborhood for a short period of time due to mobility, so that S has not recorded all the route entries announced by Y
- 2) S did not receive the previous route updates broadcasted by Y due to channel error and contention
- 3) Y has increased the lifetime of a route entry, but S is not aware of this change and has deleted it from its cache.

It is also susceptible to the blackmail attack, in which an attacker blackmails its legitimate neighbors as misbehaving nodes. However, mechanism for collaborative monitoring is not discussed here.

#### **6.5.4.2 MONITORING PACKET FORWARDING MISBEHAVIOR**

In addition to monitoring routing updates misbehavior, each node also monitors its neighbors to detect misbehavior in data packet forwarding service. This can be done in ad hoc networks through overhearing the channel in promiscuous mode in 802.11 link layer.

We currently consider three kinds of packet forwarding misbehavior, namely, packet dropping packet duplicating, and network layer packet jamming, and develop simple algorithms for each of them. Packet dropping means that a node drops the packets that it is supposed to forward for its neighbors packet duplicating means that a node duplicates the packets that it has already forwarded; and network layer packet jamming means that a node sends too many packets and occupies a significant portion of the bandwidth. The packet dropping detection algorithm is similar to the watchdog technique found in literature.

The watchdog was originally proposed for DSR, in which the sender explicitly lists the route in the data packet header. It cannot be directly applied in AODV, because if on node receives a packet, its neighbors do not know which node it should forward the packet to, and can not tell whether it has forwarded the packet in the correct manner. However, the watchdog can be extended to work with AODV-S, because each AODV- S node keeps track of the route entries announced by its neighbors, which explicitly include the next hop field. Specifically, each node overhears the channel at all time and records the headers of the recent packets it has over- heard. If it overhears one packet sent to its neighbor, say, X for forwarding, it checks its cache of the route entries announced by X and determines the next hop node to which X should forward the packet. If it does not overhear the packet being forwarded by X to the correct neighbor after Drop Time seconds, it considers this packet to be dropped. If the bandwidth corresponding to the packets dropped by X exceeds the threshold Drop Bandwidth it considers X as an attacker and broadcasts the SID (Single Intrusion Detection) packet. The packet duplicating and packet jamming detection algorithms also utilize the information obtained by overhearing the channel. If one node overhears that the bandwidth corresponding to the duplicate forwarding of packets by its neighbor X exceeds the threshold Duplicate Bandwidth or the bandwidth corresponding to the packets sent by its neighbor X exceeds the threshold Sending Bandwidth it considers this as the indication of an attack and broadcasts the SID packet.

The localized monitoring mechanism executed by each node is intrinsically inaccurate due to the inaccuracy in the information obtained by overhearing the channel. The detection accuracy is also sensitive to multiple factors, such as channel error, mobility, parameters in the detection algorithm, etc.

### 6.5.5 INTRUSION REACTION

The intrusion reaction mechanism serves as the bridge between neighbor verification and neighbor monitoring. Recall that each node keeps a TRL (Token Revocation List). When a node receives a TREV packet, it checks whether the packet is signed by SK, and whether the revoked token is already on the TRL. TREV packet that is not signed by SK or contains a token on the TRL is silently dropped. Otherwise, it adds the token into the TRL and then rebroadcasts the TREV packet. In this way, eventually every node will add the revoked token into its TRL. Meanwhile, the neighbors of an attacker deem the links between them and the attacker that are currently in use as broken, and use the path maintenance mechanism in the routing protocol to cancel out these links.

Each entry in the TRL is associated with a lifetime, which is equal to the expiration time in the corresponding token. When the token expires, none of the nodes needs to maintain this revocation information. The soft state associated with TRL entries can reduce both the storage overhead and the checking overhead when a node receives the token renewal requests from its neighbors. Recall that each node only interacts with verified neighbors. The intrusion reaction mechanism guarantees that the attacker is isolated in the network right after it is detected, and it will never be issued a new token again in the future. Although the TREV packet is flooded in the network, the communication overhead is still affordable, because the intrusion reaction process is triggered only once for each attacker or compromised node.

### 6.6 CONCLUSION

One fundamental challenge for the security design in mobile ad hoc networks is that such networks do not possess any pre-existing infrastructure support. Therefore, the security solution should be provided in a distributed manner. This proposal explores the self-organized security design for the ad hoc networks using the proposed design of the dissertation. To this end, we have presented a unified network-layer security solution that protects both routing and packet forwarding functionalities. Some nice features of this solution include fully localized design, easy support of dynamic node membership, and limited intrusion tolerance capacity, decreasing overhead over time.

SIMULATION AND RESULTS

---

## 7.1 INTRODUCTION

*Ns-2* simulator has been used to implement all the communication protocols described in above chapter. Simulations have been carried out to test the performance of the proposed system and compared with that of Centralized scheme and hierarchical scheme. The simulation environment includes a two-ray ground reflection model and IEEE 802.11 MAC protocol. The simulated network has a square shape of 1200 x 1200m where all wireless ad hoc mobile nodes share a single radio channel of 11 Mbps. The source and destination nodes associated with flows are distributed among the mobile nodes in the wireless ad hoc network. An application-layer approach was followed and has developed an UDP-like transport agent that allows for delivery of actual application data units (ADUs) and one-hop broadcast.

In order to evaluate the communication efficiency of the proposed protocol, the following metrics are used: *Success ratio* measures the ratio of the number of successful certification services over the number of attempts during the simulation time. *Average delay* measures the average latency for each node to perform a certification service, in the case of self initialization, the average time it needs to become a fully functional member of the network, from the moment it joins in. *Average number of failures* measures the number of times an entity fails on average, before successfully accomplishing its certification.

The performance of the proposed protocols is studied by running experiments in networks with sizes that range from 30 to 100 nodes. The node mobility varies from 1, 3, 5, 10, 15 and 20 m/sec. The random waypoint model in *ns-2* is used to emulate mobility patterns. In the simulations, the expiration time of the certificate is selected as five minutes, and the Coalition size  $K=5$ , except for the topologies that consist of 30 nodes, where

## 7.2 COMPARISON OF PROPOSED MECHANISM WITH CONVENTIONAL APPROACHES

This section shows why two common approaches, namely the centralized and the hierarchical approach, do not work well in large mobile networks. Certificate renewal service is used as an example to evaluate these approaches in the network simulator. We measured results in two ways, one with mobility speed set at the values varying from 1, 3, 5, 10, 15 and 20 m/sec and other with constant mobility with various network sizes.

### 7.2.1 PERFORMANCE COMPARISON WITH VARYING MOBILITY

We first examine the effectiveness of the certificate renewal service in terms of three performance parameters success ratio, average delay, number of retries, as the node speed increases from 1 m/sec to 20 m/sec and the channel error rate have the values 1% and 10%.

#### 7.2.1.1 COMPARISON OF SUCCESS RATIO

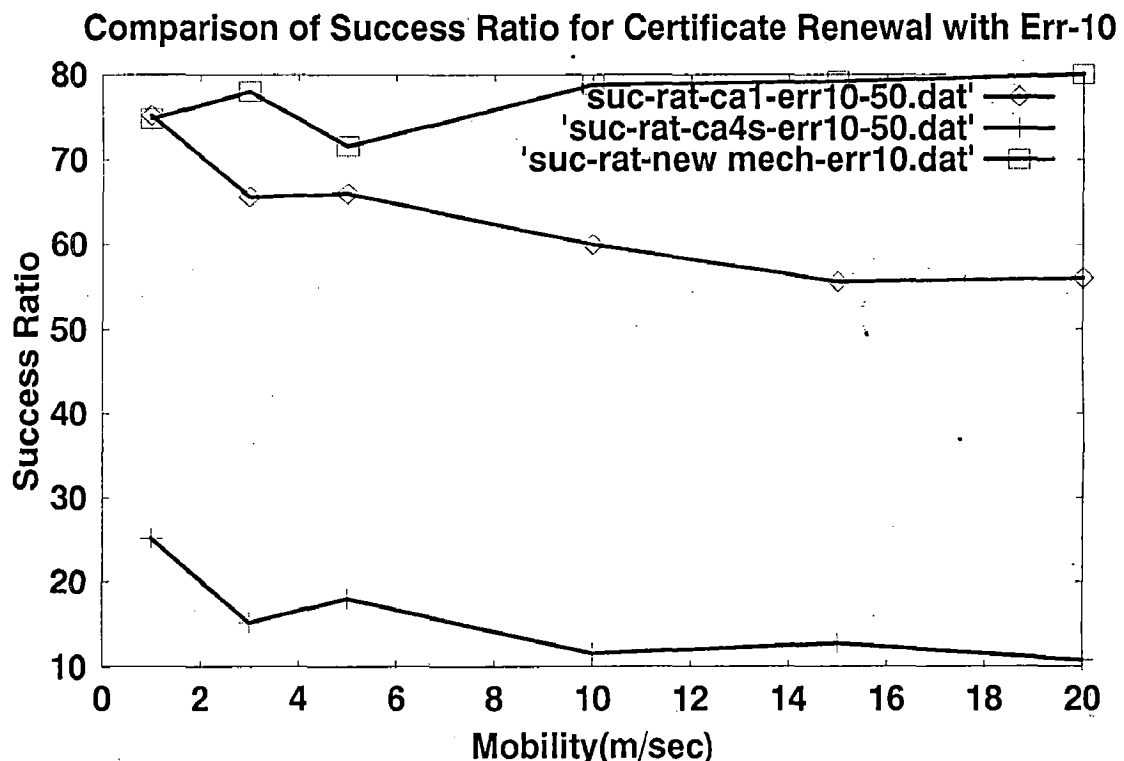


Figure 7.1 Comparison of Success Ratio for Certificate Renewal with Error rate



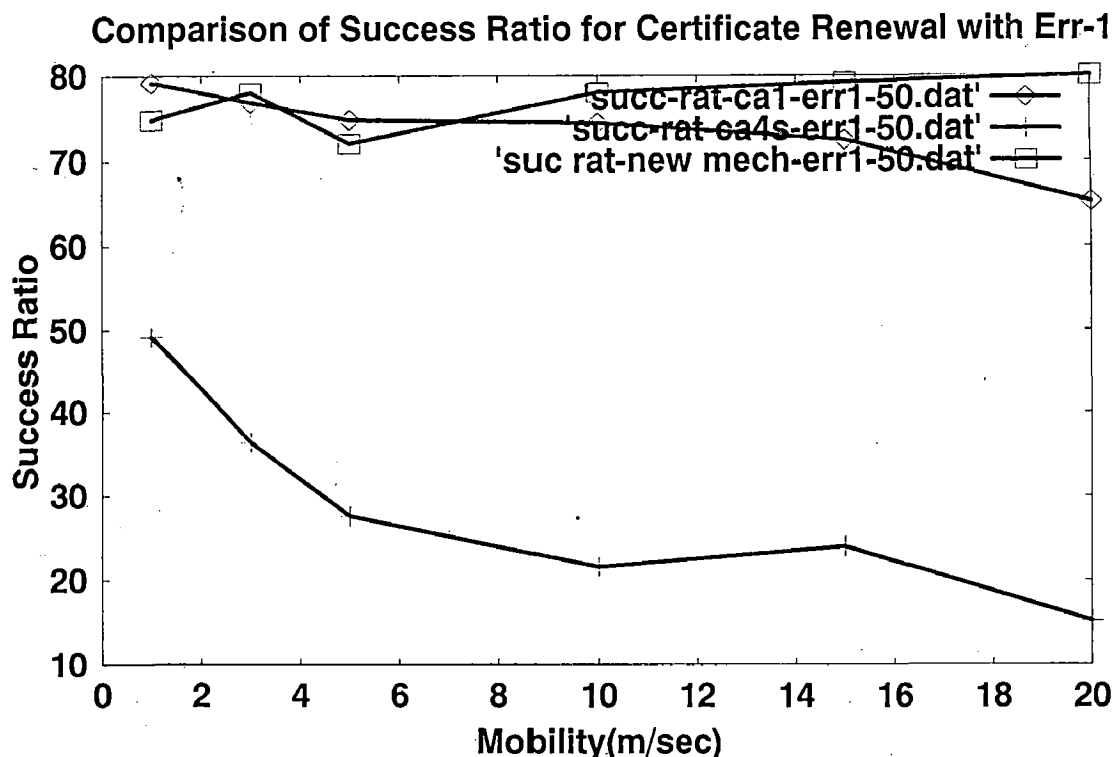


Figure 7.2 Comparison of Success Ratio for Certificate Renewal with Error rate

1

The above graphs (fig 7.1 and 7.2) show the effectiveness of the certificate renewal service, as the node speed increases from 1m/sec to 20m/sec with the different channel error rates (1% and 10%). From the graph it is clear that the Success ratio of the proposed distributed certification service is almost more than 80% while the centralized and the hierarchical solutions fail. This confirms not only the effectiveness of the proposed protocol in terms of mobility, but also service ubiquity, since during the simulation time, every node is required to renew its certificate multiple times, which means that the service should be available at any part of the network topology, at any time.

As observed from fig. 7.1 and fig 7.2, In the centralized server model, when the channel error is 1% the success ratio is 50 and where as it reduced to 25 when the channel rate is increased to 10% this is due to the multihop communication to the centralized server in highly poor channel. In contrast, the proposed new approach maintained the same value of success ratio as 78 even the channel rate has increased to 10% this is because the localized trust model needs to communicate to one hop neighbors. This shows the immunity of the proposed system.

### 7.2.1.2 COMPARISON OF AVERAGE DELAY

Comparison of Average Delay for Certificate Renewal with Err-10

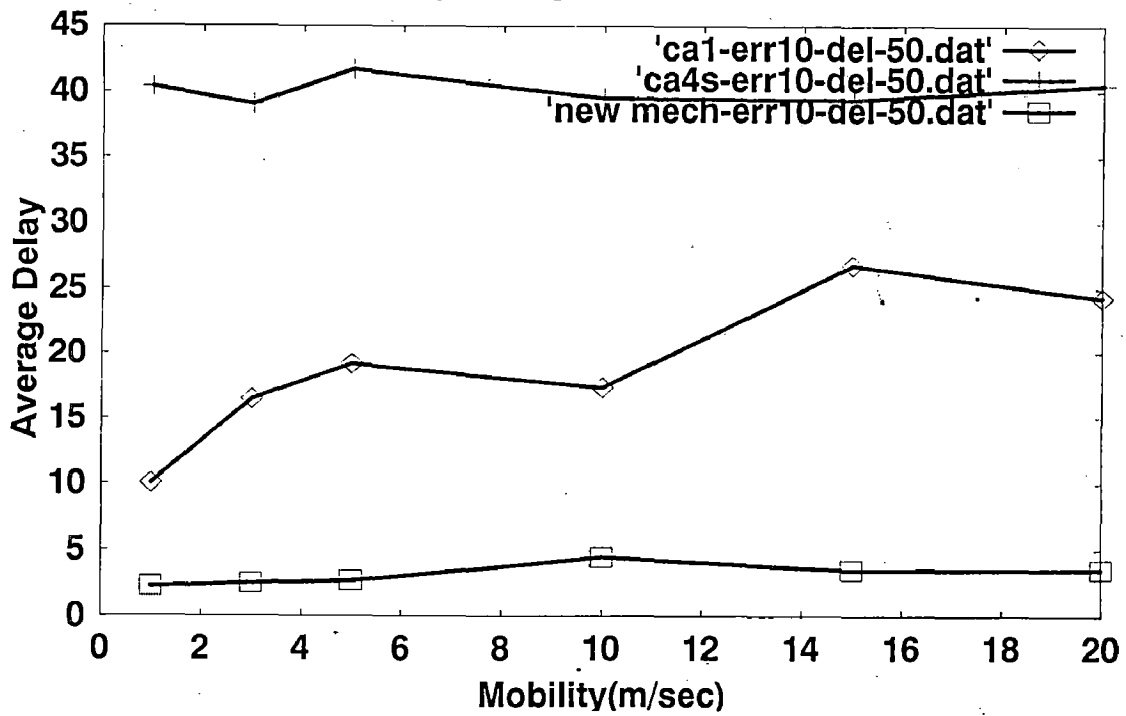


Figure 7.3 Comparison of Average Delay for Certificate Renewal with Error rate 10

Comparison of Average Delay for Certificate Renewal with Err-1

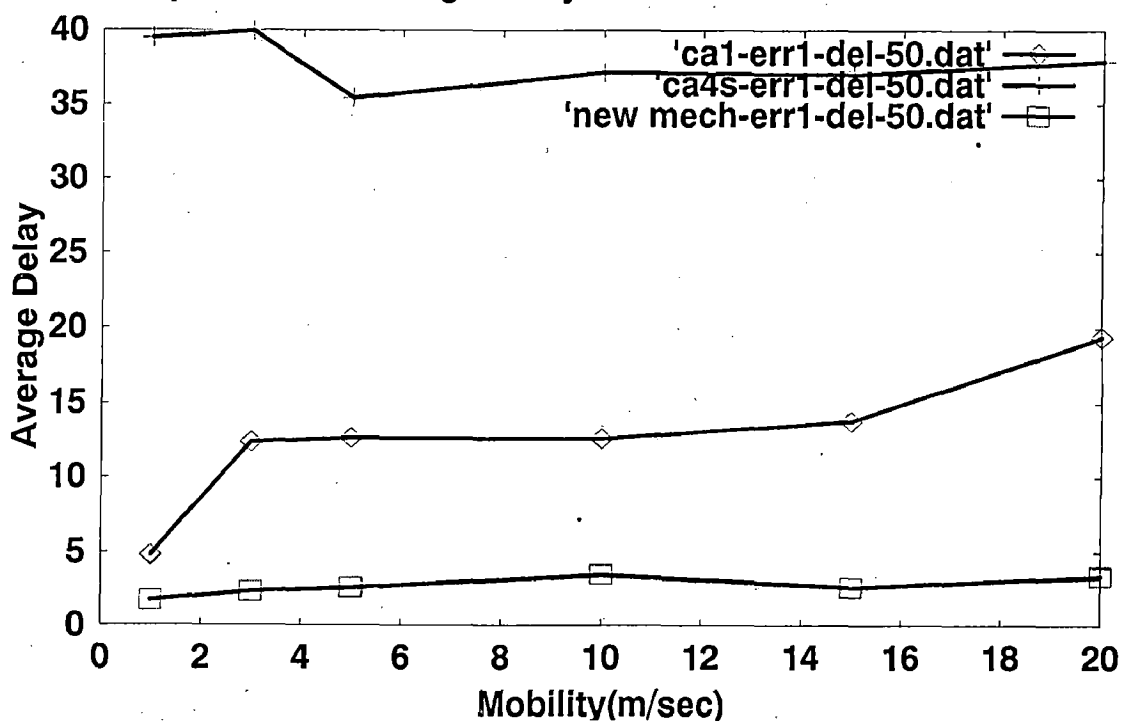
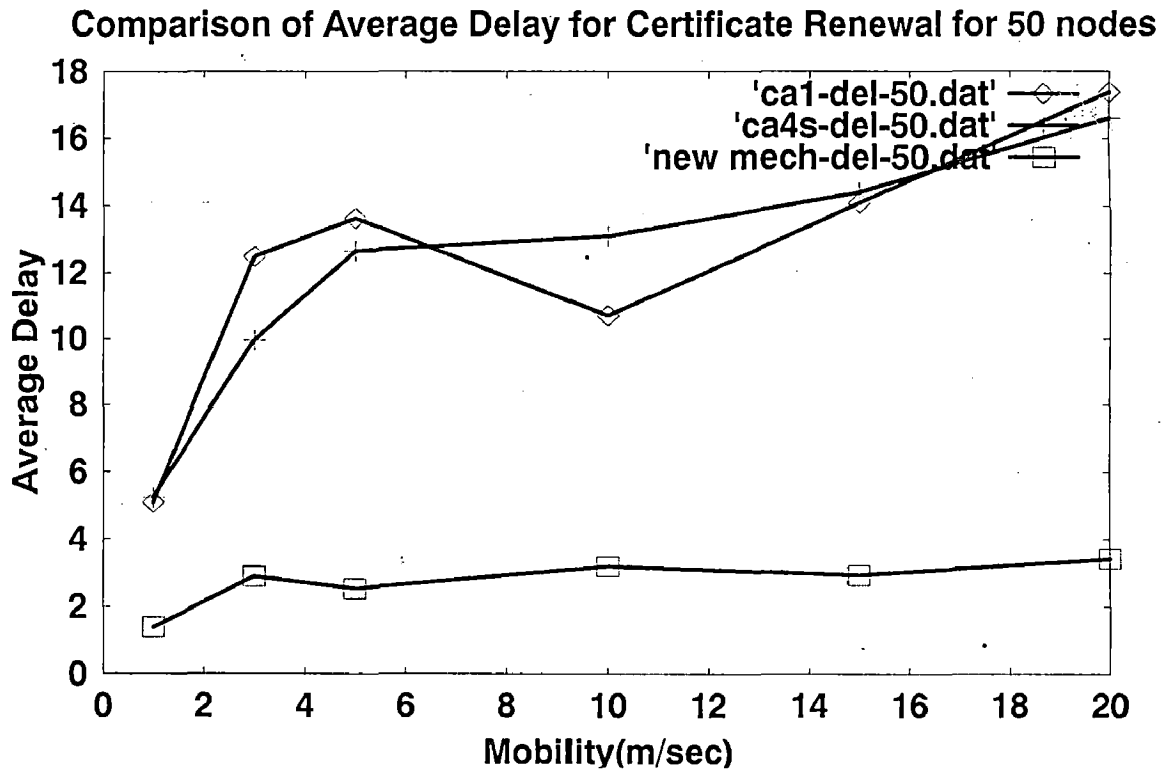


Figure 7.4 Comparison of Average Delay for Certificate Renewal with Err rate 1



**Figure 7.5 Comparison of Average Delay for Certificate Renewal for 50 nodes with out channel error**

The above graphs (Fig. 7.3, Fig. 7.4 and Fig. 7.5) evaluate the performance of the proposed protocol using another metric *Average Delay*. We measure the average delay experienced by the nodes as the node speed increases from 1m/sec to 20 m/sec for three different cases: Centralized, Hierarchical and Distributed Certification services while keeping the number of nodes constant (in this case the number of nodes is 50) for different error rates 1 and 10. We observe that the average delay almost remains unchanged as mobility speed grows from 1m/sec to 20 m/sec for the proposed solution whereas both centralized and hierarchical solutions incur much higher delay, which also greatly fluctuates, thus making it hard to predict some useful information, such as the future expiration time in certificate renewal and consequently the frequency of renewal.

It is also observed that the Average delay for a certain mobility is increasing for both centralized and hierarchical model where as the proposed mechanism maintains the average delay constant with varying channel error rates.

### 7.2.1.3 COMPARISON OF NUMBER OF RETRIES

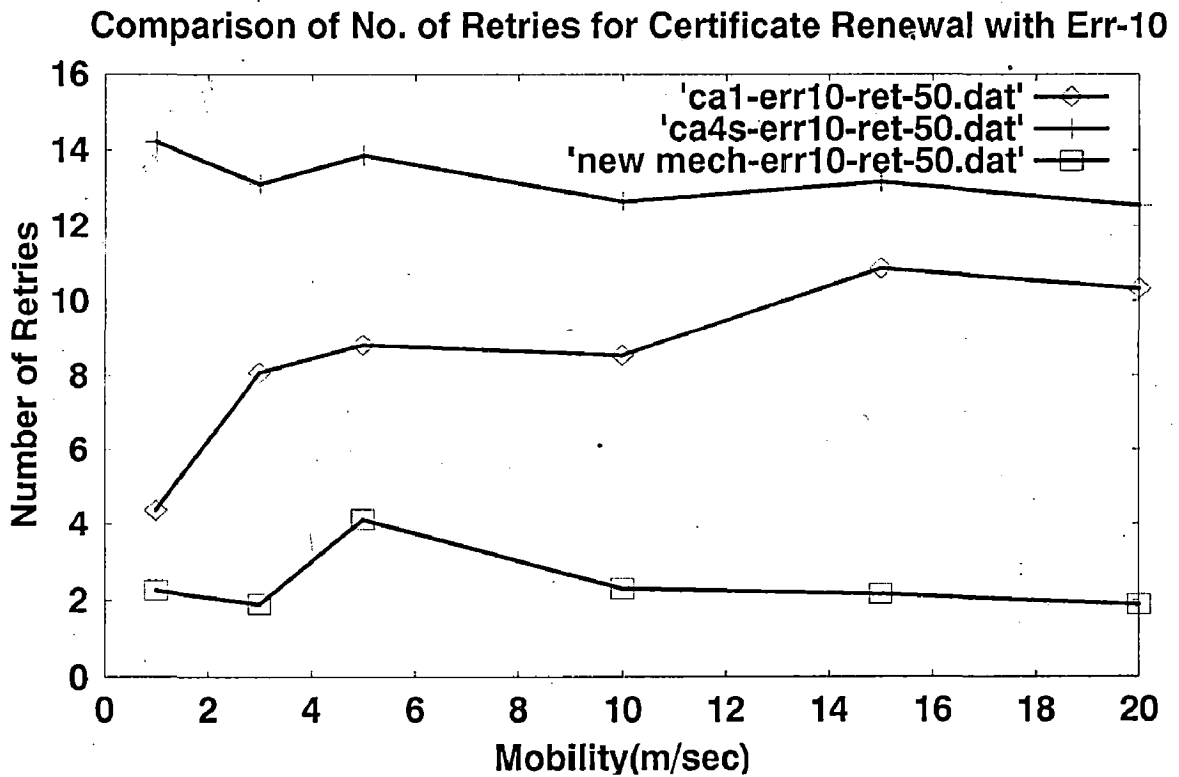


Figure 7.6 Comparison of Number of Retries for Certificate Renewal with Error rate 10

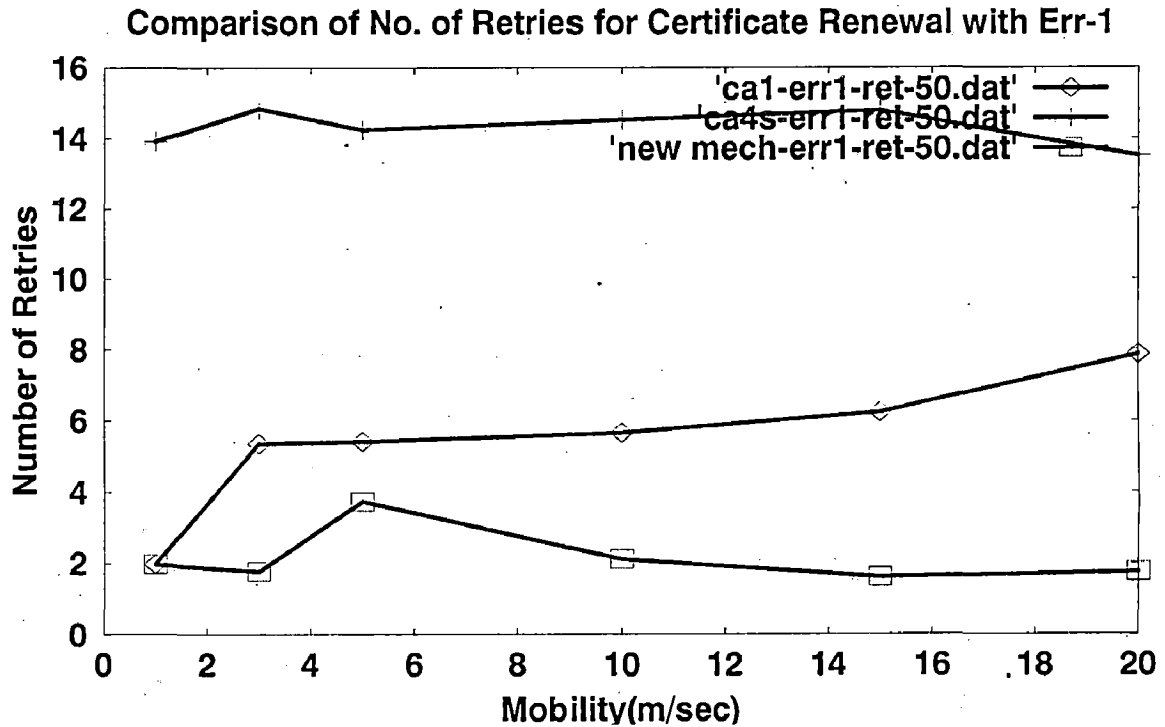
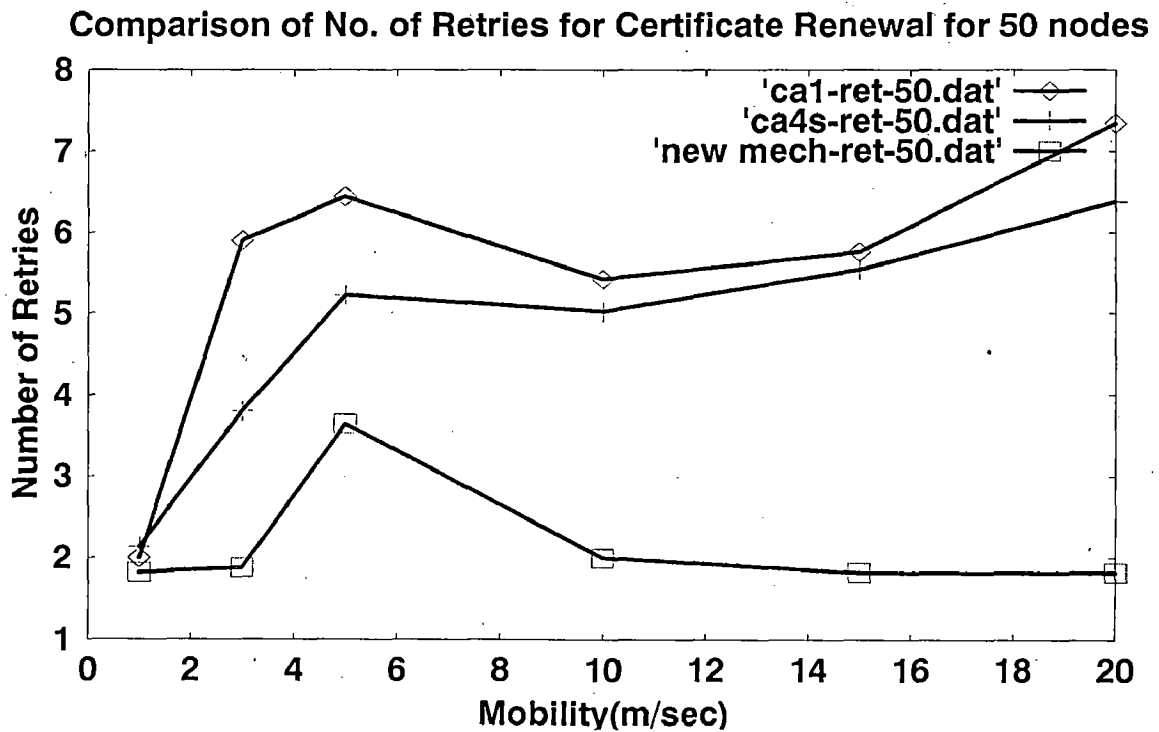


Figure 7.7 Comparison of No. of Retries for Certificate Renewal with Error Rate



**Figure 7.8 Comparison of Average No.of failures for Certificate Renewal for 50 nodes with out channel error**

The above graphs (Fig. 7.5, Fig. 7.7 and Fig. 7.8) demonstrates the robustness of the proposed distributed certification services from another perspective using the metric *Average number of failures*. We measure the number of failures each node is experiencing on average, before successfully receiving its service by varying the node mobility from 1m/sec to 20m/sec for number of nodes 50 with error rate 10. We observe that the proposal requires significantly less effort in providing the service, compared to centralized and hierarchical cases. Moreover, we observe that mobility helps the protocol. As node speed increases, the average number of failures for each node not only remains unchanged in the proposed approach but also diminishes.

It is also observed that the Number of retries for a certain mobility is increasing for both centralized and hierarchical model where as the proposed mechanism nodes to get the certificate renewed with minimum number of retries even the channel becomes more erroneous.

## 7.2.2 PERFORMANCE COMPARISON WITH VARYING NETWORK SIZE

We measured the scalability of the proposed mechanism with increasing number of nodes and mobility set with a value 5 m/sec. We evaluate effectiveness of the certificate renewal service in terms of same three performance parameters success ratio, average delay, number of retries as discussed in the above section with the increase in number of nodes from 30 to 70 and the channel error rate at 1%.

### 7.2.2.1 COMPARISON OF SUCCESS RATIO

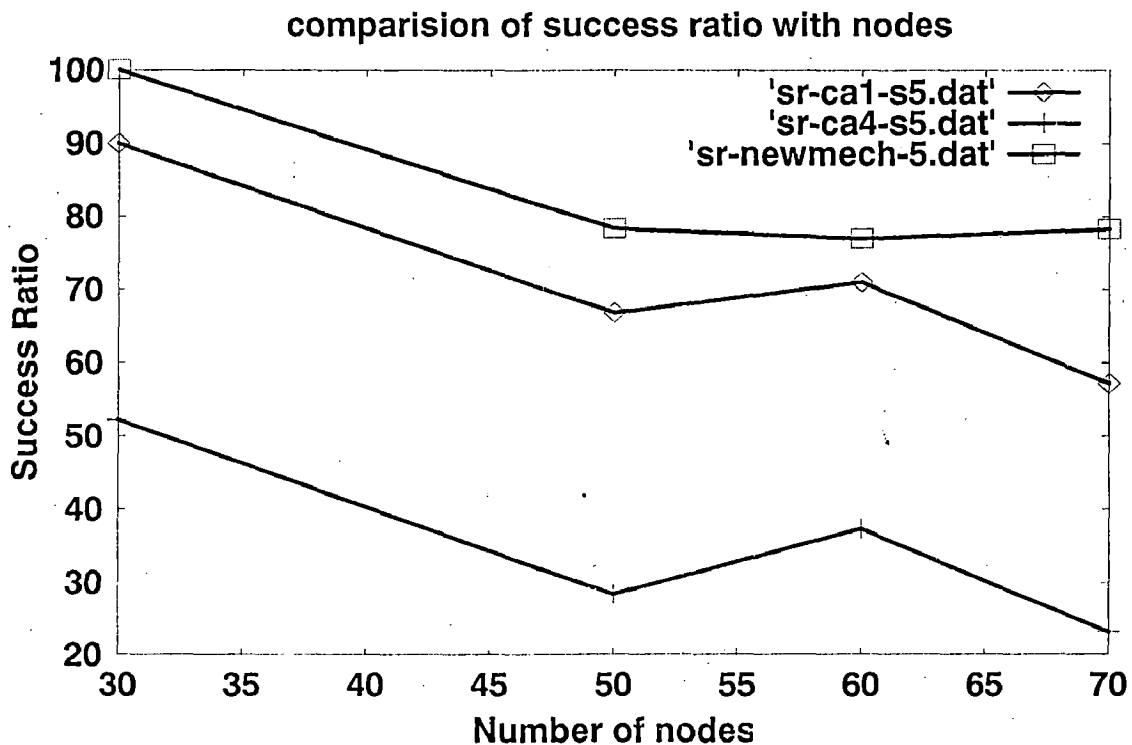


Figure 7.9 Comparison of Success Ratio with increase in number of nodes

The above graph (Fig 7.9) shows the effectiveness of the certificate renewal service, as the number of nodes increase from 30-50. From the graph it is clear that the Success ratio of the proposed distributed certification service is almost maintained around 80% while the centralized and the hierarchical solutions perform poor in large networks. This confirms the scalability of the proposed mechanism providing service ubiquity. It is also observed that, the rate of decrease in success ratio in the proposed mechanism is less than the other two mechanisms.

### 7.2.2.2 COMPARISON OF AVERAGE DELAY

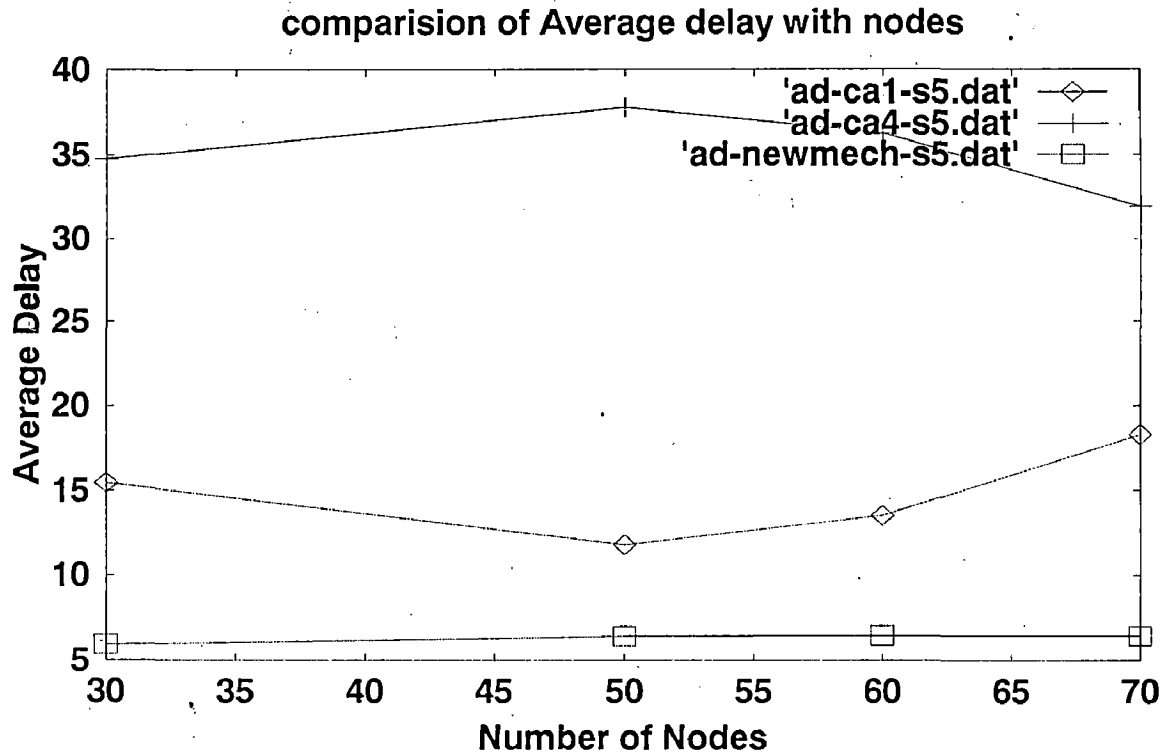


Figure 7.10 Comparison of Average Delay with number of nodes

From the above graph (Fig 7.10), We observe that the average delay required by each node to contact the local CA is much larger compared with the proposed localized approach. The proposed scheme maintains a constant and bounded delay even in large networks where as in other cases it tends to increase with the size of the network.

### 7.2.2.3 COMPARISON OF NUMBER OF RETRIES

We measured the number of failures each node is experiencing on average, before successfully receiving its service by varying the number of nodes from 30 to 70 for error rate 1%. We observe that the proposed mechanism requires significantly less effort in providing the service, compared to centralized and hierarchical cases. Moreover, the number of retries decrease with number of nodes in a contrast way with other two mechanisms. This performance can still be seen if a large channel error rate.

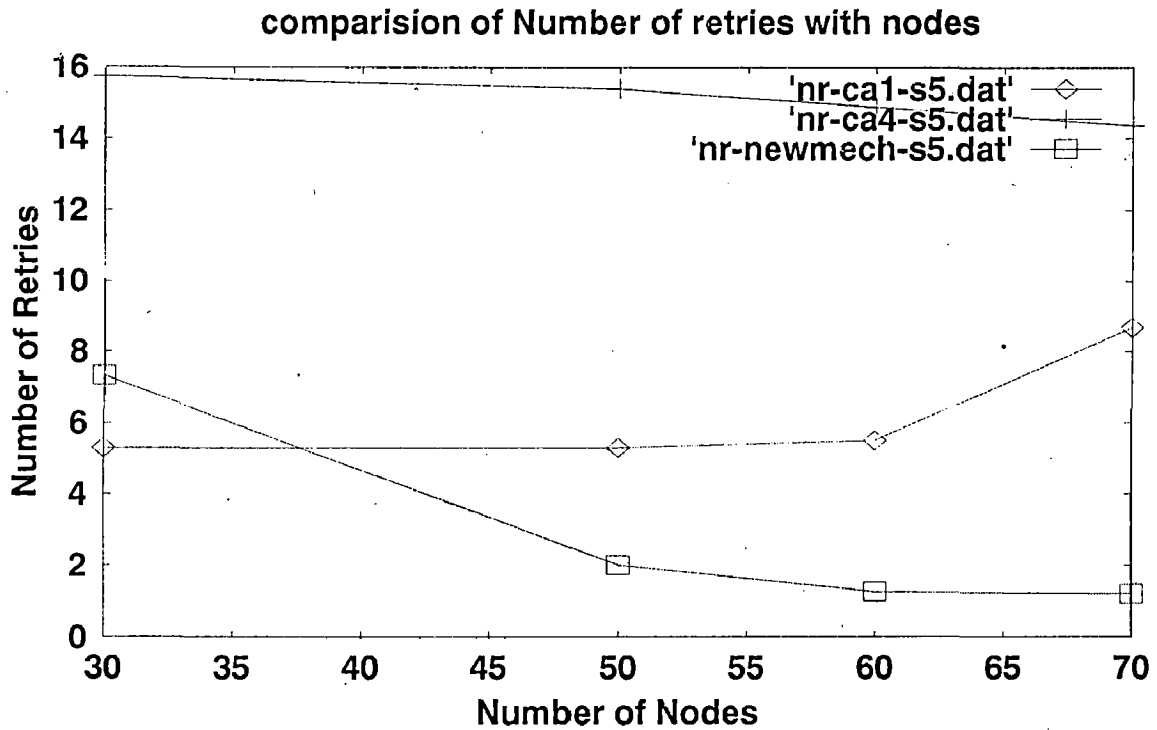


Figure 7.11 Comparison of number of retries with number of nodes

### 7.3 ANALYSIS OF PROPOSED ALGORITHMS

This section evaluates various proposed algorithms in this dissertation for providing ubiquitous security in ad hoc network. The performance parameter 'delay (latency)' is taken, which measures how fast is the proposed algorithm in various mobility environments. Firstly the self-initialization algorithm is evaluated followed by Proactive share update algorithm.

#### 7.3.1 ANALYSIS OF SELF-INITIALIZATION ALGORITHM

The below graph (Fig. 7.12) evaluates the *self initialization protocol* of the proposed certification services by considering the time needed for the nodes that haven't been already initialized by the root-of-trust to become fully functional entities ( by obtaining a secret share).In the experiment conducted,  $2 \cdot k$  nodes of the topology are assumed to have been initialized by an imaginary dealer, so that the remaining nodes be able to find a coalition of  $k$  neighbors, in order to perform self-initialization. The graph shows self-initialization latency for network topology of 50 nodes and for four different node speeds of 5,15,30 and 35m/sec. We see that the first 40% of the nodes need almost 40 seconds to self-initialize. But as soon as the sufficient number of



nodes manage to self initialize the convergence of the algorithm is pretty fast, since those nodes may in turn help others to self initialize.

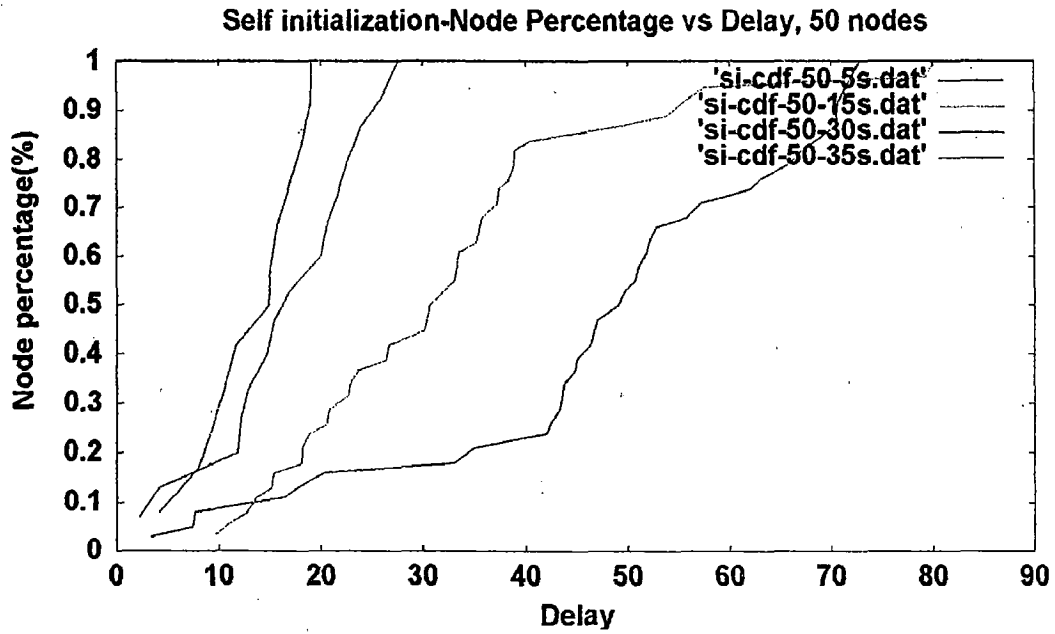


Figure 7.12 Analysis of Self-initialization protocol

### 7.3.2 ANALYSIS OF PROACTIVE UPDATE ALGORITHM

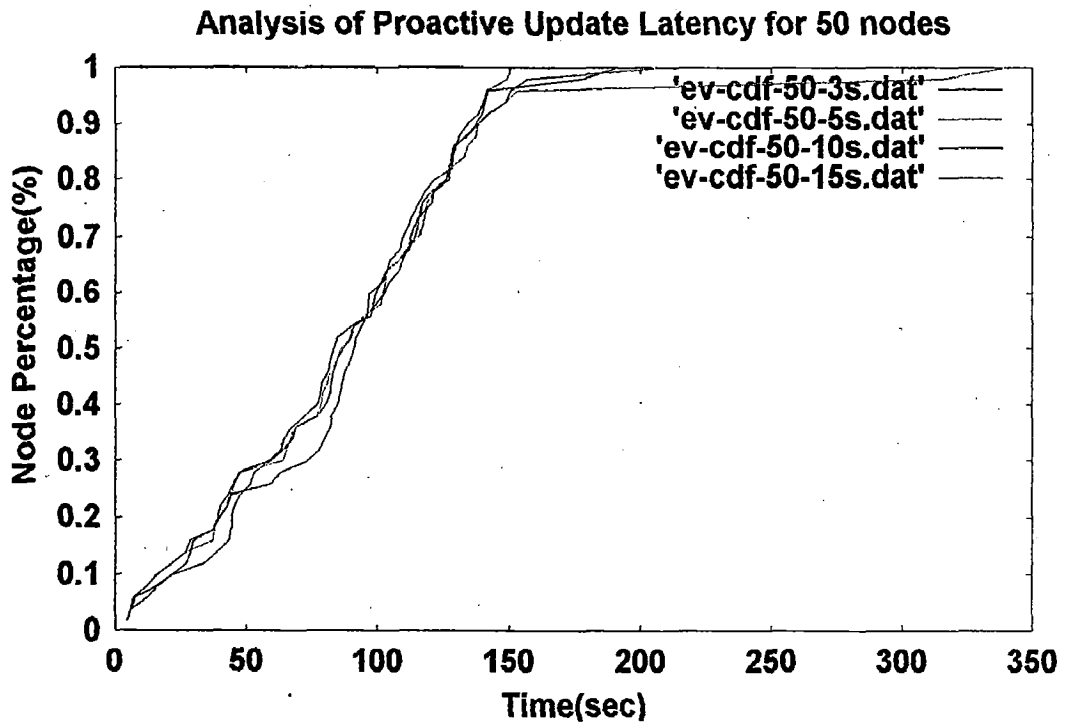


Figure 7.13 Analysis of Proactive Update Latency for 50 nodes

The above graph (Fig. 7.13) provides a detailed analysis of the *proactive update latency*, for network topology of 50 nodes for four different node speeds (3,5,10,15m/sec, that correspond roughly to low, medium and high mobility). We observe that the first 20% of the nodes needs almost 50 seconds to update. But as soon as a sufficient number of nodes manages to acquire their new secret shares, then the convergence of the algorithm is pretty fast, since those nodes may in turn help others to update their share; we reach 80% in another 80 seconds of the simulation time. We also observe that the evolution of the algorithm is similar for all mobility speeds, which shows that the proposed design is tolerant to mobility.

## CONCLUSION AND FUTURE WORK

---

In this dissertation, a solution to security support secured routing in wireless mobile networks is designed and implemented. The design has been motivated by four main factors: (a) No security system can be completely unbreakable. Therefore, the design has to work in the presence of such break-ins. (b) To maximize the service availability in each network locality; this is crucial to supporting ubiquitous services for mobile users. (c) The solution has to be fully decentralized to operate in a large-scale network. To this end, a few techniques are devised. Polynomial threshold secret sharing and proactive secret share update techniques are applied to resist break-ins. Certificate based and localized certification is used to maximize the service availability for mobile users. Finally, a self-initialization technique is devised to handle dynamic joins and leaves of networking nodes. The implementation, simulations have shown very positive results for the proposed approach. The new general architecture is also widely applicable in other contexts such as large sensor networks.

As a future work, the security can be still enhanced well with several design modifications as described below

**a) Initialization of the first  $k$  nodes** In Section 4.5, the assumption that the self-initialization process starts with  $k$  initialized nodes (i.e., they have got their polynomial shares). To initialize these very first  $k$  nodes, a dealer who knows the full certificate signing key  $SK$  and the associated polynomial  $f(x)$  of degree  $k-1$  is assumed. While this is the assumption of existing works on secret sharing [11, 12, 21, 24, 27, 30, 36], a possible mechanism to initialize the first  $k$  nodes without a dealer would be to let these  $k$  nodes generate a RSA key pair  $\{PK, SK\}$  distributedly [32, 33].

**b) Less than  $k$  neighbors** It is assumed that, a node  $v_i$  that is requesting certification services have at least  $k$  initialized neighboring nodes. However, due to high mobility and network heterogeneity, this may not always hold. Relaxing the proposed certification policies so that nodes beyond the one-hop neighborhood can serve certification requests can solve this problem. This can be implemented by proxy-

based approaches to extend the coverage of each node's certification service. However, the cost to this flexibility is the higher requirement of nodes' monitoring capability. They will have to monitor nodes that are more than one-hop away to guarantee an adversary cannot take advantage. Detailed mechanisms and methodology are out of the scope of this dissertation and it is left for future work.

**c) Obtaining the initial certificate** When a new node joins the system, It is assumed that the node already obtains an initial certificate. In essence, to issue initial certificates is the problem of registering users. Initial certificates can be obtained in two ways: (a) The node may be issued an initial certificate by an offline authority, after the authority verifies the authenticity through other means (e.g., in-person ID). (b) Any coalition of  $k$  networking nodes may be used to issue the initial certificate via collaborative admission control for this new node. The admission control policy has to be consistent with my trust model, system model and the adversary models.

**d) Cross certification** When two ad hoc networks merge, there is a need for the mechanisms for nodes originated from different networks to certify and authenticate each other. With these mechanisms a hierarchical infrastructure can be built on networks so that the network-wide activities such as initialization and sequential/parallel share updates would scale to the overall system size.

**e) Parameter  $k$  revisited** in the system model as defined in Section 3.3, It is assumed that each node has at least  $k$  legitimate neighboring nodes. This assumption is critical for the proposed certification services to be robust against the adversaries defined in Section 3.4. The parameter  $k$  also determines the availability of security services. In the current design, these three factors are coupled and represented by a single parameter  $k$ . This coupling effect limits the flexibility of the system. In some scenarios these three aspects may have conflicting goals. For instance, security may require  $k$  to be at least 10, but service availability requires  $k$  to be at most 7, and the network can only guarantee 5 legitimate neighbors. How to decouple these three aspects poses new challenges for future research.

**f) Localized trust model revisited** A localized trust model is proposed in Section 3.2, where trust is defined as agreement of any  $k$  legitimate entities, typically in a local neighborhood. In the present model  $k$  is a global criteria that is honored by each individuals. While the "localized" trust model is defined this way, another definition

would be based on specific local scenario, instead of some system-wide criteria. This new trust model would be more consistent with the common concept of being "localized". Specifically,  $k$  would be a parameter that is defined on the number of networking nodes in a specific locality. For instance, localized trust may be defined as half of local neighboring nodes' agreement and judgment. The association between security robustness and service availability can be potentially decoupled with this model. Also, this new definition of location-dependent trust model would be particularly desirable in a large heterogeneous ad hoc network interconnected by several independent autonomous parts, where a network-wide agreement on  $k$  may not be reached.

## REFERENCES

---

- [1] The mobile ad-hoc networks (MANET) working group,  
<http://www.ietf.org/html.charters/manet-charter.html>.
- [2] The Bluetooth special interest group, <http://www.bluetooth.com/>.
- [3] J. Kohl and B. Neuman, "*The Kerberos network authentication service (version 5)*," RFC- 1510.
- [4] A. Arsenault and S. Turner, "*Internet X.509 public key infrastructure*," draft-ietf-pkix-roadmap- 06.txt, 2000.
- [5] R. Housley et al., "*Internet X.509 public key infrastructure certificate and CRL profile*," RFC 2459, 1999
- [6] S. Garfinkel, "*PGP: Pretty Good Privacy*," O'Reilly & Associates Inc., USA, 1995.
- [7] A. Abdul-Rahman, "*The PGP Trust Model*," EDI-Forum: the Journal of Electronic Commerce, Vol. 3, pp.27-31, June1997.
- [8] D.B. Johnson and D.A. Maltz, "*Dynamic source routing in ad hoc wireless networks*," In Mobile Computing, edited by T. Imielinski and H. Korth, Chapter 5, Kluwer Publishing Company, pp. 153-181, 1996.
- [9] W. Stallings, "*Cryptography and Network Security*," Prentice Hall, N.J., 1999.
- [10] Adi Shamir, "*How to share a secret*", Communications of the ACM, Vol. 22 No.11, pp.612-613, Nov. 1979
- [11] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "*Proactive secret sharing*," Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology, pp. 339 - 352, 1995.
- [12] P. Feldman, "*A Practical Scheme for Non- Interactive Verifiable Secret Sharing*", Proc. of the 28th FOCS, IEEE, pp. 427-437, 1987.
- [13] M. Stadler, "*Publicly verifiable secret sharing*", In Advances in Cryptology, Proc. of Eurocrypt '96, pp 190-199, Springer-Verlag, 1996.
- [14] B. Schoenmakers, "*A simple publicly verifiable secret sharing scheme and its application to electronic voting*," In *Advances in Cryptology-CRYPTO'99*, Vol. 1666 of *Lecture Notes in Computer Science*, Berlin, pp. 148-164, Springer-Verlag, 1999.

- [15] Yongguang Zhang and Wenke Lee. "Intrusion detection in wireless ad-hoc networks". In Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000), pp. 275--283, 2000.
- [16] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. "Mitigating routing misbehavior in mobile ad hoc networks". Mobile Computing and Networking, pp. 255-265, 2000.
- [17] Y. Zhang and V. Paxson, "Detecting backdoors," 9th USENIX Security Symposium, pp.12-19, 2000.
- [18] A. Fox and S. D. Gribble. "Security on the Move: Indirect Authentication using Kerberos". In MOBICOM, pp. 155--164, 1996.
- [19] A. Santis, Y. Desmedt, Y. Frankel and M. Yung. "How to share a function securely," STOC'94, pp. 522-533,1994.
- [20] R. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," CACM, Vol.21, No.2, pp.120--126, February 1978
- [21] Y. Frankel, P. Gemmell, P. Mackenzie and M. Yung, "Proactive RSA," CRYPTO'97, pp.440-454, 1997.
- [22] T. Wu, M. Malkin, and D. Boneh. "Building intrusion-tolerant applications". In Proceedings of the 8th USENIX Security Symposium (SECURITY-99), Berkeley, CA, pp.79-92, , August 23--26 ,1999.
- [23] H. Lin and L. Harn, "Authentication protocols for personal communication systems," SIGCOMM, pp. 256-261, 1995.
- [24] N. Alon, Z. Galil and M. Yung, "Efficient dynamic-resharing verifiable secret sharing against mobile adversary," ESA'95, pp-523-537, 1995.
- [25] Y. Frankel, P. Gemmel, P. MacKenzie, M. Yung, "Optimal-resilience proactive public-key cryptosystems," FOCS '97, pp.384, 1997.
- [26] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Robust and efficient sharing of RSA functions," Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, pp. 157 - 172, 1996.
- [27] R. Canetti, S. Halevi, and A. Herzberg, "Maintaining authenticated communication in the presence of break-ins," Journal of Cryptology, pp. 61-105, 2000

- [28] N. Alon, Z. Galil and M. Yung, "Dynamic-resharing verifiable secret sharing," ESA'95, pp.523 – 537, 1995.
- [29] W. Diffie and M. Hellman, "New Directions in Cryptography", IEEE Trans. Information Theory, IT-22(6), pp.644-654, November 1976
- [30] Y. Desmedt and Y. Frankel. "Shared generation of authenticators and signatures (Extended Abstract)," CRYPTO, pp. 457-469, 1991.
- [31] Y. Frankel and Y. G. Desmedt. "Parallel reliable threshold multi-signature," Technical Report TR-92-04-02, Dept. of EECS, University of Wisconsin-Milwaukee, 1992.
- [32] Y. Frankel, P. MacKenzie and M. Yung, "Robust efficient distributed RSA key generation," ACM Symposium on the Theory of Computing, pp. 663 - 672 , 1998.
- [33] M. Malkin, T. Wu and D. Boneh, "Experimenting with shared generation of RSA keys," Internet Society's Symposium on Network and Distributed System Security (SNDSS), pp. 43-56, 1999.
- [34] L. Zhou and Z. J. Haas. "Securing ad hoc networks," IEEE Networks, Vol. 13, pp. 24-30, 1999.
- [35] L. Gong, "Increasing availability and security of an authentication service," IEEE Journal on Selected Areas in Communications, Vol.11, No.5, pp.657-662, June, 1993,
- [36] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. "Proactive secret sharing or: how to cope with perpetual leakage," Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology, pp. 339-352, November 1995.
- [37] R. Perlman, "An overview of PKI trust models", IEEE Network, pp. 38-43, vol.13, No.6, Nov-Dec,1999.
- [38] A. Abdul-Rahman and S. Hailes. *A Distributed Trust Model*. In Proceedings of the 1997 New Security Paradigms Workshop, pp.48-60, ACM, 1997.
- [39] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," IEEE Journal on Selected Areas in Communications, Vol. 15, No. 7, pp. 1265-1275, Sep. 1997.
- [40] P. Sinha, R. Sivakumar and V. Bharghavan, "CEDAR: a core-extraction distributed ad hoc routing algorithms," INFOCOM'99, pp.135-141, 1999.



- [41] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7-14, 1999.
- [42] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. H. Katz. "A comparison of mechanisms for improving TCP performance over wireless links." IEEE/ACM Transactions on Networking, pp. 756-769, December 1997.
- [43] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: a reliable transport protocol for wireless wide-area networks", Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom'99), Seattle, Washington, USA, pp. 231-241, August 15-19, 1999
- [44] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek and M. Degermark, "Routing protocols for mobile ad-hoc networks - a comparative performance analysis," ACM MOBICOM'99, pp. 153, 1999.
- [45] S. Mascolo, C. Casetti, M. Gerla, S. S. Lee, and M. Sanadidi, "TCP Westwood: congestion control with faster recovery," UCLA CSD Technical Report 200017, 2000.
- [46] IEEE Standard 802.11, Wireless LAN media access control (MAC) and physical layer (PHY) specifications, First edition, 1999-08-20
- [47] "PKI Practices and Policy Framework", ANSI X9.79, American National Standards Institute, 2000.
- [48] Perkins, C.E., Belding-Royer, E.M., Das, S.R., "Ad Hoc On Demand Distance Vector (AODV) Routing", IETF Internet Draft, <http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-12.txt>