Abstract

Mobile ad hoc networks are a new paradigm of wireless communication for mobile hosts. Hosts are always represented as different nodes in the mobile ad hoc networks. There are a lot of differences between mobile ad hoc networks and traditional networks. We have studied the characteristics of ad hoc networks and pointed out the importance of its security. The main challenge in the design of mobile ad hoc networks is their vulnerability to security attacks. Like many distributed systems, security in ad hoc networks widely relies on the use of key management mechanisms. Specific key management systems have to be developed to suit the characteristics of mobile ad hoc networks because traditional key management systems are not appropriate for them. In this presentation, we propose an authentication services that combined with the concept of trust level. In our trust model, we assumed that each node has a trust value to its neighbouring nodes. This value is used on certificate renewal and it can be combined on a trust path. We also adopted the fully distributed certificate authority approach, which means the capabilities of the CA are distributed to all nodes in the ad hoc network. This work aims at providing a scalable, distributed authentication services in ad hoc networks.

Table of Contents

1. Introduction

Mobile ad hoc networks are a new paradigm of wireless communication for mobile hosts. Hosts are always represented as different nodes in the mobile ad hoc networks. There are a number of differences between mobile ad hoc networks and traditional networks. Ad hoc networks do not rely on any fixed infrastructure. It relies on each other to keep the network connected. Also, the topology of ad hoc networks is dynamically changing and its communication is based on wireless links. Due to the above characteristics, the main challenge in the design of mobile ad hoc networks is their vulnerability to security attacks. The security issues need to be addressed to give any successful applications [1].

In this report, it points out the importance for the security in mobile ad hoc networks. Securing mobile ad hoc networks is particularly difficult with its characteristics. The problem is so broad that there is no way to devise a general solution. It is also clear that different applications will have different security requirements. As in any distributed system, security in ad hoc networks is based on the use of key management system. Specific key management systems have to be development to suit the characteristic of mobile ad hoc networks [2]. The traditional key management systems are not appropriate for ad hoc networks. In this paper, we proposed a new key management scheme and the details would be presented in the following sections. This work aims at providing a scalable, distributed authentication services in ad hoc networks.

This report firstly presents the background knowledge on key management, ad hoc networks and its security issues. Then, it discusses the related work on the current key management systems developed for ad hoc networks so far. After that, it focuses on the key management scheme that we proposed and gives details presentation and analysis on the new scheme. Finally, future work and conclusion will be stated.

2. Background

2.1. Ad Hoc Networks

2.1.1. Definition

Mobile ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration. Due to the limited transmission range of wireless network interfaces, multiple ”hops” may be needed for one node to exchange data with another across the network [3].

2.1.2 Characteristics

There are a number of characteristics in mobile ad-hoc networks. One of them is that there are dynamic topologies. Nodes are free to move arbitrarily. Thus, the network topology is typically multi-hop, so may change randomly and rapidly at unpredictable times. Another characteristic is bandwidth constrain. Wireless links will continue to have significantly lower capacity than their hardwired counterparts. Also, there is energy-constrained in the networks. Some or all of the nodes in a mobile ad-hoc network may rely on batteries or other exhaustible means for their energy. Finally, there is limited physical security. Mobile wireless networks are generally more prone to physical security threats than are fixed-cable nets. The increased possibility of eavesdropping, spoofing, and denial-of-service attacks should be carefully considered [4].

2.1.3 Applications

Examples of potential practical use of mobile ad-hoc networks may be a group of people with laptop computers at a conference that may wish to exchange files and data without mediation of any additional infrastructure in-between. It may be used in home environment for communication between smart household appliances. Ad-hoc networks are suitable to be used in areas where earthquake or other natural disasters have destroyed communication infrastructures. It perfectly satisfies military needs like battlefield survivability, operation without pre-placed infrastructure and connectivity beyond the line of

sight. For monitoring and measuring purposes a large number of small computing devices could be spread over a hostile to form a self-sustained ad-hoc network. Mobile ad-hoc networks have significant advantages above traditional communication networks. For example, use of ad-hoc networks could increase mobility and flexibility, as ad-hoc networks can be brought up and torn down in very short time. Ad-hoc networks could be more economical in some cases as they eliminate fixed infrastructure costs and reduce power consumption at mobile nodes. They are more robust than conventional wireless networks because of their non-hierarchical distributed control and management mechanisms. Also, radio emission levels could be kept at low level because of short communication links (node-to-node instead of node to a central base station). This increases spectrum reuse possibility or possibility of using unlicensed bands. Moreover, communication beyond Line Of Sight (LOS) is possible at high frequencies because of multi-hop support in ad-hoc networks. Despite the mentioned advantages and potential application possibilities, ad hoc networks are yet far from being deployed on large-scale commercial basis. Some fundamental ad-hoc networking problems remain unsolved or need optimized solutions. Although various routing protocols are suggested and tested for mobile ad-hoc networks, performance metrics like throughput, delay and protocol overhead in relation to successfully transmitted data need better optimization. This optimization would probably also depend on application type and desire to maximize the throughput or minimize the delay. One single protocol will probably not be able to work efficiently across entire range of design parameters and operating conditions. An additional complexity factor in ad-hoc network design is that differ rent layers of the system are highly interdependent.

2.1.4 Standards

2.1.4.1 IEEE 802.11

IEEE 802.11 is a digital wireless data transmission standard in the 2.4 GHz ISM band aimed at providing a wireless LAN between portable computers and between portable computers and a fixed network infrastructure. This standard defines a physical layer and a MAC layer. The most popular technology is the direct sequence spread spectrum

and can offer a bit rate of up to 11 Mbps in the 2.4 GHz band, and in the future, up to 54Mbps in the 5GHz band. The basic access method in the IEEE 802.11 MAC protocol is the Distributed Coordination Function which is a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC protocol. However, the 802.11 standard cannot do multi-hop networking as it is. The development of a number of protocols is required. The maximum data rate of IEEE 802.11 is 11Mbps. Its range is 100 meters [5].

2.1.4.2 Bluetooth

Bluetooth is a digital wireless data transmission standard operating in the 2.4 GHz Industrial, Scientific, and Medicine (ISM) band aimed at providing a short range wireless link between laptops, cellular phones and other devices. In this band are defines 79 different Radio Frequency (RF) channels that are spaced of 1MHz. The main aim of the Bluetooth Specification is to guarantee the interoperability between different applications that may run over different protocol stacks. However, in order to implement a wireless multi-hop network over Bluetooth, either or both a packet switch layer and a circuit switch layer need to be defines on top of the Bluetooth data link layer protocol. The maximum data rate of Bluetooth is 1Mbps. Its range is 10 meters. Bluetooth has lower power consumption than IEEE 802.11. Also, Bluetooth support both voice and data packet types while IEEE 802.11 just support data packet type [5].

2.1.5 Routing Protocols

There are a number of routing protocols have been developed for mobile ad Hoc networks. They can be divided into two categories, which the table-driven protocols and the source-initiated on-demand protocols. DSDV belongs to the table-driven protocols. The most popular protocols nowadays are the AODV and DSR protocols. Both of them belong to the source-initiated on-demand protocols. We will briefly describe DSDV, AODV and DSR protocols in the following sections.

2.1.5.1 DSDV

DSDV stands for Destination-Sequenced Distance-Vector Routing. It is a table-driven algorithm based on the classical Bellman-Ford routing mechanism. Table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. These protocols require each node to maintain one or more tables to store routing information, and they respond to changes in network topology by propagating updates throughout the network in order to maintain a consistent network view [6].

2.1.5.2 AODV

AODV stands for Ad Hoc On-Demand Distance Vector Routing. It builds on the DSDV algorithm. It is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes on a demand basis, as opposed to maintaining a complete list of routes as in DSDV algorithm. AODV is classified as a pure on-demand route acquisition system, since nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges. The following figure (Fig2.1a) shows how the AODV route request and route reply message flow [6].

2.1.5.3 DSR

DSR stands for Dynamic Source Routing. It is an on-demand routing protocol that is based on the concept of source routing Mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learned. The protocol consists of two major phases, which are the route discovery and route maintenance. The route discovery was initiates by broadcasting a route request packet if a node does not have a route to the destination. Route maintenance is accomplished through the use of route error packets and acknowledgements. Route error packets are generated at a node when the data link layer encounters a fatal transmission problem. The

flowing figure (Fig 2.1b) shows how the DSR route request and route reply message flow [6].

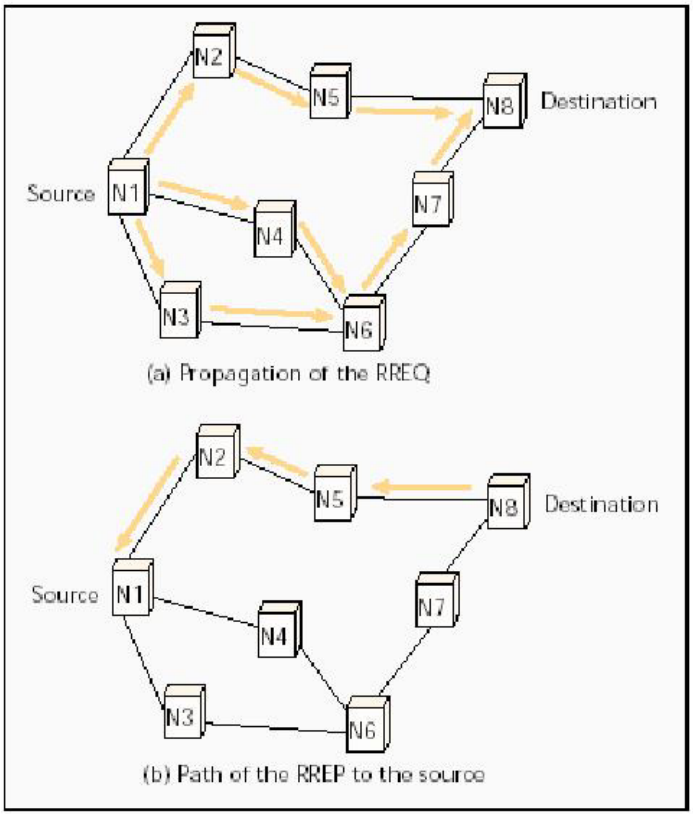

Fig. 2.1a AODV routing protocol

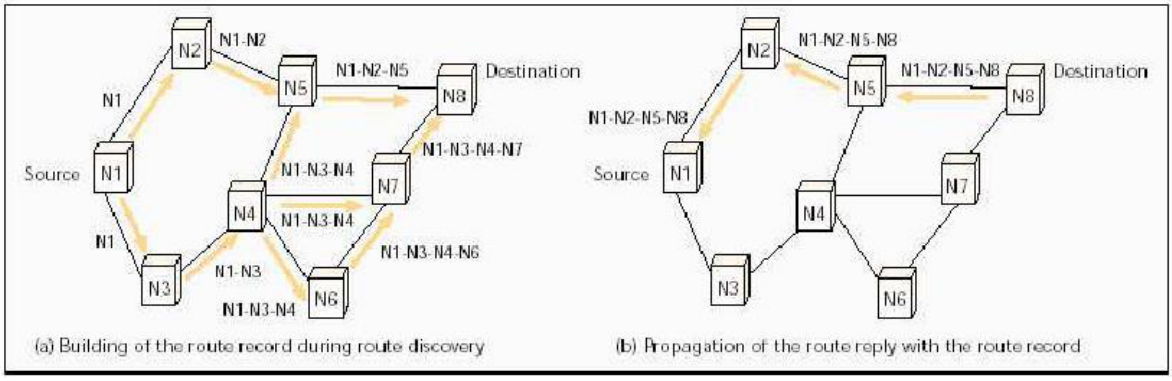

Fig 2.1b DSR routing protocol

2.2 Security Issues

2.2.1 Vulnerabilities

Due the characteristics of mobile ad hoc networks that we described in the previous section, there are a number of vulnerabilities of the networks. One characteristic is that mobile ad hoc networks have open medium, and lack of clear line of defense. The use of wireless links renders a wireless ad-hoc network susceptible to attacks ranging from passive eavesdropping to active impersonating, message replay, and message distortion. Active attacks might allow the adversary to delete messages, to inject erroneous messages, to modify messages, to impersonate a node, thus isolating availability, integrity, authentication, and non-repudiation. All these mean that a wireless ad-hoc network will not have a clear line of defense, and every node must be prepared for encounters with an adversary directly or indirectly.

Another characteristic is that there is dynamic changing topology. Mobile nodes are autonomous units that are capable of roaming independently. Nodes roaming in a hostile environment with relatively poor physical protection have non-negligible probability of being compromised. Therefore, not just external attacks should be considered, but attacks launched inside the network by compromise nodes should also be dealt with. It means that nodes with inadequate physical protection are receptive to being captured, compromised, and hijacked. It is easy to attach and hard to detect, so any node in a wireless ad-hoc network must be prepared to operate in a mode that trusts no peer.

Moreover, mobile ad hoc network has decentralized management. There is lack of centralized monitoring and management point. Decision-making in ad hoc networks is usually decentralized and many ad-hoc network algorithms rely on the cooperative participation of all nodes. Ad hoc network are supposed to operate independently of any fixed infrastructure. This makes the classical security solutions based on certification authorities and on-line servers inapplicable [7].

2.2.2 Motivation for the Attacks

From the above description, it is clear to notice that mobile ad hoc networks are easy to be attacked. However, it may still be interesting to know what is the motivation for attacking the mobile ad hoc networks. Some reason is that is it possible to gain various advantages by malicious behavior. For example, a node can get better service than cooperating nodes, gain monetary benefits by exploiting incentive measures or trading confidential information, save power by selfish behavior, extract data to get confidential information, and so on [8].

2.2.3 Types of Attacks

There are many different types of attacks can be occurred in mobile ad hoc networks. One of them is the passive denial-of-service attack. Under this kind of attacks, the misbehaving providers simply do not perform the requested function. For example, it may not participate to the Route Discovery phase of the protocol. Another type of attack is the active denial-of-service attacks. Under this kind of attacks, the malicious node prevent other providers from serving a request by communicating bogus information on reputation ratings for legitimate nodes, by performing traffic subversion or by using the security mechanism itself causing explicit Denial of Service. There are many other kinds of attacks. Most common attacks are those against routing and forwarding, such as the no forwarding or incorrect forwarding attacks, setting incorrect metrics on route for priority and remaining time in the cache, frequent route updates, and so on.

2.3 Cryptography

2.3.1 Cryptographic goals

The fundamental goal of cryptography is to address the confidentiality, data integrity, authentication, and non-repudiation in information security. Confidentiality is a services used to keep the content of information from all but those authorized to have it. Data integrity is a service, which addresses the unauthorized alteration of data. Authentication is a service related to identification. Non-repudiation is a service, which prevents an entity from denying previous comments or actions. These services can be used to prevent and detect cheating and other malicious activities [9]. In the following subsections, we will present some popular cryptographic techniques, like symmetric-key encryption, asymmetric-key encryption, digital signatures, and digital certificates.

2.3.2 Symmetric-key encryption

Symmetric key encryption involves using a single key to encrypt and decrypt data. A plain text message can be encrypted using a shared key to generate the cipher text. The plain text message can be received by decryption the cipher text with the same key. It should be noted that the key for encryption and decryption are the same in symmetric key encryption. Generally speaking, symmetric key encryption is fast and secure. However, the shared key must be distributed over a secure communication channel. The problem is that the physical medium you're sending the packets across is insecure. If it were secure, there would be no reason to encrypt the message in the first place. Anyone who might be monitoring the network could steal the encrypted packets and the key necessary for decrypting them.

2.3.3 Asymmetric-key encryption

Asymmetric-key encryption also called as public key encryption. Unlike asymmetric encryption schemes that involved parties share a common encryption or decryption key, public key encryption depends on tow different but mathematically related keys. The two different keys are a public key that's sent along with the message and a private key that is always in the possession

of the recipient. The private key is based on a derivative of the public key and only the two keys working together can decrypt the packets. The public key encryption is more secure because it only requires an authenticated channel as opposed to a secure channel that is required for the distribution of symmetric encryption keys. The down side of public key encryption is that it tends to be very slow and resource intensive. This makes it difficult to send large amounts of data using public key encryption. It is typically only used to encrypt small amount of data, like digital signatures.

2.3.4 Digital Signatures

A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and possibly to ensure that the original content of the message or document that has been sent is unchanged. Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped. The purpose of a digital signature is to provide a means for an entity to bind its identity to a piece of information held by the entity into a tag called a signature.

2.3.5 Digital Certificates

Digital certificate is an attachment to an electronic message used for security purposes. The most common use of a digital certificate is to verify that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply. An individual wishing to send an encrypted message applies for a digital certificate from a Certificate Authority (CA). The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. The CA makes its own public key readily available through print publicity or perhaps on the Internet. The recipient of an encrypted message uses the CA's public key to decode the digital certificate attached to the message, verifies it as issued by the CA and then obtains the sender's public key and identification information held within the certificate. With this information, the recipient can send an encrypted reply. The most widely used standard for digital certificates is X.509.

2.3.6 Certificate Authority

As mentioned in the previous sub-section, a certificate authority is a trusted third-party organization or company that issues digital certificates used to create digital signatures and public-private key pairs. The role of the CA in this process is to guarantee that the individual granted the unique certificate is, in fact, who he or she claims to be. Usually, this means that the CA has an arrangement with a financial institution, such as a credit card company, which provides it with information to confirm an individual's claimed identity. CAs are a critical component in data security and electronic commerce because they guarantee that the two parties exchanging information are really who they claim to be. Even though the public-key encryption looks ideal, it is possible for an adversary to defeat the system without breaking the encryption system. For example, an adversary can impersonates a communication by sending an entity an incorrect public key. It can then intercepts encrypted messages, decrypts with its private and re-encrypt the message with the correct public key of the receiver, and send it. This shows that authenticate public keys is necessary even in public-key encryption system. A public-key certificate consists of a data part and a signature part. The data part consists of the name of an entity, the public key corresponding to that entity, validity period, etc. The signature part consists of the signature of a trusted third party over the data part. A trust third party must take appropriate measures to verify the identity of A and ensure the public key to be certificated actually belongs to A in order to create a public-key certificate for A. In order for an entity B to verify the authenticity of the public key of A, B must have an authentic copy of the public signature verification function of the trust third part. In this way, entity can gain trust in the authenticity of another party's public key by acquiring and verifying the certificate [9].

3. Related Work

Traditional network authentication solutions rely on physically present, trust third-party servers, or called certificate authorities. Popular network authentication architectures include X.509 standard [10] and Kerberos [11]. There is some model on hierarchical CAs and CA delegations [12] have been proposed, but it does not address issues like service availability and robustness. However, ad hoc network is infrastructureless, there is no centralised server for key managements. There is also SPKI is a more flexible and less hierarchical security infrastructure solution [13]. However, it is devised primarily for Internet, and does not meet the requirements of mobile ad hoc network.

Pretty Good Privacy (PGP) [14, 15] is proposed following a web-of-trust authentication model, but it is unable to scale beyond a relatively small community of trust individuals. Also, the members may be unable to reach consensus on who is trusted and who is not, since independent "communities of trust webs" may be formed as a by-product. Another active research area is security function sharing [16, 17, 18, 19], a popular method is using threshold secret sharing [20]. The basic idea is distributing the functionality of the centralized CA server among a fixed group of servers. Proactive secret sharing is proposed to improve robustness by updating the secret keys periodically [21, 22, 23].

The paper written by Zhou and Hass [24] proposed the partially distributed certificate authority that makes use of a (k,n) threshold scheme to distribute the services of the certificate authority to a set of specialised server nodes. Each of these specialized server node can generate a partial certificate using their share of certificate signing key. A valid certificate can be obtained only be combining k such partial certificates. This approach basically assumes there is rich network connectivity among this small group of server nodes. Also, the server nodes better to have a multicast address because a client node needs to locate any k of the n server nodes for the certificate renewal. It may not be true that ad hoc network support multicast traffic, then the client node needs to broadcast its request and will generate a large amount of network traffic.

Similar to the partially distributed CA, the fully distributed certificate authority was proposed by Luo and Lu [25, 26, 27]. The fully distributed certificate authority approach extends the idea of the partially distributed approach by distributing the certificate services to every nodes and a threshold number (k, n) of neighbouring nodes can collaboratively act as a server to provide certification services for other nodes. It minimizes the effort and complexity for mobile nodes to locate and contact the service providers in a dynamic multi-hop wireless network. However, this approach assumes that there are k neighbours of every node, which may not be always true. Our scheme is inspired by these proposals, but extends the 1-hop neighbouring nodes taking part in certificate renewal to nodes that farther away, such as 2-hop or even 3-hop neighbours. However, the monitoring schemes on ad hoc networks usually can just detect the misbehaviour of their 1-hop neighbours, so we make use of the trust level concept for judging a node trustable or not by calculating the values on the trust chain. Therefore, nodes can decide other nodes, which are 2-hop or farther distance can be trusted or not. This makes it possible for k nodes, not direct neighbours to the requesting node to take part in the certificate renewal. It reduces the problem of not enough neighbouring nodes for certificate renewal.

Other solutions include the self-issued certificates proposed by Hubaux et al [28, 29]. It issues certificates by users themselves without the involvement of any certificate authority. In this algorithm, each user can build its own local certificate repositories for storing the certificates that they have issued. Any pair of users can find certificate chains to each other using only their certificate repositories. This solution does not require any form of infrastructure, but it lacks a certificate revocation mechanism. Also, it has problems if the number of certificates issued did not reach certain amount because it is possible that a trust chain does not exist.

Apart from public-key encryption system, distributed key management system based on symmetric encryption is also proposed [30]. This solution is suitable for nodes with low performance that are unable to perform public key encryption. The solution proposed by Balfanz et al [31] presents a mechanism for bootstrapping trust relationship in local ad hoc networks where the network nodes have no prior relationship with each other. However, it requires the nodes to be in short distance during the bootstrapping phase, so it is unsuitable for distributed ad hoc networks.

The paper from Asokan and Ginborg [32] describes a password authenticated group key agreement protocol that is an extension to the Hypercube protocol. It considers a collaborative network where a group of people wish to set up a secure wireless network during a meeting. It assume that a password can be chosen and shared within the room, then this password can be used in the password authenticated hypercube protocol for sharing a strong secret. However, this protocol assumes the participating nodes are arranged in hypercube and it is only suitable for very small ad hoc networks. Another paper [33] overviewed several existing Diffie-Hellman based protocols for group key establishment. It found that none of these protocols were found suitable for all types of ad-hoc networks mainly because they demand the network topology to follow a predestined structure.

Some related security solutions for mobile ad hoc networks also include system imprinting, tamper resistance, intrusion detection, mitigation of routing misbehaviour. System imprinting is done at node initialization to make a devices know who is its master. A paper presents the resurrecting duckling security policy model [34], which describes secure transient association of a device with multiple serialised owners. A number of papers proposed mechanisms on detecting the misbehaviours of nodes. A paper develops a viable intrusion detection system for wireless ad-hoc networks. It proposed that each node is responsible for detecting signs of intrusion locally an independently, but neighbouring nodes can collaboratively investigate in a broader range [7]. Another paper presented that trust relationships and routing decisions are made based on experienced, observed, or reported routing and forlwing behaviour of other nodes nodes. It proposed new routing protocol extensions to detect and isolate mishaving nodes, so make it unattractive to deny cooperation [8]. A similar paper also proposed to install watchdog and pathrater in the network to detect and mitigate routing misbehavior [35]. A paper proposed the components of CONFIDANT, assumed to be present in every node. CONFIDANT consists of the components, which are the monitor, the reputation system, the path manager, and the trust manager [36]. The self-organised feature of the solution is provided through fully localized design. The proposed security solution composes of four components. They are the neighbor verification, security enhanced routing protocol, neighbor monitoring, and intrusion reaction [37]. The papers from Peitro Michiardi and Rdfik Molva pointed out three possible roles that nodes can assume: the requestor, the provider and the peer role. It proposed a security

mechanism that solves the problems due to misbehaving nodes. It incorporates a reputation mechanism that provides an automatic method for the social mechanisms of reputation [38, 39].

Recently, there are a number of secure routing protocols proposed. Most of them are built on the existing routing protocols in mobile ad hoc networks, such as the DSDV, DSR and the AODV protocols. A paper proposed a protocol that can be applied to several existing routing protocols. It presented a route discovery protocol that mitigates the detrimental effects of malicious behaviour, as to provide correct connectivity information [40]. To deal with external attacks, standard schemes such as digital signatures to protect information authenticity and integrity have been considered. The use of a keyed one-way hash function with a windowed sequence number for data integrity in point-to-point communication and the use of digital signature to protect messages sent to multiple destinations was proposed [41]. A paper proposed a protocol called Ariadne. Ariadne was built on DSR and TESLA, and relies on efficient symmetric cryptography. It prevents attackers or compromised nodes from tampering with uncompromised routes consisting of uncompromised nodes, and also prevents a large number of types of Denialof-Service attacks [42]. Another paper proposed SEAD as a secure ad hoc network routing protocol based on the design of the Destination-Sequenced Distance-Vector routing protocol (DSDV). In order to support use with nodes of limited CPU processing capability, and to guard against Denial-of-Service (DoS) attacks in which an attacker attempts to cause other nodes to consume excess network bandwidth or processing time, SEAD uses efficient one-way hash functions and do not use asymmetric cryptographic operations in the protocol. This protocol can be used with any suitable authentication and key distribution schemes, but it is not straightforward [43]. One more paper looks at AODV in detail and develops a security mechanism to protect its routing information. In this paper, it assumes that there is a key management sub-system that makes it possible for each ad hoc node to obtain public keys from the other nodes of the network. Further, each ad hoc node is capable of securely verifying the association between the identity of a given ad hoc node and the public key of that node [44]. A survey paper gives an overview of potential vulnerabilitys and requirement of ad-hoc network, aand the proposed prevention, detection and reaction mechanisms for cooperative routing and thwarting attacks [45].

4. Trust-level based Authentication Services

Mobile ad hoc network can be represented as a set of nodes in a diagram. A node represents a host in the network, and they can communicate with each other. We use public key encryption in our secure network because it can obtain non-repudiation, confidentiality, integrity and authentication. However, it is possible for an adversary to defeat the system by impersonating the unsecured channel when two entities are exchanging public keys, or an adversary can alter the public file containing public keys. To prevent the above attacks, the use of public key cryptography requires the authenticity of the public keys. Public key certification by trusted third party can be adopted in a secure network. Certificate authority (CA) is a trusted third party that responsible for establishing and vouching for authenticity of public keys. A certificate binds the identity of an identity of an entity to its public key. The following figure (Fig.4a) shows the flow chart including the joining, initialising and certification of a node in our network.

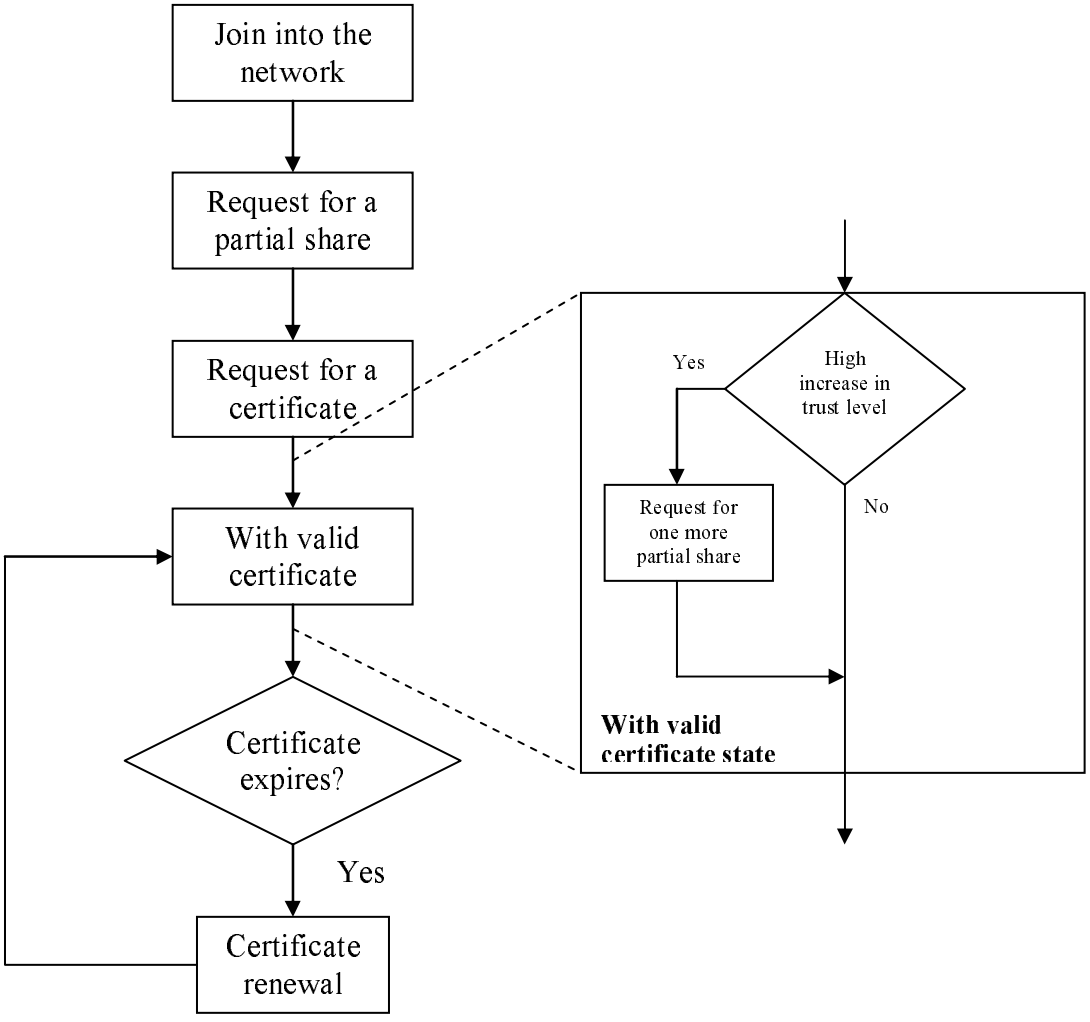

Fig.4a The flow chart of a node in our network

A node joins into the network has to request for a partial share and a certificate. The partial share will be used for initialising or certificating other nodes in the future. It is because our system adopted the fully distributed authentication approach, so each of the nodes can take part on the initialisation and certification using their partial share. Also, in order to communication with other nodes, a node must hold a valid certificate. Each certificate will have a validity period. If the certificate expires, a node has to request for certificate renewal. It should be noted that a node can holds more than one partial share, so it can has more power or importance in the network. If node finds its trust value from other nodes has been greatly increased, it can attempt to request for one more partial share.

In our system, a node can request for initialisation and certification by broadcasting its requesting its request message, like the following figure (Fig.4b):

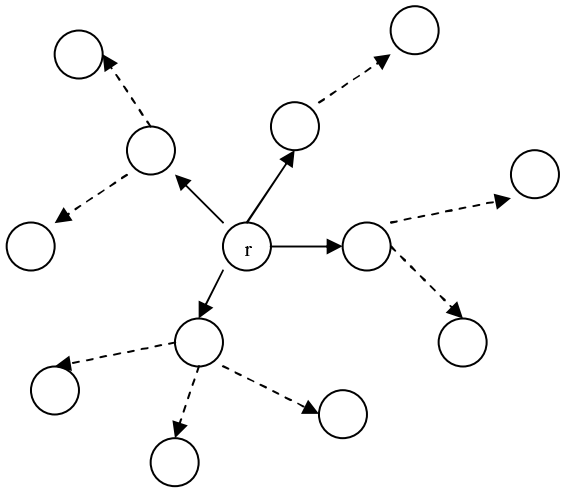

Fig.4b A node broadcasting request message

We have combined the trust level concept into our authentication services. With our trust level model, each of the nodes will give a trust value to its neighbouring nodes. This trust value to the requesting node will be used to decide whether replying or not after receiving a certificate renewal request. The trust levels given from neighbouring nodes are like the following figure (Fig.4c).

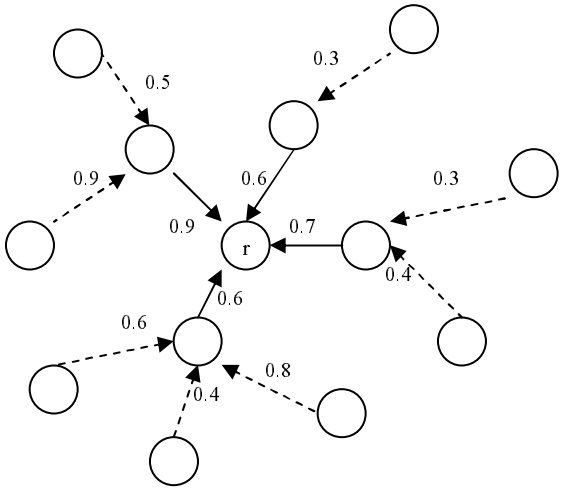

Fig.4c Trust values given by neighbouring nodes

In the following figure (Fig.4d), it shows the nodes that decided to reply to the request of certificate renewal. They are the neighbouring nodes that with high trust value to requesting node. For the non-neighbouring nodes, they are on the path of a trust chain. Trust chain means a path that all nodes along the path have a view of high trust level to its next hop.

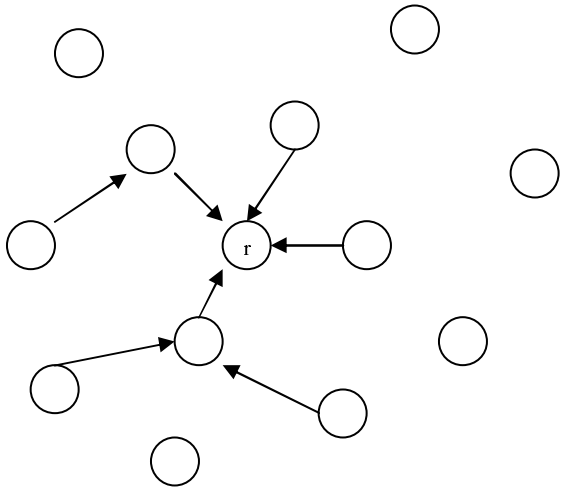

Fig.4d Nodes replying with partial certificate

A node requesting for a partial share or certificate needs to receive at least k number of reply in order to make the request successful. The following figure (Fig.4e) shows the protocol for certificate renewal as an example. Requesting for a partial share is using the same protocol as well.

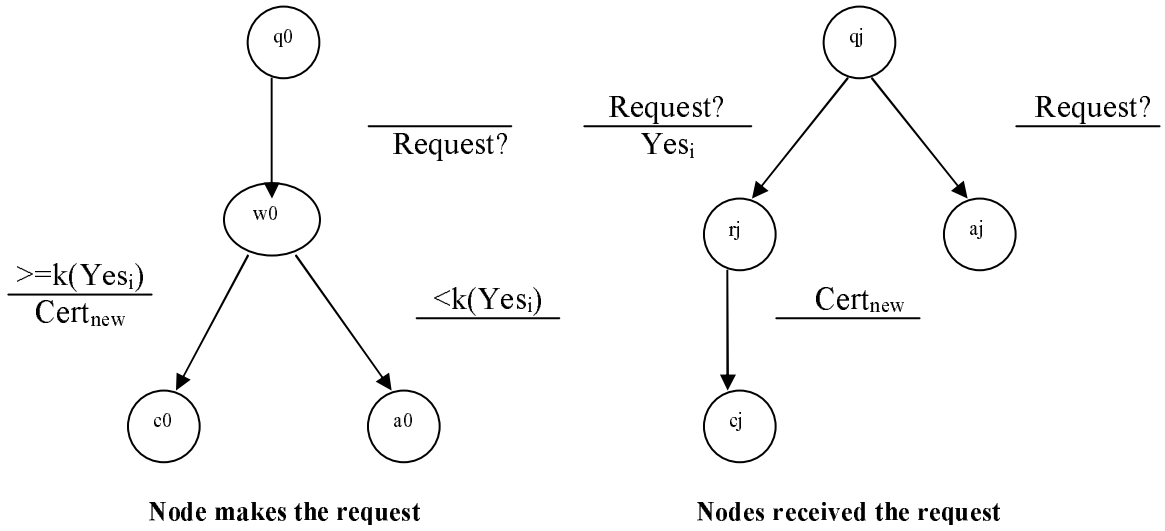

Fig.4e Protocols on certificate renewal

The following lists give the meaning of different states.

Node makes the request	
q0:	making a request
w0:	waiting for the replies
c0:	received k or more replies, request successes
a0:	received less than k replies, request fails

Nodes received the request	
qj:	receive a request
rj:	requesting node is trustable, send reply
aj:	requesting node is not trustable, no reply is sent
cj:	receive the new certificate from the requesting node

4.1. Trust Model

A trust model can define how the nodes in the network trust each other. The trust relationship between the nodes affects our key management as well. We believe that defining a good trust model can help to formalize our authentication services in the network. It can make our services design more concrete

In our trust model, we assumed that each node v_j will have a trust value to its neighbouring nodes vi. The trust value from node v_j to node v_i represent the different levels of trust that node v_j towards node v_i according to v_j's observation on the behaviour of node v_i at that moment.

Although there is no one universal value system, standardization is important for interoperability. There is a number of trust models proposed in the past, such as the discrete levels of trust model has divided the trust value into 5 levels [Trust]. In our system, we adopted the trust model that defines the trust value to be a number between zero and one. Generally, we define a node that will be trusted and given one or more partial certificate only if its trust level is above 0.5.

4.2. System Design

We adopted the fully distributed certificate authority approach, which means the capabilities of the CA are distributed to all nodes in the ad hoc network. A node is trusted in the network if it holds a valid certificate. A valid certificate must be signed by any k nodes, which are typically among the node's one-hop neighbours [26]. This approach used the (k,n) threshold secret sharing scheme proposed by Adi Shamir [46]. Apart from the above approach, we also borrow some ideas from the public key management solution proposed by Hubaux et al [28]. In this approach, two users try to find a certificate chain when they wish to authenticate each other. In our system, the certificate chain is not formed by using the certificates stored n each user's local certificate repositories, but based on the trust levels of the nodes in the chain. A certificate chain can be formed if the trust values of the nodes on the chain are high, so two nodes that may not be 1-hop neighbours can still take part in the authentication to each other. Therefore, we extended the fully distributed certificate authority approach to not limited to a corporation between only one-hop neighbours, but the nodes with more hops from

the requesting nodes. Details of this algorithm will be presented in the later part of our report.

Also, we do not restrict a node holds only one polynomial share for signing certificate. Using the weighed threshold secreting share scheme proposed by Adi Shamir [46], a node can hold more than one share in an ad hoc network if it has high trust level. This represents its importance and power in the network according to its security level. Such a node can sign more than one partial certificate to a node if it finds the node with high trust levels. The number of partial certificates node v_j signed to node v_i is directly proportional to the trust node v_j towards node v_i.

4.3. Assumptions

In this work, we consider an ad hoc wireless network with dynamic number of nodes n. The number n can change dynamically as mobile nodes join, leave, or fail over time. Besides dynamic network topology, the network also with limited physical security, bandwidth and energy constrained on nodes. It does not provide any infrastructure support. We also make the following assumptions:

(1) Each node has a unique ID.
(2) Each node can discover its one-hop neighbours.
(3) Communication within one-hop neighbours is reliable.
(4) The mobility is characterized by maximum node moving speed.
(5) Each node maintains a trust value to its each neighbour.
(6) Each node can hold a limited number of polynomial shares.
(7) Each node signs out different number of partial certificates according to the trust level of the requesting node.
(8) Partial certificates can be signed and propagated to the requesting node through a few hop counts
(9) Trust values on a path can form trust chain. Nodes along the path can trust each other even they do not meet before.

5. Certificate Issuing / Renewal

5.1. Number of Polynomial Shares per Node

In our system, each node holds a number of polynomial shares for signing certificates to its neighbours. The number of polynomial shares holds by each node is different. We define c be the maximum number of partial keys that a node can hold. Each node has its unique ID, and this node ID will be used to generate the unique partial key IDs that the node holds. The following table (Table 5.1a) illustrates the relationship between the node id and share ids:

Node ID	Share IDs
1	1, 2, …, c
2	c+1, c+2, …, 2c
3	2c+1, 2c+2, …, 3c
…	…
k	(k-1)*c+1, (k-1)*c+2, …, k*c
…	…
n	(n-1)*c+1, (n-1)*c+2, …, n*c

Table 5.1a Partial keys of node

We want to inject more flexibility into our system, while still keeping the same security of the system. We allow some nodes with high trust level in the network to hold more than one shares, so that it can gives more contribution in certificate renewal.

In the (k,n) threshold secret sharing scheme proposed by Adi Shamir [47], any k our of n users can reconstruct the secrets. In the last part of his paper, it mentions that each user has his weight on secret depending. For instance, on his position in a company, different people may hold different number of secret shares. In a (3, n) weighted threshold scheme, the manger weights 2 and workers weight 1. Then, either 1 manger and 1 worker, or 3 workers can co-operate and reconstruct the secret.

Similar scenario in our trust based fully distributed certificate renewal services, a node with high trust level weights higher. It is able to hold more number of polynomial shares, so that it can sign more than 1 partial certificate to itself and its neighbours in certificate renewal. With our scheme, it may be possible that less than k neighbouring nodes in a coalition can already complete a certificate renewals. However, we require some nodes in the coalition must have very high trust level in order to maintain the security in the network.

In the following table (Table 5.1b), We presented some calculation on the number of nodes required to form a coalition capable to do certificate renewal. We defined k as the number of partial certificates required in a successful certificate renewal. Each polynomial share can generate one partial certificate. We show the how the number of polynomial shares a node holds affects the size of the coalition required in certificate renewal. From the table, we found that if each node can hold from 1 to K polynomial shares, then the possible size of coalition ranges from K/C to K. The size of K/C represents the case that all nodes in the coalition hold maximum number of partial keys (K); while the size of K represents the case that all nodes in the coalition hold only 1 polynomial share. These are the two extreme cases in our algorithm. From our analysis, if the number of partial keys a node holds is high, the coalition can be greatly decreased. Though the coalition size can increase the performance of certificate renewal, it may make the network less secure if the situation goes too extreme. Therefore, we do not recommend a node holding too many polynomial shares.

In a (k,n) threshold scheme:

k	No. of shares a node can hold	Min. no. of nodes in a coalition	Max. no. of nodes in a coalition
5	1	5	5
5	1-2	3	5
10	1	10	10
10	1-2	5	10
10	1-3	4	10
20	1	20	20

20	1-2	10	20
20	1-3	7	20
20	1-4	5	20
K	1-C	K/C	K

Table 5.1b No. of shares per node to coalition size

In this table (Table 5.1c), we gives more information by how the average number of polynomial shares a node holds affect the average coalition size in certificate renewal.

k	Average no. of shares a node can hold	Average no. of nodes in a coalition
5	1	5
5	1.2	4.17
5	1.5	3
10	1	10
10	1.2	8.33
10	1.5	6.67
10	2	5
20	1	20
20	1.2	16.7
20	1.5	13.33
20	2	10
20	3	6.67
K	A	K/A

Table 5.1c Average no. of shares per node to average coalition size

From the above analysis, we found that the number of polynomial shares holding by a node increases a little can already gives significant decreases in the number of nodes in coalition k. Therefore, in order to maintain the security of the network, we must have a stricter rule for giving more than one polynomial shares, and for

signing more than one certificate to the neighbouring nodes. It means that a node must get relatively high trust level in order to get more than 1 polynomial share. Also, a node requesting for certificate renewal must have high trust level in order to have a node holding more than 1 polynomial share signing more than 1 partial certificate to the it.

5.2. Number of Partial Certificate in Reply

Let x be the trust value vj to vi,

mj be the maximum number of partial certificates that j can sign,

nj be the number of partial certificates that vj sign to vi.

In our design, we will divide the trust value into different ranges. The value belongs to different range will lead to different number of partial certificate to be replied. It is shown in the following table (Table 5.2a).

Trust level (vj to vi)	No. of partial certificate vj to vi
x<1/2	0
1/2<= x < ½+1/4	1
½+1/4<= x < ½+1/4+1/8	2
…	…
½+1/4+…1/(2^(K-1))<=x<½+1/4+…1/(2^K)	K-1
½+1/4+…1/(2^K)<=x<=1	K

Table 5.2a Trust level to no. of partial certificates in reply

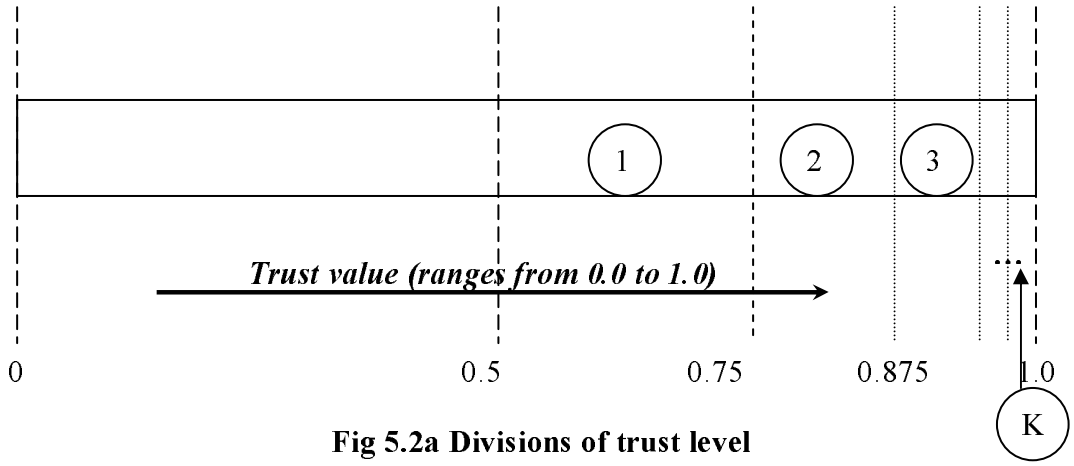

Fig 5.2a Divisions of trust level

We will divide the right most piece of partition into half each time adding a line (Fig 5.2a). For a node that only holds one polynomial share, it will just add one line and divide the trust values into two ranges (one is smaller than 0.5, one is larger than 0.5). For the trust value smaller than 0.5, the replying node will not give any partial certificate no matter how many polynomial shares it holds. For a node with k polynomial shares, we will add k lines and divide the trust value into (k+1) pieces.

Example 1

For example, a node nj now is holding 2 polynomial shares, so it can sign maximum number of partial certificates is 2.

Node vj my have different trust value to vi corresponding to the how much it trust vi. With different trust level vj to vi, vj gives vi different number of partial certificates. The corresponding trust level to number of partial certificates can be assigned are listed in the following table (Table 5.2b).

Trust level (vj to vi)	No. of partial certificate vj to vi
x<0.5	0
0.5<= x < 0.75	1
0.75<= x < =1.0	2

Table 5.2b Trust level to no. of partial certificates with maximum 2

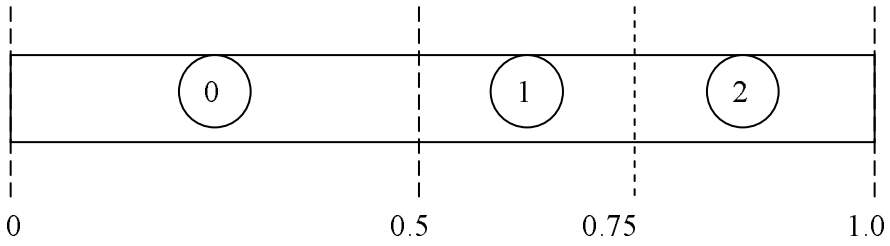

Fig 5.2b Divisions of trust level to 3 pieces

The above figure (Fig 5.2b) shows the divisions of the trust level. It is divided into 3 pieced, including the range of trust level corresponds to 0 partial certificate, 1 partial certificate, and 2 partial certificates. The number circled indicates the number of partial certificates to be returned with respect to different ranges that the trust value falls in.

Example 2

Another example, a node nj now is holding 3 polynomial shares, so it can sign maximum number of partial certificates is 3.

Node vj my have different trust value to vi corresponding to the how much it trust vi. With different trust level vj to vi, vj gives vi different number of partial certificates. The corresponding trust level to number of partial certificates can be assigned are listed in the following table (Table 5.2c).

Trust level (vj to vi)	No. of partial certificate vj to vi
x<0.5	0
0.5<= x < 0.75	1
0.75<= x < 0.875	2
0. 875<= x < =1.0	3

Table 5.2c Trust level to no. of partial certificates with maximum 3

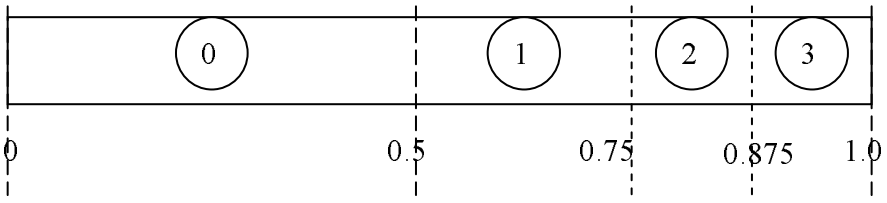

Fig 5.2c Divisions of trust level to 4 pieces

The above figure (Fig 5.2c) shows the divisions of the trust level. It is divided into 4 pieces, including the range of trust level corresponds to 0 partial certificate, 1 partial certificate, 2 partial certificate, and 3 partial certificates. The number circled indicates the number of partial certificates to be returned with respect to different ranges that the trust value falls in.

5.3. Trust Relationships of Nodes

In our system, a node requesting for certificate renewal needs the combination of k partial certificates from any other nodes. The most direct solution is to get enough partial certificates from the node's one-hop neighbours. However, this is not always possible because a node may not have enough neighbours. As we mentioned before, these k partial certificates may come nodes with hops away in our design. To make it possible, we build up a certificate chain based on the trust levels of the nodes.

A node vi broadcast its request message on certificate renewal, then vi's neighbours can further broadcast the messages to more nodes. This simple broadcast allow higher number of nodes to receive the request message, so more nodes can take part on the certificate renewal. We believed that the trust value from node that farther away to the requesting node could be calculated by integrating all related trust values on the path. This allows the nodes that receive the request message on certificate renewal can return partial certificate based on the commutated trust value. The partial certificates can be propagated to the requesting nodes hop by hop.

For example, neighbours of vi, including v1, v2, v3, and v4 receive the message (Fig.5.3a). These four neighbouring nodes may not be able to provide enough number of partial certificates, but they can broadcast the message further to their own neighbours. Nodes v5 and v6 can then receive vi's request message. In this example, nodes v1, v2, v3, v4 are 1-hop neighbours to vi, so each of them has a trust value to vi. This trust value represents the level that a neighbouring node trusts to vi. Each neighbouring node can use the algorithm proposed in section 5.2

to determine how many partial certificate to be returned to vi based on their view on vi's trust level.

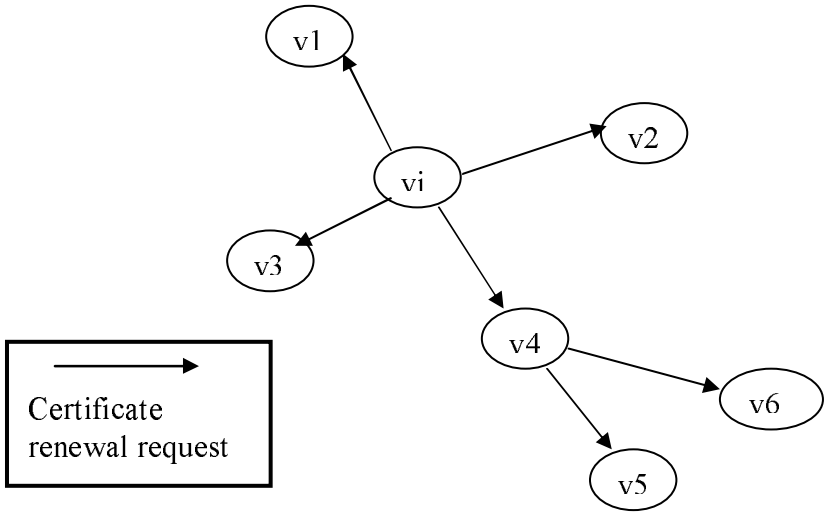

Fig 5.3a. Request for Certificate renewal from node vi

The solution looks simple for 1-hop neighbours to make this decision. However, for a node farther away, such as 2-hop neighbours, the matter becomes more complicated. It is because 2-hop neighbours do not have a record of vi, so they cannot give vi a trust value based on their observation. We proposed that the trust level between neighbouring nodes could form a trust chain. The node can obtain trust value of a non-neighbouring node by computing the trust values on the chain. For example, node v5 and v6 do not have a trust value to vi as they are not neighbours of vi, but they can calculate their trust value to vi using their trust value to node v4 and the trust value from v4 to vi.

For the calculation on the trust level, we referenced to some formulas [48]. It proposed that the value of the new relationship could be computed with the formula:

$$V1 \Theta V2 = 1 - (1-V2)^{V1}$$

We have analysed the above formula and found that it is suitable to be applied in our system. In our analysis, we consider a trust chain from node v1 to node v2, and node v2 to node vi (Fig 5.3b). The value V1 is the trust value given by node v1 to node v2, while the value V2 is the trust value given by node v2 to node vi. With our trust chain, we can calculate the trust from node v1 to node vi referencing to the trust values V1 and V2. The concept behind is that we think the trust can be propagated in a way that, node v1 trust v2 and node v1 trust vi can influence that node v1 can trust vi.

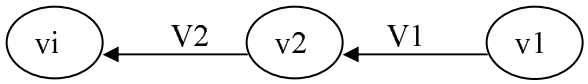

Fig 5.3b Trust chain example

V1\V2	0.3	0.6	0.9
0.3	0.1	0.24	0.49
0.6	0.19	0.42	0.75
0.9	0.27	0.56	0.87

Table 5.3a V1ΘV2

From the above table (Table 5.3a), we found that when the value of V1 is high, the resulting trust value from v1 to vi is closer to the trust value of v2 to vi. This is reasonable as V1 is high means that v2 is being trust by v1, so v1 trust more on the value V2 that it gives to vi; and vice versa.

By mathematical analysis, the formula V1ΘV2 will approach to V2 when V1 approaches to 1. Also, V1ΘV2 will be smaller when V1 becomes smaller.

Using the above formula, we can calculate the trust value from node v5 to node vi, and that from node v6 to node vi (Fig5.3c).

Trust value (v5 to vi) = $0.9\Theta 0.8 = 1 - (1-0.8)^{0.9} = 0.765$

Trust value (v6 to vi) = $0.5\Theta 0.8 = 1 - (1-0.8)^{0.5} = 0.553$

Since the trust values obtained by node v5 and v6 to node vi are above 0.5, it represents that both nodes believes vi is a good node. They can then reply node vi with their partial certificate.

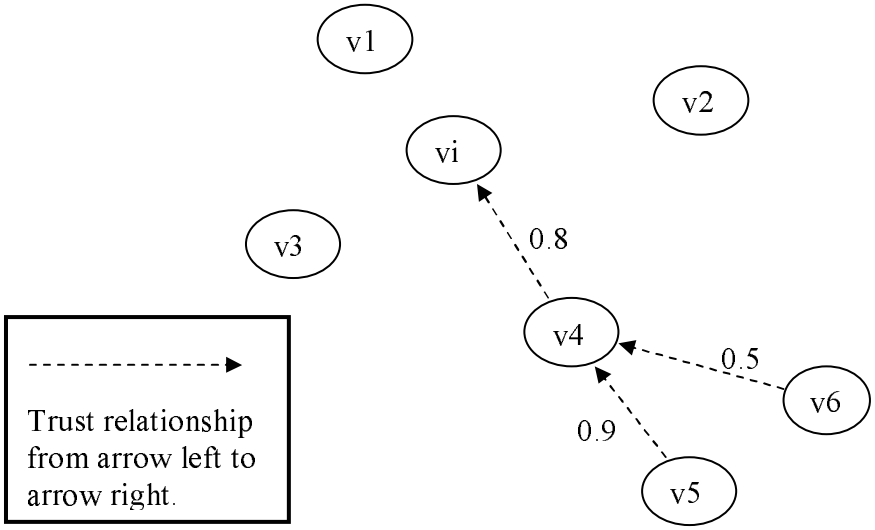

Fig 5.3c. Trust values of different nodes

Assume that both node v5 and v6 have 2 partial polynomial shares, so that each of them can sign two partial certificates at one time. The Trust value from v5 to vi is between 0.75 and 1, so node v5 will reply node vi with 2 partial certificates; while the trust value from node v6 to vi is between 0.5 and 0.75, so node v6 will reply node vi with 1 partial certificates (Fig. 5.3d). The partial certificates from node v5 and v6 can be propagated to node vi via node v4. With our algorithm, node vi can obtain more partial certificates if it has high trust level, even though it may not have enough number of one-hop neighbours. With the corporation of nodes with close distance, certificate renewal can be completed with higher flexibility and without lose of the security in the network.

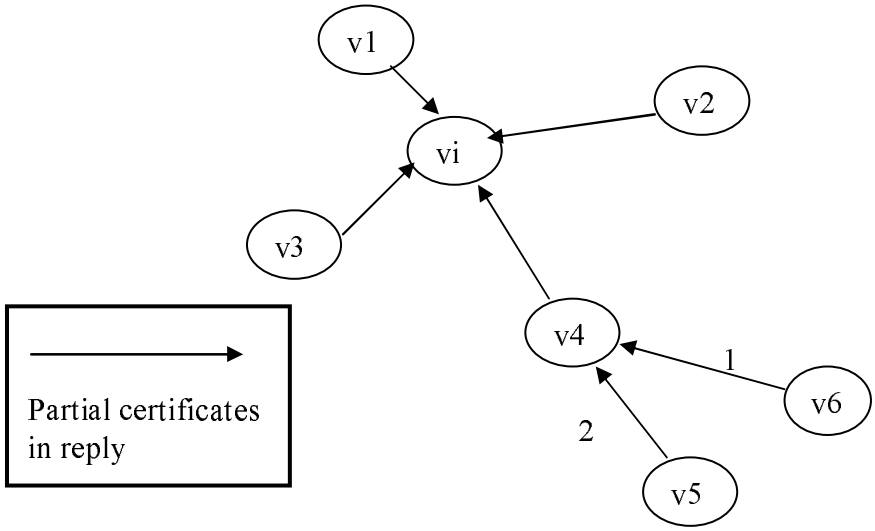

Fig5.3d. Number of partial keys in reply

5.4. Algorithm

We adopt the polynomial secret sharing to share the network secret key SK for certificate among the network. Each node v in the network holds c polynomial shares $Pa_i = f(a_i) \bmod N$, where $f(x) = SK + f_1x + \ldots + f_{k-1}x^{k-1}$ is the secret polynomial. We assumed that the certificate verification key PK and N are well known. We applied both the polynomial secret sharing and the dynamic coalescing techniques [26] in our system.

Let us go through the process of certificate issuing and renewals in the following part. When a node vi broadcasts its request for a new certificate among its neighbourhood. A neighbouring node vj that receives the request will return a number of its partial certificates according to the trust value it gives to vi.

It should be noted that the vj will use its shares start from the smallest ID. The node vi can collect a number of partial certificates from the replies. Without loss of generality, we name the shares that takes part in this certificate renewal as Pa_1, $Pa_2, \ldots, Pa_k$. One share can generate one partial certificate by directly applying its polynomial shares on certificates.

$$CERTaj = (cert)^{Paj} \bmod N$$

Upon receiving at least k such partial certificates, node vi picks k to form the coalition B. Suppose, vi chooses {$CERTa_1$, $CERTa_2$, … , $CERTa_k$}, where a1,a2, …, ak are the IDs of the corresponding polynomial shares.

$$CERT'a_j = (CERTa_j)^{Laj(0)} \mod N$$

where $La_j(0) = \prod_{r=1, r\neq j}^{k} \frac{a_r}{a_r - a_j} \mod N$

vi then multiples {$CERT'_{a1}$, $CERT'_{a2}$, … , $CERT'_{ak}$}together to generate the candidate certificate CERT':

$$CERT' = \prod_{j=1}^{k} CERT'_{aj} \mod N$$

Then node vi can employ the k-bounded coalition offsetting algorithm to recover its new certificate CERT. Details of the k-bounded coalition offsetting algorithm can be reference from the paper mentioned [25], it is also listed here:

Inputs: CERT': the candidate certificate
cert: statement of the certificate, to be signed
Output: CERT: certificate

1: $Z := (cert)^{-N} \mod N$
2: j := 0, Y := CERT'
3: while j < k do
4: Y := Y*Z mod N, j := j + 1
5: if ($cert = Y^{PK} \mod N$) then
6: break while
7: end if
8: end while
9: output Y = CERT

5.5 Communication protocol

Based on the certificate renewal algorithm we presented above, our communication protocol is simple as follow (Fig. 5.5a). The protocol is mainly divided into two parts.

(1) Request for certificate renewal

The request will first broadcast the request message on certificate renewal. The neighbours of the requesting nodes can help to further broadcast the requests. Generally 2-hop, or mostly 3-hop neighbours of the request can receive the request.

(2) Replies of partial certificates

Nodes received the request can than calculated the trust value to the requester if necessary, or they can use the value that they maintained in their own list. They can determine how many partial certificates the trust value worth, and reply the certificates to the requester. For the nodes that not decide to return any certificates have no need to send any reply messages. For the nodes will contribute their partial certificate can return more than one partial certificates in the reply.

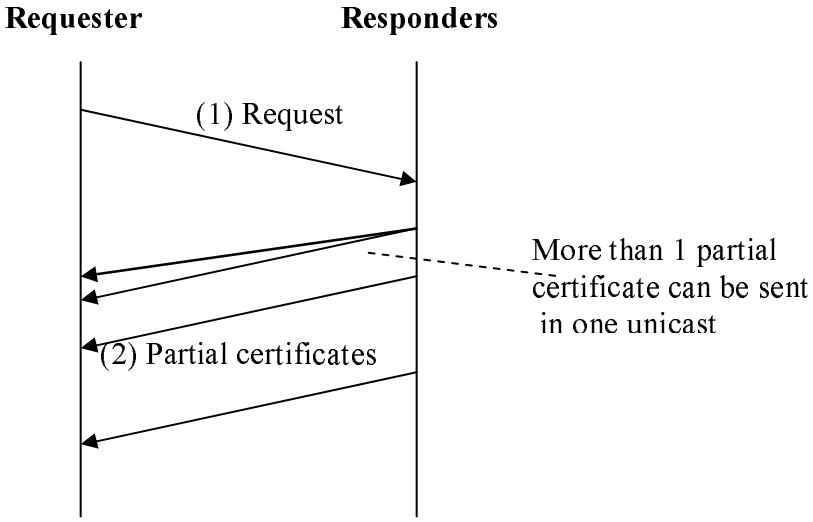

Fig. 5.5a Protocol for certificate renewal

6. More on Certificates

6.1 Validity Period of Certificates

In our system, we allow different validity period to be assigned to nodes based on their trust level. A node with high trust level receives a longer validity period for its new certificate, and vice versa. We present our general concept with the following table (Table 6.1a). It is an example showing that with the trust value below 0.5, a certificate will not be renewed. With the trust level between 0.5 and 0.75, we assume the validity period to be t. If the trust becomes even higher, then we can double or even triple the validity period. Just like we get validity period as 2t when the trust value is between 0.75 and 0.875, and validity period as 3t when the trust value is between 0.875 and 1.0 in this example.

Trust level	Validity Period
x<0.5	NIL
0.5<= x < 0.75	t
0.75<= x < 0.875	2t
0.875<= x < =1.0	3t

Table 6.1a Trust level to validity period in certificate

Since nodes only maintain trust level to their neighbours, they have no idea on the how much trust level its neighbours are giving to it. It will be great if all the nodes in the coalition k can agree on the same trust value before each certificate renewal, but it is not efficient. Therefore, we designed that there is a default validity period to a node, t. When vi broadcast the request for the renewal of its certificate, its neighbouring nodes sign our certificate with the default valid period. However, if it finds that the trust level of vi has raised to a level that worth for a longer validity period, it set the "increase validity period" field in the reply message of its partial certificate. If vi receives such an alerts more than k of its neighbours, then it can request for a longer validity period in its next certificate. For requesting a new certificate with longer validity period, it can set the "increase validity period" field in its request message. If vj receives such a request and finds vi is above the trust

level that validity period request, it serve vi. Otherwise, it does not reply that message, so other nodes can serve the request, or vi will make the request again after timeout without the "the "increase validity period" field set this time.

6.2. Time Allowance Period

Using the trust level, we can ensure the normal operation of the network for some moments that a node cannot complete certificate renewal on time. In the following figure (Fig. 6.2a), we assume that vi's certificate has expired, so it is in the time allowance period. The period Tp represents the validity period of the current certificate. The time allowance period represents the period from the expired of the last certificate to the effective moment of the new certificate.

Normally, vi will be isolated from the network after its certificate expired. However, if there is a trust relation present in the network diagram (Fig 6.2b), such ask node v4 trust vi as v4 has high trust level to vi. In the time allowance period, a neighbouring can still communication with vi as usual if it trust vi. Also, the node farther away like node v7 can build up a node chain if node v7 trust v4, and v4 trust vi. Then, v7 can still trust and communication with vi.

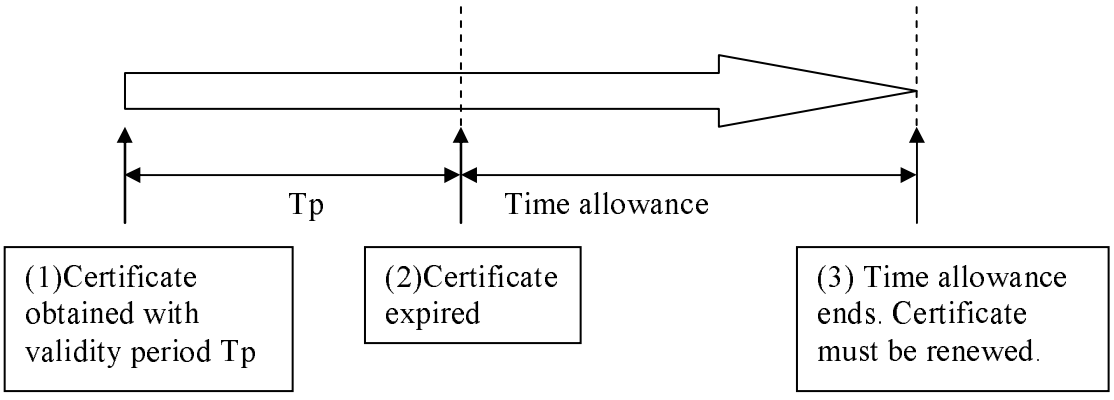

Fig 6.2a. Cycles on certificate renewal

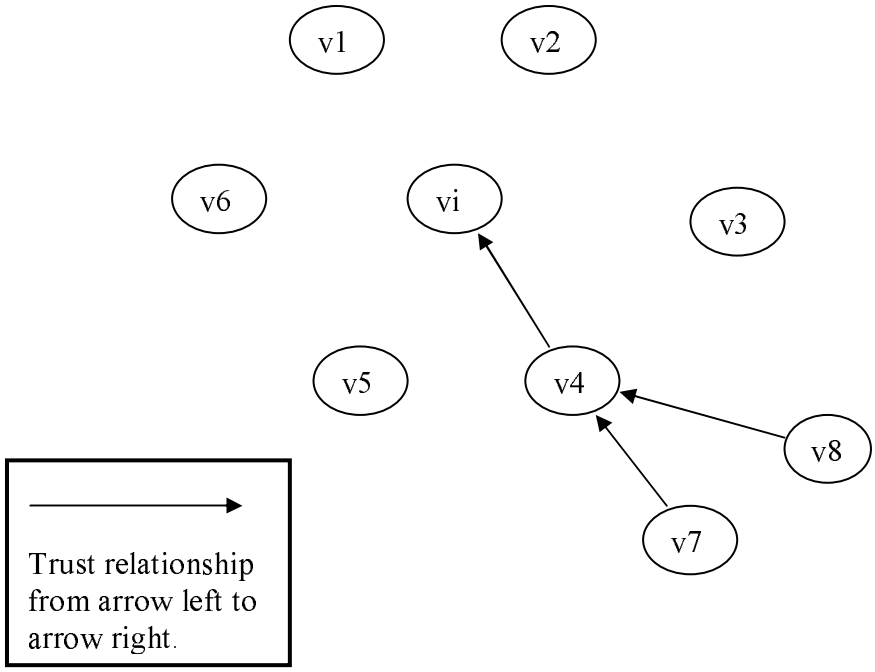

Fig 6.2b Example of trust chain in the time allowance period

We assume that vi's certificate has expired, so it is in the time allowance period. Normally, vi will be isolated from the network after its certificate expired. However, if there is a trust relation present in the network diagram, such ask node v4 trust vi as v4 has high trust level to vi. In the time allowance period, a neighbouring can still communication with vi as usual if it trust vi. Also, the node farther away like node v7 can build up a node chain if node v7 trust v4, and v4 trust vi. Then, v7 can still trust and communication with vi.

6.3. Certificate Renewal on Demand

Node vi can have very long time allowance period if its neighbours are keeping high trust value on it. Unless at a moment that some of vi's neighbour does not trust vi anymore as it may have low trust value on vi now. For example, if v4 does not trust vi anymore, it will then send a "distrust" message to vi to alert him to do certificate renewal to ensure its trust in the network (Fig 6.3a).

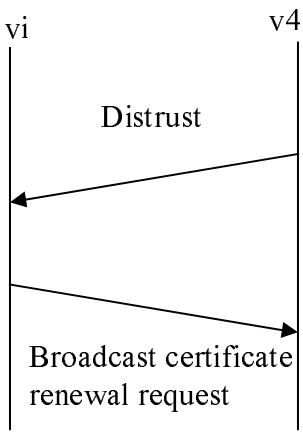

Fig 6.3a Triggering certificate renewal by the "Distrust" message

If node v4 does not trust vi any more, it will also break the trust chain that involving other nodes, like node v7 and v8. Therefore, vi is suggested to carry out renewal of certificate immediately, so it won't be isolated by any of the nodes. Also, it would be easier for it to gain enough partial certificates in the case that most of its neighbours still trust it. Otherwise, it will be too late if more and more nodes alert it with the "distrust" messages later.

Our design on the time allowance period and the on-demand request on certificate renewal aims at reducing the computational cost and increase the performance in the network, with the fully use of trust level in every node. The computational cost is high in doing certificate renewal, as the computational power of mobile devices is low. We want to minimize the cost by decreasing the number of times on certificate renewal. Relying on the trust value from neighbouring nodes, we hope can increase the performance without lowering any on the security on the network.

6.4 Machine-dependent Certificate Renewal

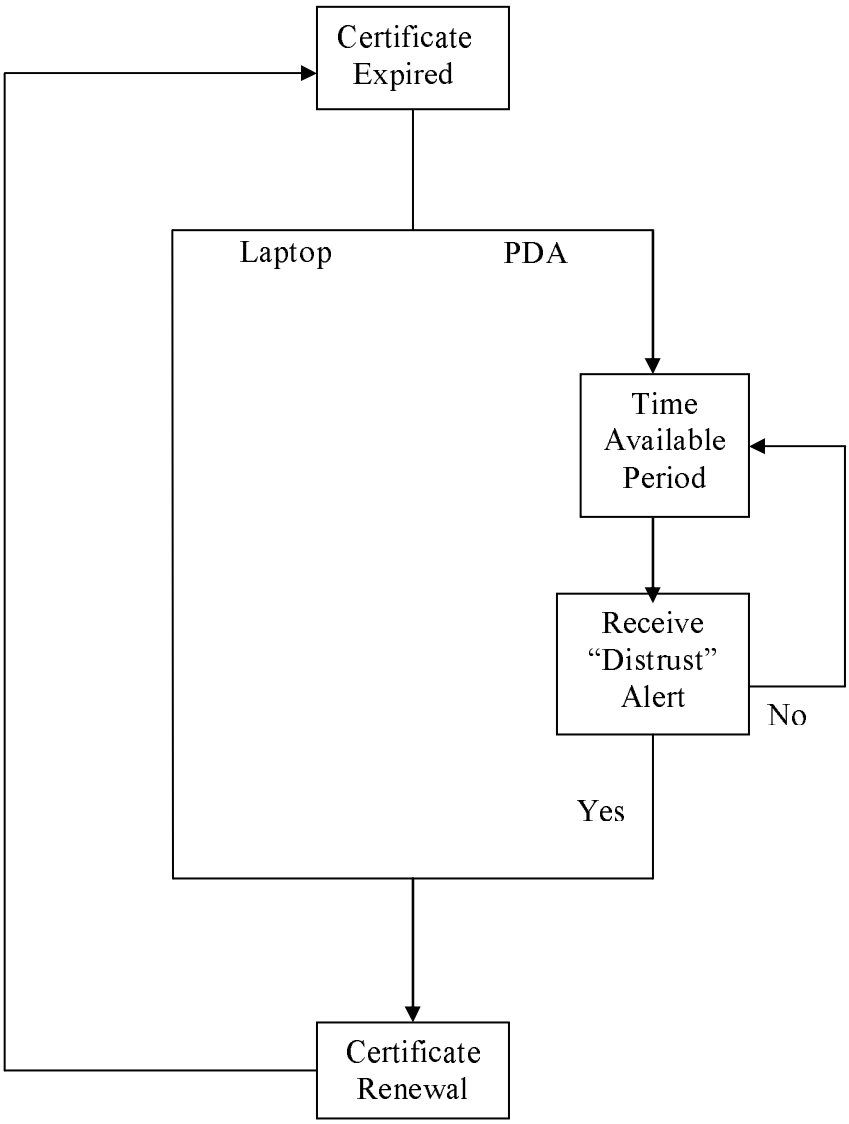

Fig.6.4a Flowchart of our machine-dependent certificate renewal policy

In our system, we recommend the mobile devices that have strong computation power, like notebook, to have certificate renewal immediately after the certificate expired. For the device that with low computation power, like handheld devices, they can delay certificate renewal until they receive some "distrust" alert from its neighbours (Fig 6.4a). This allows them to have certificate renewal less frequent and save some computational time.

7. Distributed Self-initialisation

In the above section, we have presented the certificate issuing/renewal is done by a coalition of k polynomial shares. In this section, we focus on how the polynomial shares are distributed to the nodes. We employ the following definition and mechanisms [25] to define a node being initialised, as the node possesses a valid polynomial share of SK, and assume that a dealer will initialise the first k nodes and quits. The initialized nodes collaboratively initialize other nodes, typically their neighbouring nodes.

7.1 Algorithm

A node vi broadcasts its request for initialization to its neighbours. A node vj receives the request, it checks vi's certificate and its trust level. The vj determines how many partial shares it will send to vi. Then, vj will send vi with the polynomial share IDs that it will give.

Let a_1, a_2, a_3, ... be the polynomial share IDs received by vi, the corresponding polynomial shares are Pa_1, Pa_2, Pa_3, ...

$$Pj = Pa_j * La_j(a_i) \mod N$$

where $Laj(ai) = \prod_{r=1, r \neq j}^{k} \frac{ai - ar}{aj - ar} \mod N$

The node vj can return the shares that it holds to vi. Without loss of generality, we assume a_1, a_2, a_3, ..., a_k be polynomial share IDs received by vi. By Lagrange interpolation, vi can generate a new partial share Pa_i:

$$Pa_i = f(a_i) = Pa_1 * La_1(a_i) + Pa_2 * La_2(a_i) + \ldots + Pa_k * La_k(a_i)$$

$$= \sum_{j=1}^{k} Paj * Laj(ai) = \sum_{j=1}^{k} Pj \mod N$$

Since it is insecure for node vj to send the Pj to vi directly as node vi can easily recover the partial share Paj, we employ the shuffle factor [25] approach. In our

system, we need to exchange the shuffle factors between each pair of polynomial shares in the same set. For example, dr,j can be the shuffling factor between of share r to share j. Therefore, after vi has collected all the polynomial share IDs and arranged them into c different sets, it needs to broadcast this information to all the neighbouring nodes taking part in the initialisation process of vi.

With the shuffle factors, Paj can be computed to a shuffled partial shares with a simple equation:

$$\overline{Pj} = Pj + \sum_{r=1, r\neq j}^{k} sign(ar - aj)dr, j$$

where the sign(x) = 1, when x>0;

sign(x) = 0, when x<0

Once node vi receives k shuffled partial shares from its neighbours, it can generate its

$$\sum_{j=1}^{k} \overline{Pj} = \sum_{j=1}^{k} \left(Pj + \sum_{r=1, r\neq j}^{k} sign(ar - aj)dr, j \right)$$

$$= \sum_{j=1}^{k} Pj + \sum_{j=1}^{k} \sum_{r=1, r\neq j}^{k} sign(ar - aj)dr, j$$

$$= \sum_{j=1}^{k} Pj + 0$$

$$= Pai \quad \text{(The new polynomial shares generated)}$$

7.2 Request for More Polynomial Shares

It should be noted that in our system some nodes may hold more than 1 polynomial share if its trust level is high from the view of other nodes. The default number of partial is 1 when the nodes firstly join the network and being initialised. A node may request for one more polynomial share when it know that its trust level is high from the view of other nodes. A node can know its trust level from the view of its neighbouring nodes when it received the reply of certificate renewal. It is because the validity period and the field "increase validity period" in the reply messages can give some indications. If a node has long validity period in its certificate and received the reply messages with the field "increase validity period" set from its neighbours, it can conclude that it has high trust level in the view of its neighbours. It can then request for one more polynomial share.

Having more than one polynomial share gives advantages to the node itself. It is because the polynomial shares it holds can be used in its own certificate renewal. If a node has more polynomial shares, it requires less number of neighbours to do certificate renewal. It increases the performance and flexibility. It is because waiting k partial certificates reply takes time in the certificate renewal. Also, if there are not enough neighbouring nodes, the node may wait for a long time, or cannot complete the process of certificate renewal successfully.

8. Future Work

Simulation will be carried out in the future. It helps to evaluate the performance of our authentication services. Also, variables can be set to different values in the experiment to give a better understanding on our authentication services. For example, the value k in the threshold (k,n) can be varied with the maximum number of partial secret, and the maximum hop counts that a certificate renewal request can be propagated. These experiments allow us to see the relationship between different variables and give a better understanding of our system. We can also compare the performance of our authentication services to the "fully distributed CA" approach, which without trust level concept and any of our modification.

Apart from the simulation, our authentication service is based on the use of RSA. The mathematically properties of RSA is rather complex when using with the weighted threshold secret sharing scheme. It takes rather long time for the cryptographic operations, and the communication protocol is rather complicated. We can investigate the possibility of using other less resource demanding algorithms, like elliptic curve cryptography, to obtain a less complex protocol and faster mathematical calculations.

9. Conclusion

In conclusion, we studied the characteristics of mobile ad hoc network and its security issues. An ad hoc network is a collection of nodes that do not have an infrastructure. The nodes are often mobile and communication via the wireless medium. Since nodes in the ad hoc network roam freely, the network has poor protection. Also, it lacks of clear of defence because of its open medium. Moreover, it does not have a centralized management point preventing the attacks to the network. These made ad hoc network vulnerable to both external and internal attacks. The security is in many cases dependent on proper key management. Since ad hoc network does not have any centralized resources, centralized key management approaches in traditional networks are not appropriate. We proposed a scalable, distributed authentication services to secure the mobile ad hoc networks.

We introduced a solution that combine the concept of trust level and the fully distributed certificate authorities in mobile ad hoc networks. In the previous works on fully distributed CA, a trust model has never been so well defined quantitatively from node to node basis. In our trust model, we assumed that each node has a trust value to its neighbouring nodes with our use of trust level. The trust value is used on certificate renewal and it can be combined on a trust path. The trust level concept can formalize the trust model in our system. We can make use of the trust level in our authentication services. For example, a node can determine to reply the certificate renewal request from its neighbours based on the quantitative trust level it gives to the node. Another example is that a node has never met to the requesting node can determine to reply the certificate renewal request from the trust value propagated in a trust chain. Without a formal trust level model, trust chain cannot be formed, and the certification and initialisation are greatly rely on the trust level model.

We adopted the fully distributed certificate authority approach, which means the capabilities of the certificate authorities are distributed to all nodes in the ad hoc network. Also, we have applied the weighted threshold secret sharing scheme instead of the general threshold secret sharing scheme in our system. The weighted threshold secret we applied allows a node to hold more than one partial share if it was agreed to

have high trust level. A node holds more partial share can pay more effort and be more powerful in certification and initialisation.

In the future, we will do simulation and provide the relationships between different variables in our model and information of the performance on our authentication services.

10. References

[1] "Key Management in Ad Hoc Networks", Klas Fokine, LITH-ISY-EX-3322-2002.

[2] "Security in Ad Hoc Networks", Vesa Karpijoki, Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory.

[3] "A performance comparison of multi-hop wireless ad hoc network routing protocols", David B. Johnson Yih-Chun Hu Jorjeta Jetcheva Josh Broch, DavidA. Maltz, MOBICOM, 1998.

[4] "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations", J. Macker S. Corson, Network Working Group rfc 2501, Jan 1999.

[5] "Mobile ad-hoc networks. Handbook of Wireless Networks and Mobile Computing", Silvia Giordano, Wiley, 2000.

[6] "A review of current routing protocols for ad hoc mobile wireless networks", Chai-Keong Toh, Elizabeth M. Royer, Santa Babara, IEEE Personal Communications, April 1999.

[7] "Intrusion detection in wireless ad-hoc networks", Wenke Lee, Yongguang Zhang, Mobicom, 2000.

[8] "Nodes bearing grudges: Towards routing security, fairness, and robustness in mobile ad hoc networks.", Jean-Yves Le Boudec Sonja Buchegger, Workshop on Parallel, Distributed Network-based Processing, Jan 2002.

[9] "Handbook of Applied Cryptography", A.Menezes, P. van Oorschot and S. Vanstone, CRC Press 1997, ISBN 0849385237

[10] "Internet X.509 public key infrastructure", draft-ietf-pkix-roadmap-06.txt, 2002.

[11] "The Kerberos network authentication service (version 5)", J.Kohl and B.Neuman, RFC-1510.

[12] "An overview of PKI trust models":, IEEE Network, p.38-43, vol.13, no.6, Nov-Dec 1999.

[13] "SPKI certificate theory", C.Ellison, W.Frantz, B.Lampson, R.Rivest, B.Thnomas and T.Ylonen, Internet RFC 2693, 1999.

[14] "PGP: Pretty Good Privacy", S.Garfinkel, O'Reilly & Associates Inc.,USA 1995.

[15] "The PGP trust model", A. Abdul-Rahman, EDI-Forum: the Journal of Electronic Commerce, Apr. 1997.

[16] "SDSI – a simple distributed security infrastructure," R. Rivest and B. Lampson., Working document from http://theory.lcs.mit.edu/ cis/sdsi.html

[17] "PKI practices and policy framework," ANSI X9.79, American National Standards Institute, 2000.

[18] "Increasing availability and security of an authentication service," L. Gong, IEEE Journal on Selected Areas in Communications, Vol.11, No.5, Jun. 1993

[19] "Proactive RSA," Y. Frankel, P. Gemmell, P. Mackenzie and M. Yung, CRYPTO, 1997.

[20] "Building intrusion tolerant applications," T. Wu, M. Malkin, and D. Boneh, Eighth USENIX Security Symposium, 1999.

[21] "Optimal-resilience proactive public-key cryptosystems," Y. Frankel, P. Gemmel, P. MacKenzie, M. Yung, FOCS '97, 1997.

[22] "Robust and efficient sharing of RSA functions," R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, Journal of Cryptology, 1996.

[23] "Parallel reliable threshold multi-signature," Y. Frankel and Y. G. Desmedt, Technical Report TR-92-04-02, Dept. of EECS, University of Wisconsin-Milwaukee, 1992.

[24] "Securing Ad Hoc Networks", L.Zhou and Z.J.Hass, IEEE Networks, Volume 13, Issue 6, 1999.

[25] "Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks", Haiyun Luo, Songwu Lu, Oct 2000, Technical Report UCLA-CSD-TR-200030.

[26] "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks", J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, IEEE ICNP 2001.

[27] "Self-securing Ad Hoc Wireless Networks", H. Luo, P. Zerfos, J. Kong, S. Lu and L. Zhang, IEEE ISCC 2002.

[28] "The Quest for Security in Mobile Ad Hoc Networks", J-P. Hubaux, L.Buttyan and S.Capkun, ACM 2001.

[29] "Self-Organized Public-Key Management for Mobile Ad Hoc Networks", S. Capkun, L. Buttyan and J.-P. Hubaux in Report on a Working Session on

Security in Wireless Ad Hoc Networks, ACM Mobile Computing and Communications Review (MC2R), Vol. 6, No. 4, 2002.

[30] "Secure Pebblenets", S. Basagni, K. Herrin, E. Rosti and Danilo Bruschi, ACM 2001.

[31] "Talking To Stranger: Authentication in Ad-Hoc Wireless Networks", Dirk Balfanz, DK Smetters, Paul Stewart and H. Chi Wong, Internet Society, Conference Proceding of NDSS Conference 2002.

[32] "Key Agreement in Ad Hoc Networks", N. Asokan, P. Ginzborg, Computer Communications Vol. 23, 2002.

[33] "Key Establishment in Ad-Hoc Networks", M. Hietalahti, Laboratory for Theoretical Computer Science, Helsinki University of Technology 2001.

[34] "The resurrecting duckling: Security issues for ad-hoc wireless networks", F. Stajano and R. Anderson, In Proceedings of the 7th International Aworkshop on Security Protocol.

[35] "Mitigating routing misbehavior in mobile ad hoc networks ", Kevin Lai Sergio Marti, T.J. Giuli and Mary Baker, Mobicom, 2000.

[36] "Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks)", Jean-Yves Le Boudec Sonja Buchegger, Mobihoc, June 2002.

[37] "Self-organised network-layer security in mobile ad hoc Networks", Songwu Lu-Wise 2002."Hao Yang-Xiaoqiao Meng Songwu Lu Hao Yang, Xiaoqiao Meng. Wise, 2002.

[38] "Prevention of denial of service attacks and selfishness in mobile ad hoc networks", Peitro Michiardi Rdfik Molva, Research Report, Jan 2002.

[39] "Making greed work in mobile ad hoc networks", Peitro Michiardi Rdfik Molva, Research Report RP-02-069, March 2002.

[40] "Securing ad hoc routing protocol", N.Asokan Manel Guerrero Zapata, Wise 2002.

[41] "Secure routing for mobile ad hoc networks", Panagiotis Papadimitratos and Zygmunt J. Haas,CNDS, 2002.

[42] "Securing ad hoc networks", Z.Haas L.Zhou, IEEE Network, 13(6): 24–30, Nov/Dec 1999.

[43] "Ariadne: A secure ondemand routing protocol for ad hoc networks", David B. Johnson Yih-Chun Hu, Adrian Perrig, MobiCom, 2002.

[44] "Sead: Secure efficient distance vector routing for mobile wireless ad hoc network", Adrian Perrig Yih-Chun Hu, David B. Johnson, WMCSA, 2002.

[45] "Cooperative routing in mobile adhoc networks: Current efforts against malice and selfishness", Jean-Yves Le Boudec Sonja Buchegger, 2002.

[46] "A Distributed Trust Model", Alfarez Abdul-Rahman & Stephen Hailes, 1997 New Security Paradigms Workshop, Proceedings, ACM Press, pp. 48–60, 1998.

[47] "How to Share a Secret", Adi Shamir, Communications of ACM, 1979

[48] "Valuation of Trust in Open Networks", Thomas Beth, Malte Borcherding, Birgit Klein, Proc. 3rd European Symposium on Research in Computer Security -- ESORICS '94.