Authentication & Authorization in Mobile Ad hoc Networks

Developed by
Khalid Mahmood

Supervised by
Prof. Dr. Khalid Rashid
&
Dr. Amir Qayyum

Department of Computer Science
International Islamic University, Islamabad
(2003)

International Islamic University, Islamabad
Faculty of Applied Sciences
Department of Computer Science

Dated: 21 | 7 | 2004

Final Approval

It is certified that we have read the thesis, titled "**Authentication & Authorization in Mobile ad hoc Network**" submitted by **Khalid Mahmood Reg. No. 37-CS/MS/01.** It is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad, for the Degree in **MS (Computer Science).**

Committee

External Examiner
Dr. Nazir A. Sangi,
Head, Department of Computer
Science, AIOU, Islamabad.

Internal Examiner
Mr. Sajid Hussain,
Department of Computer Science,
International Islamic University, Islamabad.

Supervisors:
1- Prof. Dr. Khalid Rashid
Dean,
Faculty of Applied Sciences and
Faculty of Management Sciences,
International Islamic University,
Islamabad.

2- Dr. Amir Qayyum
Center of Advance studies in Engineering (CASE) and
Center of Advance Research in Engineering (CARE),
F-5, Islamabad.

A dissertation submitted to the
DEPARTMENT OF COMPUTER SCIENCE
INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD
as a partial fulfillment of the requirements
for the award of the degree of
MS Computer Science

Dedicated to :

The Muslim Ummah

DECLARATION

I, hereby declare that this software, neither as a whole nor as a part thereof has been copied out from any source. It is further declared that I have developed this Software and the accompanied report entirely on the basis of my personal efforts made under the sincere guidance of our teachers. If any part of this report is proved to be copied out or found to be reported, I shall standby the consequences. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Khalid Mahmood
37-CS/MS/01

ACKNOWLEDGEMENTS

It is the blessings of Allah Almighty, Who gave me courage to accomplish this task. I am also grateful to a number of individuals whose encouragement and help made completion of this project & thesis a pleasant endeavor. In particular, I express my gratitude to kind supervisor Dr.khalid Rashid, Dean Faculty of Applied Sciences, International Islamic university, Islamabad who was the man behind the idea of "Authentication and Authorization of mobile ad hoc Networks". His motivations lead me to this success and who kept my morale high by his suggestions and appreciation. Without his precious guidance and help I could never be able to accomplish such task.

Particularly, I will not forget to express my gratitude to Dr. Amir Qayyum from CARE / CASE who always tried to keep me on right path during research work. He was available to me whenever and for whatever I consulted him.

I am also thankful to Dr. Muid Mufti from Telematix corp. who assisted me in implementation of research work in "Secure Wireless Banking" scenario and provided technical support. I am also thankful to him for providing 802.11 USB wireless cards which is core hardware requirement of this project.

And last but not the least; I would like to acknowledge the support of my family members and sincere friends. I would like to admit that I owe all my achievements to my truly, sincere and most loving parents, brothers and sisters, who mean the most to me, and whose prayers are a source of determination for me.

Khalid Mahmood
37-CS/MS/01

Project in Brief

Project Title:	Authentication & Authorization in Mobile Adhoc Networks
Objective:	To evolve an architecture of Protocol for Authentication & Authorization in Mobile ad hoc Network.
Undertaken By:	Khalid Mahmood
Supervised By:	Prof. Dr.Khalid Rashid *Dean,* *Faculty of Management Sciences and* *Faculty of Applied Sciences,* *International Islamic University, Islamabad.* Dr. Amir Qayyum *Center of Advance studies in Engineering (CASE) and* *Center of Advance Research in Engineering (CARE),* *F-5, Islamabad.*
Tool Used:	Visual C++. Net, Crystal Reports, Microsoft SQL Server, Rational Rose 2003.
Operating System:	Windows 98/ME/2000/XP
System Used:	Intel Pentium® III
Date Started:	31st October, 2002
Date Completed:	30th July 2003

Abstract

Research on mobile ad hoc networks has been focused primarily on routing protocols, whereas security issues remained mostly unexplored. This Thesis focuses on security mechanisms and notably on the key management mechanisms. We present a reliable solution for securing wireless bank transactions in bank premises, which is practical, and industry demanding requirement. In our approach devices exchange a limited amount of public information over location-constrained channel, which will later allow them to complete key exchange over wireless link. After the exchange of public keys, our banking transactions will be secure. This approach does not require any public key *infrastructure* as is required in MANET. Moreover our approach is secure against passive attacks on location-limited channel and all kinds of wireless attacks on wireless link. Instead of using traditional public key algorithms like RSA, we have included Elliptic cryptography in our approach for generating *public/private key pair because it is more* suitable for constrained environments (low memory wireless devices).we have also tried to include common wireless banking services to show the usefulness of this authentication protocol.

Table of Contents

ABBREVIATIONS

BRAN:	Broadband Radio Access Networks
CA:	Certification Authority
CDMA:	Code Division Multiple Access
CRL	Certificate revocation list
CTS:	Clear-to-send
DH:	Diffie-Hellman
DSSS:	Direct Sequence Spread Spectrum
ECC:	Elliptic Curve Cryptography
ECDH:	Elliptic Curve Diffie-Hellman
ECDSA:	Elliptic Curve Digital Signature Algorithm
EKE:	Encrypted Key Exchange Protocol
ETSI:	European Telecommunications Standards Institute
HCI:	Host Controller Interface
IKE:	Internet Key Exchange protocol
IR:	Infrared
KDC:	Key distribution center
KTC:	Key translation center
LAN:	Local Area Network
LMP:	Link Manager Protocol
L2CAP:	Logical Link Control and Adaptation Protocol
MANET:	Mobile Ad hoc Network
MFC:	Microsoft Foundation Classes
OO:	Object Oriented

OFDM:	Orthogonal Frequency Division Multiplexing
PAN:	Personal Area Networks
PDA:	Personal Digital Assistant
PKI:	Primary key infrastructure
RTS:	Short ready-to-send
RA:	Registration authority
SSL:	Exchangeable Image File
SDP:	Service Discovery Protocol
SIG:	Special Interest Group
TTP:	Trusted third party
WEP:	Wired Encryption Protocol
WLAN:	Wireless Local Area Network

Chapter 1

Introduction

1. INTRODUCTION

Ad-hoc networks are an emerging area of mobile computing. There are various challenges that are faced in the Ad-hoc environment. These are mostly due to the resource poorness of these networks. The definition of Ad hoc Networking continues to evolve broader from its literary meaning - Network for a particular purpose. The scenario, envisaged in the recent past, has become an immediately applicable, if not lagging behind. But our greatest fear is deploying this next generation technology without the absolutely necessary security measures put in place. They are usually set up in situations of emergency, for temporary operations or simply if there are no resources to set up elaborate networks. Ad-hoc networks therefore throw up new requirements and problems in all areas of networking. The solutions for conventional networks are usually not sufficient to provide efficient Ad-hoc operations. The wireless nature of communication and lack of any security infrastructure raise several security problems.

Typically the network nodes are interconnected through wireless interfaces and unlike traditional networks lack specialized nodes, i.e. routers, that handle packet forwarding. Instead every node in the network functions as a router as well as an application node and forwards packets on behalf of other nodes. Figure 1.1 shows such an example, node A is not within reach of node C, however by using node B as an intermediate node, A and C are able to communicate.

Ad hoc networks have the ability to form "on the fly" and dynamically handle the joining or leaving of nodes in the network. An example is when three people with ad hoc networking enabled PDA's come within communication range of each other. The three PDA's could then automatically create an ad hoc network used to exchange data.

Ad hoc Networking could be described as an evolution from Mobile IP (The convergence of mobile and data technology), through Packet Radio Networks. Further research is ongoing on the advancement to MANET (Mobile Ad hoc Networking) and Ubiquitous Computing. The English dictionary defines ad hoc as an adjective meaning "Formed for or concerned with one specific purpose; Improvised and often impromptu".

Ad hoc networking enables wireless devices to network with one another, as needed, even when access to the Internet is unavailable. It enables a wide range of powerful applications, from instant conferencing between notebook PC users to emergency and military services that must perform in the harshest conditions. Wireless communications without routers, base stations, or Internet Service Providers; an ad-hoc network might consist of several home-computing devices, plus a notebook computer that must exist on home and office networks without extra administrative work. Key applications can be conferencing, home networking, emergency services, Personal Area Networks, Bluetooth, and many more. Addressing the key challenges of ad hoc networking - resource management, scalability, and especially security is of the essence.

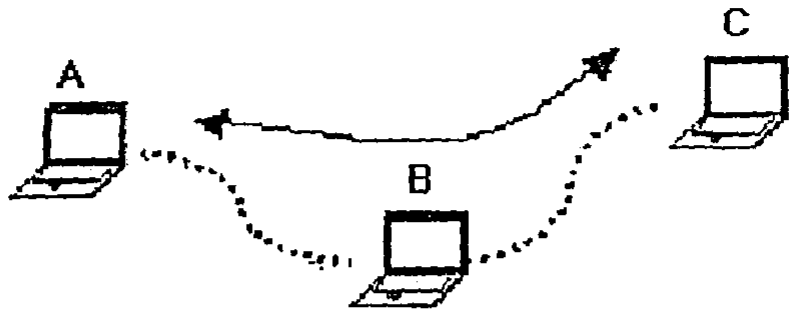


Figure 1.1 Wireless Ad hoc network

1.1 Characteristics of Ad hoc Network

When designing protocols for ad hoc networks, whether it be routing protocols or security protocols it is important to consider the characteristics of the network and realize that there are many "flavors" of ad hoc networks. Mobile ad hoc networks generally have the following characteristics:

Dynamic network topology

The network nodes are mobile and thus the topology of the network may change frequently. Nodes may move around within the network but the network can also be partitioned into multiple smaller networks or be merged with other networks.

Limited bandwidth

The use of wireless communication typically implies a lower bandwidth than that of traditional networks. This may limit the number and size of the messages sent during protocol execution.

Energy constrained nodes

Nodes in ad hoc networks will most often rely on batteries as their power source. The use of computationally complex algorithms may not be possible. This also exposes the nodes to a new type of denial of service attack, the "Sleep deprivation torture attack" that aims at depleting the nodes energy source.

Limited physical security

The use of wireless communication and the exposure of the network nodes increase the possibility of attacks against the network. Due to the mobility of the nodes the risk of them being physically compromised by theft, loss or other means will probably be bigger than for traditional network nodes.

In many cases the nodes of the ad hoc network may also have limited CPU performance and memory, e.g. low-end devices such as PDA's, cellular phones and

embedded devices. As a result certain algorithms that are computationally or memory expensive might not be applicable.

Besides the characteristics mentioned above that are due to the nature of ad hoc networking the following aspects that depend more on the application should also be considered.

Network origin: spontaneous vs. planned

Spontaneous: nodes with no prior relationship.

Planned: nodes with a prior relationship, e.g. belonging to the same company, military unit etc.

Network range: localized vs. distributed

Localized: the network nodes are within physical range, e.g. the same room.

Distributed: the nodes are distributed over a large area without the possibility of physically interacting.

Node capabilities: uniform vs. diverse

Uniform: all nodes have approximately the same capabilities in terms of power source, CPU performance and memory size etc.

Diverse: the nodes capabilities differ significantly, certain nodes may be high-end computers while other are embedded devices.

Network transiency: short term vs. long term

Short term: Nodes come together once and create an ad hoc network, when finished no knowledge is kept about the other nodes. These networks typically only persist during a relatively short time period, i.e. less than a few hours.

Long term: The same nodes will probably be part of the same ad hoc network multiple times and therefore save information about the other nodes for future use. These networks will persist during a longer time period. This also includes ad hoc networks that may only persist during a short time period, but that instead are created frequently.

1.2 Possible Applications of MANET

To motivate the development of ad hoc networking protocols, there needs to be applications where the properties of ad hoc networking are beneficial. This section will cover some applications where this is the case. Although some have been implemented many are still in the early research phase.

Military Tactical Networks

The first application of ad hoc networking was in the military domain. Ad hoc networking enables battlefield units to communicate anywhere and anytime, without the requirement of any fixed infrastructure. The fact that every node forwards packets also provides for a robust network. The loss of any one unit will not disrupt the network since there will (hopefully) be other units that can still provide packet forwarding services. Examples of military applications are the Tactical Internet [1] and the Saab NetDefence concept [8].

Personal Area Networks

The concept of personal area networks is about interconnecting different devices used by a single person, e.g. a PDA, cellular phone, laptop etc. In this case the PDA or the laptop will connect with the cellular phone in an ad hoc fashion. The cellular phone can then as an example be used to access Internet.

Another example could be when a person holding a PDA comes within communication range of a printer. If both the PDA and the printer were ad hoc enabled the PDA could automatically get access to the printing services.

Sensor Networks

Sensor networks are ad hoc networks consisting of communication enabled sensor nodes [9]. Each such node contains one or more sensors, e.g. movement-, chemical- or heat sensors. When a sensor is activated it relays the obtained information through the ad hoc network to some central processing node where further analysis and actions can be performed.

Such sensor networks may consist of hundreds or thousands of sensors and can be used in both military and non-military applications, e.g. surveillance, environmental monitoring etc. Sensor networks differ significantly from the other types of ad hoc networks described in this section. The most significant difference is the small size, extremely limited power resources and processing power of the sensor nodes.

Collaborative Networking

This application of ad hoc networking may be the most intuitive. The simplest example is when a group of people are attending a meeting and need to share information between their laptops or PDA's. If these devices were ad hoc enabled they could dynamically set up a network consisting of the meeting participants and thus enable the sharing of the information. Without ad hoc networking, a great deal of configuration and setup would be required to accomplish this task.

Disaster Area Networks

Ad hoc networking allows for the quick deployment of a communication network in areas where no fixed infrastructure is available or where the fixed infrastructure has

been destroyed by natural disasters or other events. Thus such networks could be used to improve the communication among rescue workers and other personnel and thereby support the relief efforts.

1.3 Network Security

When discussing network security, three aspects can be covered; the services required, the potential attacks and the security mechanisms. The security services aspect includes the functionality that is required to provide a secure networking environment while the security attacks cover the methods that could be employed to break these security services. Finally the security mechanisms are the basic building blocks used to provide the security services.

1.3.1 Security Services

In providing a secure networking environment some or all of the following services may be required [3]:

Confidentiality

Ensures that transmitted information can only be accessed by the intended receivers.

Authentication

Allows the communicating parties to be assured of the others identity.

Integrity

Ensures that the data has not been altered during transmission.

Non-repudiation

Ensures that parties can prove the transmission or reception of information by another party, i.e. a party cannot falsely deny having received or sent certain data.

Availability

Ensures that the intended network services are available to the intended parties when required. Depending on the capabilities of any potential attacker different mechanisms may be used to provide the services above.

1.3.2 Security Attacks

Security attacks can be classified in the following two categories [3] depending on the nature of the attacker.

1.3.2.1 Passive attacks

The attacker can only eavesdrop or monitor the network traffic. Typically this is

the easiest form of attack and can be performed without difficulty in many networking environments, e.g. broadcast type networks such as Ethernet and wireless networks.

1.3.2.2 Active attacks

The attacker is not only able to listen to the transmission but is also able to actively alter or obstruct it. Furthermore depending on the attacker's actions, the following subcategories can be used to cover the majority of attacks.

- **Eavesdropping**

This attack is used to gain knowledge of the transmitted data. This is a passive attack which is easily performed in many networking environments as mentioned above. However this attack can easily be prevented by using an encryption scheme to protect the transmitted data.

- **Traffic analysis**

The main goal of this attack is not to gain direct knowledge about the transmitted data, but to extract information from the characteristics of the transmission, e.g. amount of data transmitted, identity of the communicating nodes etc. This information may allow the attacker to deduce sensitive information, e.g. the roles of the communicating nodes, their position etc. Unlike the previously described attack this one is more difficult to prevent.

- **Impersonation**

Here the attacker uses the identity of another node to gain unauthorized access to a resource or data. This attack is often used as a prerequisite to eavesdropping. By impersonating a legitimate node the attacker can try to gain access to the encryption key used to protect the transmitted data. Once this key is known by the attacker, she can successfully perform the eavesdropping attack.

- **Modification**

This attack modifies data during the transmission between the communicating nodes, implying that the communicating nodes do not share the same view of the transmitted data. An example could be when the transmitted data represents a financial transaction where the attacker has modified the transactions value.

- **Insertion**

This attack involves an unauthorized party, who inserts new data claiming that it originates from a legitimate party. This attack is related to that of impersonation.

- Replay

The attacker retransmits data previously transmitted by a legitimate node.

- **Denial of service**

This active attack aims at obstructing or limiting access to a certain resource. This resource could be a specific node or service or the whole network.

1.3.3 Security Mechanisms

Most of the security services previously mentioned can be provided using different cryptographic techniques. The following subsections give an overview of which techniques are used to provide each of the services.

Confidentiality

The confidentiality service can be of two different types. The most common type of confidentiality requirement is that transmitted information should not be exposed to any unauthorized entities. A more strict confidentiality requirement is that the very existence of the information should not be revealed to any unauthorized entities.

The first type of confidentiality requirement only requires protection from eavesdropping attacks and can be provided using an encryption scheme. The stricter requirement implies that the service must also provide protection against traffic analysis. Such a service will typically require additional mechanisms along with some encryption scheme.

Integrity

The integrity service can be provided using cryptographic hash functions along with some form of encryption. When dealing with network security the integrity service is often provided implicitly by the authentication service.

Authentication

Authentication can be provided using encryption along with cryptographic hash functions.

Non-repudiation

Non-repudiation requires the use of public key cryptography to provide digital signatures. Along with digital signatures a trusted third party must be involved.

Availability

The availability is typically ensured by redundancy, physical protection and other non cryptographic means, e.g. use of robust protocols.

1.4 Cryptographic Background

There are two types of cryptographic techniques.

- Symmetric encryption
- Asymmetric encryption or Public key encryption

1.4.1 Symmetric Encryption

Symmetric encryption is illustrated in figure 1.2. The plain text message *m* is encrypted using the shared key *k*, resulting in the ciphér text *c*. To recover the plain text message the cipher text is decrypted using the same key used to for the encryption. Symmetric encryption schemes can be used to provide confidentiality, integrity and authentication. The shared key must be distributed over a secure communication channel.

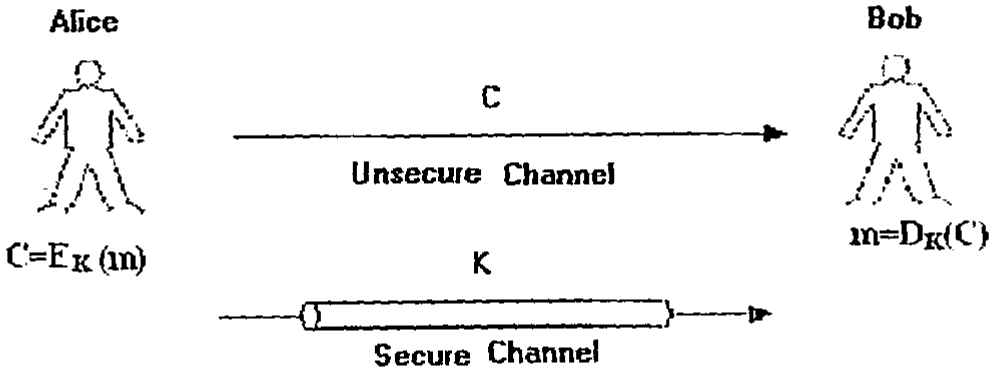


Figure 1.2 Symmetric encryption scheme

1.4.2 Public Key Encryption

Unlike conventional encryption schemes where the involved parties share a common encryption/ decryption key, public key encryption schemes depend on the use of two different but mathematically related keys. One of the keys is used for encryption and the other for decryption.

The public key encryption scheme is illustrated in figure 1.3. Bob generates a pair of keys, his public/private key pair pk_{Bob} /sk_{Bob}. The public key is related to the private key, but in such a way that the private key cannot be derived from it without additional information.

If Alice wants to send an encrypted message to Bob, she first needs to obtain his public key. As the name implies Bob's public key does not need to be kept secret, however it must be authenticated, i.e. Alice must be assured that the public key she believes belongs to Bob is really his. Once Alice has Bobs authentic public key pk_{Bob}, she

encrypts the plain text message m using it. The resulting cipher text c can then only be decrypted using Bob's private key sk_{Bob} which only Bob knows.

Compared with symmetric encryption, public key encryption has a weaker requirement for the communication channel over which the key distribution is performed. Public key encryption only requires an authenticated channel as opposed to a secure channel that is required for the distribution of symmetric encryption keys. Public key encryption can also provide non-repudiation along with confidentiality, integrity and authentication. However, public key encryption requires much more computational resources than symmetric encryption and therefore has much lower performance. Therefore public key encryption is typically only used to encrypt small amounts of data, e.g. symmetric encryption keys and digital signatures.

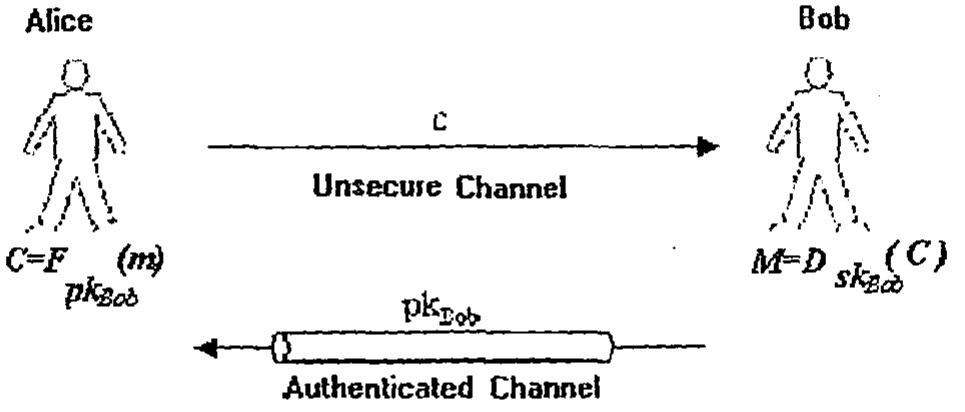


Figure 1.3 Public key encryption scheme

1.4.2.1 Diffie-Hellman

The Diffie-Hellman (DH) algorithm was the first public key algorithm published. However, it is limited to securely exchanging keys that can subsequently be used to provide the security services mentioned above.

The DH algorithm, illustrated in figure 1.4, requires two public parameters, a prime p and a generator g of Z_p. A generator of Z_p is an integer g such that $g, g2, \ldots, gp - 1 \pmod p$ generate the values 1 through $p - 1$ in some order. To exchange a shared key Alice and Bob generate the random secrets x_{Alice} and x_{Bob}. Bob then sends to Alice and Alice sends to Bob.

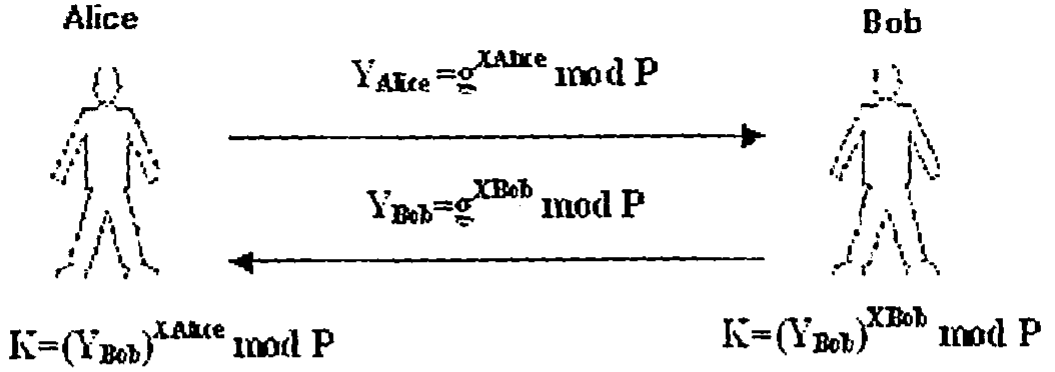


Figure 1.4 Diffie-Hellman key exchange

1.4.2.2 RSA

RSA is a public key encryption algorithm that can be used to provide confidentiality, integrity, authentication and non-repudiation services. To encrypt a message m or decrypt a cipher text c, the following calculations are performed:

$$c = m^e \bmod n$$

$$m = m^e \bmod n = m^{ed} \bmod n$$

If the algorithm is intended to be used to provide confidentiality the values n and e are made publicly known while d is kept secret. Therefore the public key $pk = \{e, n\}$ and the private key $sk = \{d, n\}$. For user A to encrypt a message intended for user B, B's public key pk_B is used for the encryption, $C=E_{pksm}(m)= m^e \bmod n$. Since only B has knowledge of the secret key sk_B it alone can decrypt the cipher text and recover the plain text, $m= D_{skb}(c)= c^d \bmod n$.

1.4.3 Digital Signature

A digital signature is a data structure that provides proof of origin, i.e. authentication and integrity, and depending on how it is used, it can also provide non-repudiation. Alice wants to send a message to Bob, however she doesn't want it to be modified during transmission and Bob wants to be sure that the message really came from Alice. What Alice does is that she computes a hash digest of the message which she encrypts with her private key sk_{Alice}. She then sends both the message and the encrypted digest which is here signature. Bob can then verify the signature by computing the hash digest of the message he received and comparing it with the digest he gets when decrypting the signature using Alice's public key pk_{Alice}. If the digests are equal Bob knows that Alice sent the message and that it has not been modified since she signed it.

1.4.4 Digital Certificate

Public key cryptography is very useful, but in the presence of active attackers a problem arises. Consider the following, Alice wants to send a secret message to Bob, so she encrypts the message using Bobs public key pk_{Bob} that she has retrieved from a server. However the key that Alice retrieved actually belongs to an attacker. The secret message which was intended for Bob can now be decrypted and read by the attacker.

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

Figure 1.5 shows the information in an X.509 certificate. The serial number is used to uniquely identify the certificate, and issuer name is the name of the trusted party who has issued the certificate.

The validity field specifies how long the certificate is valid. The subject is the entity being identified by the certificate, i.e. the entity whose public key is being certified. The next two fields contain the public key being certified and information about what it is certified to be used for (e.g. encryption, signatures etc.). The extensions field can be used to specify any additional information about the certificate. The signature field contains the certificates signature along with information about the hash algorithm used etc.

1.4.5 Secret Sharing

Secret sharing allows a secret to be shared among a group of users (share holders) in such a way that no single user can deduce the secret from his share alone. Only by combining (a sufficient number of) the shares can the secret be reconstructed. A secret sharing scheme where k out of n share holders are needed to reconstruct the secret is referred to as a *(k, n)* threshold scheme.

1.4.5.1 Shamir's Secret Sharing

This (k, n) threshold secret sharing scheme proposed by Adi Shamir [11] is based on polynomial interpolation and works as follows. The secret S is to be shared among the n shareholders identified by idi, i = 1... n.

To reconstruct the secret Lagrange interpolation is used. With the knowledge of a minimum of k shares the polynomial f(x) can be reconstructed and the secret recovered by calculating f(0). It is important that no shareholder gains knowledge of any share other than his own. Otherwise he could potentially gain knowledge of k shares and then be able to reconstruct the secret himself. Therefore a trusted party is

needed to perform the reconstruction of the secret, i.e. the shareholders provide their shares to the trusted party who performs the action requiring the secret, e.g. the signing of certificates etc.

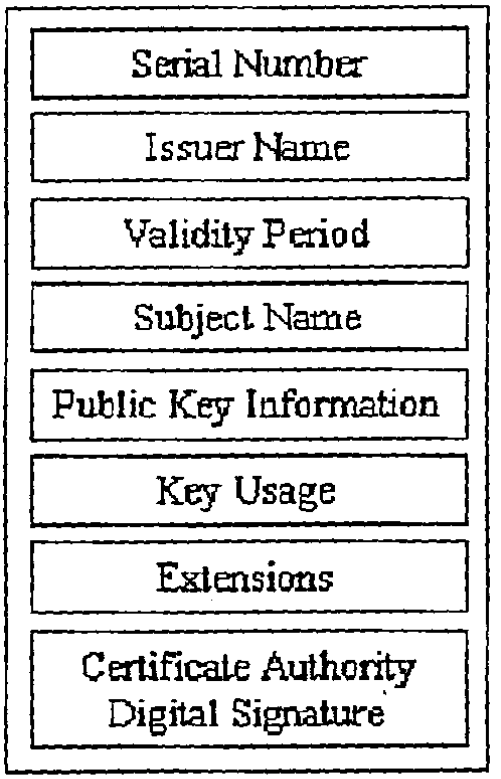


Figure 1.5 X.509 certificate format

1.4.5.2 Proactive Secret Sharing

In the secret sharing scheme described above the secret is protected by distributing it among several shareholders. However, given sufficiently long time an attacker could compromise k shareholders and obtain their shares, thereby allowing him to reconstruct the secret. To defend against such attackers proactive secret sharing schemes update the shares on a regular basis. An attacker must then compromise k shareholders between the updates since only k shares belonging to the same update period can be used to reconstruct the secret.

The updated shares $S_{I,updated}$ then be calculated as, $f_{new}(id_i), i = 1,..., k$. However, in practice it is enough to calculate the shares of the update polynomial, S_I $i = 1,..., k$ and securely distribute them to the respective shareholders. Each shareholder then adds it to its original share to obtain the updated share, i.e. $S_{I, updated} = S_I + S_I \pmod p$.

1.4.5.3 Verifiable Secret Sharing

If any shareholder wishes to prevent the reconstruction of the secret, he can provide an invalid share, e.g. a random value, to be used for the reconstruction. The Lagrange interpolation will then result in the reconstruction of a value, different from the secret S. Verifiable secret sharing mechanisms are used to prevent this type of denial of service attack.

1.5 Key Management

Most of the mechanisms used to provide the security services require the use of some kind of cryptographic keys that need to be shared between the communicating parties. As stated by Menezes et al [2], the purpose of key management is to:

- Initialize system users within a domain.
- Generate, distribute and install keying material.
- Control the use of keying material.
- Update, revoke and destroy keying material.
- Store, backup/recover and archive keying material.

Threats that must be dealt with by the key management system include:

- Compromise of the confidentiality of keys.
- Compromise of the authenticity of the keys.
- Unauthorized use of keys, e.g. the use of keys which are no longer valid.

1.5.1 Trusted Third Parties

A trusted third party (TTP) is an entity trusted by all users of the system and is often used to provide the key management services mentioned above. Depending on the nature of their involvement they can be categorized as in-line, on-line or off-line [2]. Figure 1.6 illustrates these different categories. An in-line TTP participates actively in-between the communication path of the two users while an on-line TTP participates actively but only for management purposes, the actual communication between the users is direct. An off-line TTP communicates with the users prior to their setting up a communication link. During the actual protocol run the off-line TTP is not active, in fact it does not even need to be connected to the network.

Examples of trusted third parties are key distribution centers (KDC), key translation centers (KTC) and certificate authorities (CA). The KDC and KTC are symmetric key management systems and the CA is a public key management system.

KDC's and KTC's are used to simplify the key management. Instead of each user having to share a secret key with every other user they only need to share one with the TTP. This brings down the number of keys that need to be managed from $n(n-1)/2$ to n, where n is the total number of users. Figure 1.7 illustrates how such a TTP is used.

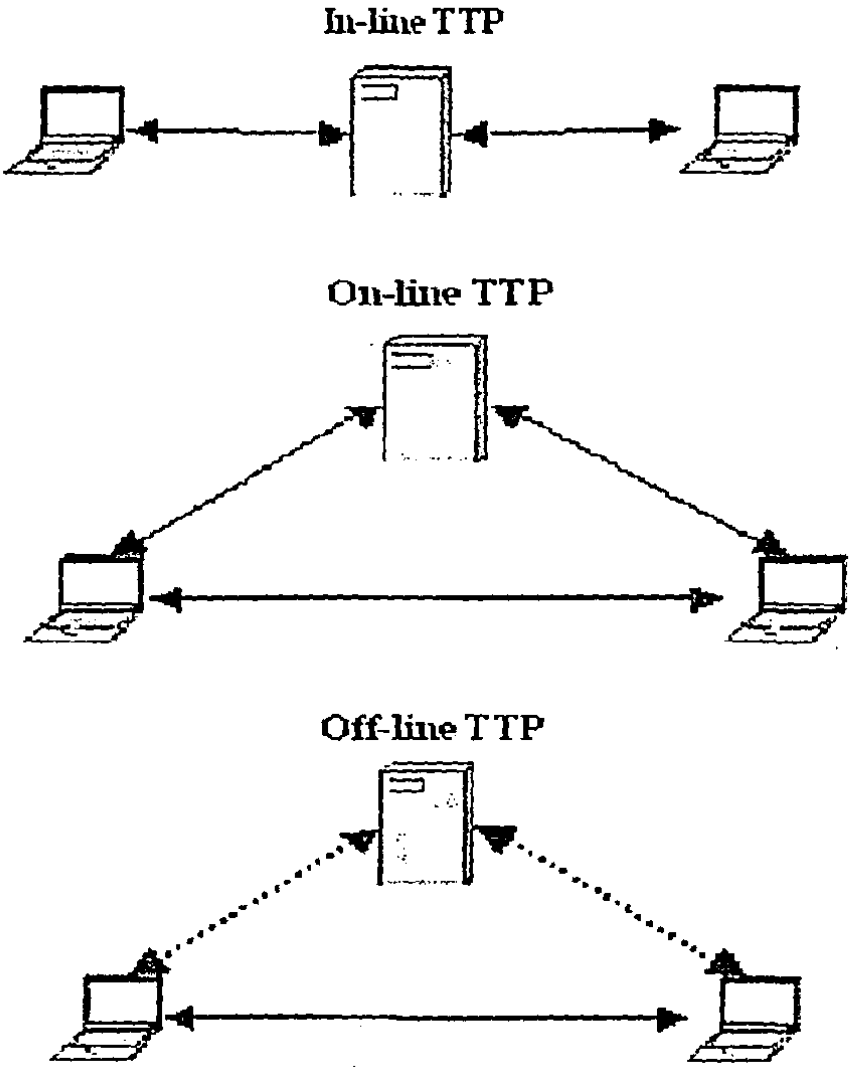


Figure 1.6 Categories of trusted third parties

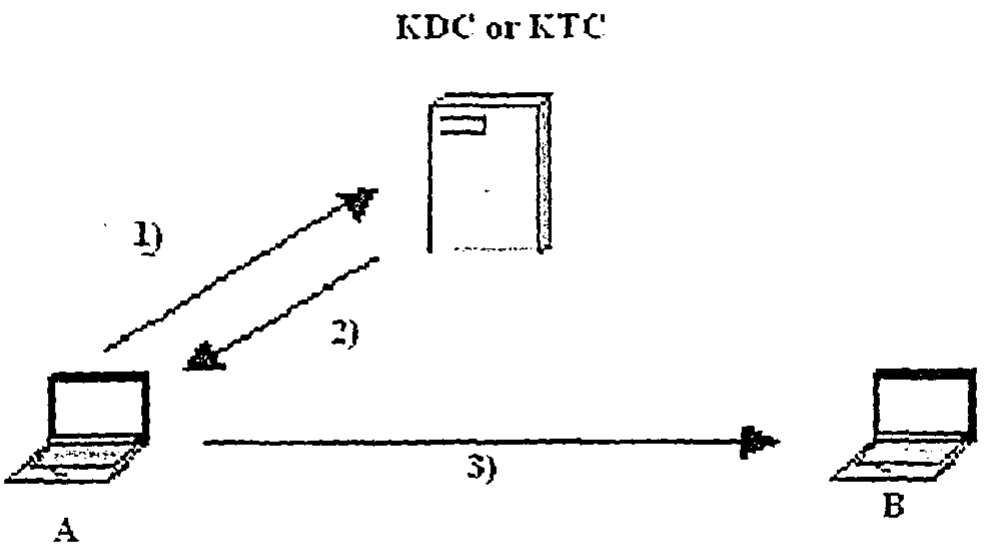


Figure 1.7 Function of a KDC or KTC

1) *User A requests to share a secret key with user B. If the TTP is a KDC it generates the key to use, otherwise user A provides it. This communication is encrypted using the key shared by user A and the TTP.*
2) *The TTP encrypts the session key with the key it shares with user B and returns it to user A*
3) *User A sends the encrypted session key to user B, who can decrypt it and thereafter use it to communicate securely with user A*

1.5.2 Public Key Infrastructure

The use of public key cryptography requires that the authenticity of the public keys can be established. A straightforward approach requires that any two users that wish to communicate must exchange their public keys in a authenticated manner and this would require the initial distribution of *n(n-1)* public keys. However, by having a trusted third party issue certificates to each of the users only the public key of the TTP needs to be distributed to each of the users.

A PKI provides the mechanisms needed to manage such certificates and consists of the components illustrated in Figure 1.8.

An end entity is either a user of a certificate or the subject to which a certificate has been issued. The certification authority (CA) is the component responsible for issuing and revoking certificates while the registration authority (RA) is responsible for establishing the identity of the subject of the certificate and the mapping between the subject and its public key. The registration function can be implemented by the CA and therefore the RA is an optional component.

The following basic services should be provided by the PKI components described above:

- Registration
- Initialization
- Certification
- Key update
- Revocation
- Certificate and revocation notice distribution

Other services that may also be provided by the PKI include key recovery, key generation, Cross-certification, secure time stamping and non-repudiation. The following subsections briefly describe each of the basic services mentioned above.

1.5.2.1 Registration

The registration service establishes the mapping between an end entity and its public key. This typically includes providing the public key to the RA along with any information that is required for the certificate, e.g. name, e-mail address, organization etc. The RA may also require that the end entity prove that it possesses the corresponding private key, e.g. by generating a digital signature.

Next the RA needs to verify the information received from the end entity. This could e.g. be done by requiring that the end entity in person provide proof of identity such as a driver's license or id-card. Finally when the RA has verified the identity of the end entity it contacts the CA and request the generation of the certificate.

1.5.2.2 Initialization

Before an end entity can use the services provided by the PKI it has to be initialized with certain information. The most important item required is the CA's certificate which contains the public key which is needed to verify any certificates issued by the CA. Other information could be addresses of certificate repositories and other PKI components that the user may need to contact etc. The initialization also includes the generation of the end entities public/private key pair.

1.5.2.3 Certification

Upon receiving a certification request from the RA, the CA generates and signs the certificate. This process includes filling in the certificate form with the information provided by the RA and adding any additional information required, e.g. any extensions used etc. The CA then digitally signs the certificate using its private key sk_{CA}.

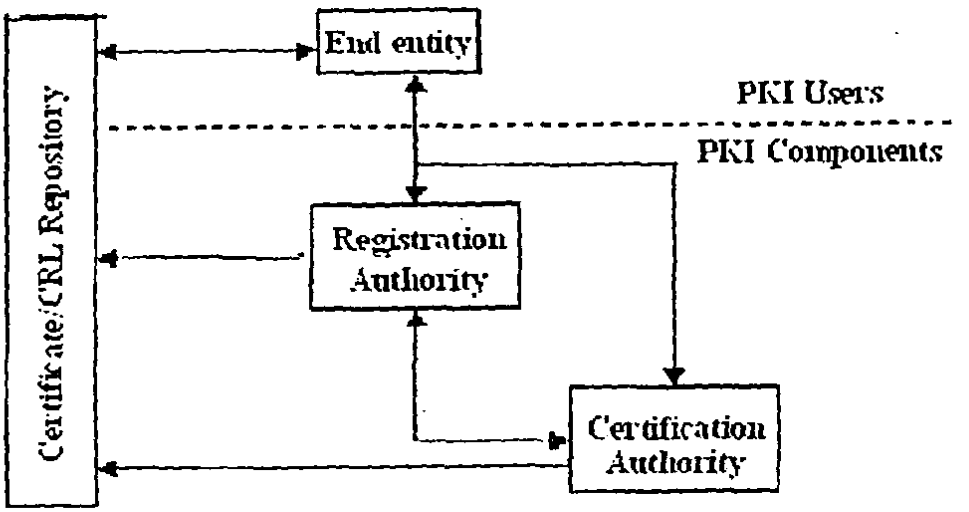


Figure 1.8 Main components of a PKI

1.5.2.4 Key update

Key pairs are typically only valid for a limited time, ranging from days to years depending on the application. The key update service provides for the transition to a new key pair and the issuing of the corresponding certificate.

1.5.2.5 Revocation

The CA is responsible for maintaining the status of the certificates it has issued. E.g. if a certificate becomes invalid due to the compromise of the private key the CA needs to revoke the certificate. Certificates may also need to be revoked if e.g. any information in the certificate becomes invalid, e.g. subject name, organization etc.

1.5.2.6 Certificate and Revocation Notice Distribution

After a certificate has been issued it needs to be made available to the owner and to other users who wish to use it. After the CA generates a certificate it can distribute it in a number of ways, e.g. by making it available on a publicly accessible server or by providing it to the certificate owner directly. In case a certificate is revoked the PKI must provide a mechanism that informs certificate users about this. A common method used is that the CA on a regular basis publishes a certificate revocation list (CRL) that lists all the certificates that have been revoked. Certificate users can then use the CRL to check whether or not a certain certificate has been revoked. Figure 1.9 shows an example of the contents of a CRL.

A problem with CRL's is that the time between the compromise and the notification of a certificate being revoked can be significant since CRL's are only published at regular intervals. A different approach to revocation notification is the use of on-line revocation

notification mechanisms. These allow the certificate users to query the CA in real-time for the status of a particular certificate.

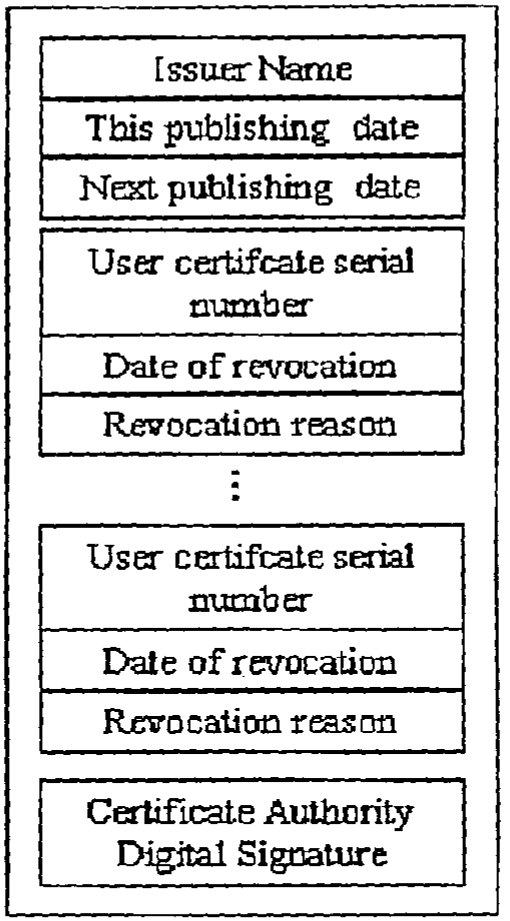


Figure 1.9 Example of CRL contents

1.6 Wireless Network Standards

The most influential standards-making body in the world of WLANs is the Institute of Electrical and Electronic Engineers (IEEE) a professional association for engineers, which is headquartered in the USA. The IEEE 802 Standards Committee is responsible for Local Area Network and Metropolitan Area Network standards. The Committee comprises various Working Groups focusing on particular areas. The 802.11 Working Group looks at WLAN standards. Within the 802.11 Working Group there are various Task Groups. Some of these, such as 802.11a, 802.11b and802.11g, concentrate on air interface standards. Others, such as 802.22e, 802.11f and 802.11i, concentrate on issues that cut across multiple air interfaces.

The European Telecommunications Standards Institute (ETSI) is also active in setting WLAN standards through its Broadband Radio Access Networks (BRAN) project. ETSIBRAN is responsible for the HiperLAN2 standard. ETSI is a trade association, most of whose members are equipment manufacturers, network operators or service providers.

Finally, the Bluetooth Special Interest Group (SIG) is, as one might guess, responsible for specifying Bluetooth, which is usually defined as a Personal Area Network standard rather than a Local Area Network standard. The Bluetooth SIG is an ad hoc group of companies with an interest in Bluetooth, led by nine promoter companies. The IEEE is not involved in setting the Bluetooth standard, but it is basing its own 802.15.1 Wireless Personal Area Network standard on Bluetooth.

The following sections discuss the standards developed by these groups in more detail.

We can classify Air Interface Standards into two types

- The 2.4 GHz Spectrum includes 802.11 b, Bluetooth and 802.11g.
- The 5 GHz Spectrum includes 802.11 a, HiperLan2 and 802.11h.

1.6.1 IEEE 802.11b

802.11b is today's most widely-used wireless LAN technology. 802.11b equipment that has been certified for interoperability by the Wireless Ethernet Compatibility Alliance (WECA) carries the Wi-Fi logo, so the standard itself is often referred to as Wi-Fi. 802.11b operates in the 2.4 GHz frequency band using a modulation system known as Direct Sequence Spread Spectrum (DSSS), which is similar to that used by CDMA (Code Division Multiple Access) cellular systems. There are three non-overlapping channels available for 802.11b networks in the license-exempt 2.4 GHz band. That enables users to run up to three access points in the same physical area.

802.11b is usually quoted as having a speed of 11 Megabits per second (Mbps). This is the maximum speed of the system under perfect conditions. However, as the distance between the user and access point or the amount of background noise or interference increases, the speed reduces in stages down to as low as 1 Mbps. The actual data transfer rate is also lower than the quoted speed because of system overheads. For example, an 802.11b link running at 11 Mbps will typically transfer data at a rate of 5-7 Mbps.

IEEE 802.11 Layers

The IEEE 802.11 standard places specifications on the parameters of both the physical (PHY) and medium access control (MAC) layers of the network. The PHY layer, which actually handles the transmission of data between nodes, can use either direct sequence spread spectrum, frequency-hopping spread spectrum, or infrared (IR) pulse position modulation. IEEE 802.11 makes provisions for data rates of either 1 Mbps

or 2 Mbps, and calls for operation in the 2.4 - 2.4835 GHz frequency band (in the case of spread-spectrum transmission), which is an unlicensed band for industrial, scientific, and medical (ISM) applications, and 300 - 428,000 GHz for IR transmission. Infrared is generally considered to be more secure to eavesdropping, because IR transmissions require absolute line-of-sight links (no transmission is possible outside any simply connected space or around corners), as opposed to radio frequency transmissions, which can penetrate walls and be intercepted by third parties unknowingly. However, infrared transmissions can be adversely affected by sunlight [5], and the spread-spectrum protocol of 802.11 does provide some rudimentary security for typical data transfers.

The MAC layer is a set of protocols which is responsible for maintaining order in the use of a shared medium. The 802.11 standard specifies a carrier sense multiple access with collision avoidance (CSMA/CA) protocol. In this protocol, when a node receives a packet to be transmitted, it first listens to ensure no other node is transmitting. If the channel is clear, it then transmits the packet. Otherwise, it chooses a random "backoff factor" which determines the amount of time the node must wait until it is allowed to transmit its packet. During periods in which the channel is clear, the transmitting node decrements its backoff counter. (When the channel is busy it does not decrement its backoff counter.) When the backoff counter reaches zero, the node transmits the packet. Since the probability that two nodes will choose the same backoff factor is small, collisions between packets are minimized. Collision detection, as is employed in Ethernet, cannot be used for the radio frequency transmissions of IEEE 802.11. The reason for this is that when a node is transmitting it cannot hear any other node in the system which may be transmitting, since its own signal will drown out any others arriving at the node.

Whenever a packet is to be transmitted, the transmitting node first sends out a short ready-to-send (RTS) packet containing information on the length of the packet. If the receiving node hears the RTS, it responds with a short clear-to-send (CTS) packet. After this exchange, the transmitting node sends its packet. When the packet is received successfully, as determined by a cyclic redundancy check (CRC), the receiving node transmits an acknowledgment (ACK) packet. This back-and-forth exchange is necessary to avoid the "hidden node" problem, illustrated in Figure 3. As shown, node A can communicate with node B, and node B can communicate with node C. However, node A cannot communicate node C. Thus, for instance, although node A may sense the channel to be clear, node C may in fact be transmitting to node B. The protocol described above alerts node A that node B is busy, and hence it must wait before transmitting its packet.

1.6.2 Bluetooth

Another 2.4 GHz band solution is called Bluetooth™. Bluetooth operates at lower power points than 802.11b or 802.11g, which gives it a shorter effective range (commonly 10 meters). This seeming limitation actually makes it more suitable for use in small battery-powered handheld equipment. Due to this lower power consumption, Bluetooth is viewed as an excellent solution for completely unwired phone headsets, links between printers and laptops, etc. Bluetooth devices communicate in groups known as piconets consisting of one master device and up to seven slaves. Each piconet hops 1600 times a second in a pseudo-random fashion among 79 different frequencies. This is important to understand since as the number of Bluetooth piconets in a given area increases, the chance of two piconets causing a data collision by trying to use the same frequency at the same time increases. In practice it is typically possible to run 8-10 piconets in a given area without losing a significant amount of data throughput.

Bluetooth is an open standard for short-range radios and is the primary technology used in ad hoc networks. These products are gaining interest due to their low-cost and low power requirements. This technology provides a small wireless network called a "personal area network" (PAN) as it is usually only transmitted in a 30' area or less. These products have much the same abilities of other wireless networks in that they can provide voice and data, eliminate cables, bridge two networks and has support for PDAs, mobile phones, printers, faxes, microphones and even earpieces. The main function of Bluetooth is to provide an ad hoc network to synchronize between personal devices.

These networks are usually temporary networks that change as ad hoc devices enter and exit the coverage area. One device is a "master" and can support up to seven devices within its network.

All of these devices use the same frequency, but can be in multiple networks. This means a slave device in one network can be a master in another network, extend the distance and support more users. An example would be a master laptop talking to two other master laptops in a different network.

Bluetooth also operates in the 2.4GHz ISM band, which is similar to WLAN and other 802.11 devices. Bluetooth uses a different modulation and employs a different spread spectrum. Theoretically, it has a bandwidth of 1Mbps, 79 different radio channels; frequency changes of 1,600 times per second and little interference with a single WLAN products. However, in reality the network cannot support such data rates because the forward error correction (FEC) may interfere with 802.11 networks for a short time. The second generation of Bluetooth should support 2Mbps, but look for faster rates in the future.

Bluetooth provides three types of power classes. These classes provide power and range in figure 1.10.Each has its usage. Class 1 is a cable replacement for a mouse or keyboard. Class 2 is to Connect a laptop to PDA, PDA to PDA, or laptop to laptop. Class 3 can compete with 802.11b and could be used for all of the above, or it can provide wireless connectivity for a whole building.

Type	Power Level	Operating Range
Class 3 Devices	100mW	Up to 100 meters
Class 2 Devices	10mW	Up to 10 meters
Class 1 Devices	1mW	0.1-10 meters

Figure 1.10 Bluetooth classes of range and power

Some of the features of Bluetooth technology are:

- Signals can be transmitted through walls and briefcases, thus eliminating the need for line-of-sight.
- Devices do not need to be pointed at each other, as signals are omni-directional.
- Both synchronous and asynchronous applications are supported, making it easy to implement on a variety of devices and for a variety of services, such as voice and Internet.
- Devices have a range of about 10 meters and up to eight devices can link to form a piconet. Piconets communicate with each other easily and the chips can change frequency at about 1600 hops per second. This phenomenon is known as frequency hopping, and provides protection against interference.
- Governments world wide regulate it, so it is possible to utilize the same standard wherever one travels.

Bluetooth transmissions also interfere with 802.11b transmissions. The interference can be quite serious if a Bluetooth device is placed right next to an 802.11b device (hence laptops that contain both technologies ensure only one transmits at a time) but the interference does not have much impact on the throughput of either technology if the separation is a few meters or more (so one person using a Bluetooth headset with their cell phone is unlikely to prevent everyone else in an office from using the 802.11b WLAN).

Bluetooth Specification Protocol Stack

Each layer is briefly described below and Figure 1.11 shows Bluetooth Specification Protocol Stack.

- The **Radio layer** defines the requirements for a Bluetooth transceiver operating in the 2.4 GHz ISM band

- The **Baseband layer** describes the specification of the Bluetooth Link Controller (LC) which carries out the baseband protocols and other low-level link routines.

- The **Link Manager Protocol (LMP)** is used by the Link Managers (on either side) for link set-up and control.

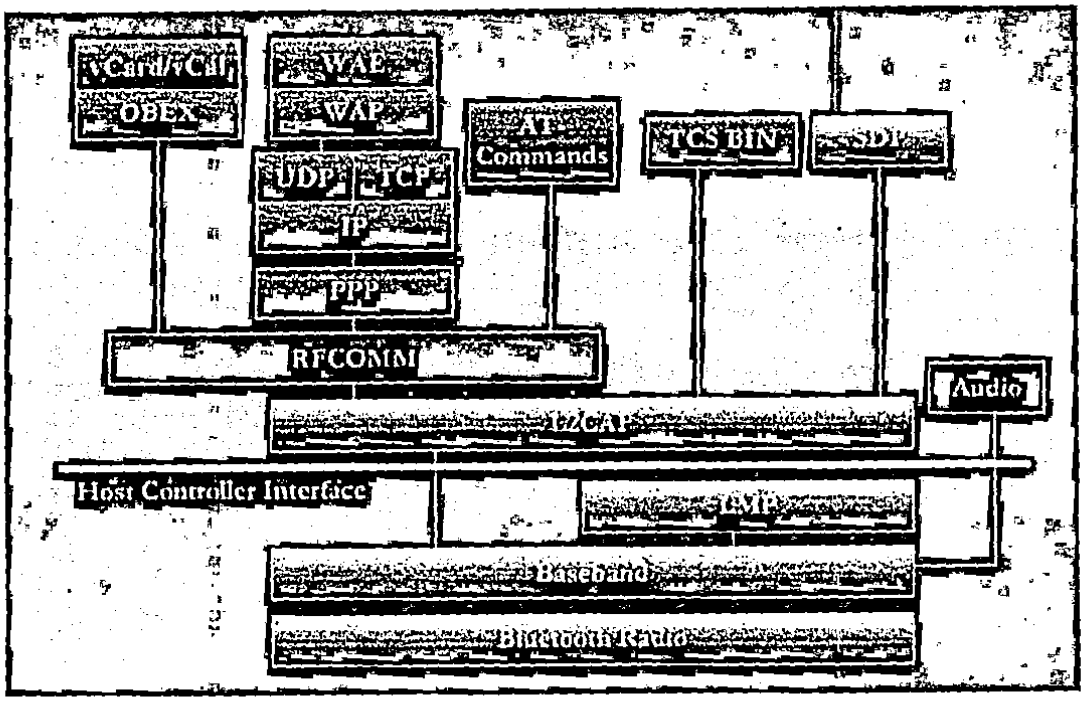


Figure 1.11 Bluetooth specification protocol stack

- The **Host Controller Interface (HCI)** provides a command interface to the Baseband Link Controller and Link Manager, and access to hardware status and control registers.

- **Logical Link Control and Adaptation Protocol (L2CAP)** supports higher level protocol multiplexing, packet segmentation and reassembly, and the conveying of quality of service information.

- The **RFCOMM** protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10.

- The **Service Discovery Protocol (SDP)** provides a means for applications to discover which services are provided by or available through a Bluetooth device. It also allows applications to determine the characteristics of available services.

1.6.3 802.11g

802.11g is a draft standard developed jointly by Texas Instruments and Intersil. 802.11g allows operation at up to 54 Mbps in the 2.4 GHz band. The draft standard states that 802.11g devices must support existing 802.11b modulation techniques to ensure that an 802.11b client device can talk to an 802.11g access point and vice versa. For higher data rates the draft standard specifies Orthogonal Frequency Division

Multiplexing (OFDM) as the mandatory modulation technique. OFDM has also been selected for 802.11a (see the following section on 5 GHz air interface standards). Two other optional modulation techniques are also included in the standard, but these techniques may not be implemented in commercial products. The 802.11g standard is expected to be ratified in January 2003 but "pre-standard" client cards and access points are already widely available.

1.6.4 802.11a

Contrary to what its name implies, 802.11a is a newer standard than 802.11b. Wireless vendors started shipping 802.11a equipment in late 2001 and it is expected to become widely available during 2002. 802.11a is designed to run at speeds of up to 54 Mbps in the license-exempt 5 GHz band. WECA also certifies 802.11a equipment for compatibility. 802.11a uses a modulation technique referred to as Orthogonal Frequency Division Multiplexing (OFDM) which uses multiple sub-carriers operating at slightly different frequencies.

There is more license-exempt spectrum available at 5 GHz than at 2.4 GHz and so it is possible to have more non-overlapping channels to increase coverage and density. In addition there are fewer existing sources of interference. However, 802.11a systems generally have a shorter range than 802.11b.

1.6.5 HiperLAN2

In Europe, the frequency regulations specify that WLAN systems operating in the 5 GHz band must support Transmit Power Control (i.e. they must be able to turn their "volume" controls up and down automatically) and Dynamic Frequency Selection (i.e. they must be able to move to a different part of the frequency band to avoid interference). The European HiperLAN2 standard supports both these features but 802.11a does not. For this reason, 802.11a is not generally approved for use in Europe, although it can be used under certain circumstances in particular countries.

1.6.6 802.11h

However, the 802.11h Task Group is working on a new version of the standard that would support Transmit Power Control and Dynamic Frequency Selection. Consequently, vendor support for HiperLAN2 is waning and it seems quite likely that the majority of 5 GHz systems in Europe will be 802.11h rather than HiperLAN2.

Chapter 2

Existing Security Models in MANET

2. Existing Security Models in MANET

The major problem in providing security services in such infrastructure less networks is how to manage the cryptographic keys that are needed. Various security models for ad hoc networks are presented in this chapter.

2.1 Partially distributed certificate Authority

This solution proposed by Zhou and Hass [10] uses a *(k, n)* threshold scheme to distribute the services of the certificate authority to a set of specialized server nodes. Each of these nodes is capable of generating a partial certificate using their share of the certificate signing key sk_{CA}, but only by combining k such partial certificates can a valid certificate be obtained. The solution is suitable for planned, long-term ad hoc networks. Since it is based on public key encryption it requires that the all the nodes are capable of performing the necessary computations. Finally it assumes that subsets of the nodes are willing or able to take on the specialized server role.

The system contains three types of nodes; client, server and combiner nodes. The client nodes are the normal users of the network while the server and combiner nodes are part of the certificate authority. The server nodes are responsible for generating partial certificates and storing certificates in a directory structure allowing client nodes to request for the certificates of other nodes. The combiner nodes which are also server nodes are responsible for combining the partial certificates into a valid certificate. Although not stated implicitly by the authors the system also has an administrative authority which will be termed the dealer. The dealer is the only entity in the system that has knowledge of the complete certificate signing key sk_{CA}.

Every node in the network has a public/private key pair and it is the responsibility of the dealer to issue the initial certificate for the nodes public key as well as distributing the public key pk_{CA} of the certificate authority which is needed to verify the certificates.The certificate authority as a whole has a public/private key pair, pk_{CA} /sk_{CA} of which the public key is known to all network nodes. The private key sk_{CA}, is shared among the server nodes according to Shamir's secret sharing scheme. Figure 2.1 illustrates the different components of the system.

2.1.1 Certificate Issuing

Before any node may join the network it must first obtain a valid certificate from the dealer off-line. A the same time the node should be supplied with the CA's certificate along with any other parameters that are required.

2.1.2 Certificate Renewal

The certificates are only valid for a certain amount of time and therefore need to be renewed before they expire. When a node wishes to renew its certificate, it must request a certificate renewal from a minimum of k server nodes. If the request is granted, each of these k server nodes generates a partial certificate with a new expiration date. These partial certificates are then sent to a combiner, which could be one of the k servers, which then combines the partial certificates.

If any of the servers are compromised they may generate an invalid partial certificate which they then send to the combiner. The certificate produced by the combiner will then also be invalid. This type of denial of service attack is prevented by verifying the validity of the certificate before accepting it. If the combiner detects that the certificate is invalid it will request a new set of partial certificates until a valid certificate is obtained. If a node changes its private/public key pair it will need to update its certificate with the new public key, this is accomplished in a similar way as the renewal.

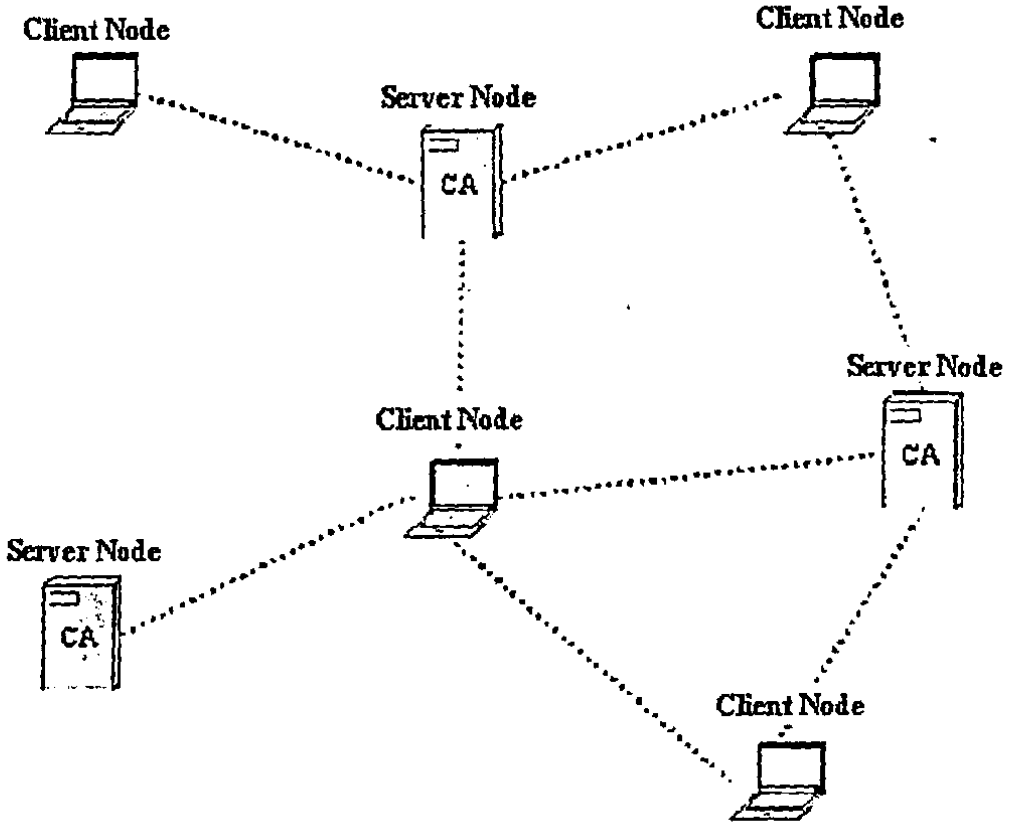


Figure 2.1 System architecture showing three server nodes of which one is also a combiner

2.1.3 Certificate Retrieval

The server nodes are responsible for storing the certificates of all nodes in the network. This allows any nodes requiring the public key of any other nodes to simply request the corresponding certificate from any of the server nodes.

This service requires that all nodes must register their certificate with the servers when they initially join the network. The servers must also have a mechanism of synchronizing their certificate directories in the case of updates and renewal.

2.1.4 System Maintenance

The maintenance of the certificate authority consists of two main parts; the issuing of the initial shares and the proactive update of the shares to protect against mobile adversaries. The share update can also allow the system to change its configuration, e.g. from a *(3, 8)* to a *(2, 5)* threshold scheme. This could be useful e.g. if some of the servers have been compromised or for some other reason are unavailable.

During the bootstrapping of the network the dealer generates n shares of the CA's private key sk_{CA} and supplies each of the n servers with a share.At periodic intervals the servers update their shares of the CA's private key sk_{CA}. At the beginning of the update, each server generates a random (n, k) sharing of 0 and distributes a share to each of the other servers, each of these shares are called sub-shares. Each server now has n sub-shares from different servers which are added to their old share giving them their new updated share.

At periodic intervals the servers update their shares of the CA's private key sk_{CA}. At the beginning of the update, each server generates a random (n, k) sharing of 0 and distributes a share to each of the other servers, each of these shares are called sub-shares. Each server now has n sub-shares from different servers which are added to their old share giving them their new updated share. Figure 2.2 illustrates the share update mechanism.

During the share update a malicious server can potentially launch a denial of service attack against the distributed CA by generating invalid sub-shares. When the other servers use these to update their old shares the updated share will also be invalid, effectively preventing the correct generation of certificates. To prevent such an attack the authors propose using a verifiable secret sharing scheme. This allows the servers to validate each of the sub-shares before using them.

Due to the highly dynamic topology of ad hoc networks, all servers might not be connected during the share update. Therefore, mechanisms must be in place to handle such situations. An example would be if the network is segmented into two parts. The servers in each part may update their shares independently of each other. If the network

later merges, the shares held by the servers will be inconsistent. A mechanism that deals with this is mentioned by the authors but no details about it are given.

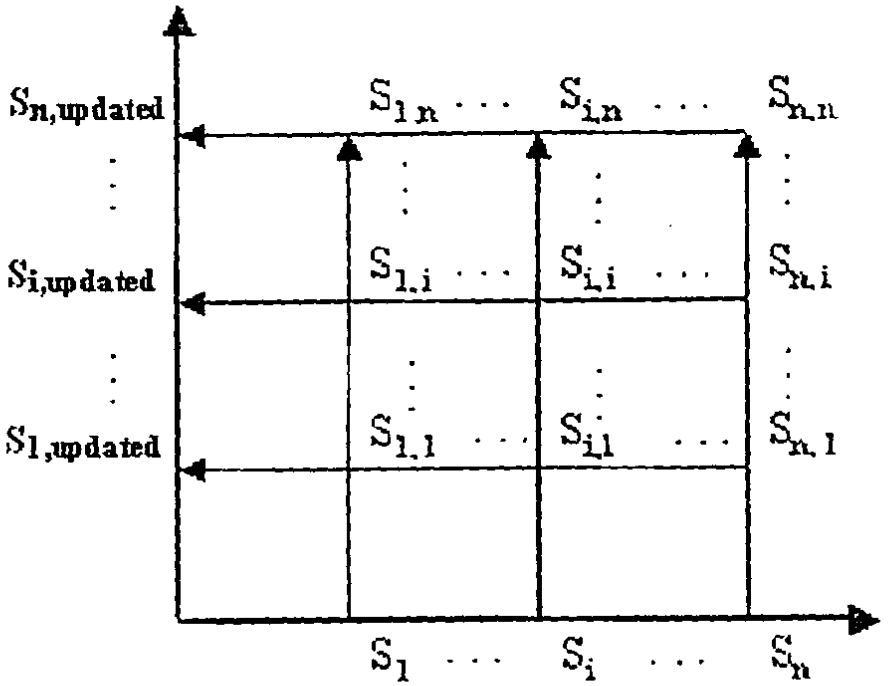


Figure 2.2 Illustration of the share update mechanism

2.2 Fully Distributed Certificate Authority

This solution is first described by Luo and Lu in [12] and later analyzed by Luo et al in [13] and [14]. Its uses a *(k, n)* threshold scheme to distribute an RSA certificate signing key to all nodes in the network. It also uses verifiable and proactive secret sharing mechanisms to protect against denial of service attacks and compromise of the certificate signing key.

Similar to the solution presented in section 2.1, this solution is aimed towards planned, long term ad hoc networks with nodes capable of public key encryption. However, since the service is distributed among all the nodes when they join the network, there is no need to elect or choose any specialized server nodes. The availability of the service is based on the assumption that every node will have a minimum of k one-hop neighbors and that the nodes are provided with a valid certificate prior to their joining the network. The system then provides services to maintain and update these initial certificates.

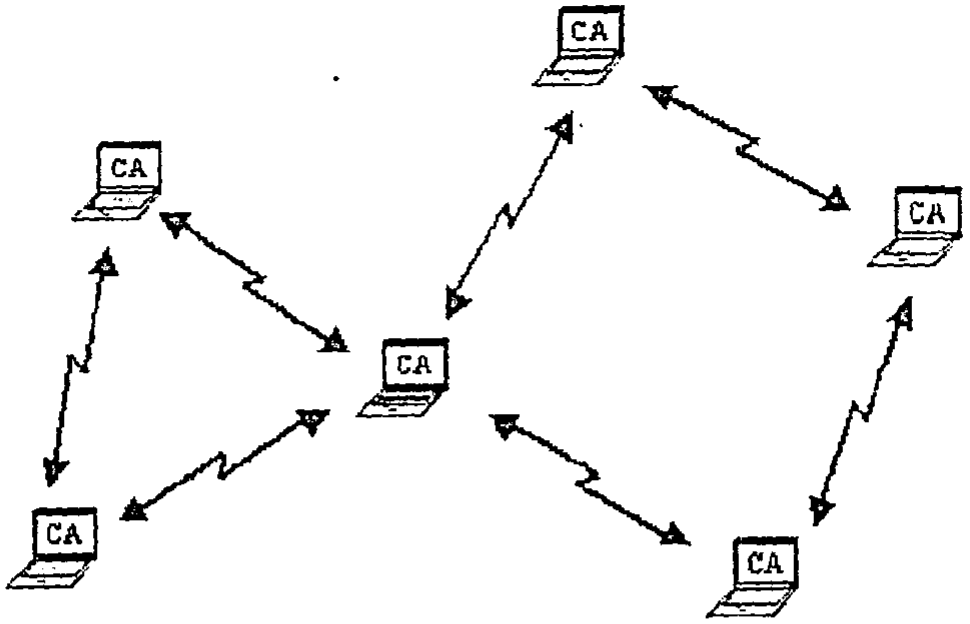


Figure 2.3 Fully distributed CA service where all nodes in the network are equals and each hold a share of the signing key

2.2.1 System Maintenance

This section describes the steps required to setup and maintain the distributed CA service. The maintenance is required to handle the joining of new nodes and to protect the service against attackers who try to compromise the CA service.

2.2.1.1 System Bootstrapping

During the system bootstrapping phase, the administrative authority responsible for the ad hoc network (known as the dealer) initializes the first k nodes. The initialization includes providing the nodes with their own certificates $cert_{id}$, the CA's certificate $cert_{CA}$ and their shares of the CA's secret key sk_{CA}. The bootstrapping phase is illustrated in figure 2.4. After the bootstrapping of the first k nodes the dealer is only responsible for the registration, initialization and initial certification of any new nodes joining the network.

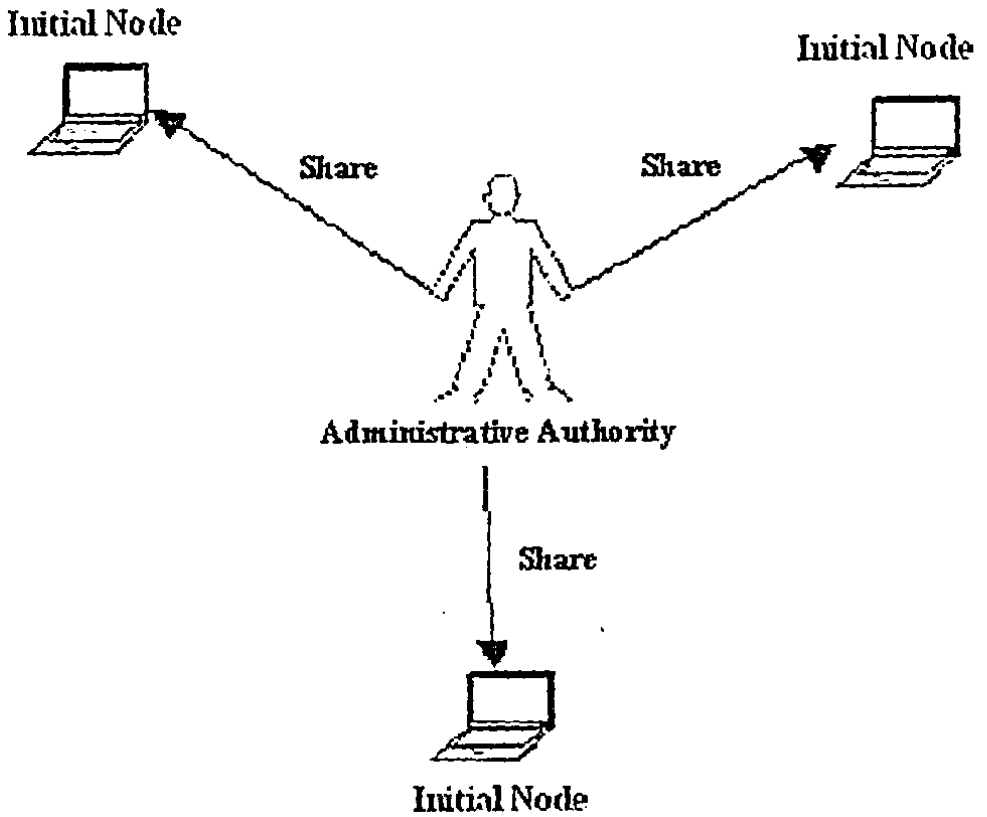


Figure 2.4. Share initialization during the bootstrapping phase. The dealer provides each of the initial nodes with their share

2.2.1.2 Share Initialization

Any new nodes joining the network are incorporated into the distributed CA by being provided with their own share of the CA certificate signing key sk_{CA}. Since the dealer is no longer part of the network this share distribution mechanism needs to be handled by the nodes that have already been initialized as illustrated in figure 2.5.

However the joining node can only be allowed to know the value of the sum of the k sub-shares, not the value of the sub-shares themselves. The reason for this is that is a publicly known value and therefore the Si can be derived, thereby revealing the secret shares of the nodes in the coalition. After being initialized with its share of the certificate signing key sk_{CA} the node has become part of the CA and can participate in the provision of the CA's service, including certificate renewal and revocation as well as initializing any joining nodes with their share of the certificate signing key sk_{CA}.

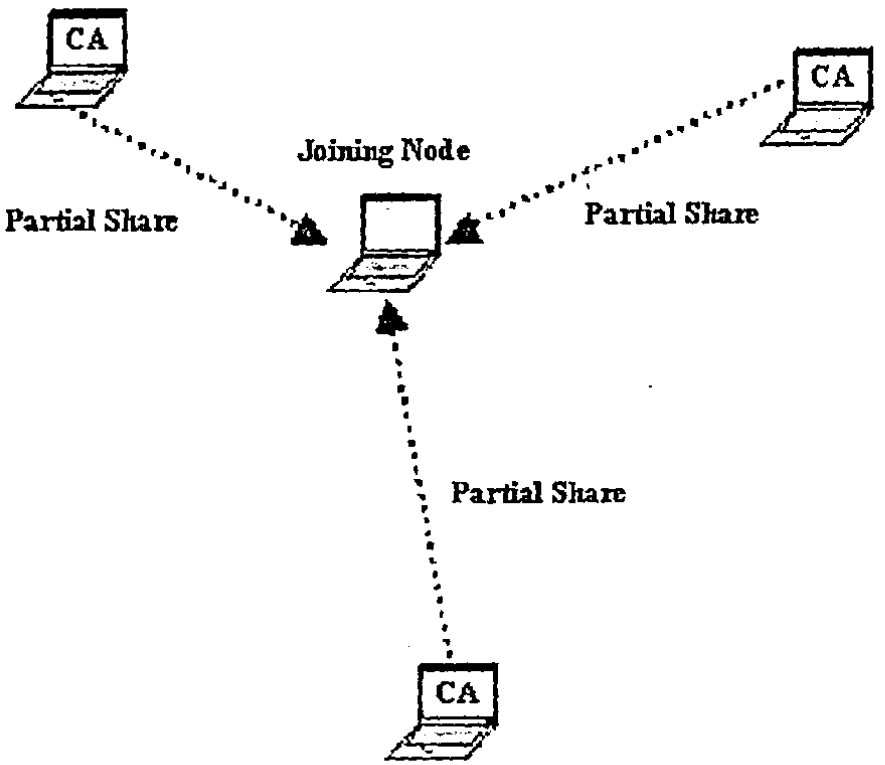


Figure 2.5 Share initialization during operational phase, due to joining node. Those nodes already part of the CA service generates a new share from their shares and initialize the new node.

2.2.1.3 Share Update

Proactive secret sharing is required to protect against attackers that (given enough time) can compromise k or more nodes and thereby be able to reconstruct the shared secret, in this case the certificate signing key sk_{CA}. As illustrated in figure 5.5 the lifetime of the network is divided up in time periods, where each time period consists of two phases, the operational phase and the share update phase. During the operational phase nodes can renew their certificate and request share initialization. During the share update phase all nodes that have been initialized update their shares in a distributed manner.

During the share update phase the following three steps are performed:

- Collaborative generation of the update polynomial $f_{update}(x)$.
- Distribution of the update polynomial to all network nodes.
- Distributed evaluation of the share update of all nodes p.

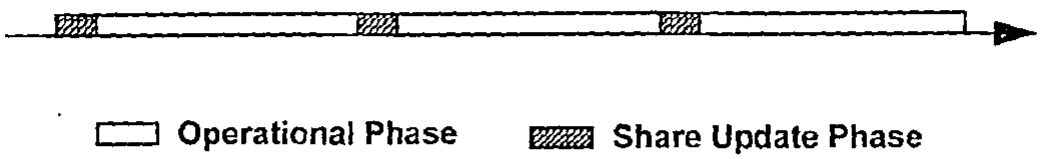


Figure 2.6 Different phases during the lifetime of the ad hoc network

At the beginning of the share update phase each node initiates the update process with the probability where is a known estimate of the total number of nodes in the network. The node that decides to initiate the share update then locates a coalition of k nodes that generate the update polynomial. Each of the polynomial's coefficients is then encrypted, signed, and flooded in the network. At this point every node in the network has received which it has authenticated by verifying the signatures. This prevents an attacker from being able to flood a false update polynomial.

Each node p in the network can now generate its update share; however since the polynomial is encrypted it must request the evaluation of its update share from a coalition of k nodes. Each of the nodes in the coalition returns a partial update share to the requesting node p who adds them to obtain its complete update share. Adding this update share to the nodes current share provides the updated share of sk_{CA}.

During the share update phase two different versions of shares can be present since not all nodes update their shares simultaneously. However, as long as the nodes in the coalition use the same version of their shares, the services required can be performed. Therefore the nodes keep the old share until the share update phase is complete. At this point they destroy the old share and thereafter only use the updated share until the next share update phase.

2.2.2 Certificate Issuing

This solution is based on the assumption that all nodes have been initialized, registered, and issued a valid certificate before they join the network. The distributed CA never issues new certificates; it only manages certificates once they have been initially created. The responsibility of initializing, registering, and certifying new nodes belongs to the administrative authority responsible for setting up the network, i.e. the dealer.

2.2.3 Certificate Renewal

Since certificates are only valid for a limited time period they must be renewed before they expire. When a node p wishes to renew its certificate *cert* it requests a certificate renewal from a coalition of k of its one-hop neighbors. Each node i in the coalition then first checks that the old certificate has not already expired and that it has

not been revoked. If they agree to serve the request they each generate a new partial certificate and returns it to the requesting node *p*. Node *p* then combines the *k* partial certificates to obtain its updated certificate *certupdated*.

2.2.4 Certificate Revocation

The certificate revocation mechanism is based on the assumption that all nodes monitor the behavior of their one-hop neighbors and maintain their own certificate revocation lists. If a node discovers that one of its neighbors is misbehaving it adds its certificate to the CRL and floods an accusation against the node. Any node receiving such an accusation first checks its CRL to verify that the accusation didn't originate from a node whose certificate has been revoked. If the accuser's certificate has been revoked the accusation is ignored. However, if the accusation originated from a valid node, the accused node is marked as suspect. When thresholds of legitimate accusations, i.e. k accusations, against the same node are received the accused node's certificate is revoked.

2.3 Self Issued Certificates

This solution, like PGP, deals with the problem of distributing public keys in an authenticated manner. Unlike traditional PKI solutions, in PGP the public keys aren't certified by some trusted third party, e.g. a CA. Instead each user has the capability of certifying the public keys of other users. It is then up to each user to determine how much trust to place in a specific certificate.

In PGP, public key servers, i.e. certificate directories are used to distribute certificates; however in ad hoc networks no such servers are available and therefore the solution proposed by Hubaux et al relies on the users to distribute and store the certificates themselves. Each user stores a small number of certificates that have been issued. When two users wish to authenticate each others' public keys, they try to find a certificate chain using only the certificates stored in their combined local certificate repositories.

The certificate selection algorithm proposed by the authors, the Shortcut Hunter algorithm is based on a phenomenon known as the small-world phenomenon and it only provides a probabilistic guarantee of obtaining a certificate chain as described above.

2.4 Secure Pebblenets

This solution proposed by Basagni et al [17] provides a distributed key management system based on symmetric encryption. The solution provides group authentication, message integrity and confidentiality. This solution is suitable for planned and distributed, long-term ad hoc networks consisting of low performance nodes that are unable to perform public key encryption, e.g. sensor networks.

All network nodes share a secret group identity key k_{GI} which is used to provide authentication and to derive additional keys used to provide confidentiality. The group identity key's crypto-period lasts for the whole duration of the network while the keys used to provide confidentiality are updated on regular intervals.

The network's lifetime is divided up in time periods, each consisting of three phases as illustrated in figure 2.7. During the operational phase the nodes use the group identity key k_{GI} to provide authentication and message integrity, and a traffic encryption key k_{TEK} to provide confidentiality.

To enable the distributed update of the traffic encryption key, the network is segmented into clusters during the cluster generation phase. Each cluster has a node designated as the cluster head. During the key update phase one of the cluster heads is elected key manager and is responsible for generating a new traffic encryption key and distributing it to the other cluster heads. Each of the cluster heads then distributes the new traffic encryption key to its cluster members.

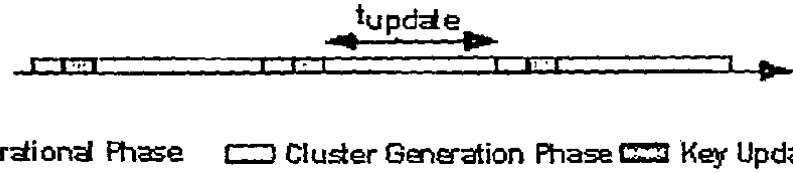


Figure 2.7 The different phases during the network lifetime.

2.4.1 System Requirements

This section specifies the system requirements that are either explicitly mentioned by the authors or that are implied. The solution is intended to be used in large ad hoc networks consisting of nodes with limited processing, storage and power resources. Therefore public key cryptography is not feasible. Due to the complexity involved in key distribution, authentication is limited to group membership since this only requires that all nodes in the network share one cryptographic key. If authentication of individual nodes is required, n . (n – 1) /2 symmetric keys need to be managed.

The network nodes have tamper-resistant storage where the group identity key k_{GI} is stored. This prevents any attacker from being able to gain access to it by theft or other means. It is also assumed that all nodes are honest and that no malicious nodes are present in the network. Since the network lifetime consists of a number of time periods during which the nodes update the encryption key, the nodes must maintain a loosely synchronized clock.

2.4.2 Cluster Generation Phase

During the cluster generation phase each node decides, depending on its weight w and the weights of its one hop neighbors, if it is to be a cluster head or an ordinary cluster member. The cluster heads have been selected, they must discover each other and setup a cluster head backbone which subsequently will be used to distribute the new traffic encryption key.

The whole cluster generation phase consists of three main steps:

- Neighbor discovery
- Cluster head selection
- Cluster head backbone creation

2.4.3 Key Update Phase

During the key update phase one of the cluster heads selected in the previous phase will become the key manager and generate a new traffic encryption key that will be distributed among the other cluster heads.

To select a key manager all cluster heads first decide whether they are a potential key manager. This is done by checking whether any of its neighboring cluster heads has a higher weight. If not, the node decides it is a potential key manager. After an exponential delay with the average, the potential key manager generates a new traffic encryption key which is subsequently distributed to all other cluster heads over the cluster backbone. If another potential key manager receives this message it will do one of the following:

- If it has not already generated a new traffic encryption key it will cancel its timer and accept the message.
- If it has generated a new traffic encryption key it will discard the received key if it has a higher weight than the cluster head that has generated the received key.
- If it has generated a new traffic encryption key but has a lower weight than the cluster head that generated the received key will accept the received key as the new traffic encryption key.

The new traffic encryption key is then distributed by each cluster head to its cluster members.

2.5 Password Authenticated Key Exchange

This scheme describes key agreement protocols that can be used to secure the communication in a collaborative networking scenario. A number of such group key agreement protocols, also known as conference keying protocols, have been devised. However, the majority of them require that the group members are authenticated prior to the execution of the protocol, e.g. using certificates.

Asokan and Ginzborg [19] present a password authenticated group key agreement protocol that is an extension to a previous group key agreement protocol called the Hypercube protocol. The Hypercube protocol is based on the Diffie-Hellman key agreement protocol described in section.

The extensions to the Hypercube protocol described in [19] include password authentication and improved fault tolerance. The focus of this description is on the password authentication extension. Asokan and Ginzborg consider a collaborative networking scenario where a group of people wish to set up a secure wireless network during a meeting. The meeting members have no means of authenticating the other members using e.g. digital certificates. Therefore a simple password is chosen and e.g. written on a whiteboard. Using this weak password the members can engage in the password authenticated Hypercube protocol which results in them sharing a strong secret. The key agreement protocol needs to be authenticated to protect against active attacks, e.g. man-in-the-middle attacks. In [20] a number of group key establishment protocols are compared along with the Hypercube protocol. However, neither of these protocols provide authentication.

2.5.1 The Hypercube Protocol

The nodes participating in the protocol are arranged as the vertices in a d-dimensional cube, a hypercube. The protocol then consists of d rounds of two-party DH key exchange. During each round $j = 1,..., d$ a node performs the two-party key exchange with its neighbor in the j:th dimension. In the first round each node i uses his own secret x_i as the exponent. In the following rounds the key obtained from the previous round is used as the secret exponent.

The protocol can be generalized to 2^d group members and the members are then arranged in a d-dimensional cube and the same steps as described above are performed. The authors of the original Hypercube protocol also propose a protocol called the 2^d Octopus protocol that can handle the case when the number of group members isn't an even power of two, see [21] for more details. The extensions to the Hypercube protocol presented by Asokan and Ginzborg [19] also handle the case where the number of group members is not an even power of 2.

2.5.2 Password Authentication Extension

A protocol known as Encrypted Key Exchange (EKE) provides a two-party password authenticated DH key exchange. The two parties wishing to exchange a strong cryptographic key agree on a simple password *p*. Using this password they encrypt their public DH values and send to other party. The receiving node can then decrypt the public value and calculate the shared secret *k*.

After receiving the message in step 1, *B* can extract and calculate the shared key *k* as $k = (g^{X}{}_{A})^{X}{}_{B} \bmod p$. Likewise *A* can calculate the shared key after receiving the message sent in step 2. To verify that both parties share the same view of the shared key, i.e. that both know the simple password *p* and could decrypt the messages in steps 1 and 2, they exchange challenges c_A and c_B. Asokan and Ginzborg simply propose using EKE instead of the basic DH key exchange in each round of the Hypercube protocol.

2.6 Limitations of Existing Security models

Scope and limitation of every model is described one by one in the following sections.

2.6.1 Scope of Password-Based key agreement

Password-Based key agreement model perfectly works for small groups. Authentication is done outside the IT system, e.g. the group members authenticate themselves by showing their passport or common knowledge. This model does not suffice anymore for more complicated environments, though. Groups of people who do not know each other or number of people who want to have confidential exchanges without bringing in knowledge of the rest of the group be able to eavesdrop on the channel, are two examples. Another problem arises for large groups or groups at different locations. The secure channel to distribute the initial password is not available anymore. It seems that existing support infrastructure is required to set up a secure channel.

This is a group-oriented solution since it doesn't allow for authentication of individual nodes. An issue that isn't mentioned is that of joining and leaving nodes. Consider as an example a meeting where all but one of the meeting members start the meeting and perform the key exchange. When the missing member arrives he needs to join the network. A mechanism for providing him with the encryption key or generating a new encryption key must be present. Similarly a member that leaves a meeting should not be able to eavesdrop from outside the meeting room. A method for updating the encryption key upon the departure of a member should also be available.

The Hypercube protocol used assumes that the participating nodes are arranged in a hypercube. A mechanism must also be provided to decide upon this internal ordering of the nodes.

2.6.2 Scope of Resurrecting Duckling scheme

The Resurrecting Duckling scheme is an appropriate model for a well-defined hierarchy of trust relationships. It particularly suits cheap devices that do not need a display or a processor to perform public-key operations. It perfectly works for a set of home devices.

However, more flexible ad-hoc networks may not contain explicit trust relationships between each pair of nodes or to a centralized entity like the mother duck. Deploying a comprehensive network consisting of a hierarchy of a global mother duck and multiple subsidiary local mother duck is very similar to a public-key infrastructure, where the mother duck correspond to Certification Authority (CA), with all its advantages and drawbacks.

Even the battlefield scenario raises some problems. Here the soldiers are siblings and obey their mother, the general. If one soldier device wants to authenticate to another device it has to present its credentials. The second device can then check the Credentials by using its policy. But what happens if all soldiers' do not use the same credentials, i.e. the same secret key to prevent it to be stolen by the enemy. If all devices use the same key the other side might invest considerable effort doing some physical attack [9] to recover the key because it would compromise all nodes. Since the devices cannot hold a list of all valid credentials it seems that a further authentication method is needed.

2.6.3 Scope of Partially Distributed Key Management

This solution requires that a server- and organizational/administrative infrastructure is available and therefore is only applicable to a subset of ad hoc network applications. Viewed from a functional standpoint the solution has a number of faults or weaknesses of which the lack of a certificate revocation mechanism is the most critical. Any solution based on certificates should, considering the risk of compromise in ad hoc networks, provide such a mechanism.

Also the solution requires that the server nodes store all of the certificates issued. This requires a synchronization mechanism that propagates any new certificates to all the servers. It also must handle the case when the network has been segmented and later re-joined. Consider the scenario illustrated in figure 2.8. In a) all the nodes are interconnected in the same ad hoc network and keeping the certificate repositories synchronized is no problem. Later in b) the network is segmented into two ad hoc networks and in each of these, subsequent certificate renewals occurs. In c) when the two network segments re-join the server nodes will have inconsistent certificate repositories which require synchronization.

Another issue that arises in this scenario is what happens if the server nodes in the two segments have performed a share update. The servers then must synchronize their shares so that they have a consistent view of the shares of the CA's certificate signing key

sk_{CA}. The availability of the CA service is highly dependent on the threshold parameters k and n. These must therefore be chosen carefully to obtain an acceptable trade-off between availability, security, and cost. The larger k is, the higher security is achieved at the cost of availability.

An issue related to addressing is how client nodes know how to locate the server nodes. Since the client node is interested in locating any k of the n server nodes, the CA service could be given a multicast address. A client node that requires a CA service could then send the request to this multicast address. Those server nodes that choose to serve the request then reply with their unicast address. Another option if the ad hoc network doesn't support multicast traffic is that the client node broadcasts the request. This approach however will potentially generate a large amount of network traffic.

2.6.4 Scope of Fully Distributed CA Policy

Similar to the partially distributed CA this solution requires an organizational/administrative infrastructure to provide the registration and initialization services. The main benefit of this solution is its availability and that it, unlike the other certificate based solution proposed, provides a certificate revocation mechanism. Since all nodes are part of the CA service, it is sufficient that a requesting node has k one-hop neighbors for the CA service to be available. Also since only a node's one-hop neighborhood is involved in any service request the addressing issue mentioned in the previous section is avoided. The amount of network wide traffic is also limited.

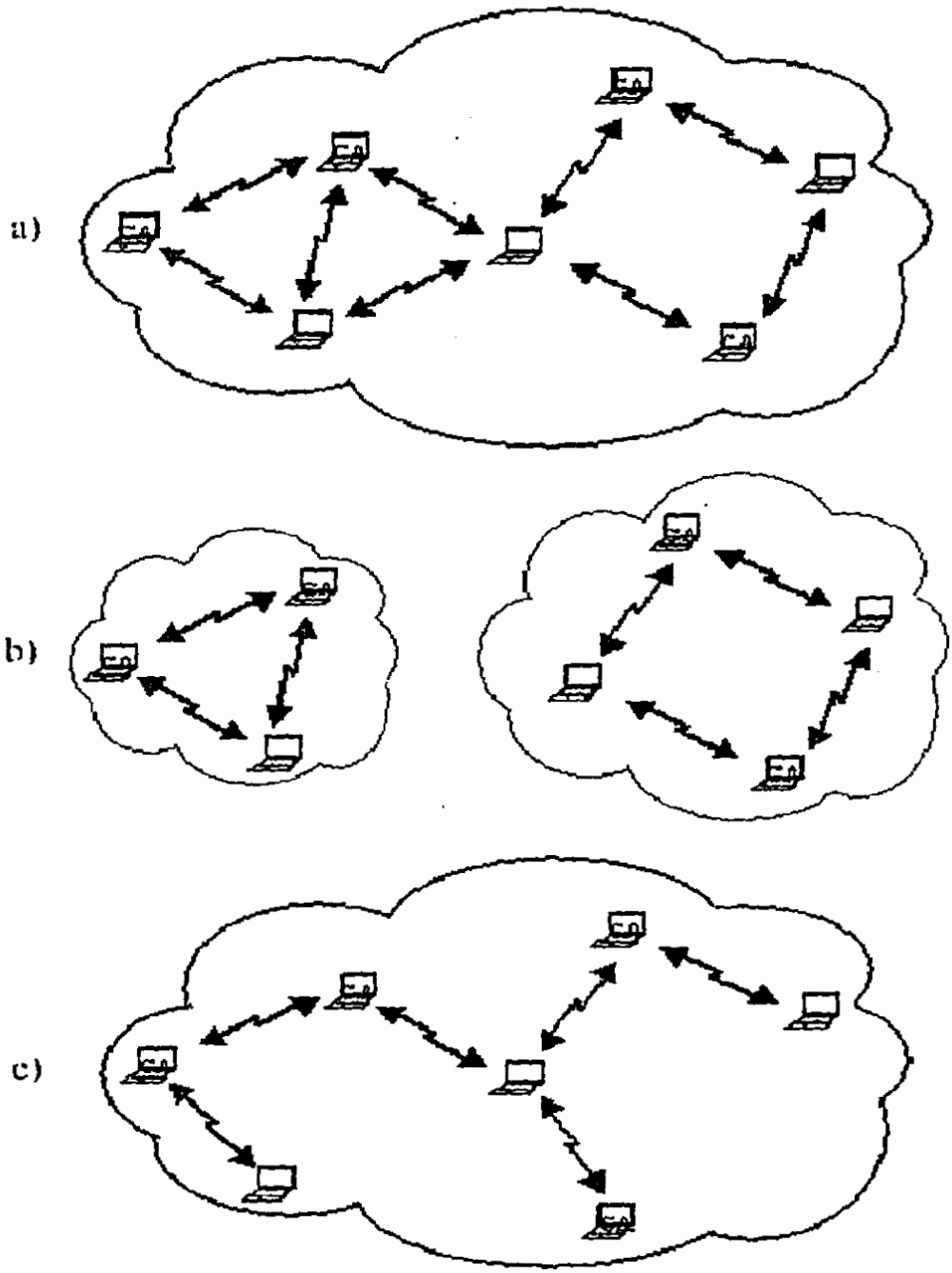


Figure 2.8 Synchronization problem due to segmentation.

The cost of achieving this high availability is a set of rather complex maintenance protocols, e.g. the share initialization and the share update protocols. A larger number of shares are also exposed to compromise since each node has its own share as compared to

only the specialized server nodes in the partially distributed solution. The threshold parameter k therefore may need to be chosen larger since an attacker may be able to compromise a larger number of shares between each share update. This in turn affects the availability of the service. The solution must also provide for a synchronization mechanism in the case of network segmentations as illustrated in figure 2.8.

The certificate revocation method proposed assumes that each node is capable of monitoring the behavior of all its one-hop neighbors, e.g. by monitoring the network traffic they generate. This assumption however may be too strong in certain ad hoc networks.

2.6.5 Scope of Self Issued Certificates Policy

The main benefit of this solution is that it doesn't require any form of infrastructure neither routing, server or organizational/administrative. However it lacks a certificate revocation mechanism. Also like PGP it has problems during its initial stages before the number of certificates issued reaches a critical amount. Consider the situation illustrated in figure 2.9. There is a "chain of friendship" connecting Alice and Bob, however so far only Chris and Jane have issued any certificates and therefore no certificate chain exists between Alice and Bob. Not until the other users have issued certificates to their friends will Alice and Bob be able to communicate securely.

This solution also assumes that all users are, what in PGP terminology is called trusted introducers or even meta-introducers. A trusted introducer is a user that is trusted to introduce other users, i.e. issue certificates to other users. A meta-introducer is a trusted introducer that is trusted to introduce other trusted introducers e.g. in figure 2.9, Chris may issue a certificate to Dave, but he doesn't trust any certificates issued by Dave. On the other hand he trusts any certificates issued by Trent. This makes Trent a trusted introducer. If Chris even trusts the certificates issued by users to which Trent has issued certificates, e.g. Jane, then Trent is a meta-introducer. Requiring that all users must accept all other users as meta-introducers is a strong requirement that most probably will not be acceptable in many ad hoc networks.

Finally the Shortcut Hunter algorithm has been tested on authentic PGP trust graphs, i.e. a database of PGP certificates available on Internet. However in an ad hoc network no such global knowledge of all certificates will be available e.g. in step 4 of the algorithm, the edge *(w, z)* is chosen such that the vertex *z* has the highest number of shortcuts. Determining how many shortcuts a node has however requires knowledge of the surrounding trust graph. The solution doesn't mention this issue which needs to be dealt with if the solution is to function in a real life ad hoc network.

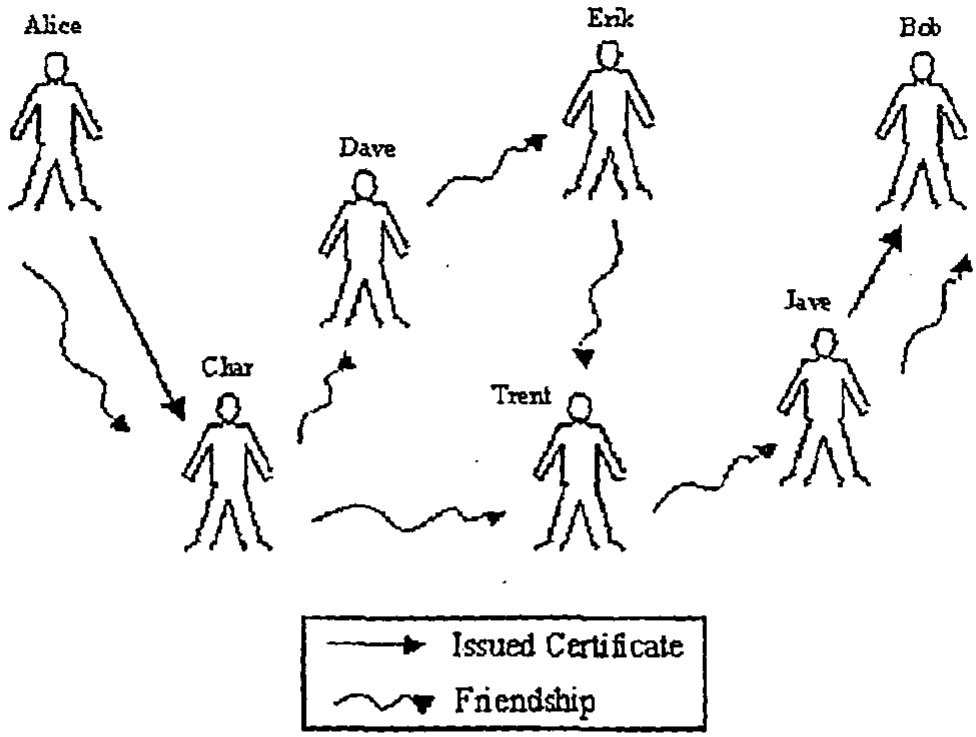


Figure 2.9 Problem of certificate propagation.

2.6.6 Scope of Secure Pebblenets Policy

This solution based on symmetric cryptography requires an organizational/administrative infrastructure that initializes the network nodes with the shared group identity key k_{GI} and additional parameters such as t_{update}.

The main weakness of this solution is that it requires that the nodes maintain a tamper-resistant storage. Such a requirement excludes the use of standard networking devices since these typically don't include any tamper-resistant memory. If the group identity key is compromised then all the network nodes need to be re-initialized with a new group identity key .Finally since only group authentication is supported this solution is not applicable in applications where the communication is peer-to-peer.

Chapter 3

Problem Definition

3. Problem Definition

Ad hoc networking is a wireless networking paradigm for self-organizing networks that until recently has mainly been associated with military battlefield networks. However, with the availability of wireless technologies such as Bluetooth and 802.11 and the development of the next generation networks, civilian applications that exploit the advantages of ad hoc networking are being envisioned.

So far most of the research that has been done on ad hoc networking has focused on routing. Other issues such as security and network addressing have received considerably less attention and these issues need to be addressed before any successful applications will appear.

In wireless ad hoc networks security to be achieved is difficult because:

- Wireless links are susceptible to passive as well as active attacks mainly due to open media
- Nodes with inadequate physical protection can be compromised
- Sporadic nature of connectivity
- Dynamically changing topology
- Absence of Certification authority
- Lack of centralized monitoring or management point
- Cooperative nature of algorithms
- Sleep deprivation torture

3.1 Threats in Ad hoc networks

The traditional key management solutions presented in chapter 1 have a number of requirements that make them unsuitable or even unusable in ad hoc networks. The use of a trusted third party to provide key management services implies that an organizational or administrative infrastructure is present. Such an infrastructure may be present in certain ad hoc networking applications, e.g. military networks or sensor networks but in many it will not, e.g. collaborative networking applications.

Even in the presence of an organizational/administrative infrastructure, solutions based on the use of a trusted third party face problems. The obvious problem is that they

rely on the availability of a central server. However, due to the lack of routing infrastructure and highly dynamic network topology in ad hoc networks, such availability cannot be assumed. Finally the very existence of a server infrastructure cannot be assumed in certain ad hoc network applications, e.g. in spontaneous ad hoc networks formed when two or more PDA's come within range of each other. There are several threats in ad hoc networks. Ad hoc networks use wireless data transmission. This makes them susceptible to passive eavesdropping, message replaying, message distortion and active impersonation. Mobile nodes have low physical security. They can be easily compromised, for example, in the battlefield. This means that attacks can come also from inside the ad hoc network. Therefore we cannot trust one centralized node, because if this node would be compromised the whole network would be useless. One problem is scalability. Ad hoc networks can have hundreds or even thousands of mobile nodes. This possesses challenges to security mechanisms. The most important threats that mobile ad hoc network have to face are [3]

- Attacks on basic mechanisms
- Attacks on security mechanisms

3.1.1 Attacks on basic mechanisms

These are Attack on basic mechanism of the ad hoc network, such as routing. Prevention of these attacks requires security mechanisms that are often based on cryptographic algorithms. Routing mechanisms are more vulnerable in ad hoc networks than in conventional networks since in ad hoc network each device acts as a relay. This means, that an adversary who hijacks an ad hoc node could paralyze the entire network by disseminating false routing information. A less dramatic but subtler malicious behavior is node selfishness. Moreover, weakness in protocols can be exploited to perform malicious neighbor discovery.

3.1.2 Attacks on security mechanisms

Attack on security mechanisms and especially on the key management mechanism. Key management is certainly not a problem limited to ad hoc networks; however, because of the peculiarities of ad hoc networks, its solution requires specific attention. Examples of attacks on security mechanism are: Public keys can be maliciously replaced; some keys can be compromised; if there is a distributed trusted server, it can fall under the control of a malicious party.

3.2 Complexities associated with securing MANET

In most applications where security matters, authenticity is an essential prerequisite. Granting resources to, obeying an order from, or sending confidential information to a principal of whose identity we are unsure is not the best strategy for protecting availability, integrity and confidentiality.

The ad hoc network environment introduces an advanced security problem: the absence of an online server. When a node comes within range, we cannot connect to an authentication server to obtain a Kerberos ticket or to check the validity of an exhibited certificate: suddenly, the traditional solutions for wired networks no longer apply.

A more serious threat however, and one that complicates the matter of key establishment is that of active attacks. Due to the lack of routing infrastructure nodes participate in the forwarding of messages on behalf of other nodes as described in chapter one. A direct consequence of this is that an attacker can easily modify messages and perform man-in-the-middle attacks. Therefore unauthenticated key establishment protocols such as Diffie-Hellman cannot be used.

Since the nodes of ad hoc networks are often mobile and more exposed than traditional network nodes (e.g. desktop computers) the risk of compromise through theft or loss should generally be considered higher. Therefore any key management solution for ad hoc networks should have a revocation mechanism. This also implies that the straightforward approach of improving the availability of a TTP by distributing it to multiple nodes is inappropriate. The reason for this is that it weakens the security since it exposes more TTP's to attackers. If any of them are compromised the security of the whole network is compromised. In the case where the TTP is a CA a more suitable approach is to use a threshold secret sharing scheme to share the secret among a number of nodes. The compromise of up to *k-1* of these still leaves the security of the service intact.

The threats mentioned above are not the only threats in ad hoc networks and they also exist in traditional networks. However, the effort required to perform the attacks is considerably smaller in ad hoc networks than in traditional networks. Figure 3.1 and figure 3.2 illustrate the difference between a man-in-the-middle attack in a traditional network and an ad hoc network. To perform the attack in the traditional network the attacker must first gain control of the router connecting Alice and Bob before he is able to perform the attack against Alice and Bob's communication.

In an ad hoc network however, Alice and Bob use the attacker (but they don't know he's an attacker) as an intermediate node. The attacker can then easily inserts new messages or delete or modify Alice and Bob's messages. If ad hoc networks become widely used it could even be assumed that programs will appear that allow even novice computer users to perform such active attacks. Such a scenario could be compared with the so called "script kiddies", i.e. virus authors and other attackers that use available programs to automatically create new viruses.

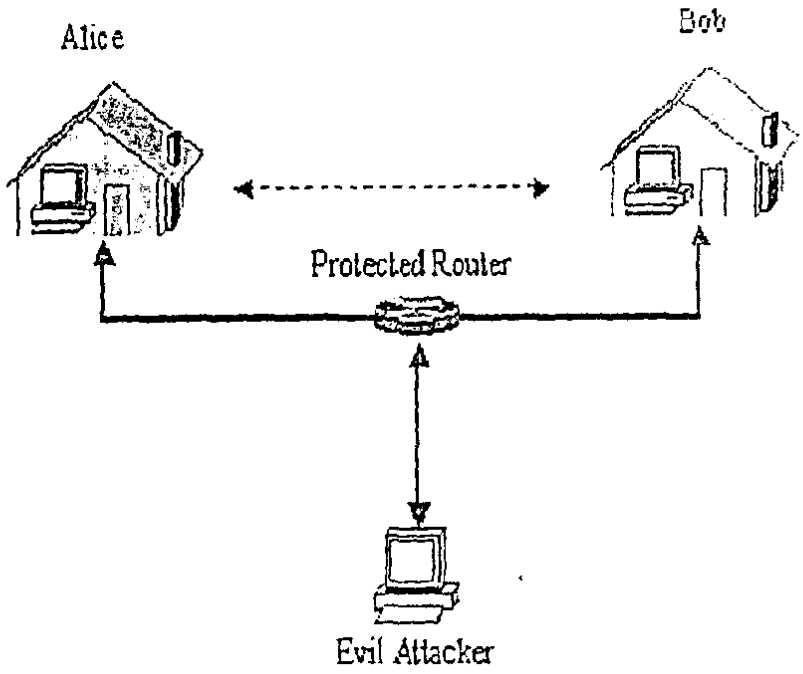


Figure 3.1 Man-in-the-middle attack in a traditional network

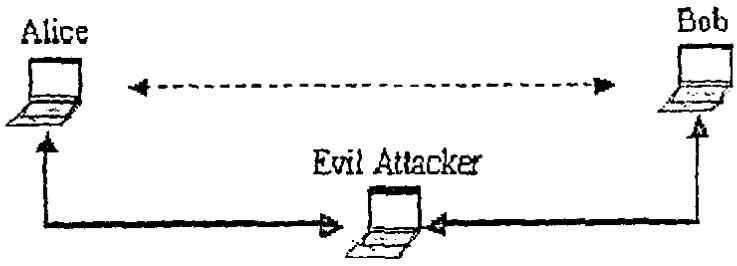


Figure 3.2 Man-in-the-middle attack in an ad hoc network

3.3 Security Issues in WLAN

The following problems are mostly observed in Wireless LAN.

- The default configurations of the wireless "servers" (Access Points) are insecure. They are set to be "open" to make them easy to deploy and use out of the box.

- The physical transport is invisible and therefore it is difficult to understand and control its boundaries.

- There are interoperability issues among Access Points. That is, because the security and configuration features vary by vendor, if you have more than one type of Access Point (even if it's different types of Access Points from the same vendor) you're going to have to understand (in detail) what the compatibility issues are among them.

- Many wireless setups are installed by end-users and not by IT or security professionals.

- The standard data encryption protocol (WEP) that is used on almost every Access Point in the market has been proven to be insecure.

3.4 Summary

Keeping in the view the security issues of MANET and WLAN there should be a protocol which can run on both kinds of networks. Also this protocol should overcome the problems in existing Authentication techniques.

Chapter 4

Proposed system Model

4. Proposed system Model

Our solution is based on demonstrative identification [18]. To establish initial trust, two nodes first exchange authentication data, termed pre-authentication data over a location-limited channel, which is separate from the main wireless links, see figure 4.1. Examples of such channels are infrared, physical contact, audio etc.

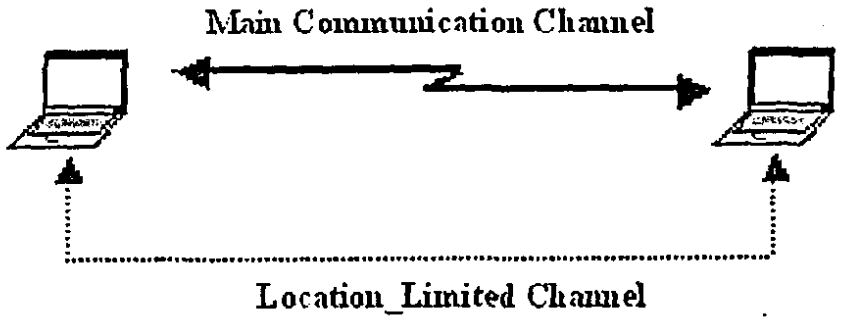


Figure 4.1 Separate location-limited channel between two nodes.

The concept of demonstrative identification can be illustrated with the following example. Consider a user with a PDA who wishes to connect to one of several printers. By using e.g. an infrared channel the user can identify the printer by going up to it and directing the infrared device towards it. Due to the characteristics of infrared communication the user can be assured that the authentication data exchanged originated from the chosen printer.

Once the pre-authentication data has been exchanged, e.g. public keys, any traditional key agreement protocols, e.g. SSL/TLS/Diffie-Hellman can be used over the main communication channel to establish a common key used to protect the communication.

4.1 System Requirements

This solution requires that all network nodes be equipped with the same location-limited communication device, e.g. infrared or audio and it only permits the key-exchange in local ad hoc network, i.e. ad hoc networks where the nodes are in close proximity to each other.

4.2 Preliminaries

Illustration of concepts (introduced above) is given below.

4.2.1 Location-limited channel

The two devices exchange public information during physical contact. We call this physical contact a *location-limited channel*. Location-limited channels have the property that human operators can precisely control which devices are communicating with each other. The notion of location-limited channels was introduced by Stajano and Anderson (although they did not use that name) [18], as a part of their "Resurrecting Duckling" model of interaction in ad-hoc networks. They use secret data exchanged over a contact channel to bootstrap a particular authentication and key exchange protocol ("imprinting" between a "mother" or control device, and a "duckling").

Inspired by Anderson and Stajano [18], we propose bootstrapping secure wireless communication through pre-authentication over a location-limited channel. Location limited channels are separate from the main wireless link, and have special security properties by virtue of the media over which data travels. In order to be used for pre-authentication, a candidate location-limited channel must have two properties.

First, it must support demonstrative identification; that is, identification based on physical context (the printer in front of me, all the PDA's in this room, *etc.*). Communication technologies that have inherent physical limitations in their transmissions are good candidates. For example, audio (both in the audible and ultrasonic range), which has limited transmission range and broadcast characteristics, can be used by a group of PDA's in a room to demonstratively identify each other. For situations that require a single communication endpoint (*e.g.*, the printer across the room), channels with directionality such as infrared are natural candidates. It is these demonstrative properties that allow communication across a location-limited channel to "name" a target device or group of devices.

The second property required of a location-limited channel is *authenticity* – that it is impossible (or difficult) for an attacker to transmit in that channel, or at least to transmit without being detected by the legitimate participants. As we will see below, this property is sufficient to ensure that information exchanged over the location-limited channel will to allow the parties involved to securely authenticate each other over the wireless link, even in the presence of potential attackers.

A third property that was required in previous work is *secrecy* – that the channel be impervious (or resistant) to eavesdropping. For example, Anderson and Stajano [18] use *secret* data, such as a symmetric key, exchanged across the location-limited channel to allow participants to authenticate each other. As a result, that authentication protocol is vulnerable to a passive attacker capable of eavesdropping on the location-limited channel, thereby obtaining the secrets necessary to impersonate one of the legitimate participants. A location-limited channel used to exchange such secret pre-authentication data must therefore be very resistant to eavesdropping.

4.2.2 Pre-authentication

We can divide the "Duckling" protocol of Stajano and Anderson into two parts. In the first part, duckling and mother exchange secret information over a particular location-limited channel). In the second phase, the duckling uses this secret data to recognize and authenticate its mother when she contacts it over the wireless link; the duckling is willing to be controlled by any "mother" that can present the right authentication data. We refer to the first phase as a *pre-authentication* exchange. The data that is exchanged over the location-limited channel during pre-authentication will then be used for subsequent authentication of the parties on the wireless link.

We generalize this idea of *pre-authentication* to secure arbitrary peer-to-peer ad-hoc interactions using a wide variety of key exchange protocols, and provide detailed blueprints for its use. We introduce the use of public key cryptography in this context, and are thereby able to remove the secrecy requirement on location-limited channels used to authenticate key exchange protocols. This allows us to broaden the types of media suitable for use as location-limited channels to include, for example, audio and infrared. More importantly, it allows us to expand the range of key exchange protocols which can be authenticated in this manner to include almost any standard public-key-based protocol. As a result, our approach can be used with an enormous range of devices, protocols, and applications. At the same time, our approach is significantly more secure than previous approaches, as we force an adversary to mount an active attack on the location-limited channel itself in order to successfully subvert an ad-hoc exchange.

If we can remove that requirement that pre-authentication data be secret, and instead only require that it be *authentic*, we can increase our security dramatically. Because legitimate participants would only communicate with entities from whom they had received pre-authentication data, we would now require an attacker to perform an *active* attack – to be able to transmit not only in the main wireless medium, but also in the location-limited channel. Because of the physical limitations of transmission on location-limited channels, it is significantly harder for an attacker to passively eavesdrop on them, not to mention to actively transmit. For such an active attack to succeed, the attacker must not only transmit on the location-limited channel, but must do so without being detected by any legitimate participant. To be effective, such detection does not require that we correctly identify the devices transmitting on the location-limited channel. Instead, it only requires one's ability to count: if you know that both you and your intended communication partner have successfully initiated communication (*e.g.*, the lights on the target device blink, the human using the other laptop indicates the communication attempt was successful), and you (or your proxy device) know that only two participants have attempted to inject messages into the location-limited channel, then you know you must be talking to whom you think you are. If something appears to be wrong, you can simply abort the communication protocol.

The difficulty of monitoring a pre-authentication for such unwanted participation depends on the type of channel used and the number of legitimate parties involved. The more directed the channel and the smaller the number of parties, the easier it is to

monitor. Note that, because of the physical limitations of the channels used and this monitoring requirement, it is only possible to use our techniques to pre-authenticate devices that are physically co-located at the time of first introduction.

We therefore propose that any physically limited channel suitable for demonstrative identification, on which it is difficult to transmit without being detected by at least one legitimate participant (human or device), is a candidate for use as a pre-authentication channel. Such candidates include: contact, infrared, near-field signaling across the body (see [20]), and sound (both audible [16] and ultrasound).

The amount of data exchanged across the pre-authentication channel is only a small fraction of that sent across the main wireless link, and so we can use channel media capable only of low data rates.

4.3 The Basic Protocol

For two nodes A and B capable of public key cryptography to exchange a secret encryption key the following steps are performed:

- Using the location-limited channel node A sends $Hash(pk_A)$ to node B and node B sends $Hash(pk_B)$ to node A.
- Switching over to the main wireless channel the nodes now exchange their public keys. The received keys pk_A * and pk_B * are the authenticated by verifying $Hash(pk_A^*) = Hash(pk_A)$ and $Hash(pk_B^*) = Hash(pk_B)$.
- If the verification was successful any public key based key-exchange protocols, e.g. SSL/TLS/Diffie-Hellman can now be used to exchange a shared secret.

4.4 Usage of Elliptic curve cryptography

ECC is a public key cryptography technology commonly applied in wireless applications because its small size and high speed are well suited to low-end CPUs and mobile appliances. Elliptic curve cryptography (ECC) takes a kind of mathematical shortcut to generate encryption and decryption keys that are significantly smaller than those generated by other methodologies, such as the commonly used RSA algorithm, but just as secure. The algorithm makes all the difference with bandwidth-constrained wireless networks, such as data link technology used between systems in an aircraft and on the ground.

ECC offers the smallest key size and the highest strength per bit of any known public-key cryptosystem. This stems from the discrete logarithm problem in the group of points over an elliptic curve. Among the different fields that can underline elliptic curves,

integer fields F(p) and binary polynomial fields GF(2m) have shown to be best suited for cryptographically applications. In particular, binary polynomial fields allow for fast computation in both software and hardware implementations.

Small key sizes and computational efficiency of both public- and private-key operations make ECC not only applicable to hosts executing secure protocols over wired networks, but also to small wireless devices such as cell phones, PDA's and Smartcards. To make ECC commercially viable, its integration into secure protocols needs to be standardized. As an emerging alternative to RSA, the US government has adopted ECC for the Elliptic Curve Digital Signature Algorithm (ECDSA) and specified named curves for key sizes of 163, 233, 283, 409 and 571 bit. Additional curves for commercial use were recommended by the Standards for Efficient Cryptography Group (SECG) . However, only few ECC-enabled protocols have been deployed in commercial applications to date.

Today's dominant secure Internet protocols such as SSL and IPsec rely on RSA and the Diffie-Hellman key exchange. (ECC), offers the highest strength per bit of any knows public-key cryptosystem today. ECC not only uses small keys for equivalent strength compared to traditional public key cryptosystems like RSA, the key size disparity grows a security needs increase. This makes it especially attractive for constrained wireless devices because smaller keys result in power, bandwidth and computational savings.

Most Internet security protocols (e.g. SSL, IPsec) employ a public-key cryptosystem to derive symmetric-keys and then use fast symmetric-key algorithms to ensure confidentiality, integrity and source authentication of bulk data. RSA is the most commonly used public-key cryptosystem today. The security of a system is only as good as that of its weakest component; for this reason, the work factor needed to break a symmetric key must match that needed to break the public-key cryptosystem used for key establishment. Due to expected advances in cryptanalysis and increases in computing power available to an adversary, both symmetric and public-key sizes must grow over time to offer acceptable security for a fixed protection life span. However, the true benefit and performance impact of any cryptosystem is closely tied to how it is used within a security protocol. In particular, it is imperative that expected performance improvements at the protocol level be carefully weighed against the usual costs associated with deploying any new technology.

At the foundation of every public key cryptosystem is a hard mathematical problem that is computationally infeasible to solve. For instance, RSA and Diffie-Hellman rely on the hardness of integer factorization and the discrete logarithm problem respectively. Unlike these cryptosystems which operate over integer fields, the Elliptic Curve Cryptosystems (ECC) operates over points on an elliptic curve.

The fundamental mathematical operation in RSA and Diffie-Hellman is modular integer exponentiation. However, the core of elliptic curve arithmetic is an operation called scalar point multiplication, which computes Q = kP (a point P multiplied k times

resulting in another point Q on the curve). Scalar multiplication is performed through a combination of point-additions (which add two distinct points together) and point-doublings (which add two copies of a point together).

The security of ECC relies on the hardness of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), which states that given P and Q = kP, it is hard to fined k. While a brute-force approach is to compute all multiples of P until Q is found, k would be so large in a real cryptographic application that it would be infeasible to determine k in this way.

Besides the curve equation, an important elliptic curve parameter is the base point, G, which is fixed for each curve. In the Elliptic Curve Cryptosystem, the large random integer k is kept private and forms the secret key, while the result Q of multiplying the private key k with the curve's base point G serves as the corresponding public key. Not every elliptic curve offers strong security properties and for some curves the ECDLP may be solved efficiently.

Elliptic Curve Diffie-Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA) are the Elliptic Curve counterparts of the Diffie-Hellman key exchange and Digital Signature Algorithm, respectively.

In ECDH key agreement, two communicating parties A and B agree to use the same curve parameters. They generate their private keys, kA and kB and corresponding public keys QA = kA:G and QB = kB:G. The parties exchange their public keys. Finally each multiplies its private key and the other's public key to arrive at a common shared secret kA:QB = kB:QA = kA:KB:G.

4.5 Use of SSL as key exchange protocol

Secure Sockets Layer is the most widely deployed and used security protocol on the Internet today. The protocol has withstood years of scrutiny by the security community and is now trusted to secure virtually all sensitive web-based applications ranging from online banking and stock trading to e-commerce.

SSL offers encryption, source authentication and integrity protection for data exchanged over insecure, public networks. It operates above a reliable transport service like TCP and has the exibility to accommodate different cryptographic algorithms for key agreement, encryption and hashing. However, the specification does recommend particular combinations of these algorithms, called cipher suites, which have well-understood security properties. For example, a cipher suite such as RSA-RC4-SHA would indicate RSA as the key exchange mechanism, RC4 for bulk encryption, and SHA for hashing.

Public-key cryptographic operations are the most computationally expensive portion of SSL processing. SSL allows the re-use of a previously established master

secret resulting in an abbreviated handshake that does not involve any public-key cryptography, and requires fewer and shorter messages. However, a client and server must perform a full handshake on their first interaction. Moreover, practical issues such as server load, limited session cache and naïve load balancers can adversely impact the ability to use an abbreviated handshake. Therefore, speeding up the public-key operations in SSL still remains a very active area for research and development.

The two main components of SSL are the Handshake protocol and the Record Layer protocol. The Handshake protocol allows an SSL client and server to negotiate a common cipher suite, authenticate each other, and establish a shared master secret using public-key cryptographic algorithms. The Record Layer derives symmetric-keys from the master secret and uses them with faster symmetric-key algorithms for bulk encryption and authentication of application data.

Public-key cryptographic operations are the most computationally expensive portion of SSL processing. SSL allows the re-use of a previously established master secret resulting in an abbreviated handshake that does not involve any public-key cryptography, and requires fewer and shorter messages. However, a client and server must perform a full handshake on their first interaction. Moreover, practical issues such as server load, limited session cache and naïve load balancers can adversely impact the ability to use an abbreviated handshake. Therefore, speeding up the public-key operations in SSL still remains a very active area for research and development.

Client authentication is optional. Only the server is typically authenticated at the SSL layer and client authentication is achieved at the application layer, e.g. through the use of passwords sent over an SSL-protected channel. However, some deployment scenarios do require stronger client authentication through certificates.

4.6 Deploying Proposed MANET Protocol in Banking

The basic purpose of this project is to test the proposed protocol but we have kept in mind that to motivate the development of ad hoc networking protocols, there needs to be applications where the properties of ad hoc networking are beneficial. This is sole purpose of selecting banking environment involving 802.11 or Bluetooth technology. Also it is possible that we can test our protocol in Wireless LAN as well as in ad hoc mode which will set base point in removing security issues of WEP Actually wireless banking can be adopted using concept of wireless LAN using 802.11 wireless mode but wireless LAN itself introduces security problems. So introducing ad hoc mode in banking will open new direction of research. The following problems are mostly observed in Wireless LAN.

- The default configurations of the wireless "servers" (Access Points) are insecure. They are set to be "open" to make them easy to deploy and use out of the box.

- The physical transport is invisible and therefore it is difficult to understand and control its boundaries.

- There are interoperability issues among Access Points. That is, because the security and configuration features vary by vendor, if you have more than one type of Access Point (even if it's different types of Access Points from the same vendor) you're going to have to understand (in detail) what the compatibility issues are among them.

- Many wireless setups are installed by end-users and not by IT or security professionals.

- The standard data encryption protocol (WEP) that is used on almost every Access Point in the market has been proven to be insecure.

Keeping in view these points we have selected banking scenario for implementation of our security protocol for MANET. Here is brief description of our scenario.

Imagine a situation in which bank customers are standing in a queue and two or three bank employees are dealing them in making transactions. The customers have to spend a lot of time and also have to wait for their turn in the queue. The bank needs more employees for giving better service to its customers. This scenario, both for the bank and the customers, can be improved by deploying ad hoc networking using Resurrecting Duckling Policy, with some extensions in our idea. Here, we have also taken inspiration from work of D. Balfanz.

Our idea is elaborated as: A user (Bank customer) enters the bank, with his laptop or PDA, to perform some transactions. The bank has mounted the Infrared Bar Code at one side in the bank and also owns a wireless radio link network (e.g. Bluetooth or 802.11). The user or customer enters the bank premises, would walk up to the Infrared bar code and briefly establish physical contact between infrared bar code and his laptop or PDA. This process is termed as Pre-authentication[1], which is done using a link-constrained channel. During Pre-authentication, their hash of public keys will be exchanged. Now the user can sit in the bank premises and his laptop or PDA can then perform standard SSL/TLS key exchange with the bank server over the wireless link (e.g. Bluetooth or 802.11), since he owns a secret key to perform secure transactions and can establish an authenticated and secret communication channel.

Such an exchange of pre-authenticated data ensures the user that he wants to communicate with the device (infrared bar code) by using a special, link-constrained side channel to exchange a small amount of cryptographic information. That information can be used to authenticate standard key exchange protocols performed over wireless link.

[1] First phase in which duckling and mother exchange some secret information over location-limited channel.

Once the pre-authentication is completed, the devices proceed to establish a secure connection between them over the main wireless link. To this end, they can use any established public-key-based key exchange protocol which requires them to prove possession of a particular private key (e.g., SSL/TLS, SKEME IKE, etc.), which will correspond to the public key committed to in the pre-authentication step.

After secret key exchange, the customer or user will send his/her account number along with the password or secret code (which can be issued at the time of opening an account in the bank and can be changed at any time through bank website) to the bank server. The A/C No. plus code or password will be encrypted with secret key (Session key) exchanged during pre-authentication. After this phase, the user can perform his transactions securely over the wireless link (e.g. Bluetooth or 802.11).

4.6.1 Using ECC in SSL and its implementation in Banking

In our approach, we have integrated ECC into the Secure Socket Layer Protocol. We chose SSL because it is the most popular and trusted security protocol on the Web. In the form of HTTPS (HTTP secured using SSL), SSL is single handedly responsible for the widespread adoption of e-commerce and many emerging wireless devices too now have SSL capabilities.

We have added ECC support to OpenSSL[27], the most widely used open source implementation of SSL. We have used the OpenSSL0.9.6b cryptographic library for our implementation. We selected ECC in place of RSA because ECC offers the highest strength per bit of any known public-key cryptosystem. ECC not only uses small keys for equivalent strength compared to traditional public key cryptosystems like RSA, the key size disparity grows a security needs increase. This makes it especially attractive for constrained wireless devices because smaller keys result in power, bandwidth and computational savings.

4.6.2 Possible Wireless Banking Services using Ad hoc mode

Wireless Banking is the ability to use a wireless device (web-enabled cellular phone or personal digital assistant) to access your financial information. "Wireless" means that you do not have to be physically connected to the web, either through a network connection or "wired" modem connection.

Following services are implemented in which security is main goal. Implementation involves my own research work of Authentication in Mobile ad hoc Network.

- Transfer money to another customer's account.
- Obtain a summary of your deposit and VISA account balances.

- Review a list of up to 10 of your account's most recent transactions.
- Pay bills immediately.
- Transfer funds between your deposit and VISA accounts.
- Transfer funds between your brokerage and banking accounts.
- Issuance of Secure Credit Card Tickets which can be used only once.

4.7 Architecture of Proposed Model

Steps involved in developing secure Wireless Banking are:

- Pre-Authentication.
- Post-Authentication.
- Account Validation.
- Usage of Bank Services.

4.7.1 Pre-Authentication

We can summarize the basic scheme for pre-authentication as follows.

A) Pre-authentication over location-limited channel.

- X ⟶ Y: H (KU_x)
- Y ⟶ X: H (KU_Y)

B) Authentication over wireless channel with SSL/TLS.

X ⟶ Z: TLS_CLIENT_HELLO
..... And so on.

The various symbols denote:

X: Customer's Laptop

Y: Bar code device

Z: 802.11 node mounted in Bank

KUx, KU_Y: Public key belonging to X and Y respectively

H (KUx), H (KUY): one-way hash of encoding of corresponding keys.

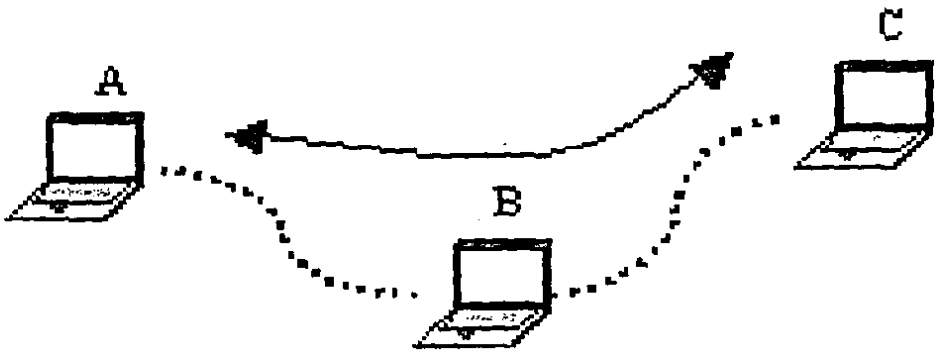


Figure 4.2 shows Pre-Authentication [dotted line] between two nodes

4.7.2 Post-Authentication

Then actual key will be exchanged on Wireless channel using ECC in SSL. When Pre-authentication and Post authentication are completed then transaction will be encrypted using ECC.

4.7.3 Account Validation

Account id and Password are encrypted and are used to transfer over main wireless link between 802.11 enabled devices.

4.8 Advantages of our Approach

This idea of pre-authentication has been generalized to secure arbitrary peer-to-peer ad hoc interactions using a wide variety of key exchange protocols. We have introduced the use of Elliptic Curve Cryptography and can remove the secrecy requirements on link-constrained channels used to authenticate key exchange protocols. More importantly, it allows us to expand the range of key-exchange protocols, which can be authenticated in this manner to include almost any standard public-key-based protocol. As a result, this approach can also be used with an enormous range of devices, protocols and applications (as one of them is discussed).

This approach is significantly more secure than previous approaches; as we force an adversary to mount an active attack on the location-constrained channel itself in order to successfully subvert an ad-hoc exchange. Previous approaches (e.g., use of unauthenticated Diffie-Hellman key exchange) are either vulnerable to either active attacks in the main wireless channel, or, in the case of Anderson and Stajano [5, 6], to passive (eavesdropping) attacks in the location-limited side channel.

Because legitimate participants would only communicate with entities from which they had received pre-authentication data, we would now require an attacker to perform an active attack – to be able to transmit –not only in the main wireless medium, but also in the location-limited channel. Because of the physical limitations of transmission on location-limited channels, it is significantly harder for an attacker to passively eavesdrop on them, not to mention to actively transmit. For such an active attack to succeed, the attacker must not only transmit on the location-limited channel, but must do so without being detected by any legitimate participant.

The difficulty of monitoring a pre-authentication for such unwanted participation depends on the type of channel used and the number of legitimate parties involved. The more directed the channel and the smaller the number of parties, the easier it is to monitor. Note that, because of the physical limitations of the channels used and this monitoring requirement, it is only possible to use our techniques to pre-authenticate devices that are physically co-located at the time of first introduction.

4.8.1 Advantages of testing protocol in banking

Consumers are looking for convenience and businesses are looking for new ways to provide it. Without doubt, wireless technologies have a high gee-whiz factor. They provide always-on network connectivity, but don't require a network cable. Office workers can roam from meeting to meeting throughout a building constantly connected to the same network resources enjoyed by wired, desk-bound coworkers. Home or remote workers can set up networks without worrying about how to run wires through houses that never were designed to support network infrastructure.

To motivate the development of ad hoc networking protocols, there needs to be applications where the properties of ad hoc networking are beneficial. Although some have been implemented many are still in the early research phase. Mobile ad hoc network is the latest field of research and its applications are limited to military tactical networks, personal area networks, sensor networks and disaster area networks. We have tried to introduce this field in banking mainly due to two reasons. WLAN is suffering security problems in industrial field and our approach will be effective alternative. Secondly there is no such application of MANET in industry so at least our approach will be foundation for finding MANET application in industries and institutions. For example we can deploy MANET in big shopping malls, Hospitals, Stock exchanges etc. It is hoped that the industry and financial institutions as a whole can progress forward more rapidly.

Wireless devices may actually prove less expensive to support than traditional networks for employees that need to connect to corporate resources in multiple office locations. Large hotel chains, airlines, convention centers, Internet cafes, etc., see wireless LANs as an additional revenue opportunity for providing Internet connectivity to their customers. Wireless is a more affordable and logistically acceptable alternative to wired LANs for these organizations. For example, an airline can provide for-fee wireless network access for travelers in frequent flyer lounges – or anywhere else in the airport.

- **Cable Replacement**

There would be no wires involved in banks for exchanging information.

- **Wireless Synchronization**

Wireless technologies provide automatic synchronization among PDA's, laptops, mobile phones and other devices. This automation occurs without the knowledge of the users as devices get within range of one another.

- **Security**

802.11 provide security only over the radio link, from each device to all other devices.

- **Confidentiality**

The intent is to prevent information eavesdropping. The encryption of the data is to prevent the payload of the packet from eavesdropping.

- **Authentication**

This addresses the identity of each communicating device. The sender sends an encrypted authentication request frame to the receiver. The receiver sends an encrypted challenge frame back to the sender. Both perform a predefined algorithm. The sender sends its findings back to the receiver, which in turn either allows or denies the connection.

- **Authorization**

This allows control of resources.

Chapter 5

System Analysis

5. System Analysis

The methodology that is used for Object Oriented Analysis and Design is Objectory or Object Oriented Software Engineering approach (Jacobson Method).It has the following four models:

- The Requirements Model
- The Analysis Model
- The Design Model
- The Implementation Model

Good analysis helps in making perfect software. There are two main parts of the analysis.

- Requirement Analysis
- Domain analysis

5.1 Requirement Analysis

The purpose of analysis is to provide a model of the system's behavior. In conducting the project, object oriented approach is adopted. Object oriented analysis is a method of analysis that examine the requirements from the perspective of the classes and object found in the vocabulary of the problem domain. In the requirement analysis we define use case diagrams containing use cases, actors. A first step in analysis is to extract scenarios, or use cases that describe the behavior of a system from and external user's perspective.

5.2 Domain Analysis

Conceptual domain analysis yields common ground for each specific analysis. Object oriented analysis notions lend themselves for capturing generic concepts at multiple levels of granularity. Ensembles, sub ensembles, classes, and generic relationships are all candidates for describing an application domain. Requirements domain analysis may lead to an Object oriented engineering effort. This entails the construction of design fragments of the generic elements identified by a requirement domain analysis. These designs can be implemented and added to a domain-specific code library.

5.3 A Unified Approach to Object-Oriented Analysis

It is a method of analysis that examines the requirements of end-user from the perspective of objects and classes found in the vocabulary of problem domain.

Over the past decade, Grady Booch, James Rumbaugh, and Ivar Jacobson have collaborated to combine the best features of their individual object-oriented analysis and design methods into a unified method. The result, called the Unified

Modeling Language (UML), has become widely used throughout the industry. UML allows a software engineer to express an analysis model using a modeling notation that is governed by a set of syntactic, semantic, and pragmatic rules.

In UML a system is represented using five different "views" that describe the system from distinctly different perspectives. Each view is defined by a set of diagrams. The following views are presented in UML:

- **User Model View.** This view represents the system (product) from the user's (called actors in UML) perspective. The use-case is the modeling approach of choice for the user model view. This important analysis representation describes a usage scenario from the end-user's perspective.
- **Structural Model View.** Data and functionality are viewed from inside the system. That is, static structure (classes, objects, and relationships) is modeled.
- **Behavioral Model View.** This part of the analysis model represents the dynamic or behavioral aspects of the system. It also depicts the interactions or collaborations between various structural elements described in the user model and structural model views.
- **Implementation Model View.** The structural and behavioral aspects of the system are represented as they are to be built.
- **Environment Model View.** The structural and behavior aspects of the environment in which the system is to be implemented are represented

In general, UML analysis modeling focuses on the user model and structural model views of the system. UML design modeling addresses the behavioral model, implementation model, and environmental model views.

5.4 The Object-Oriented Analysis Process

The OOA process doesn't begin with a concern for objects. Rather, it begins with an understanding of the manner in which the system will be used—by people, if the system is human-interactive; by machines, if the system is involved in process control; or by other programs, if the system coordinates and controls applications. Once the scenario of usage has been defined, the modeling of the software begins.

A series of techniques is used to gather basic customer requirements and then define an analysis model for an object-oriented system

5.4.1 Steps for Object Oriented Analysis

- Obtain "Complete" Requirements.
- Describe system-context interaction.
- Delineate subsystems
- Develop vocabulary by identifying instances with their classes, ensembles, and relationships.

- Elaborate classes and relationships by identifying their generic static structure and describing their generic dynamic dimension.
- Construct a model in which the dynamics of objects are wired together.

These steps are connected by transformation—elaboration relationships. The output of the last step, the model, feeds naturally into the design phase.

5.5 Use case Model

The Use case Model uses actors and use cases. These concepts are simply an aid to defining what exists outside the system (Actors) and what should be performed by the system (Use case).In Use case model the actors are identified, Use cases are identified and a use case model is constructed.

5.5.1 Use Cases

A use case is a specific way of using the system by using some part of the functionality. Each use case constitutes a complete course of events initiated by an actor and it specifies the interaction that takes place between an actor and the system. A use case is thus a special sequence of related transaction performed by and actor and system in a dialogue. The collected use cases specify all the existing ways of using the system.

5.5.2 Actors

There are three actors in this use case diagram

- Bank customer
- Administrator
- Mobile ad hoc node

5.5.3 Use Case Analysis

Use case analysis is performed to identify portion of system performing specific task. In use case analysis use cases, actors interacting with those use cases, and boundaries are identified. A use case comprises a course of events begun by an actor and it specifies the interaction between actor and the system. All the use cases specify existing ways to use the whole system. It is interaction of actors with external or other system with system being designed in order to achieve a goal. Use case describes the functionality of the product to be constructed.

5.5.4 Use cases in the System

Following use cases exist in this domain.

- Exchange ECC Key.
- Exchange hash of ECC Key.
- Generate Key
- Login
- Transfer Amount
- Generate secure disposable Credit Card.
- Logout
- Save Transaction
- Check Balance
- Submit Bill
- Create Account
- Statement Generation
- Maintain Log file.

5.6 Use Case Diagram

Use case diagram of system includes uses cases, actors and their relationship. Use case diagram of both systems including PDA client and Bank nodes is shown below.

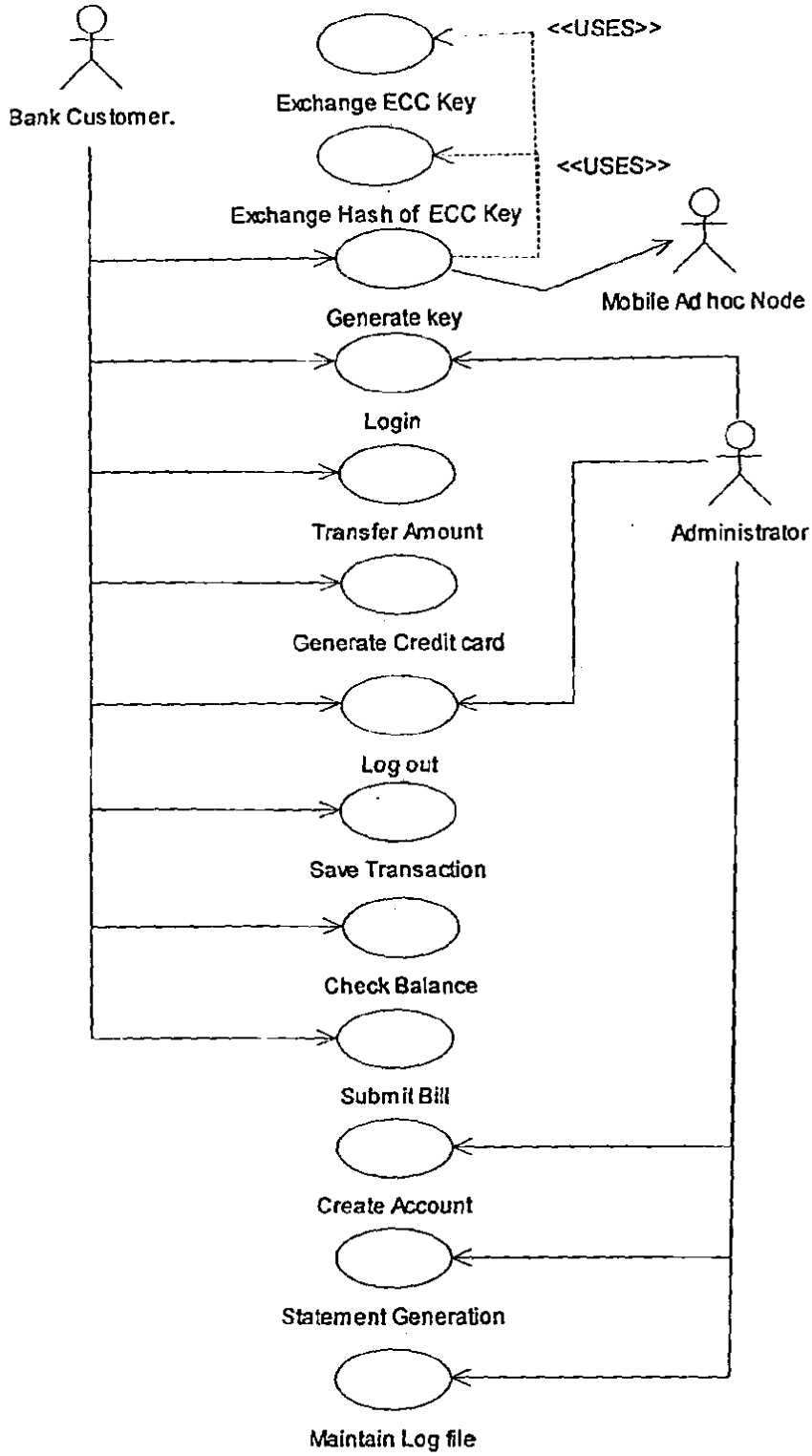


Figure 5.1 Use case diagram of Wireless Banking

5.7 Use Case Specification in system (Expanded Use-Case Format)

5.7.1 Login

Use Case Name: Login
Actor Involved: Administrator, Bank User.
Purpose: This use case lets the customer login to their account for further access to Wireless banking services. Administrator also uses this to manage services.
Trigger: User Wants to Access his account and administrator uses for managing account.
Pre Condition: Bank user will first create account, then he has to generate and exchange key. Otherwise User may not Login to the Application.
Post Condition: Actor has Access to All Services
Basic Course of Action:

User Request	System Response
1- Open the Login dialog	
2- Enter User Name & Password	
	3-System will accept user name and password
4- Request for login	
	5-System will get login data from user.
	6-System will validate Information.
	7- Actor is redirected to main application page.

Alternative Condition: *None*
Error Condition: when user enters wrong Login Information, System prompts that the login name and password does not match..
Author: Khalid Mahmood.

5.7.2 Log Out

Use Case Name: Log Out
Actor Involved: Administrator, Bank User
Purpose: Used for Logout to the System
Trigger: Bank user wants to end the Application or administrator wants to log out.
Pre Condition: Bank customer should be Login first
Post Condition: *The system doesn't let any Access to the Services.*

Basic Course of Action:

User Request	System Response
1- Actor Requests to exit application by clicking Cross button.	
	2- System will turn off the application.

Alternative Condition: None
Error Condition: None
Author: Khalid Mahmood.

5.7.3 Generate ECC Key

Use Case Name: Generate Key.
Actor Involved: Bank User, any mobile ad hoc node.
Purpose: It will be used for Post-Authentication.
Trigger: Nodes involved in communication want successful completion of post-Authentication.
Pre Condition: Bank Customer should start the application and must have his Account. For Ad hoc node it is necessary that it should receive customer PDA key.
Post Condition: Actor goes to post authentication mechanism.
Basic Course of Action:

User Request	System Response
1- Actor Requests to generate key by Pressing generate ECC key button.	
	2- System will show hash of PDA's key.
	3- Mobile node system will receive PDA's key.
	4- Mobile node will also generate its own ECC key and send back to PDA.

Alternative Condition: None.
Error Condition: None.
Author: Khalid Mahmood.

5.7.4 Exchange ECC Key

Use Case Name: Exchange ECC Key.
Actor Involved: Bank User, any mobile node.
Purpose: Public key will be exchanged through wireless media Using SSL/TLS protocol between Customer's PDA /Laptop and adhoc wireless node. Both Nodes will exchange their public keys and these keys will authenticated by hash of the keys already exchanged. This process is called post authentication.
Trigger: Actor wants to show successful completion of post authentication.
Pre Condition: Actor starts the application and must have his account. Also there should be successful completion of pre authentication.

Post Condition: Actor goes to Login dialog.
Basic Course of Action:

User Request	System Response
1- Bank customer requests to Exchange key by pressing Exchange key button in process of post authentication.	
	2- System will show "successful key Exchange" message on both nodes.

Alternative Condition: If one of entity in communication system is out of order then other node will display message that connection failed.
Error Condition: None
Author: Khalid Mahmood.

5.7.5 Exchange Hash of ECC key.

Use Case Name: Exchange Hash of ECC key.
Actor Involved: Bank User
Purpose: It uses Generate key use case. After generation of key, hash of key is calculated and both nodes will exchange their hash of keys with each other via location limited channel. This process is called pre-authentication.
Trigger: Actor wants to show successful completion of pre authentication.
Pre Condition: Actor starts the application and must have his account.
Post Condition: Actor goes to post authentication.
Basic Course of Action:

User Request	System Response
	1- System will calculate hash of its ECC key.
	2- one of Ad hoc node will send his hash to other node.
	3- This node will also accept hash of other node's ECC key.

Alternative Condition: If one of entity in communication system is out of order then other node will display message that connection failed.
Error Condition: None
Author: Khalid Mahmood.

5.7.6 Transfer Amount

Use Case Name: Transfer Amount.
Actor Involved: Bank User
Purpose: This uses case deals with transfer of amount from one account to other when user gets authenticity to use these services.

Trigger: Actor want to show successful transfer of amount from one account to other account.
Pre Condition: There should be successful completion of pre authentication and post authentication. Login and Password should also be validated.
Post Condition: Actor can use other available services.

Basic Course of Action:

User Request	System Response
1- Actor Requests to transfer amount by choosing transfer amount service from service menu .	
	2- A new dialog will appear.
3-User enters his 'Personal Account NO'.	
4-User enters the 'Transferring Account NO.' in which he wants to transfers amount.	
5-User enters the amount to be transferred.	
6-User enters Current Date.	
7- User presses OK button.	
	8-system displays a message box asking for surety to commit transaction.
9-User will press 'yes' button	
	10-Transaction will commit.

Alternative Condition: None.
Error Condition: if user enters invalid data error message will be displayed and control of focus would remain in same field.
Author: Khalid Mahmood.

5.7.7 Submit Bill

Use Case Name: Submit Bill.
Actor Involved: Bank User
Purpose: This uses case deals with transfer of amount from personal account to some organizational account when user gets authenticity to use these services.
Trigger: Actor want to show successful transfer of amount from one account to other account.
Pre Condition: There should be successful completion of pre authentication and post authentication. Login and Password should also be validated.
Post Condition: Actor can use other available services.
Basic Course of Action:

User Request	System Response
1-Actor Requests to submit bill by Choosing Submit bill service from service menu.	

2- A new dialog will appear.

3-User enters his 'Personal Account NO'.
4-User selects the 'organization' for which he wants to submit bill.
5-User enters the amount to be transferred.
6-User enters Current Date.
7- User Presses OK button.

8-system displays a message box asking for surety to commit transaction.

9-User will press 'yes' button

10-Transaction will commit.

Alternative Condition: None.
Error Condition: If user enters invalid data error message will be displayed and control of focus would remain in same field.
Author: Khalid Mahmood.

5.7.8 Generate secure disposable Credit Card.

Use Case Name: Generate secure disposable credit card
Actor Involved: Bank User
Purpose: This uses case deals with generation of secure disposable credit card.
Trigger: Actor want to show successful accomplishment of generation of secure credit card.
Pre Condition: There should be successful completion of pre authentication and post authentication. Login and Password should also be validated.
Post Condition: Actor can use other available services.

Basic Course of Action:

User Request	System Response
1. Actor Requests to submit bill by Choosing Submit bill service from Service menu.	
	2- A new dialog will appear.
3- User enters his 'Personal Account NO'.	
4-User enters the amount of Credit.	
5-User enters Current Date.	
6- User Press OK button.	
	7-system displays a message box asking for

	surety to commit transaction.
8-User will press 'Yes' button	
	9-System will commit transaction.

Alternative Condition: None.
Error Condition: If user enters invalid data error message will be displayed and control of focus would remain in same field.
Author: Khalid Mahmood.

5.7.9 Create Account.

Use Case Name: Create Account.
Actor Involved: Administrator
Purpose: To create new bank customer account
Trigger: Actor wants to show successful creation of new account.
Pre Condition: Actor starts the application and must be successfully login.
Post Condition: None.

Basic Course of Action:

User Request	System Response
1-Bank administrator requests to create new account by choosing create new account submenu.	
	2- System will submit entries to wireless bank database.

Alternative Condition: None.
Error Condition: None
Author: Khalid Mahmood.

5.8 Domain Analysis

In domain analysis concepts of project are represented. Conceptual diagrams are drawn to represent functionality of project which contains main concepts and their relations and their attributes.

5.8.1 Conceptual Diagrams

Conceptual diagram explains the main concept of the Authentication of Mobile ad hoc network, its functionality and relationship between different functions.

Conceptual diagrams of SSL/TLS Component, PDA client and mobile banking node are discussed in following sections. Following are conceptual diagrams drawn in this system.

- Conceptual Diagram of mobile node and admin, Bank component
- Conceptual Diagram of Key Exchange protocol SSL/TLS component
- Conceptual Diagram of PDA/Laptop client component

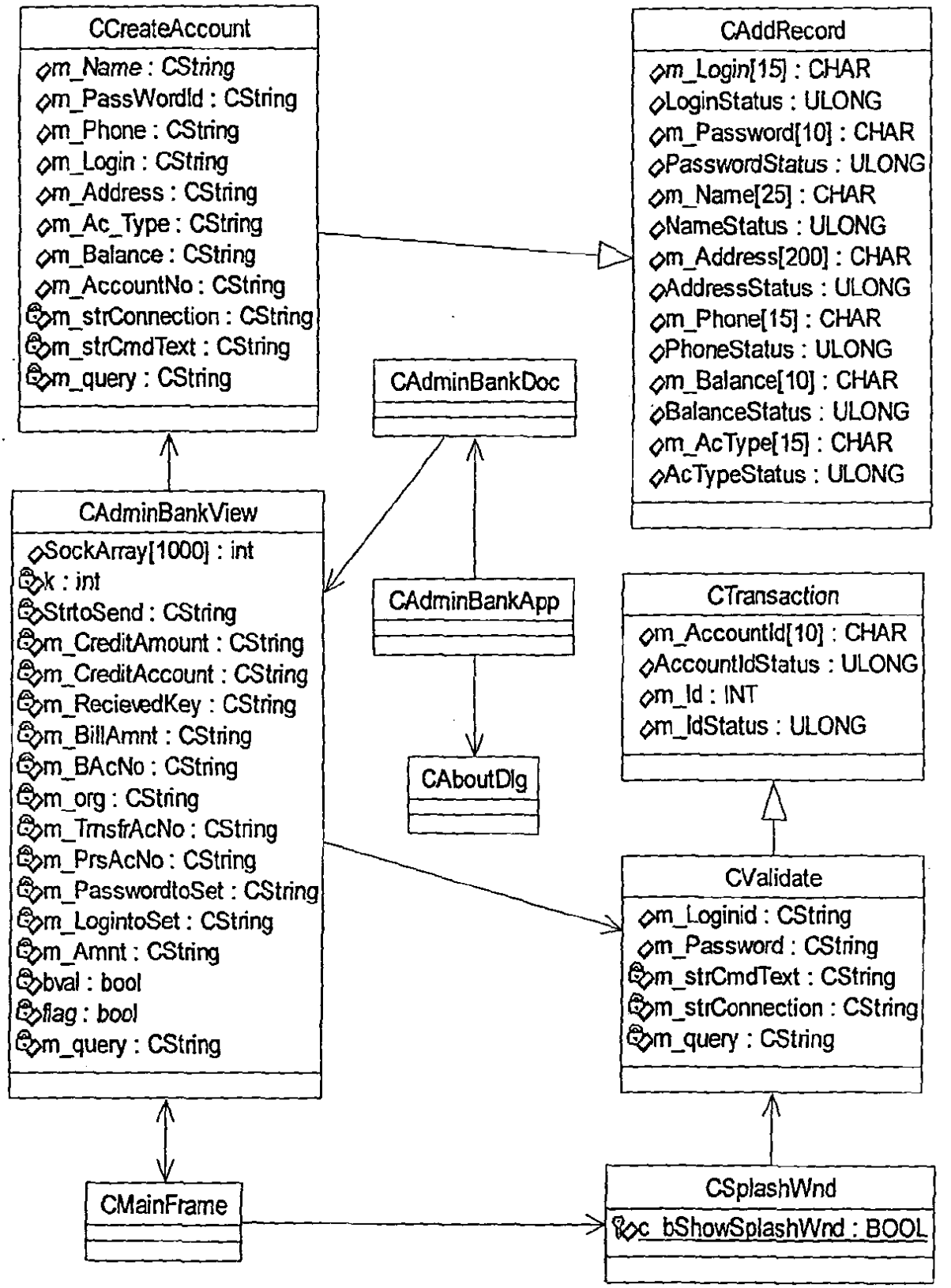


Figure 5.2 Conceptual Diagram of mobile node and adminBank

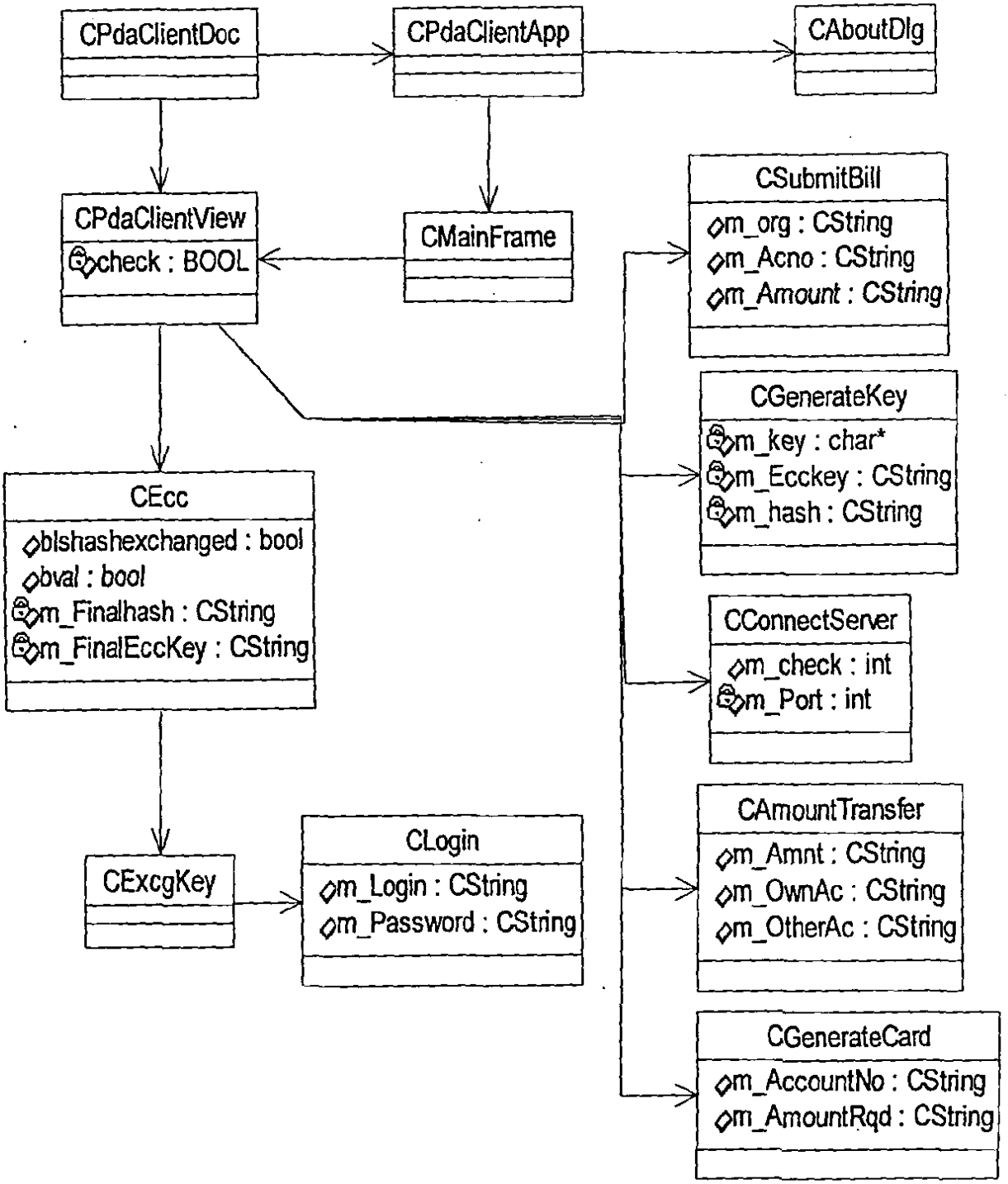


Figure 5.3 Conceptual Diagram of PDA Client

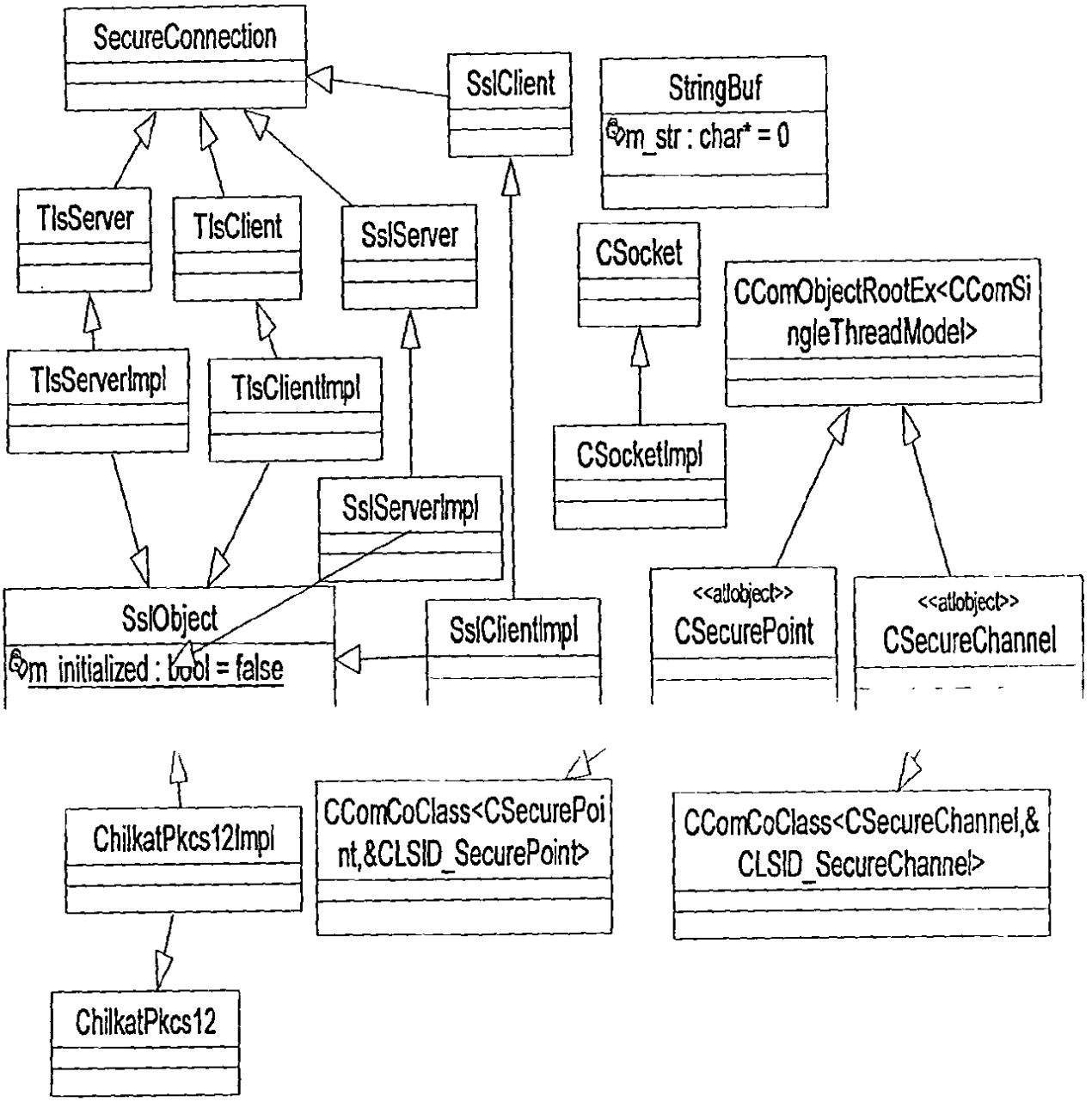


Figure 5.4 Conceptual Diagram of SSL/TLS Key Exchange Protocol

Chapter 6

System design

6. System design

The purpose of the design is to create architecture for the evolving implementations. Object oriented design is a method of design encompassing the process of objects oriented decomposition and a notation for depicting logical and physical as well as static and dynamic models of system under design.

The design phase focuses on defining the software to implement the application. The design object is to produce a model of the system, which can be used later to build the system. The design goal is to find the best possible design within the limitations imposed by the requirements and the physical and social environment in which the system will operate.

We shall define the software Design and Overview of Object Oriented technology and which tools we have used and why. Object oriented design contain UML Diagrams which will contain the specification of all classes and use cases.

6.1 Object Oriented Design of System

System design is the process of expanding what was learned during domain analysis into a working implementation. Design is a series of decisions on what is the most cost-effective implementation that will carry out the system charter and lead to reuse among many systems. Object oriented design deals with classes that are how a class interacts with each other which are found during requirement analysis. In this we'll also choose the tool in which system should be developed.

6.2 Design Patterns

The best engineers in any field have an uncanny ability to see patterns that characterize a problem and corresponding patterns that can be combined to create a solution. Throughout the OOD process, a software engineer should look for every opportunity to reuse existing design patterns (when they meet the needs of the design) rather than creating new ones.

6.2.1 Describing a Design Patterns

All design patterns can be described by specifying the following information:

- Name of the pattern
- Intent of the pattern
- "Design forces" that motivate the pattern
- Solution that mitigates these forces

- Classes that are required to implement the solution
- Responsibilities and collaboration among solution classes
- Guidance that leads to effective implementation
- Example source code or source code templates
- Cross-references to related design patterns

The design pattern name is itself an abstraction that conveys significant meaning once the applicability and intent are understood. Design forces describe the data, functional, or behavioral requirements associated with part of the software for which the pattern is to be applied. In addition forces define the constraints that may restrict the manner in which the design is to be derived. In essence, design forces describe the environment and conditions that must exist to make the design pattern applicable. The pattern characteristics (classes, responsibilities, and collaborations) indicate the attributes of the design that may be adjusted to enable the pattern to accommodate a variety of problems. These attributes represent characteristics of the design that can be searched (e.g. via a database) so that an appropriate pattern can be found. Finally, guidance associated with the use of a design pattern provides an indication of the ramifications of design decisions.

6.2.2 Using Patterns in Design

In an object-oriented system, design patterns can be used by applying two different mechanisms: inheritance and composition. Using inheritance, an existing design pattern becomes a template for new subclass. The attributes and operations that exist in the pattern become part of the subclass.

Composition is a concept that leads to aggregate objects. That is, a problem may require objects that have complex functionality (in the extreme, a subsystem accomplishes this). The complex object can be assembled by selecting a set of design patterns and composing the appropriate object (or subsystem). Each design pattern is treated as a black box, and communicates among the patterns.

6.3 Object-Oriented Design Process

UML design modeling addresses the structural model, behavioral model, implementation model, and environmental model views.

6.4 Structural Model

Data and functionality are viewed from inside the system. That is, static structure (classes, objects, and relationships) is modeled.

6.5 Class Diagrams

The development phases produce candidate classes and relationships. After selecting concise and evocative names we must describe each class with attributes. Although each class must have a unique name, class should be distinguishable on the basis of their attribute characterizations. A rule of thumb is if two classes have identical attributes, then they are most likely the same.

In Design Model we give shape to our components for implementation. Different classes are created. These classes represent the functions and attributes to be used in implementation. The class diagrams will show the relationship among different classes. Following Class diagrams are drawn below.

1. Class Diagram of mobile node and admin bank component.
2. Class Diagram of PDA/Laptop client component.
3. Class Diagram of Key Exchange Protocol SSL COM component

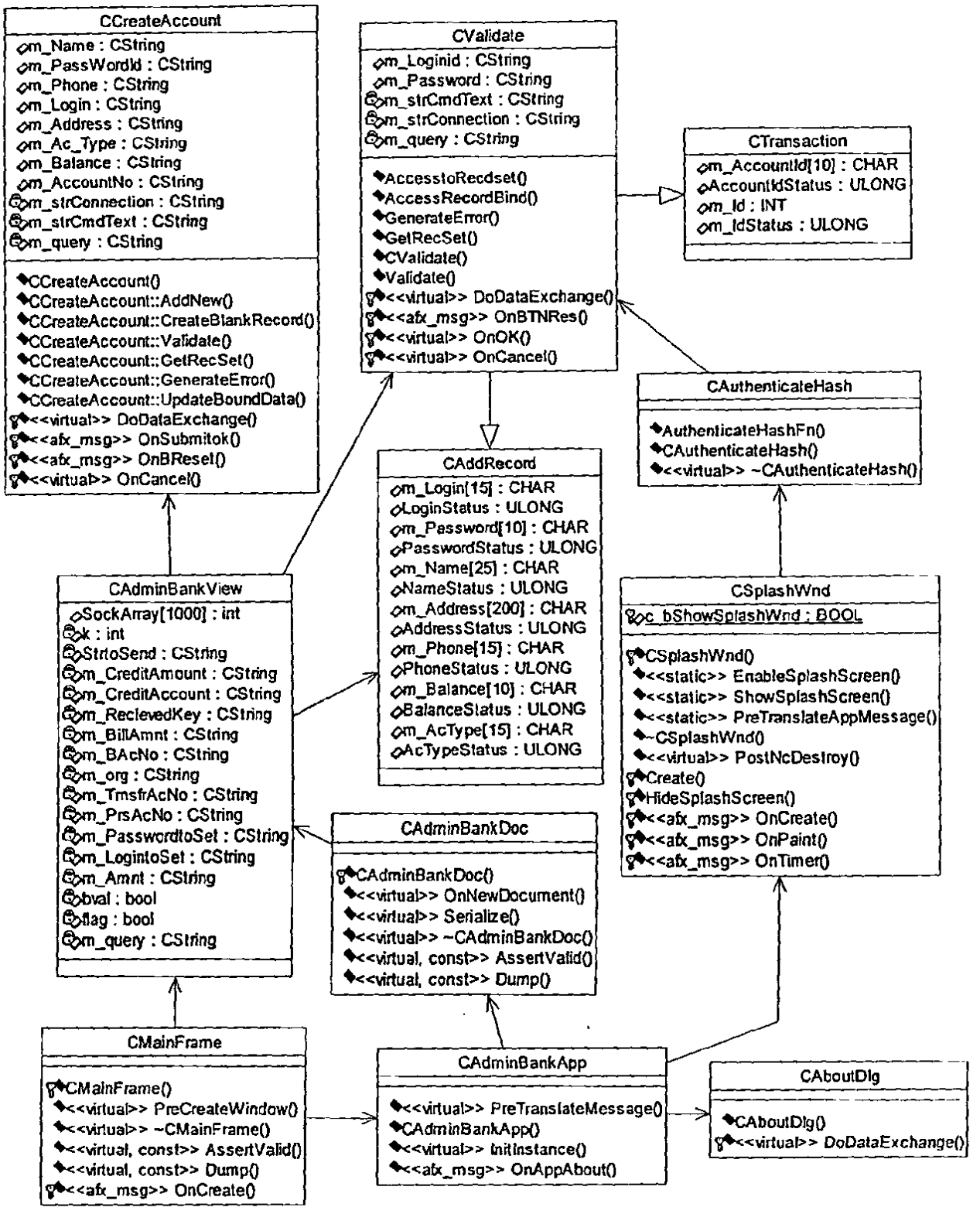


Figure 6.1 Class Diagram of mobile node and admin. Bank component

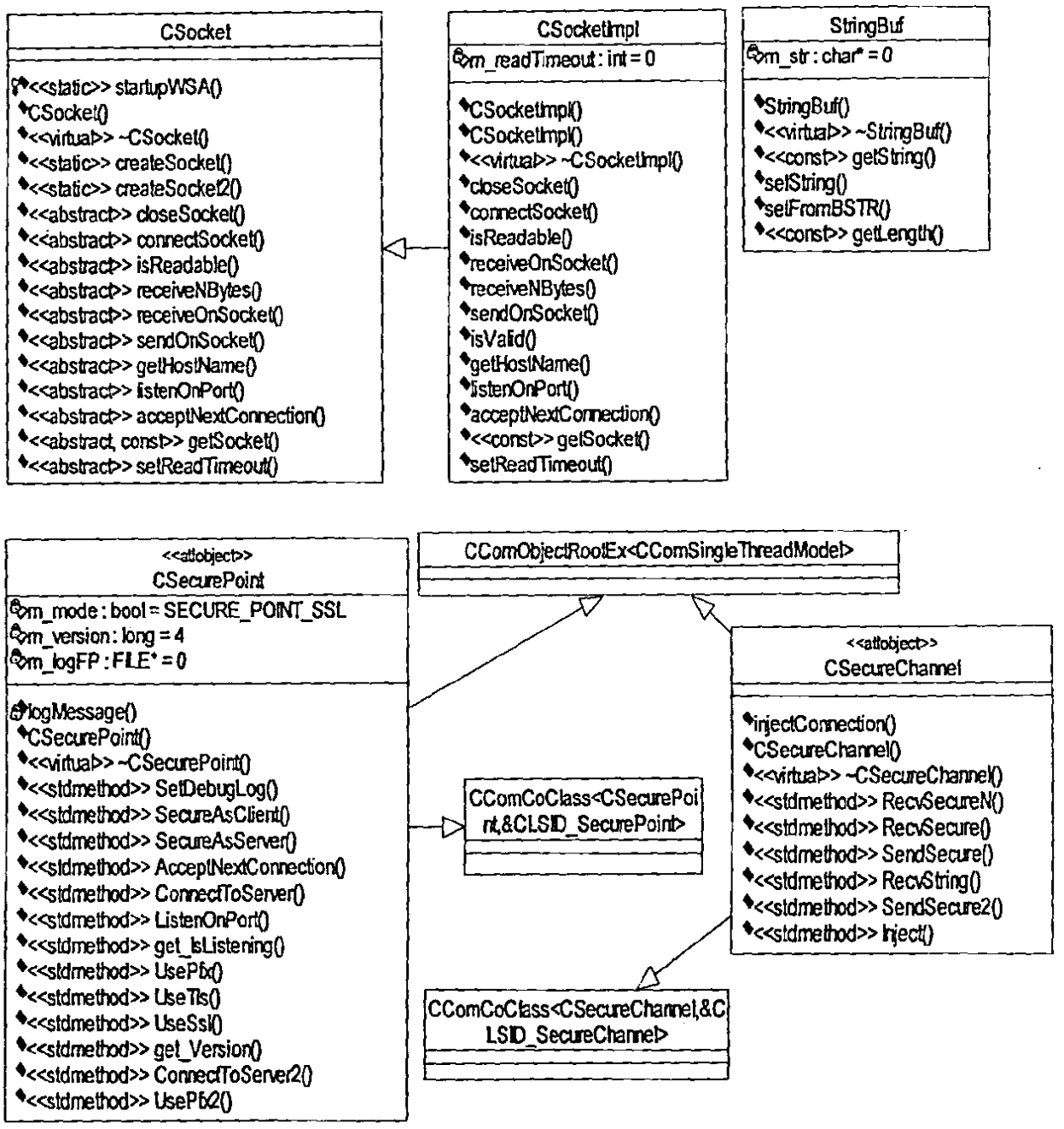


Figure 6.2 Class Diagram of SSL Server COM Component

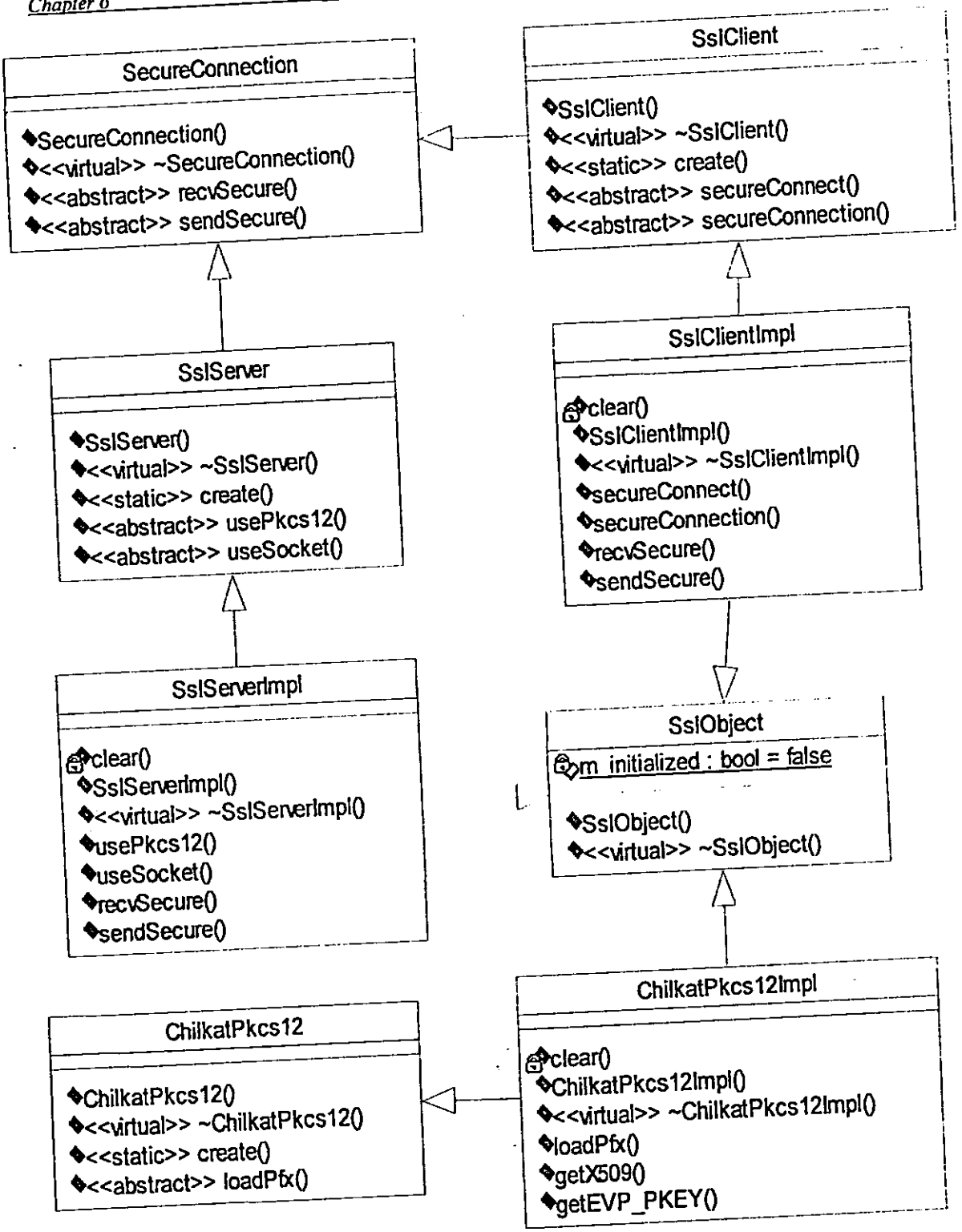


Figure 6.3 Class diagram of COM component

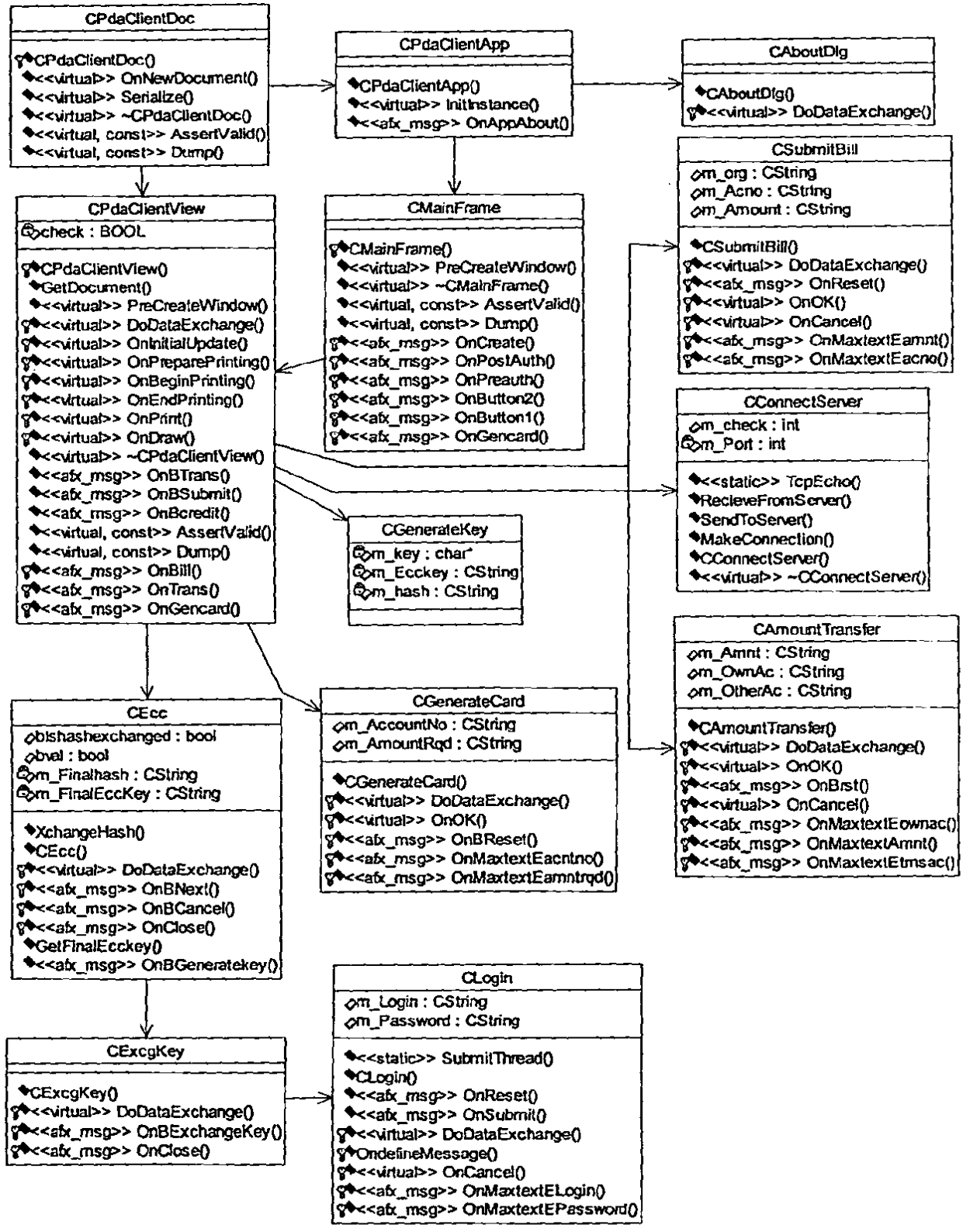


Figure 6.4 Class diagram of PDA/Laptop (Client side which is also mobile)

6.6 Activity Diagrams

It gives the pictorial representation for *algorithm for function.* Activity diagram is used to represent activities present in use cases. Basic need is that we need to make procedural design in UML. Operations in use cases in sequences are represented in activity diagram. Activity diagram are useful when we want to describe a behavior which is parallel, or when we want to show how behaviors in several use cases interact. Following Figures will describe activity diagrams of above mentioned *use* cases. Following diagrams explain the activities in system.

1. Activity Diagram of Authentication in Mobile ad hoc Network using Location limited channel.
2. Activity Diagram of Account Validation Process for Bank Customer.
3. Activity Diagram of Submit Bill by Customer on Mobile ad hoc Network.
4. Activity Diagram of Transfer of Amount.
5. Activity Diagram of Create users by administrator.

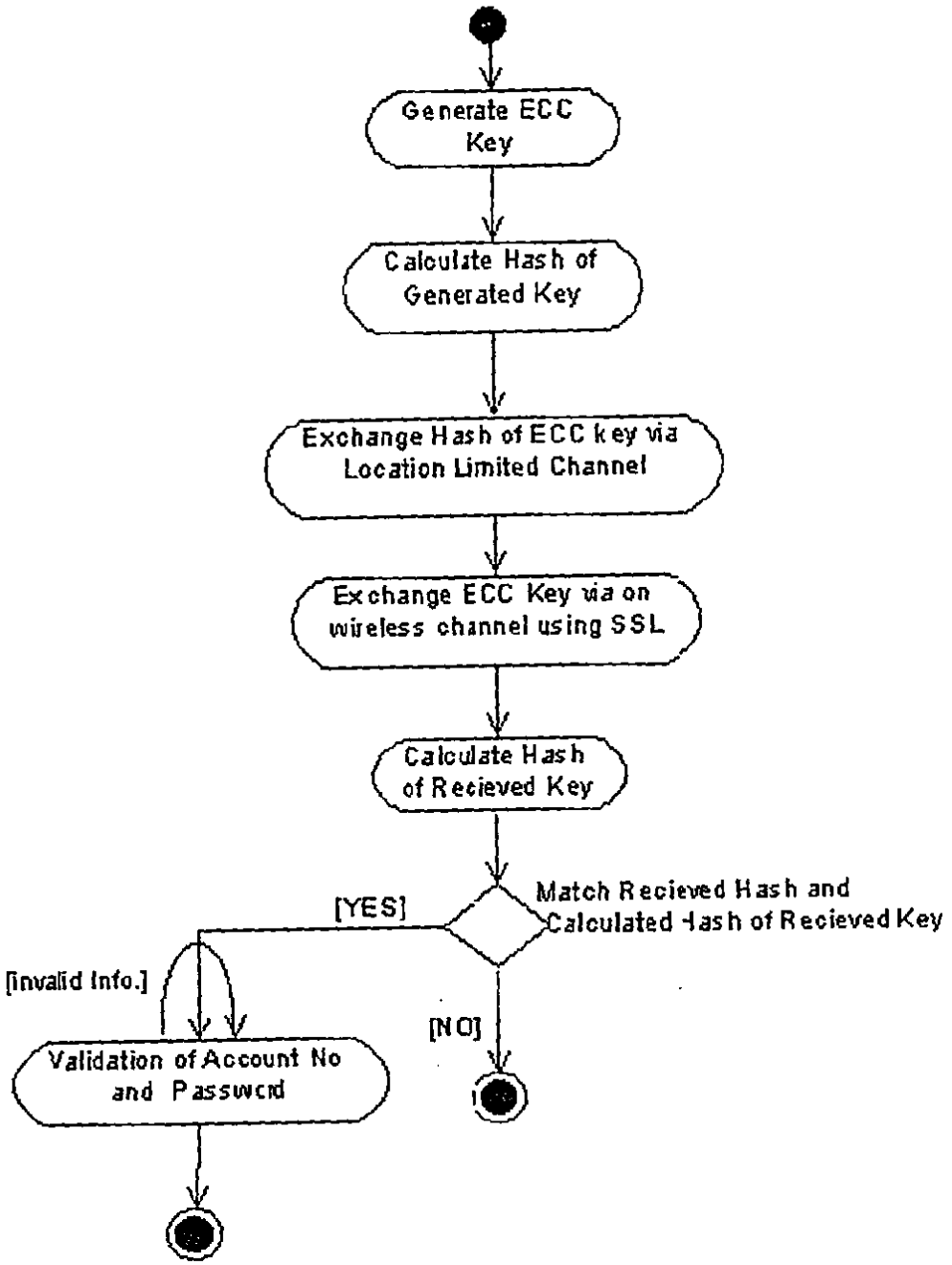


Figure 6.5 Activity Diagram of Authentication in Mobile ad hoc Network using Location limited channel

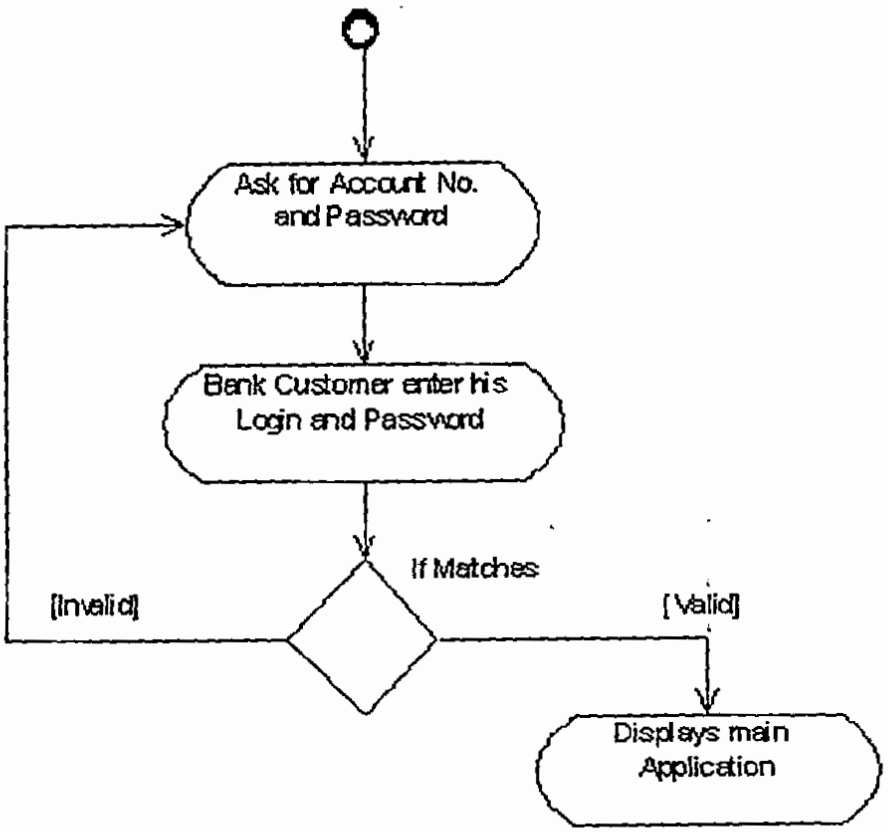


Figure 6.6 Activity Diagram of Account Validation Process for Bank Customer

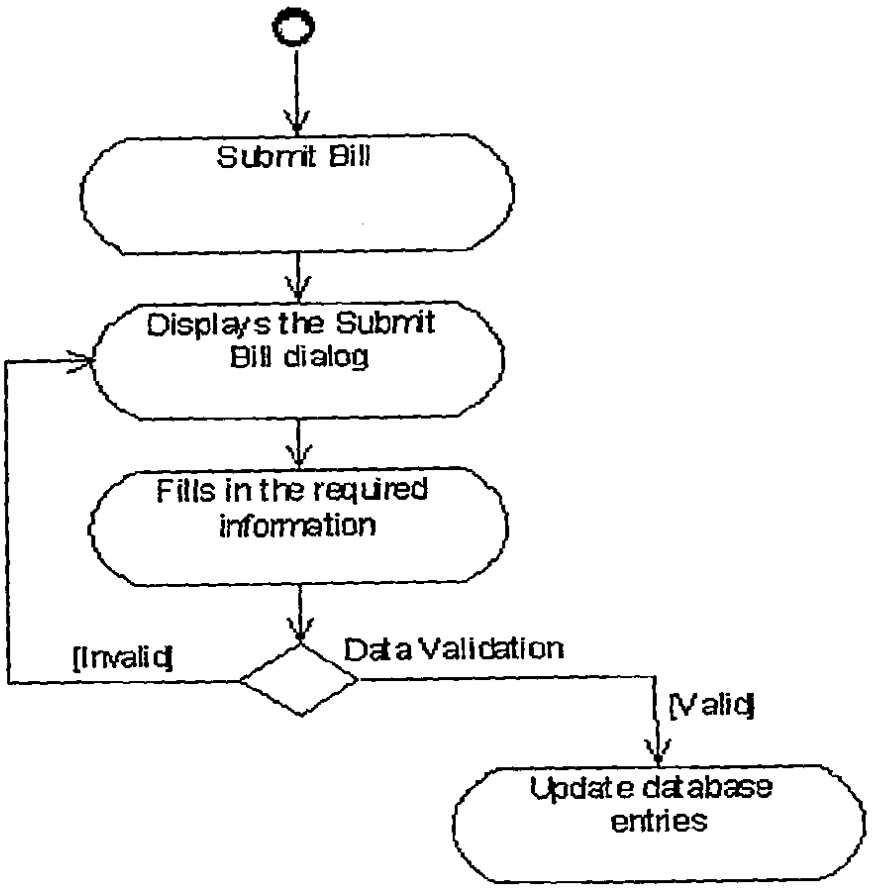


Figure 6.7 Activity Diagram of Submit Bill by Customer on Mobile ad hoc Network

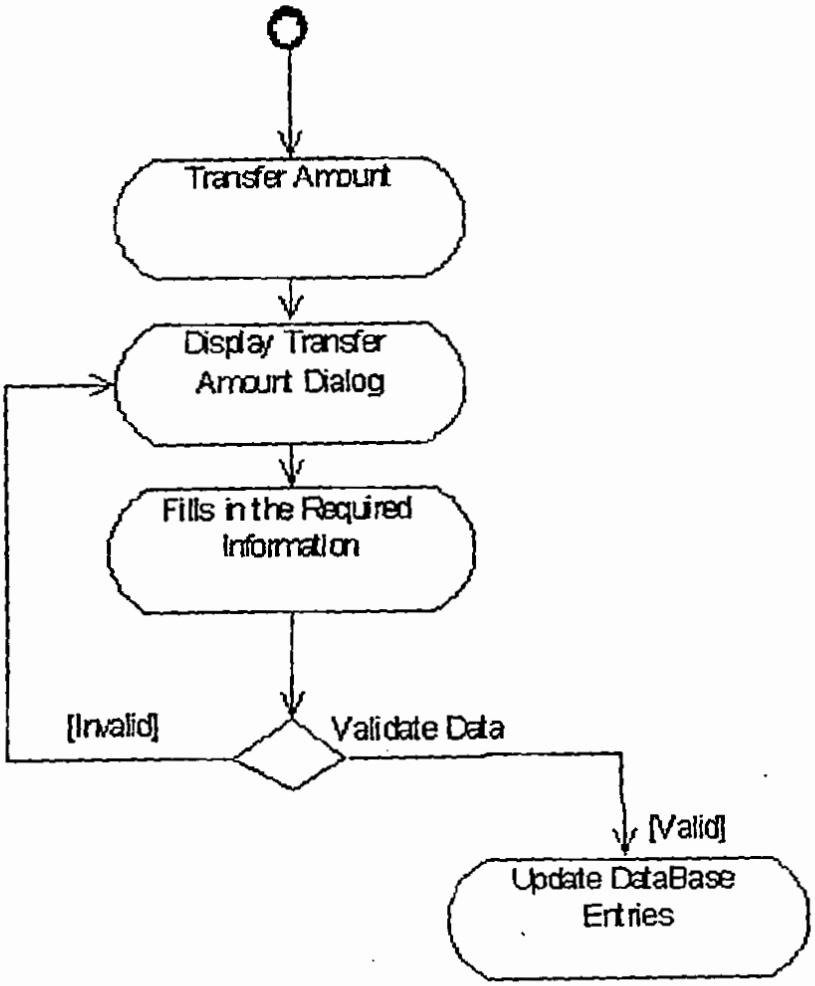


Figure 6.8 Activity Diagram of Transfer of Amount from one account to another

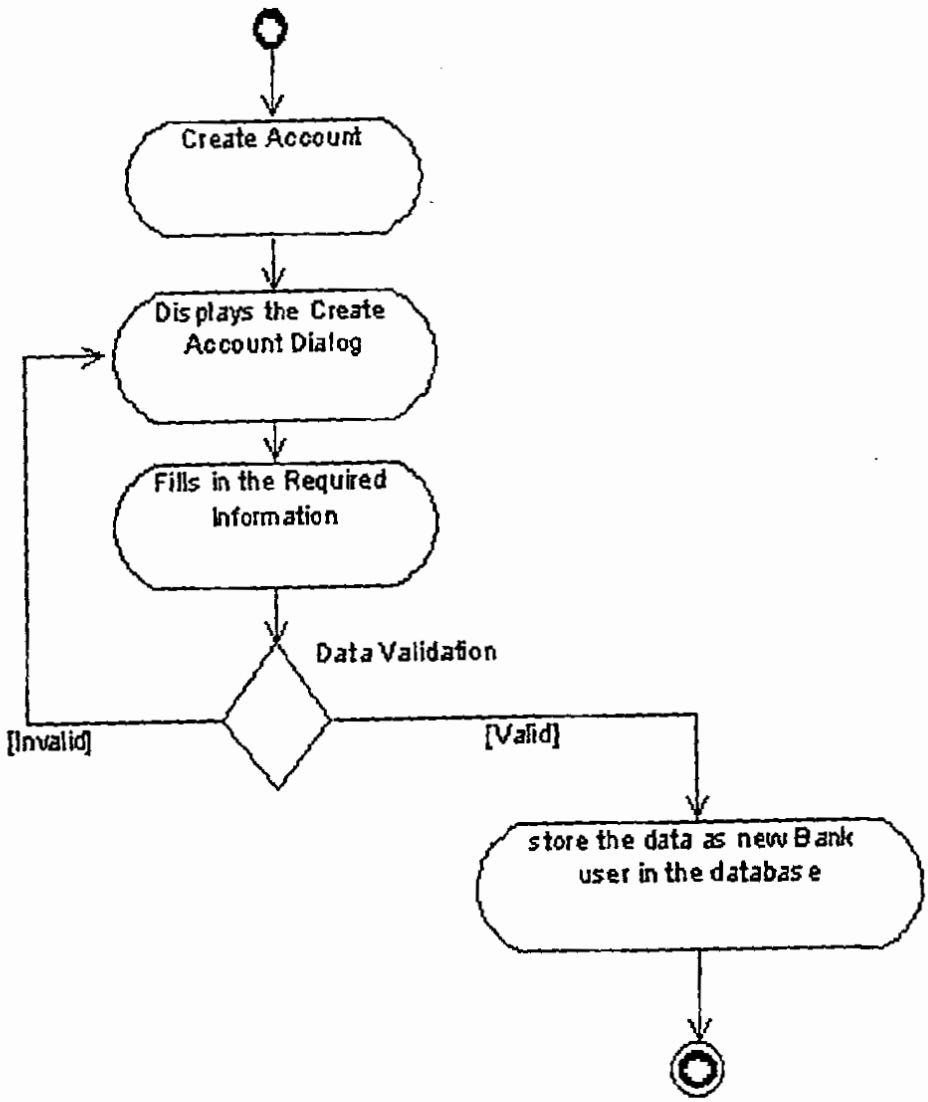


Figure 6.9 Activity Diagram of Create users by administrator

6.7 Behavioral Model

This part of the analysis model represents the dynamic or behavioral aspects of the system. It also depicts the interactions or collaborations between various structural elements described in the user model and structural model views.

6.7.1 Interaction Diagrams

An Interaction diagram shows, step-by-step, one of the flows through a use-case. There are two types of Interaction diagrams:

- Sequence Diagram represents dynamic behavior which time oriented. It can show the focus of control.
- Collaboration Diagram represents dynamic behavior which message oriented. It can show the data flow.

6.8 Sequence Diagrams

Sequence Diagrams are used to show the flow of functionality through a use case. For one use case diagram there can be multiple Sequence Diagram. Sequence Diagrams are time dependent and tell which operation will be executed first. Sequence Diagram defines a pattern of interaction among objects arranged in chronological order. These show the objects participating in interaction by the order of their life times and the message being sent from one object to the other.

A Sequence diagram is an interaction diagram, which is ordered by time; it is read form the top to the bottom. We can read this diagram by looking at the objects and messages. The objects that participate in the flow are shown in rectangles across the top of the diagram. The actor objects, involved in the use-case are also shown in the diagram.

Each object has a lifeline, drawn as a vertical dashed line below the object. A message is drawn between the lifelines of two objects to show that the objects communicate. Each message represents one object making a function call of another. Messages can also be reflexive, showing that an object is calling one of its own operations.

6.8.1 Authentication of PDA client by mobile node:

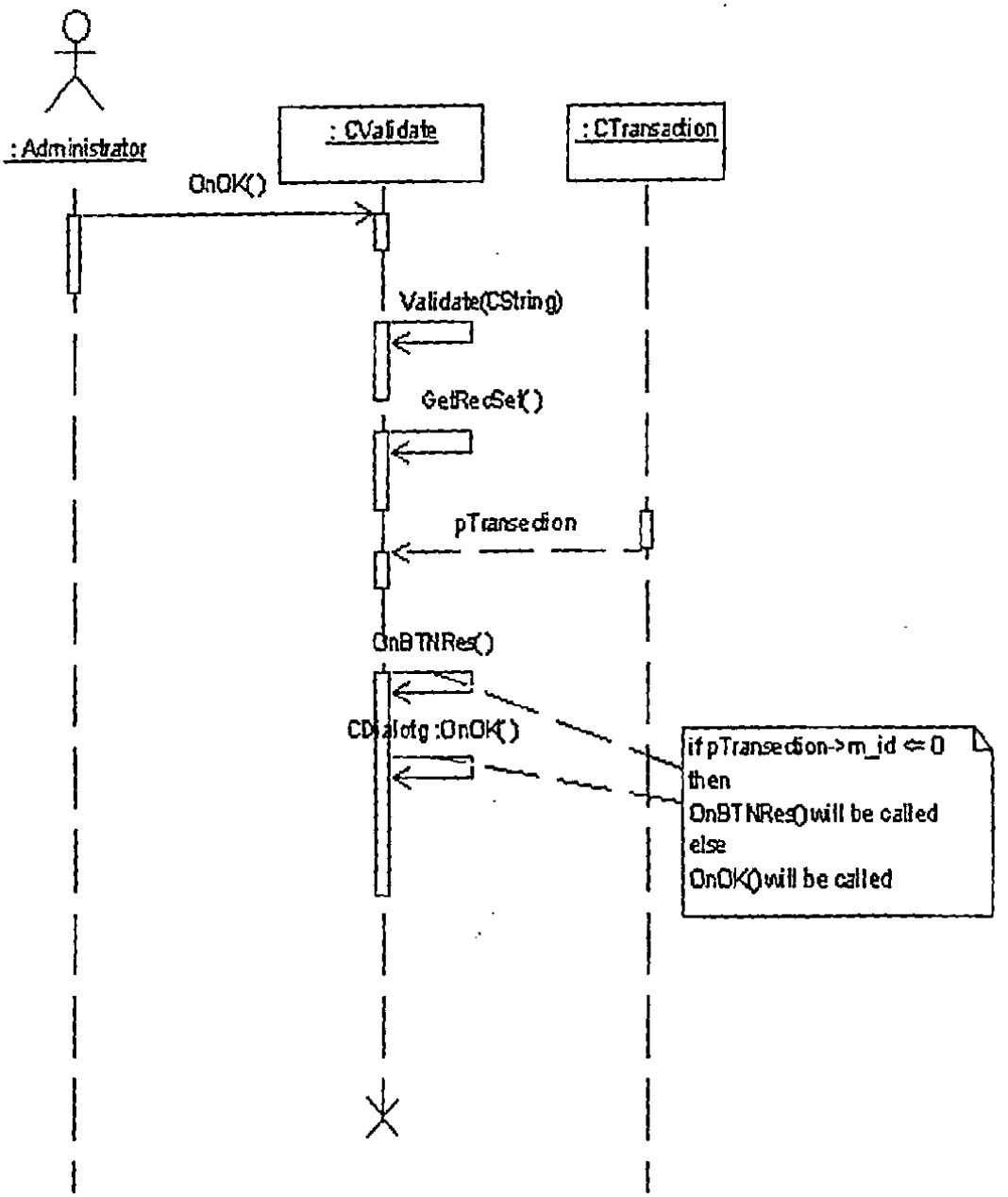


Figure 6.10 Sequence diagram of Authentication of PDA/Laptop client by mobile node

6.8.2 Create new user by administrator in admin.bank side.

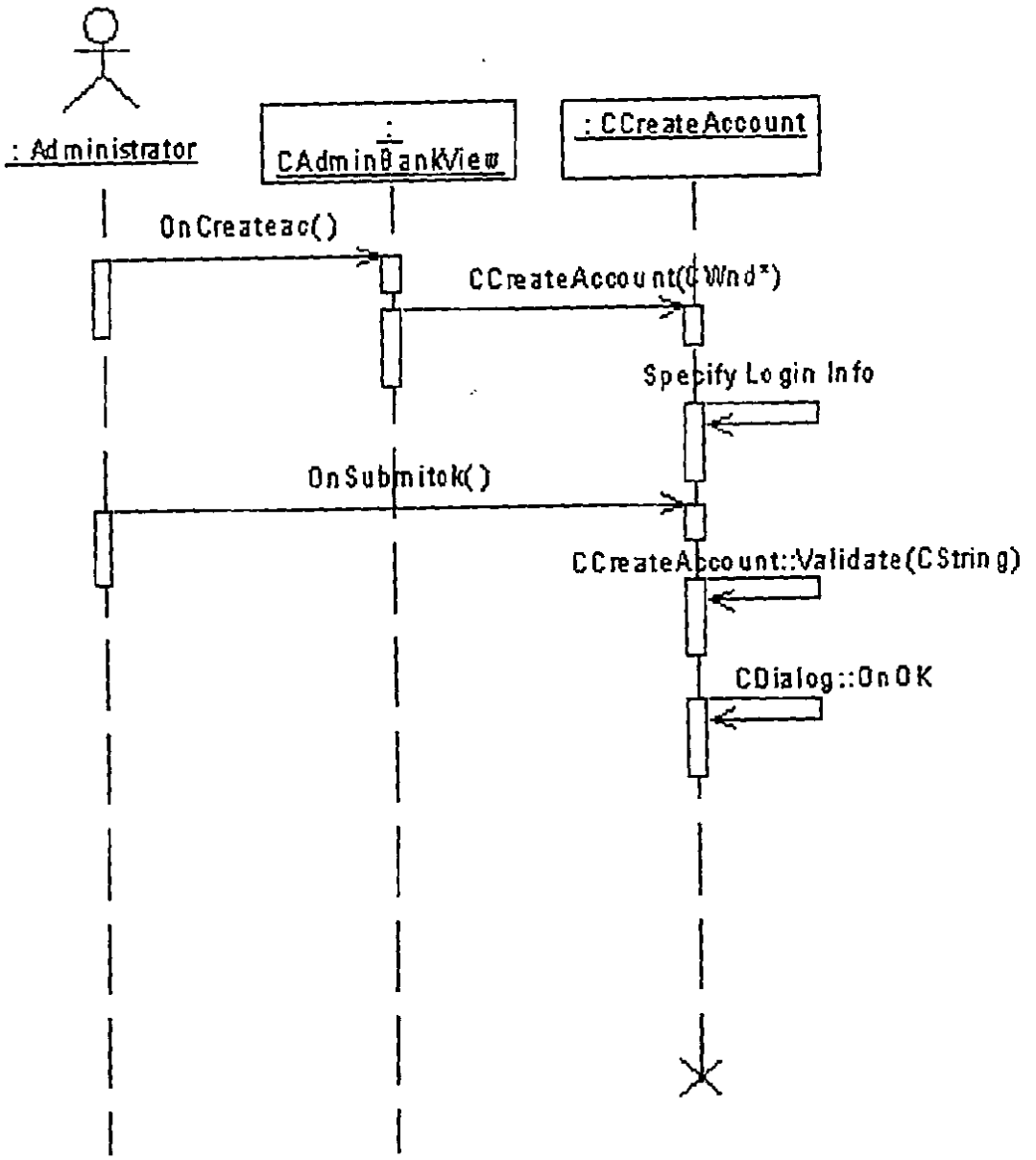


Figure 6.11 Sequence diagram of create new user by administrator

6.8.3 Generation of disposable secure credit card by bank Customer having PDA /Laptop.

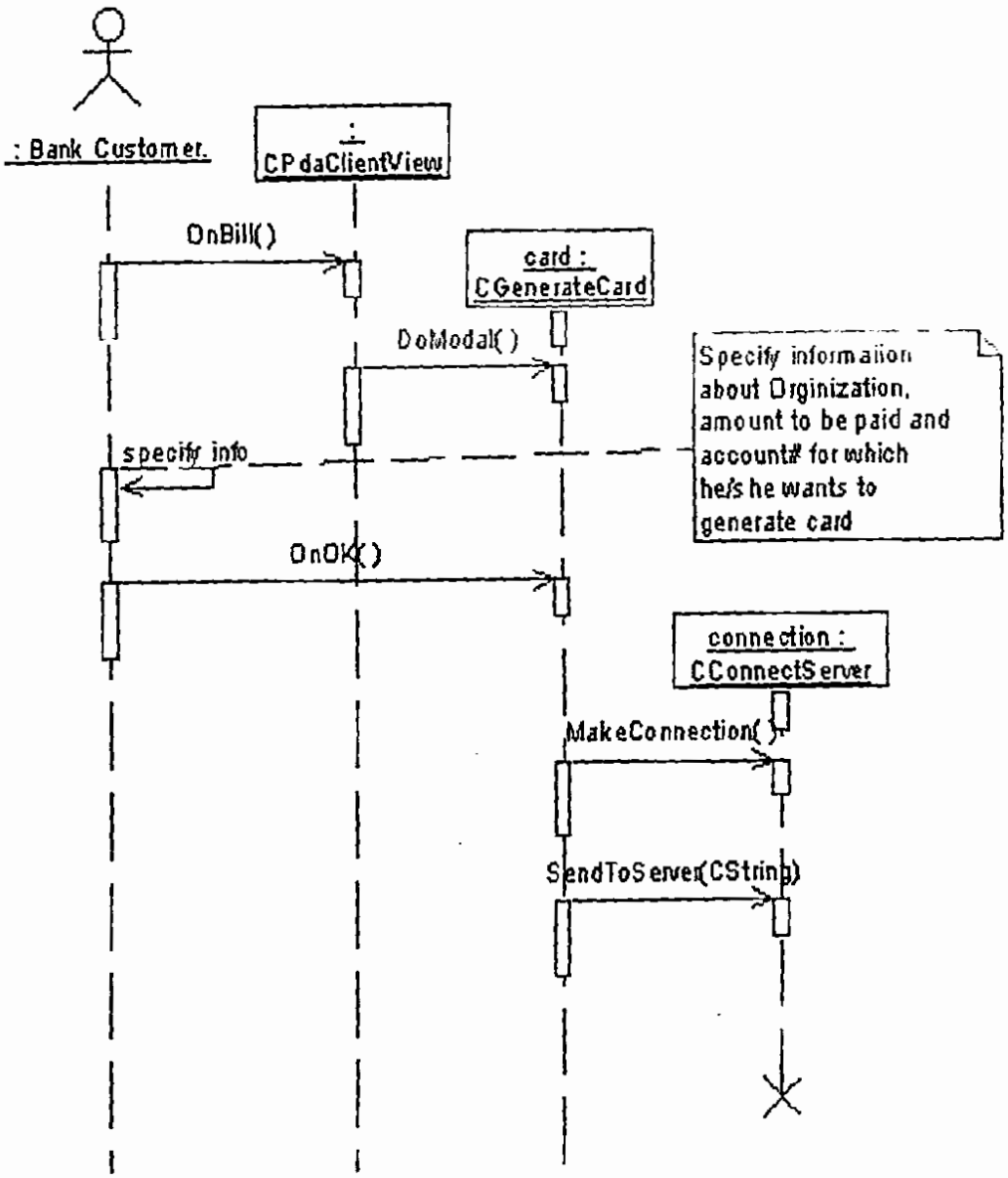


Figure 6.12 Sequence diagram of Generation of disposable secure credit card by bank Customer having PDA /Laptop

6.8.4 Generation of key by all mobile ad hoc node and PDA/Laptop Client.

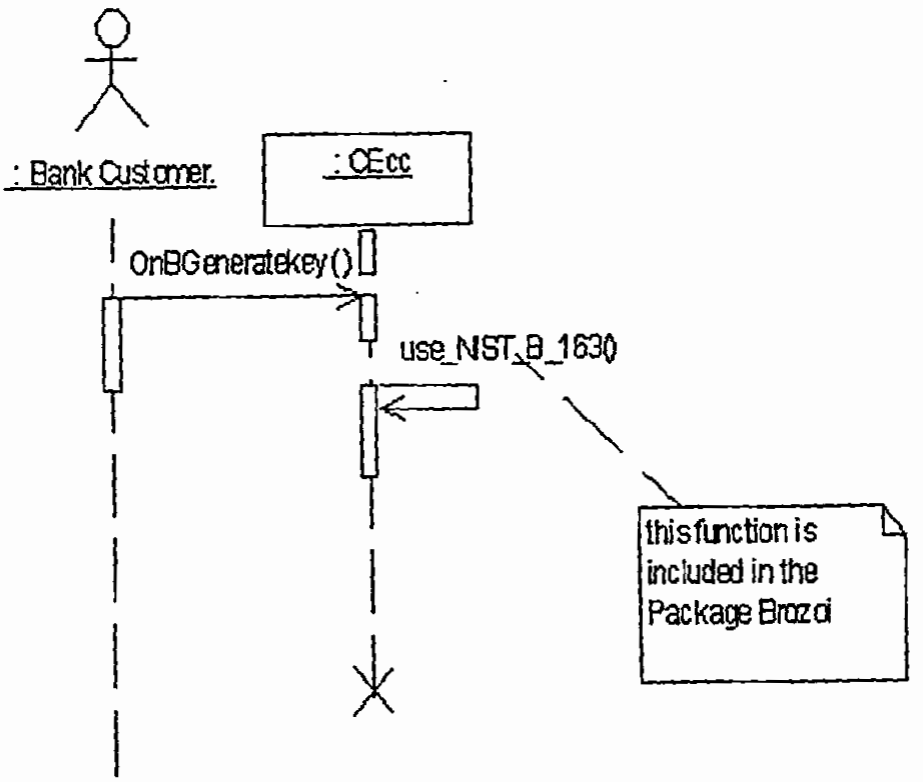


Figure 6.13 Sequence diagram of Generation of key by all mobile ad hoc node and PDA/Laptop Client

6.8.5 Post authentication by Client and mobile node by exchanging ECC keys.

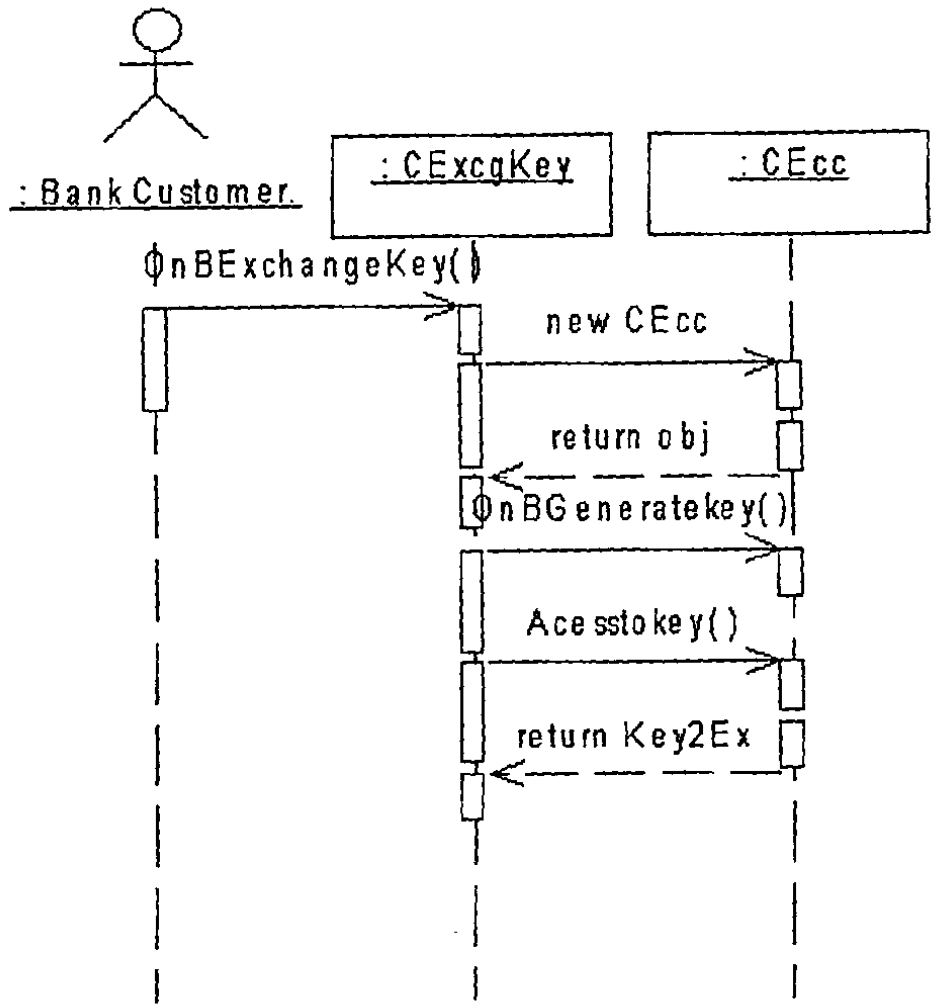


Figure 6.14 Sequence diagram of Post authentication by Client and mobile node by exchanging ECC keys

6.8.6 Pre-authentication done by PDA/Laptop and mobile node by exchanging hash of ECC keys.

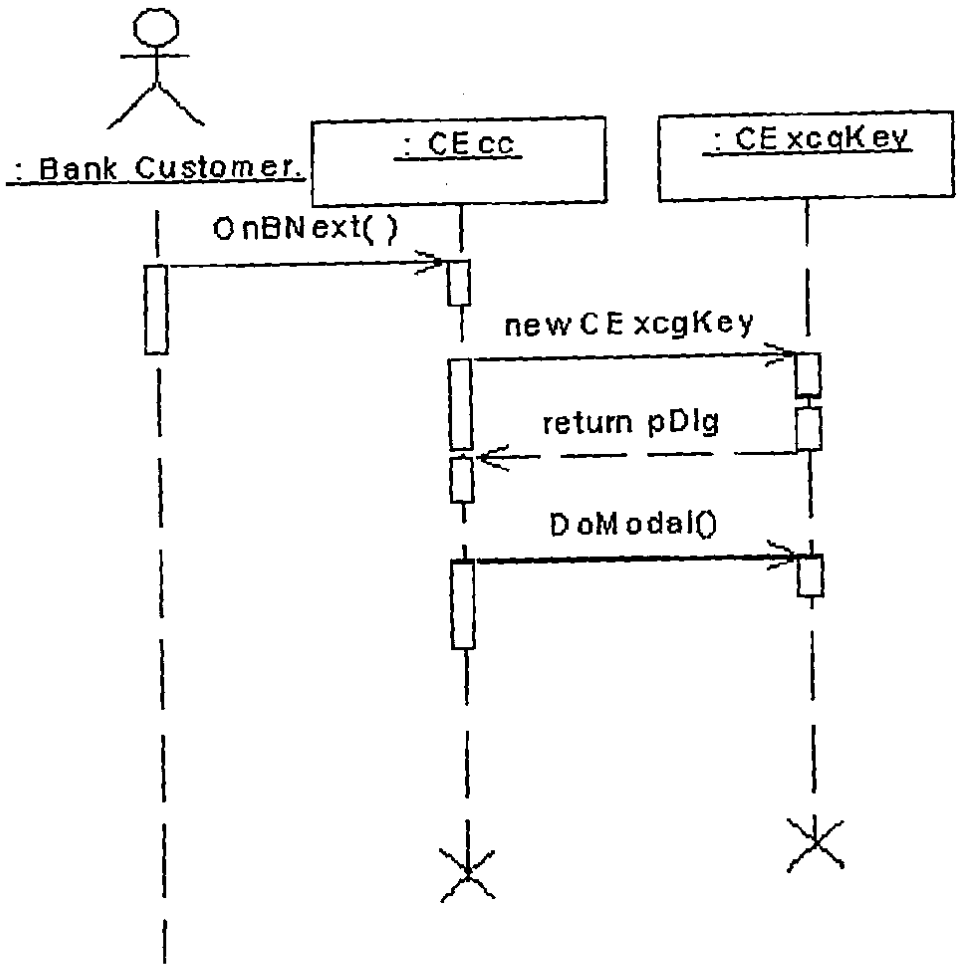


Figure 6.15 Sequence diagram of Pre-Authentication between nodes by exchanging hash of ECC keys

6.8.7 Submit bill by bank customer using his PDA/Laptop in wireless bank building.

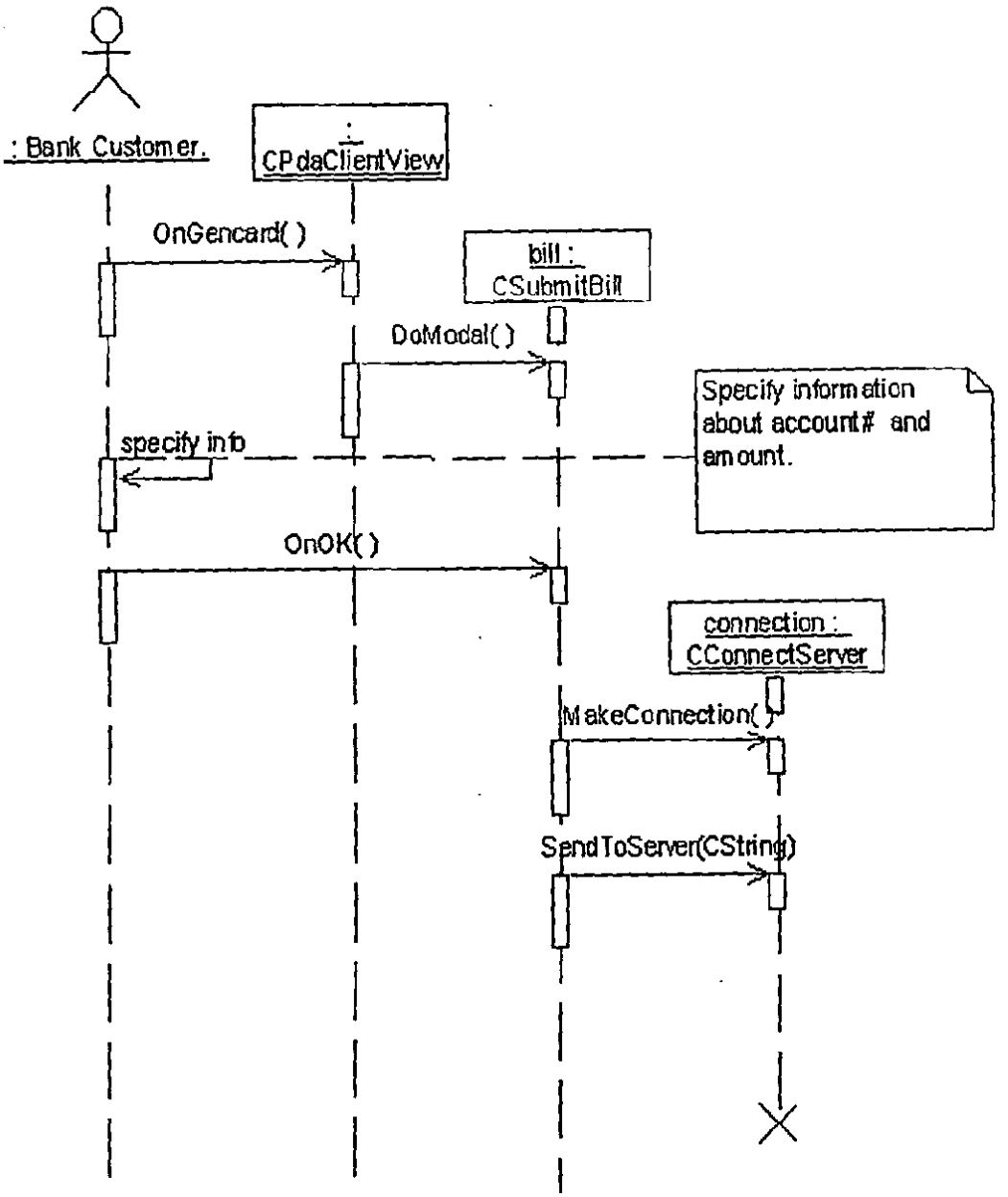


Figure 6.16 Sequence diagram of Submit bill by bank customer using his PDA/Laptop in wireless bank building

6.8.8 Transfer of amount from personal account to another account.

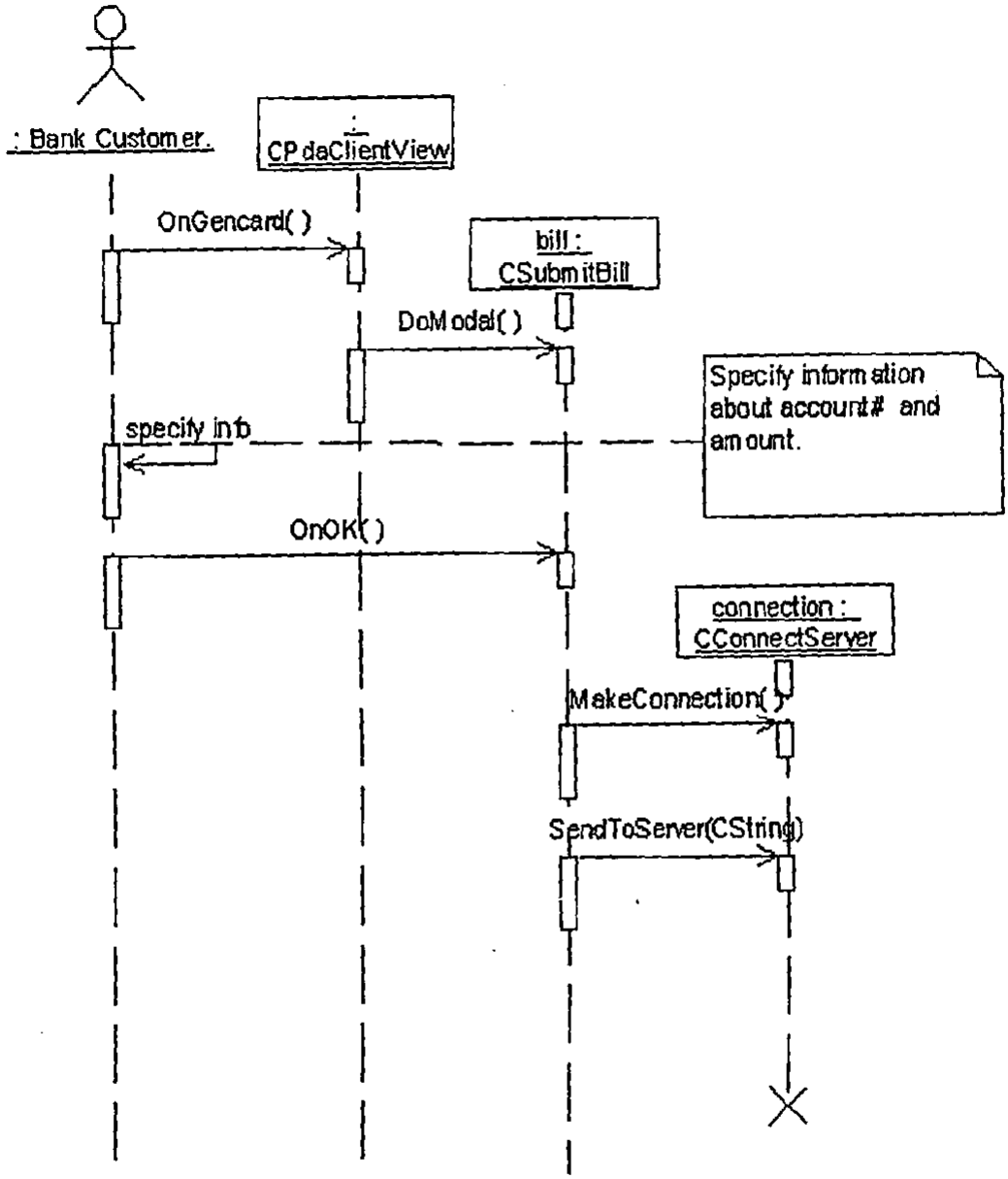


Figure 6.17 Sequence diagram of transfer of amount from personal account to another account

6.9 Collaboration diagrams.

6.9.1 Authentication of PDA client by mobile node:

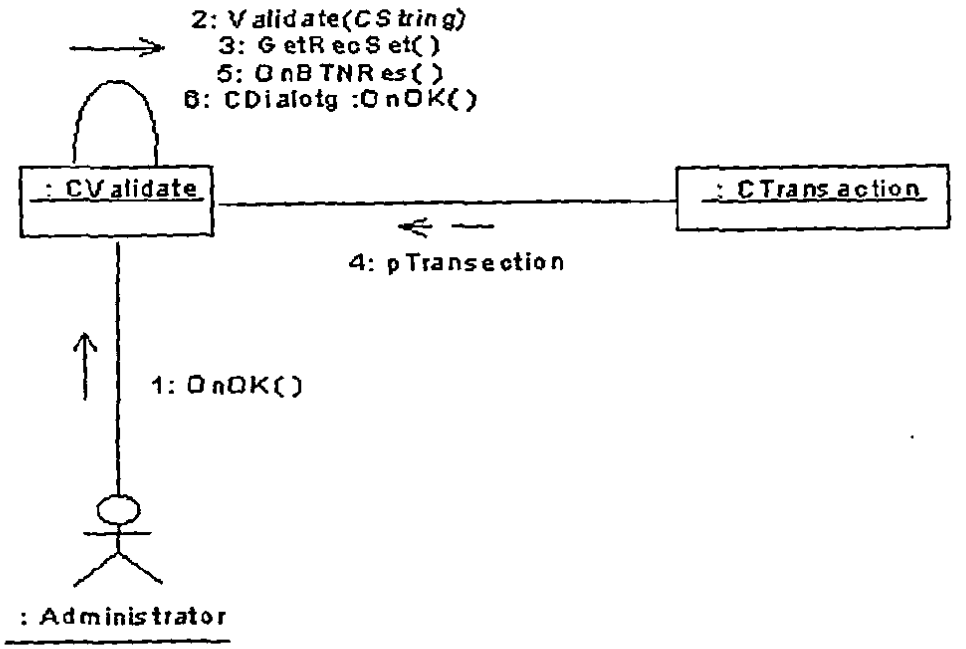


Figure 6.18 Collaboration diagram of Authentication of PDA/Laptop client by mobile node

6.9.2 Start Application by Bank Administrator

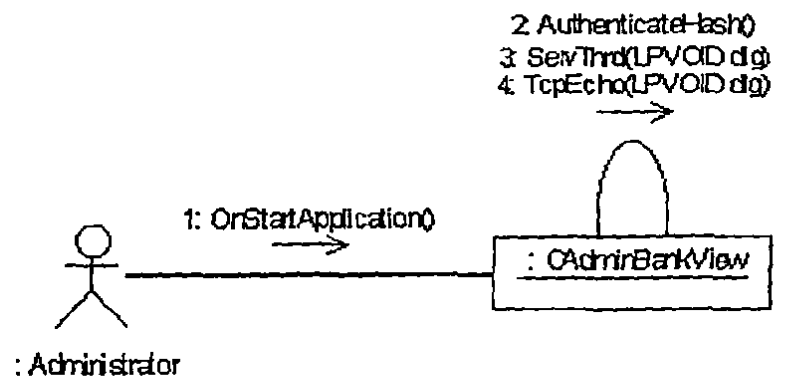


Figure 6.19 Collaboration diagram of start application by administrator

6.9.3 Create Account by Administrator

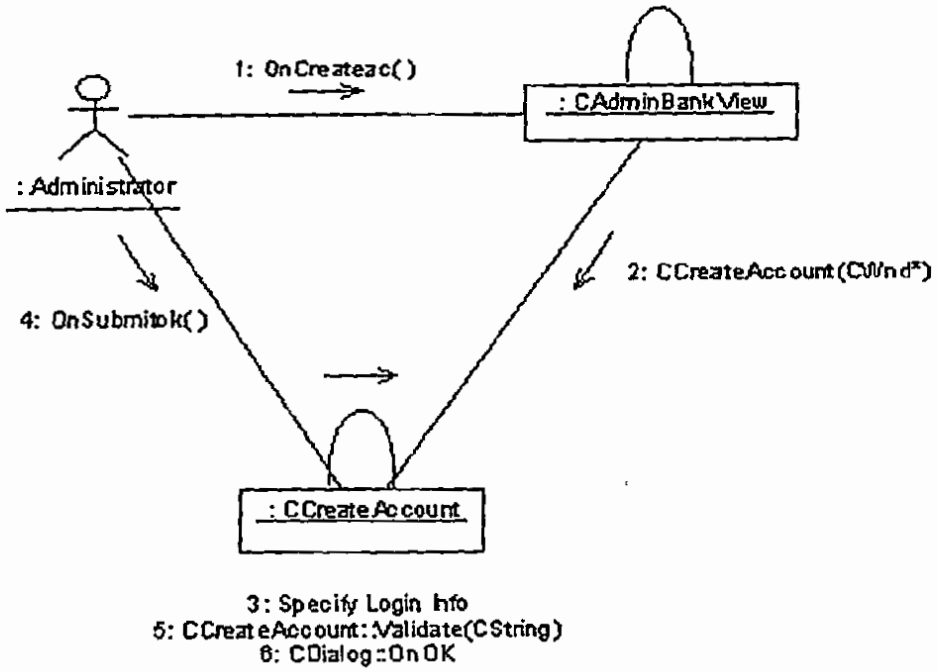


Figure 6.20 Collaboration diagram of Create account

6.9.4 Generate card by Bank customer having his PDA/Laptop

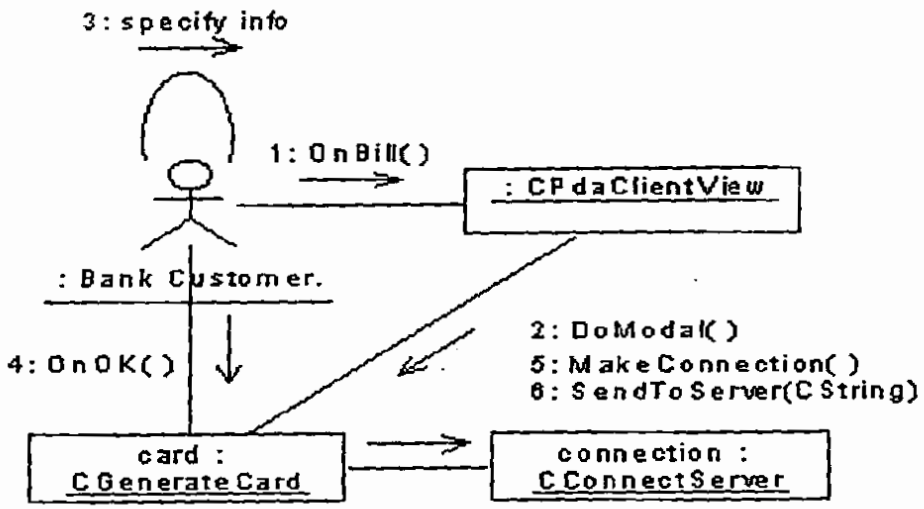


Figure 6.21 Collaboration diagram of Generate card by Bank customer

6.9.5 Generate key by Bank customer having his PDA/Laptop

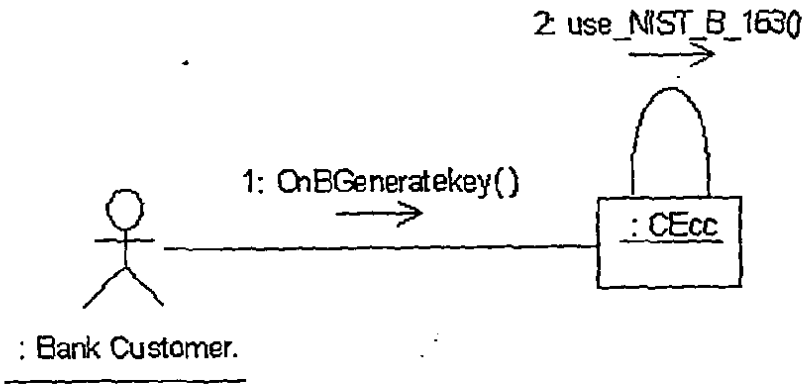


Figure 6.22 Collaboration diagram of Generate key by Bank customer

6.9.6 Post-Authentication between customer having PDA/Laptop and other mobile node.

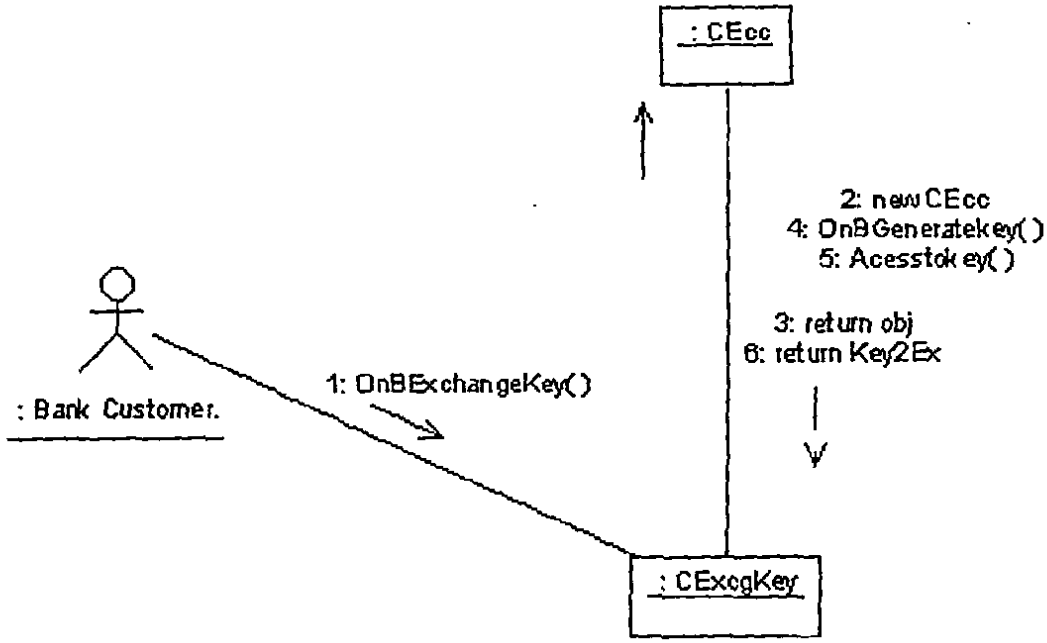


Figure 6.23 Collaboration diagram of Post-Authentication between customer having PDA/Laptop and other mobile node

6.9.7 Pre-authentication between PDA/Laptop and any Mobile ad hoc Node

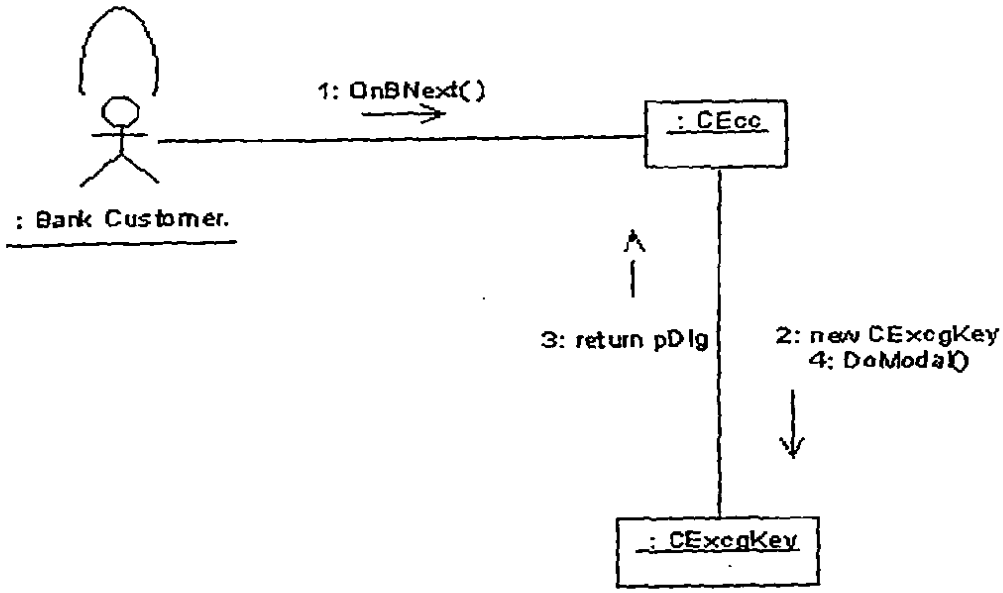


Figure 6.24 Collaboration diagram of Pre-authentication between PDA/Laptop and any Mobile ad hoc Node

6.9.8 Submit bill by Bank Customer

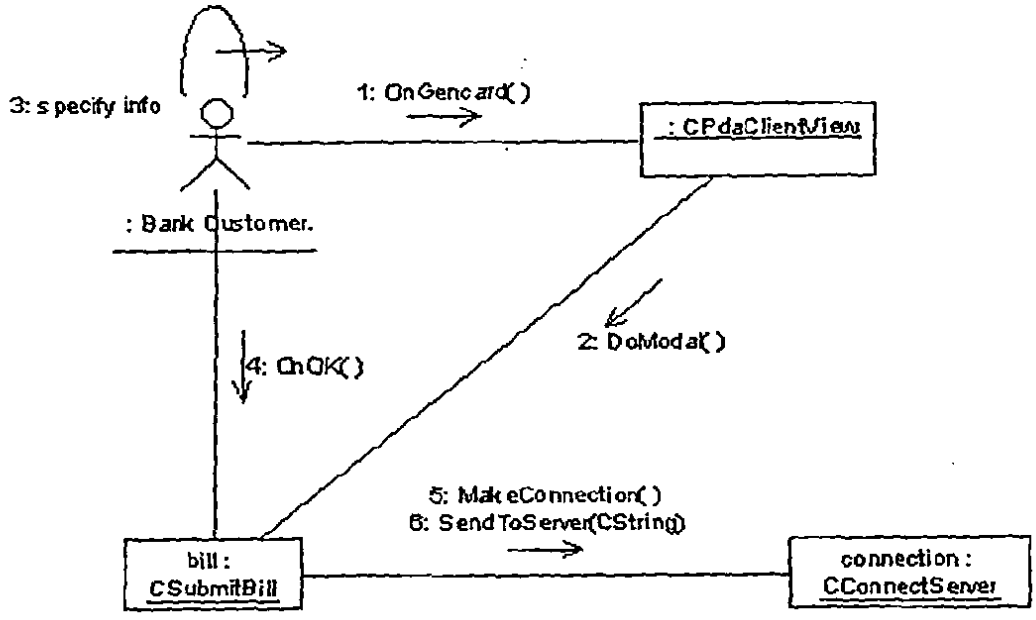


Figure 6.25 Collaboration diagram of Submit bill by Bank Customer

6.9.9 Transaction between two accounts

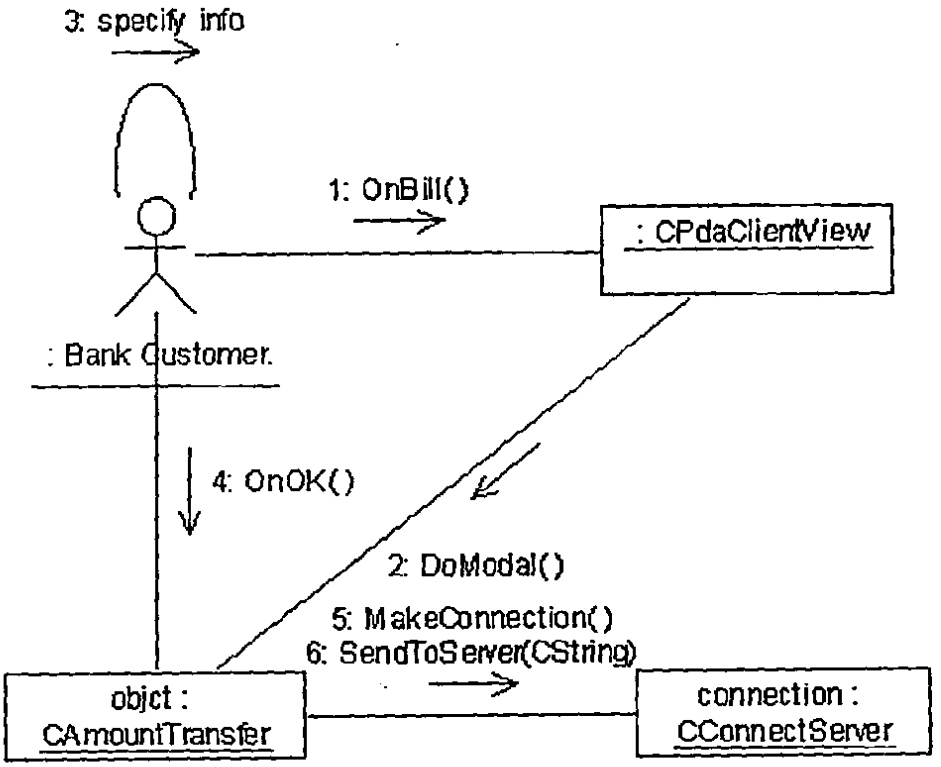


Figure 6.26 Collaboration diagram of Transaction between two accounts

6.10 Database Design

In this chapter we will discuss about the database design of Distributed Hotel Management System and we will also discuss about their relationship.

6.10.1 Entity Relationship Diagram (ERD)

First of all what is Entity? Any thing a person, a place or thing that have some attributes is called Entity and the entity relationship diagram (ERD) depicts relationship between data objects. The attributes of each data object noted in the ERD can be described using a data object description. The relationships between objects can be different according to the conditions .The relationships between data objects can be

- One-to-One (1:1)
- One-to-many (1:M)
- Many-to-Many (M:M)

Following are the data models for the objects of our domain and then the relationship between those objects.

6.10.2 Entities for Domain

Following are main entities for mobile adhoc networks.

- Bank User
- Login
- Accounts
- Credit Card
- Account type
- Transaction

6.10.3 ER Diagram of Wireless Bank Database

Figure 6.27 is given below in which ERD of Wireless Bank database showing Entities, their attributes and relationship between entities.

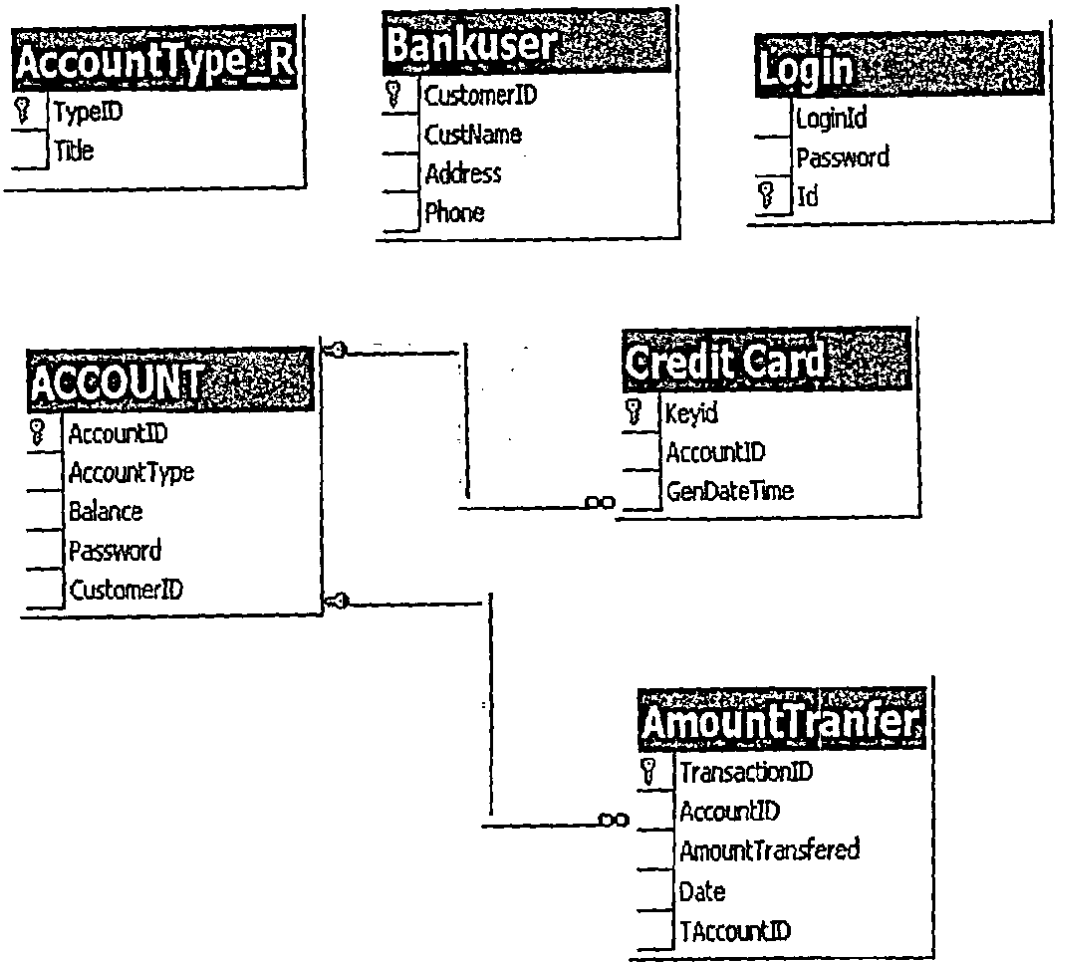


Figure 6.27 ERD of Wireless Bank database showing Entities, their attributes and relationship between entities

In Figure shown below there is detailed table view of Wireless Bank database showing Entities, their attributes, attribute's properties and relationship between entities.

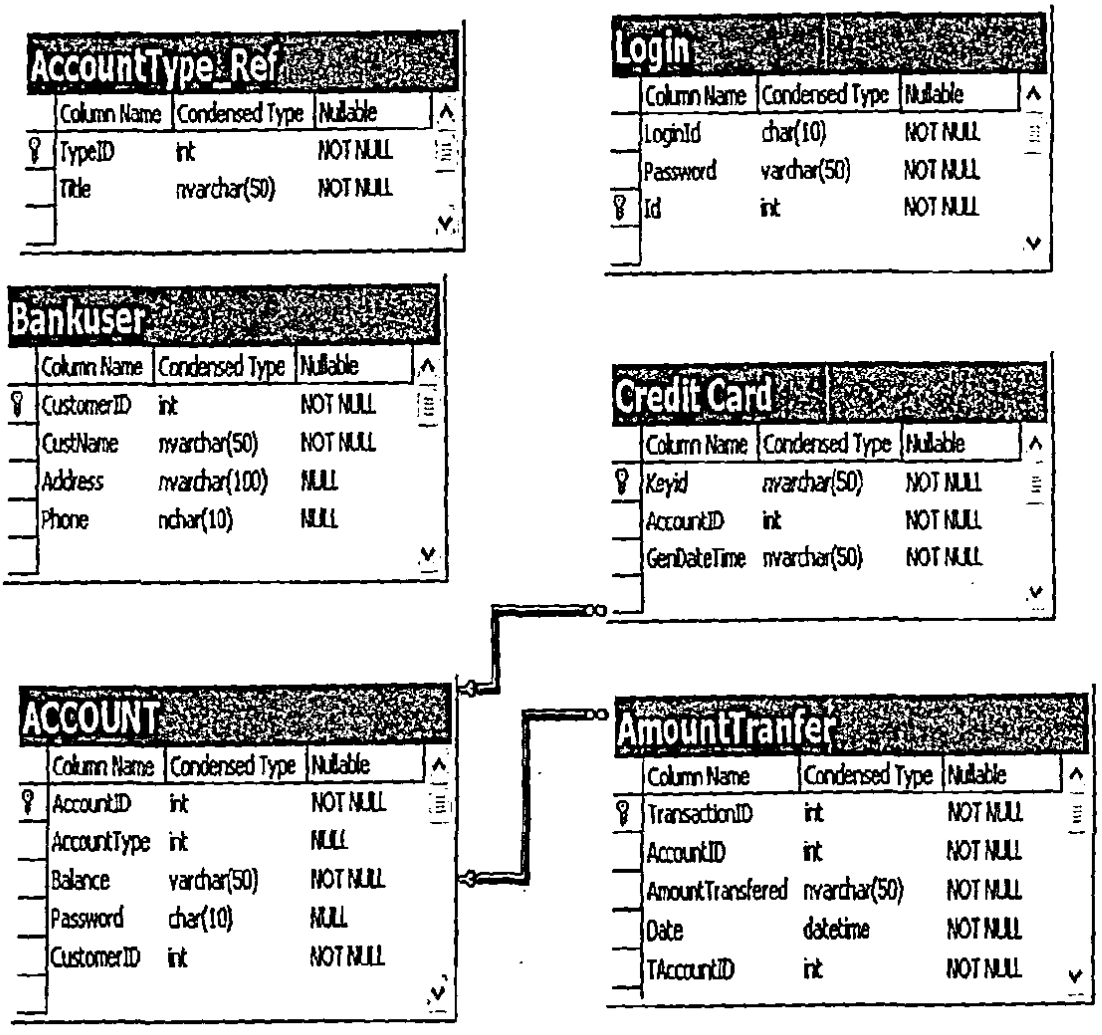


Figure 6.28 Detailed table view of Wireless Bank database showing Entities, their attributes, attribute's properties and relationship between entities

Chapter 7

IMPLEMENTATION

7. IMPLEMENTATION

This is very important phase in the software engineering pyradigm because no matter how efficiently analysis has been done? Or how brilliantly the design has been prepared? Although programming is an outgrowth of analysis and design, all the programming and implementation skill has to be applied here, because any inefficiency on part of the programmer will hammer the quality of the software. Another important aspect of this phase is that, although this phase is succeeded by the testing phase, but during the implementaion phase the programmer is best equiped for glass box testing of the software, because at this stage he has the access to the code.

7.1 Implementation Tools

Our software is developed using two tools Microsoft Visual C++ and Microsoft SQL Server and crystal reports.

7.2 Implementation of Some Important Functionalities

As previously described, there are five components to manage all the authentication and services provided in application. Let's look at each of the module's implementation details.

7.2.1 Key Exchange SSL Protocol implementation

We will call recvSecure() function when we have to receive key from other node. Similarly if we are sending key then we will call sendSecure() in our main application of mobile node as well as PDA client side. The implementation detail of both functions are given below.

7.2.1.1 RecvSecure()

```
bool SslServerImpl::recvSecure(unsigned char *data, int &nbytes)
{
if (!m_sock)
  {
  nbytes = 0;
  return false;
  }
```

```
if (!m_ssl)
  {
  nbytes = 0;
  return false;
  }
nbytes = SSL_read(m_ssl, data, nbytes);
if (nbytes <= 0)
  {
  nbytes = 0;
  return false;
  }
return true;

}
```

7.2.1.2 sendSecure

bool SslServerImpl::sendSecure(const unsigned char *data, int nbytes, CLog *cLog, ProgressMonitor *pm)

```
{
if (!m_sock)
  {
  return false;
  }
if (!m_ssl)
  {
  return false;
  }
```

```
int nWritten = SSL_write(m_ssl,data,nbytes);
if (nWritten <= 0)
  {
  return false;
  }
if (nWritten != nbytes)
  {
  return false;
  }
return true;
}
```

7.2.1.3 Socket routines implemented in SSL

```
bool CSocketImpl::connectSocket(const char *hostname, int port)
  {
  if (m_hSocket == INVALID_SOCKET) return false;
  SOCKADDR_IN sockAddr;
  ZeroMemory(&sockAddr, sizeof(sockAddr));
  sockAddr.sin_family = AF_INET;
  sockAddr.sin_port = htons((u_short)port);
  sockAddr.sin_addr.s_addr = inet_addr(hostname);
  if (sockAddr.sin_addr.s_addr == INADDR_NONE)
      {
      LPHOSTENT lphost;
      lphost = gethostbyname(hostname);
      if (lphost != NULL)
```

```
            sockAddr.sin_addr.s_addr = ((LPIN_ADDR)lphost->h_addr)->s_addr;
        else
          {
          return false;
          }
        }
    int retval = connect(m_hSocket,(SOCKADDR*)&sockAddr, sizeof(sockAddr));
    if (!retval)
        {
        return true;
        }
    else {
        return false;
        }
    }
bool CSocketImpl::isValid(void)
  {
  return m_hSocket != INVALID_SOCKET;
  }
void CSocketImpl::closeSocket(void)
  {
  if (m_hSocket != INVALID_SOCKET) {
        closesocket(m_hSocket);
        m_hSocket = INVALID_SOCKET;
        }
```

```
}
bool CSocketImpl::isReadable(void)
{
  if (m_hSocket == INVALID_SOCKET) return false;
  timeval timeout = {0, 0};
  fd_set fds;
  FD_ZERO(&fds);
  FD_SET(m_hSocket, &fds);
  int nStatus = select(0, &fds, NULL, NULL, &timeout);
  if (nStatus == SOCKET_ERROR)    {
      return false;
      }
  else {
      return true;
      }
}
bool CSocketImpl::sendOnSocket(const unsigned char *data, int nbytes)
{
  if (m_hSocket == INVALID_SOCKET) {
    return false;
    }
  while (nbytes)
    {
    int retval = send(m_hSocket,(const char *)data,nbytes,0);
    if (retval == SOCKET_ERROR)
```

```
            {
            return false;
            }
          nbytes -= retval;
          data += nbytes;
          if (retval == 0)
            {
            Sleep(50);
            }
          }
        return true;
      }
      bool CSocketImpl::receiveNBytes(unsigned char *data, int nbytes)
        {
        bool success = true;
        int waitCount = 0;
        while (nbytes)
          {
            bool timedOut = false;
          int n = nbytes;
          if (!receiveOnSocket(data,n,timedOut))
            {
            success = false;
            break;
            }
```

```
}
    else if (n == 0)
      {
      Sleep(100);
      waitCount++;
      if (waitCount > 20) break;
      }
    else
      {
      waitCount = 0;
      nbytes -= n;
      data += n;
      }
  }
 return success;
 }
bool CSocketImpl::receiveOnSocket(unsigned char *data, int &nbytes, bool
&timedOut)
{
 timedOut = false;
 if (m_hSocket == INVALID_SOCKET)
   {
   return false;
   }
 if (m_readTimeout)
   {
```

```
timeval tv;
tv.tv_sec = m_readTimeout;
tv.tv_usec = 0;
fd_set fdSet;
FD_ZERO(&fdSet);
FD_SET(m_hSocket,&fdSet);
int s = select(0,&fdSet,0,0,&tv);
if (s != 1)
  {
  if (s == 0)
    {
        timedOut = true;
        return false;
    }
  else if (s == SOCKET_ERROR)
    {
      m_hSocket = INVALID_SOCKET;
      return false;
    }
  }
}
int retval = recv(m_hSocket, (char *)data, nbytes, 0);

if (retval == 0)
    {
```

```
            m_hSocket = INVALID_SOCKET;
            return false;
            }
        else if (retval == SOCKET_ERROR)

            return false;
            }
        else {
            nbytes = retval;
            return true;
            }
}
bool CSocketImpl::listenOnPort(int port)
  {
  struct sockaddr_in sock;
  sock.sin_family = PF_INET;
  sock.sin_port = htons(port);
  sock.sin_addr.s_addr = 0;
  if (bind(m_hSocket, (struct sockaddr *) &sock, sizeof(sock)))
    {
    // Use GetLastError to get the last Windows error.
    return false;
    }
  if (listen(m_hSocket, 25))
    {
```

```
        return false;
        }
    return true;
    }
// If 0 is returned, there was an error.
CSocket *CSocketImpl::acceptNextConnection(void)
    {
    struct sockaddr_in fromsock;
    int nRC = sizeof(fromsock);
    SOCKET client;
    client = accept(m_hSocket, (struct sockaddr *) &fromsock, &nRC);
    if (client == INVALID_SOCKET)
        {
        return 0;
        }
    else
        {
        return new CSocketImpl(client);
        }
    }
```

7.2.2 ECC key Generation implementation

Generation of an ECC private key and public key is provided here. These two routines are called by all nodes as well as PDA Client.

```
ECPrivKey::ECPrivKey (const EC_Domain_Parameters& ecdp) {

        dp = ecdp;

        s = GenRandom (dp.m);

        s %= dp.r;

}

ECPrivKey& ECPrivKey::operator= (const ECPrivKey& sk) {

        if (this != &sk) { // avoid self assignment

                dp = sk.dp;

                s = sk.s;

        }

        return *this;

}

ECPubKey::ECPubKey (const ECPrivKey& sk) {

        dp = sk.dp;

        Curve E (dp.a, dp.b);

        W = E.mul (sk.s, dp.G);

}

bool ECPubKey::valid () {

        if (W.isZero ())

                return false;

        F2M x = W.x;

        F2M y = W.y;
```

```
F2M y2_xy = y*y + x*y;

F2M x3_ax_b = x*x*x + dp.a*x*x + dp.b;

if (y2_xy != x3_ax_b)

        return false;

Curve E (dp.a, dp.b);

Point O = E.mul (dp.r, W);

if (!O.isZero ())

        return false;


return true;

}
```

7.2.3 Hash Calculation Using MD5

All the nodes and PDA client having ECC key first calculate hash of their key by using MD5. Later this hash is used in Pre-authentication.

```
CString CMD5Checksum::GetMD5(BYTE* pBuf, UINT nLength)

{

        AfxIsValidAddress(pBuf,nLength,FALSE);

        //calculate and return the checksum

        CMD5Checksum MD5Checksum;

        MD5Checksum.Update( pBuf, nLength );

        return MD5Checksum.Final();

}
```

7.2.4 Functionality of services provided by PDACLIENT

All names of routines show their functionality. All functionality starting from Pre-authentication to services are given below.

```
void CEcc::OnBGeneratekey() {

CGenerateKey key;

key.GenEccKey();

m_FinalEccKey +='K';

m_FinalEccKey+=key.GetEccKey();

UpdateData(FALSE);

char* tkey = key.GetTempKey();

key.CalcHash(tkey);

m_Finalhash=key.GetHash();

XchangeHash();

m_BGen.EnableWindow(FALSE);

m_BNext.EnableWindow(TRUE);

CConnectServer con;

con.MakeConnection();

con.SendToServer(m_FinalEccKey);

}

void CConnectServer::MakeConnection(){

struct sockaddr_in sin;

sock=socket(PF_INET,SOCK_STREAM,0);

if(sock==INVALID_SOCKET)    {

        WSACleanup();

        return;
```

```
                }
                memset(&sin,0,sizeof sin);
                sin.sin_port=htons(m_Port);
                sin.sin_addr.s_addr=inet_addr("127.0.0.1");
                sin.sin_family=AF_INET;
        if(connect(sock,(struct sockaddr*) &sin,sizeof sin)==SOCKET_ERROR)  {
                        closesocket(sock);
                        WSACleanup();
                                AfxMessageBox("Connection Failed");
                                exit(0);
                        return;
                }
        }
        void CConnectServer::SendToServer(CString strin)
        {
        send(sock,strin.GetBuffer(strin.GetLength()),strin.GetLength(),0);
        }
        void CGenerateCard::OnOK()
        {
                UpdateData();
                CString string ="";
                string +='C';
                string+=m_AccountNo;
                string+="µ";
                string+=m_AmountRqd;
```

```
                    CConnectServer connection;
                    connection.MakeConnection();
                    connection.SendToServer(string);
                }
            }
                    void CLogin::OnSubmit()
                    {
                        CString m_str="";
                        m_str +="L";
                        UpdateData(TRUE);
                        m_str +=m_Login;
                        m_str +="µ";
                        m_str += m_Password;
                        MessageBox(m_str,NULL,MB_OK);
                        ConnectServer conn;
                        conn.MakeConnection();
                        conn.SendToServer(m_str);
                        conn.RecieveFromServer();
                        OnOK();
                    }
                    void CSubmitBill::OnOK() {
                        UpdateData();
                        CString string ="";
                        string+='B';
                        string+=m_org;
```

```
                string+="µ";
                string+=m_Amount;
                string +="µ";
                string+=m_Acno;
                CConnectServer connect;
                connect.MakeConnection();
                connect.SendToServer(string);
        }
```

7.2.5 Functionality implemented in mobile nodes and AdminBank

```
void CAdminBankView::OnStartapplication()
{
        if (bval) {
                bval = FALSE;
MessageBox("Application Started",NULL,MB_OK | MB_ICONINFORMATION);
                AuthenticateHash();
        }
        struct sockaddr_in sin;
        int retcode;CString name;CString pass;
        socket_descriptor=socket(PF_INET,SOCK_STREAM,0);
        memset(&sin,0,sizeof sin);
        sin.sin_family=AF_INET;
        sin.sin_port=htons(1112);
        sin.sin_addr.s_addr =INADDR_ANY;
        retcode =bind(socket_descriptor,(struct sockaddr*) &sin,sizeof sin);
        if(retcode==SOCKET_ERROR)
```

```
        MessageBox("Bind failed",MB_OK);
        listen(socket_descriptor,5);
        AfxBeginThread(ServThrd,this);
}
UINT CAdminBankView::TcpEcho(LPVOID dlg)
{
        CAdminBankView* d=(CAdminBankView*) dlg;
        d->StrtoSend="";
        while(1)
        {
                length = recv(d->new_socket,buff,sizeof(buff),0);
                buff[length]='\0';
                switch (buff[0])
                {
                        case 'L':
                        {
                                str1="";str2="";
                                for(int x=1;x<strlen(buff);x++)
                                {
                                if (buff[x] =='µ') break;
                                str1+=buff[x];
                                }
                                for(int y=x+1;y<strlen(buff);y++){
                                str2+=buff[y];
                                }
```

```
                    d->m_LogintoSet=str1;

                    d->m_PasswordtoSet=str2;

CString m_Logquery =_T("Select Id from Login where LoginId='" + d->m_LogintoSet + "'
and Password='" + d->m_PasswordtoSet + "'");

                    CValidate acc;

                    acc.Validate(m_Logquery);

                    CTransaction* pRs=acc.GetRecSet();

                    if (pRs->m_Id <=0)

                    d->StrtoSend+="0";

                    else

                    d->StrtoSend+="1";

                send(d->new_socket,d->StrtoSend,d->StrtoSend.GetLength(),0);

                break;

                }

                case 'T':

                {

                //same logic but in database changing accounts value

                }

                    case 'B':

                {

                    str1="";str2="";str3="";

                    for(int a=1;a<strlen(buff);a++)

                    {

                        if (buff[a] =='µ')

                            break;

                        str1+=buff[a];
```

```
                }
                for(int b=a+1;b<strlen(buff);b++)
                {
                        if (buff[b] =='µ')
                                break;
                        str2+=buff[b];
                }
                for(int c=b+1;c<strlen(buff);c++)
                {
                        if (buff[c] =='µ')
                                break;
                        str3+=buff[c];
                }
                d->m_org=str1;
                d->m_BAcNo=str3;
                d->m_BillAmnt=str2;
                break;
        }
        case 'K':
        {       tr1="";
                for(int N=1;N<strlen(buff);N++)
                {
                str1+=buff[N];
                }
                        d->m_RecievedKey = str1;
```

```
                    d->AuthenticateHash();
                    break;
                }
                case 'C':
                {
                    str1="";str2="";
                    for(int T=1;T<strlen(buff);T++)
                    {
                        if (buff[T] =='µ')
                            break;
                        str1+=buff[T];
                    }
                    for(int S=T+1;S<strlen(buff);S++)
                    {
                        str2+=buff[S];
                    }
                    d->m_CreditAccount=str1;
                    d->m_CreditAmount=str2;
                    d->MessageBox("Printer Not attached");
                        break;
                }
            }
        }//end of while
            return 1;
        }
```

```
UINT CAdminBankView::ServThrd(LPVOID dlg)

{

        struct sockaddr_in new_sin;

        int     addrlen =sizeof(new_sin);

        CAdminBankView* d=(CAdminBankView*) dlg;

        memset(&new_sin,0,sizeof new_sin);

        while(1)

        {

    d->new_socket = accept(d->socket_descriptor,(struct sockaddr*)
&new_sin,&addrlen);

                if(d->new_socket==INVALID_SOCKET)

                {

                        d->MessageBox("Invalid socket");

                        continue;

                }

                AfxBeginThread(TcpEcho,dlg);

        }

        return 0;

}




void CCreateAccount::OnSubmitok()

{

        UpdateData(TRUE);
```

```
m_query = _T("insert into Bankuser(CustName,Address,phone) values('" + m_Name +
"','" + m_Address + "','" + m_Phone +"')");

                Validate(m_query);

CString m_queryLogin = _T("insert into Login(LoginId,Password) values(' " + m_Login +
"','" + m_PassWordId+ "')");

                Validate(m_queryLogin);

                CString m_queryAccount;

        m_queryAccount = m_queryAccount + _T( "declare @AccountType int "
        +"declare @CustomerId int " +"set @AccountType = (Select TypeId from
        AccountType_Ref where Title = '" + m_Ac_Type + "') " + "set @CustomerId =
        (Select CustomerId from BankUser where CustName = '" + m_Name + "') "
        +"insert into Account values (" + "@AccountType" + "," + "'" + m_Balance + "',"
        +"'" + m_PassWordId + "'," +"@CustomerId )"

                Validate(m_queryAccount);

                }

int CValidate::Validate(CString query)

{// validation of Administrator

m_strCmdText = query; m_pRs=NULL;  m_piAdoRecordBinding=NULL;

::CoInitialize(NULL);

try{

m_pRs.CreateInstance(__uuidof(Recordset));

Open((LPCTSTR)m_strCmdText,(LPCSTR)m_strConnection,adOpenDynamic,adLockOp
timistic,adCmdUnknown);

if(FAILED(m_pRs-QueryInterface(__uuidof(IADORecordBinding),(LPVOID
*)&m_piAdoRecordBinding))){_com_issue_error(E_NOINTERFACE);

                        return FALSE;

                }

                m_piAdoRecordBinding->BindToRecordset(&m_rsRecSet);

        }

        catch(_com_error &e)
```

```
                {// error code

                }

                return 0;

        }

        void CValidate::OnOK()

        {

                UpdateData(TRUE);

m_query =_T("Select Id from Login where LoginId='" + m_Loginid + "' and Password='" +
m_Password + "'");

                Validate(m_query);

                CTransaction* pRs;

                 pRs = GetRecSet();

if (pRs->m_Id <=0){MessageBox("You are Not Valid User","Invalid
user",MB_ICONINFORMATION | MB_OK);

        OnBTNRes();

        }

else{

MessageBox("Authentication completed","Valid user",MB_ICONINFORMATION |
MB_OK);

        }

}
```

Chapter 8

Testing

8. Testing

System testing is an essential step for the development of a reliable and error-free system. Testing is the process of executing a program with the explicit intension of finding errors i.e., making the program fail and test cases are devised with the purpose in mind. A test case is a set of data items that the system processes as normal input. A successful test is the one that finds an error.

8.1 Testing Process

Test consists of a number of test cases, where different aspects of the part of the project under test are checked. Each test case tells what to do, what data to use, and what results to expect. When conducting the test, the results including deviations from the planned test cases are not in a test protocol. Normally a deviation indicates an error in the system (although some times the test case is wrong, and the system is right). An error is noted and described in a test report for removal or directly removed by the programmer who developed that part.

8.2 General Types of Errors

Error can be of following types:

- Functional error (e.g. function is not working correctly or missing).
- Non-Functional error (e.g. performance is slow)
- Logical error (e.g. error in algorithm, user interface errors is not considered as a logical error).

8.3 Testing Strategies

The following basic strategies were used for testing :

1. Specification Testing
2. Black Box Testing
3. White Box Testing
4. Regression Testing
5. Acceptance Testing
6. Assertion Testing
7. Unit Testing

8. System Testing

Each of the strategy are discussed as following

8.3.1 Specification Testing

Even if the code testing is performed exclusively, it doesn't ensure against progra failure. Code testing doesn't answer whether the code meets the agreed specificatio document. It doesn't also determine whether all aspects of the design are implemented.

Therefore, examining specifications stating what program should do and how it shou behave under various conditions performs specification testing. Test cases are develope to test the range of values expected including both valid and invalid data. It helps finding discrepancies between the system and its original objective. During this testi phase, all efforts were made to remove programming bugs and minor design faults.

8.3.2 Black Box Testing

In Black Box testing only the functionality was tested without any regard to the coc written. If the functionality, which was expected from a component, is provided the black box testing is completed.

8.3.3 White Box Testing

In White Box testing internal code written in every component was tested and it w checked that the code written is efficient in utilizing various resources of the system lil memory or the utilizing of input output.

8.3.4 Regression Testing

In Regression testing the software was tested against the boundary conditions. Vario input fields were tested against abnormal values and it was tested that the software do not behave abnormally at any time.

8.3.5 Acceptance Testing

In acceptance testing the software was checked for completeness that it is read Normally the quality assurance department performs the acceptance testing that t software is ready and can be exported.

8.3.6 Assertion Testing

In assertion testing the software is tested against the possible assertions. Assertio are used to check the program and various locations that whether the state of the progra at a particular point is the same as expected or not.

8.3.7 Unit Testing

In unit testing we checked that all the individual components were working proper Before integration of the entire components unit testing is essential because it gives

confidence that all the components individually are working fine and ready to b integrated with the other ones.

8.3.8 System Testing

When all the units were working properly and unit testing was performed then comes th time for system testing where we checked all the integrated components as a whole an looked for possible discrepancies, which could have arisen after the integration.

8.4 System Evaluation

The objectives of the system evaluation are to determine whether the desire objectives have been accomplished or not. Determining the merits and demerits of th proposed system over the existing system is also covered in the system evaluation. This i concerned with the detailed study of the developed system, from implementation point o view. At the end, some suggestions for the improvement of the system are coded.

8.5 Test Plan

Test plan provides an overview of the testing effort for the product.

8.5.1 Authentication & Authorization in MANET Test Plan

- **Introduction**

Our testing effort descriptions summarize IEEE 829-1983 for Software Test Documentatio which attempts to define a common set of test documents, to be used across the industry.

- **Features to be Tested**

Our Test Design Specification (Section 8.6) will summarize the features to be tested.

- **Item Pass/Fail Criteria**

Item will be considered as pass if the test lead to an expected result or the behavior of th module is according to expectations. If otherwise the item will be considered as failed.

- **Suspension Criteria and Resumption Requirements**

If the test leads to a BSOD (Blue Screen of Death) or a severe error message from th operating system or a lethal software crash, further testing will be ceased. The test will b redone, incase the bug can't be reproduced.

- **Test Deliverables**

  None

- **Environmental Needs**

  1. A Pentium® III or higher machine.
  2. Windows 2000 (Family)

3. Microsoft® Visual C++ 6.0
4. Microsoft® Platform SDK

8.6 Test Design Specification

This specifies how a feature or group of features will be tested according to Standar 829.

8.6.1 Authentication in MANET Design Specs

- **Features to be Tested**

This specification includes:

- Encryption , decryption & key Generation in ECC
- SSL key Exchange Protocol with embedded ECC
- MD5 Hash Calculation
- Exchange of Hash
- Exchange of Public keys
- Banking services which are used by protocol
- Testing of others features implemented in application

- **Test Identification**

Our Test Case Specifications (Section 8.7) will define each test associated with this design.

- **Feature Pass/Fail Criteria**

A feature or a combination of features will be considered as pass if the tests lead to a expected result, or the behavior of the module is according to expectations. If otherwise th feature will be considered as failed.

8.7 Test Case Specification

This defines a test case. According to the Standard 829, the test case specification include the following sections:

8.7.1 Encryption, decryption & key Generation in ECC

Key generation was done with different key sizes.

- **Output Specifications**

Dialog having Edit Box will show ECC key and its validity was checked in decryption.

- **Environmental Needs**
    1. A Pentium® III or higher machine.
    2. Windows 2000 (Family)
- **Special Procedural Requirements**

  Encryption & decryption requires Valid ECC key

8.7.2 Boundary Condition Test for banking service modules

All fields were tested in all services with respect to Boundary condition. Some problem were found and later corrected.

- **Output Specifications**

  All successful, valid and authenticated transactions were done.
- **Environmental Needs**
    1. A Pentium® III or higher machine.
    2. Windows 2000 (Family)

8.7.3 MD5 Hash Calculation

Hash calculation of ECC key was done.

- **Output Specifications**

128 bit hash was calculated..

- **Environmental Needs**
    1. A Pentium® III or higher machine.
    2. Windows 2000 (Family)
- **Special Procedural Requirements**

  It requires valid ECC key

8.7.4 SSL key Exchange Protocol with embedded ECC

Generated ECC Key was exchanged using standard SSL protocol with embedded ECC.

- **Output Specifications**

  Key was successfully exchanged on wireless media using wireless media.
- **Environmental Needs**
    1. A Pentium® III or higher machine.

2. Windows 2000 (Family)

- **Special Procedural Requirements**

  All nodes should have valid ECC key

8.7.5 Testing of others features implemented in application

Other features were tested by applying load test, boundary condition tests. Some bu were reported but later on corrected e.g. creation of Account in fig. 8.1

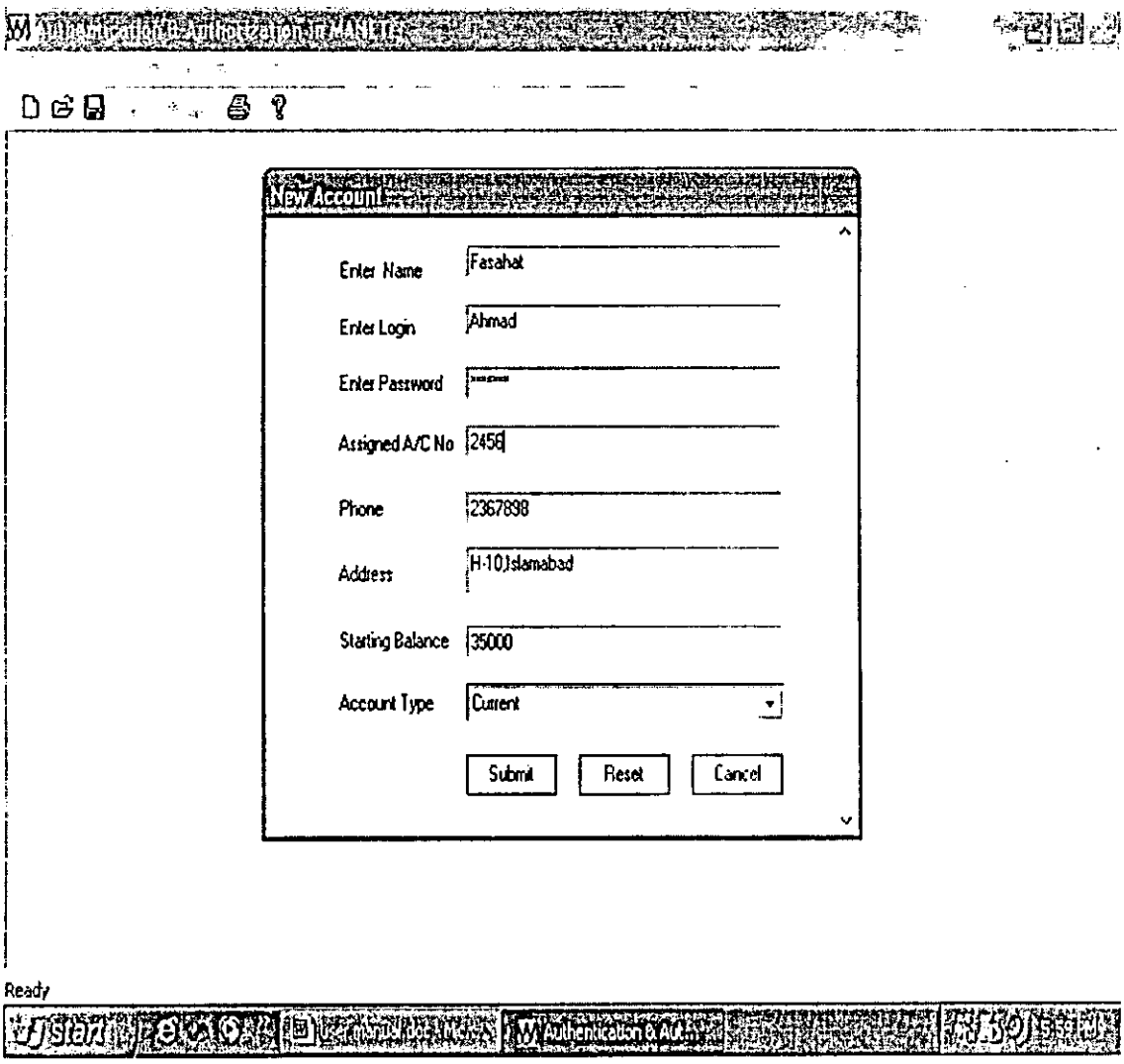


Fig 8.1 Dialog showing creating New Bank Customer Account.

Chapter 9

Conclusion

9. CONCLUSION

To motivate the development of ad hoc networking protocols, there needs to be applications where the properties of ad hoc networking are beneficial. Although some have been implemented many are still in the early research phase. Mobile ad hoc network is the latest field of research and its applications are limited to military tactical networks, personal area networks, sensor networks and disaster area networks. We have tried to introduce this field in banking mainly due to two reasons. WLAN is suffering security problems in industrial field and our approach will be effective alternative. Secondly there is no such application of MANET in industry so at least our approach will be foundation for finding MANET application in industries and institutions. For example we can deploy MANET in big shopping malls, Hospitals, Stock exchanges etc. It is hoped that the industry and financial institutions as a whole can progress forward more rapidly.

For all wireless ad hoc applications we have presented and generalized new approach for peer-to-peer authentication in mobile ad hoc networks. We have constructed our schema on previous work by Anderson, Stajano and others, and presented to perform pre-authentication over location-limited channels. Our schema is public key infrastructure less and resolves naming problem that plagues traditional authentication systems.

Unique location-limited channels: Instead of limiting the concept of imprinting a duckling device with its mother's secret key, we propose to use location-limited channel to bootstrap a wide range of key-exchange protocols. This approach can be experimented to audio, infrared, and contact-based channels, but other media are certainly imaginable.

Composed pre-authentication protocols: we provide a composed recipe for enhancing existing key exchange protocols with a pre-authentication step. And we explained how passive as well as active attacks can be detected by human user or by the system.

No dependency on Public key infrastructure: Key exchange and key agreement protocols depend on authentication step to verify the user. We have suggested a way to solve this problem. A reliance on pre-existing third party naming and trust infrastructures is unnecessary if one can briefly bring communicating parties within close physical proximity. In such a case, our pre-authentication protocols can be used in place of a PKI.

Bibliography & References

Bibliography & References

[1] C. Perkins, Ad Hoc Networking, Addison-Wesley 2001, ISBN 0201309769
[2] A. Menezes, P. van Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC Pr 1997, ISBN 0849385237
[3] W. Stallings, Cryptography and Network Security: Principles and Practice, 2nd ed., Prentice-H 1999, ISBN 0138690170
[4] A. Nash, W. Duane, C. Joseph and D. Brink, PKI: Implementing and Managing E-Securi McGraw-Hill 2001, ISBN 0072131233
[5] N. Asokan and P. Ginzboorg, "Key Agreement in Ad Hoc Networks", Computer Communicatio Volume 23, Pages 1627-1637
[6] J. Mackar and S. Corson, RFC 2501, "Mobile Ad hoc Networking (MANET): Routing Protoc Performance Issues and Evaluation Considerations", IETF 1999
[7] F. Stanjo, R. Anderson, "The Resurrecting Duckling: Security Issues in Ad-Hoc Wirele Networks", Security Protocols, 7th International Workshop Proceedings, Lecture Notes in Compu Science, Springer-Verlag 1999
[8] Saab Net Defence, Available on-line 2002-05-06 http://www.saab.se/future/node2567.asp
[9] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Next Century Challenges: Mobile Networking for *Sm Dust*", ACM Press 1999
[10] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks", IEEE Networks, Volume 13, Issue 6 199
[11] A. Shamir, "How to Share a Secret", Communications of ACM 1979
[12] H. Luo and S. Lu, "Ubiquitous and Robust Authentication Services for Ad Hoc Wirele Networks", Technical Report 200030, UCLA Computer Science Department 2000
[13] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing Robust and Ubiquitous Secur Support for Mobile Ad-Hoc Networks, IEEE ICNP 2001
[14] H. Luo, P. Zerfos, J. Kong, S. Lu and L. Zhang, "Self-securing Ad Hoc Wireless Networks", IE ISCC 2002
[15] J-P. Hubaux, L. Buttyán and S. Capkun, "The Quest for Security in Mobile Ad Hoc Network ACM 2001
[16] S. Garfinkel, PGP: Pretty Good Privacy, O'Reilly & Associates 1995, ISBN 1-56592-098-8
[17] S. Basagni, K. Herrin, E. Rosti and Danilo Bruschi, "Secure Pebblenets", ACM 2001
[18] D. Balfanz, D. K. Smetters, P. Stewart and H. Chi Wong, "Talking To Strangers: Authentication Ad-Hoc Wireless Networks", Internet Society, Conference Proceeding of NDSS Conference 2002
[19] N. Asokan, P. Ginzborg, "Key Agreement in Ad Hoc Networks", Computer Communicatio Volume 23, 2000
[20] M. Hietalahti, "Key Establishment in Ad-Hoc Networks", Laboratory for Theoretical Compu Science, Helsinki University of Technology 2001
[21] K. Becker and U. Willie, "Communication complexity of group key distribution", Proceedings 5th ACM Conference on Computer and Communications Security, ACM Press 1998.
[22] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, "SPINS: Security Protocols for Sens Networks", ACM SIGMOBILE 2001
[23] Pierre-Alain Muller, "Instant UML" WROX Press.
[24] Ivor Horton, "Beginning Visual C++ 6",WROX Press.
[25] Yashavant P. Kanetkar, "VC++, COM and Beyond", BPB publication.
[26] Beginning Windows NT Programming, WROX Press.
[27] Kate Gregory," QUE Special Edition Using Visual C++ 6", Macmillan Computer Publishing,
[28] Richard Grimes, "ATL COM Programmer's Reference" WROX Press, First Edition, 1998.
[29] Richard Grimes, "Professional ATL COM Programming" WROX Press, First Edition, 1998.
[30] Grimes, Stockton, Reilly and Templeman, "Beginning ATL COM Programming", WROX Pre First Edition, 1998.
[31] MSDN online, http://msdn.microsoft.com

Appendix A

User Manual

A-User Manual

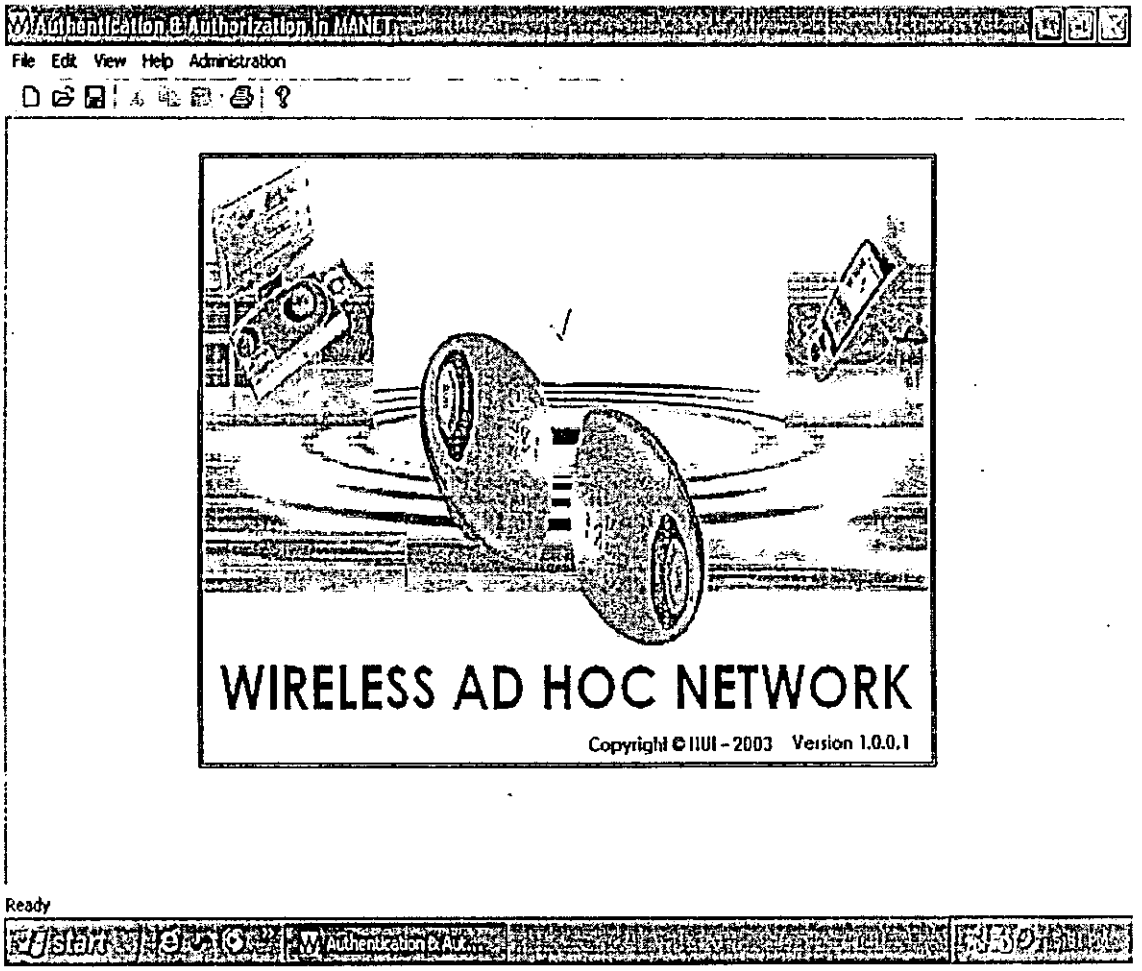


Fig A-1 Splash screen appearing in start of application.

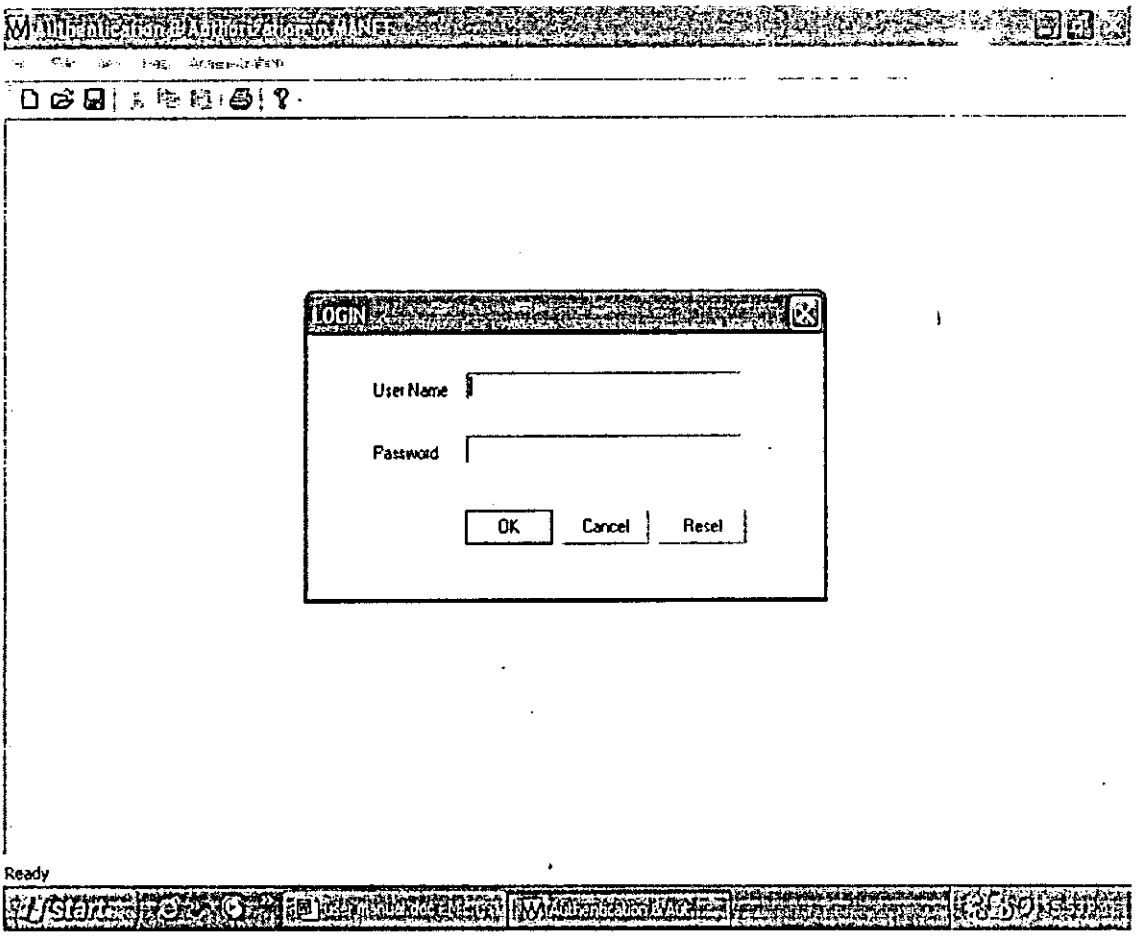


Fig A-2 Login dialog for Administrator.

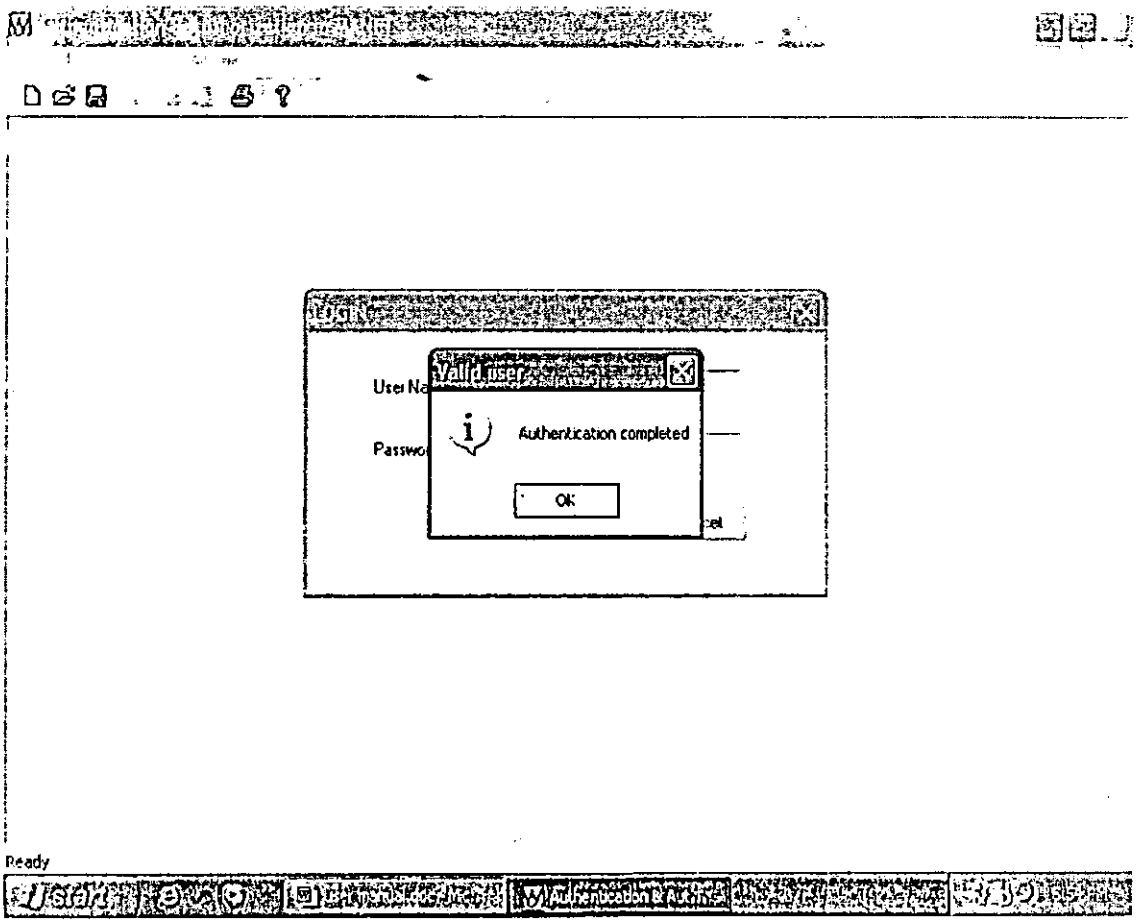


Fig A-3 Dialog showing validation of administrator.

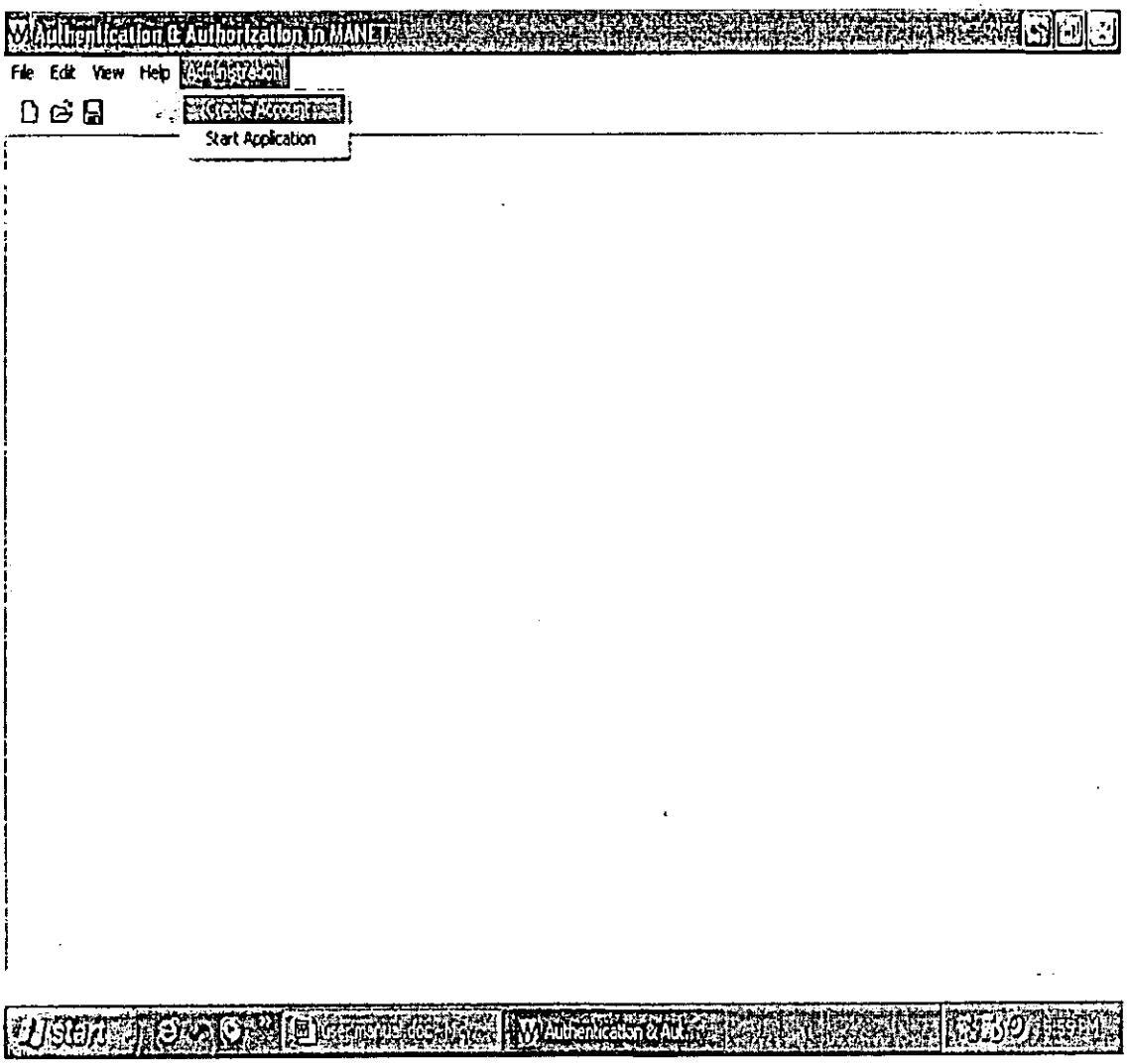


Fig A-4 Administrator creating new Bank Customer's Account.

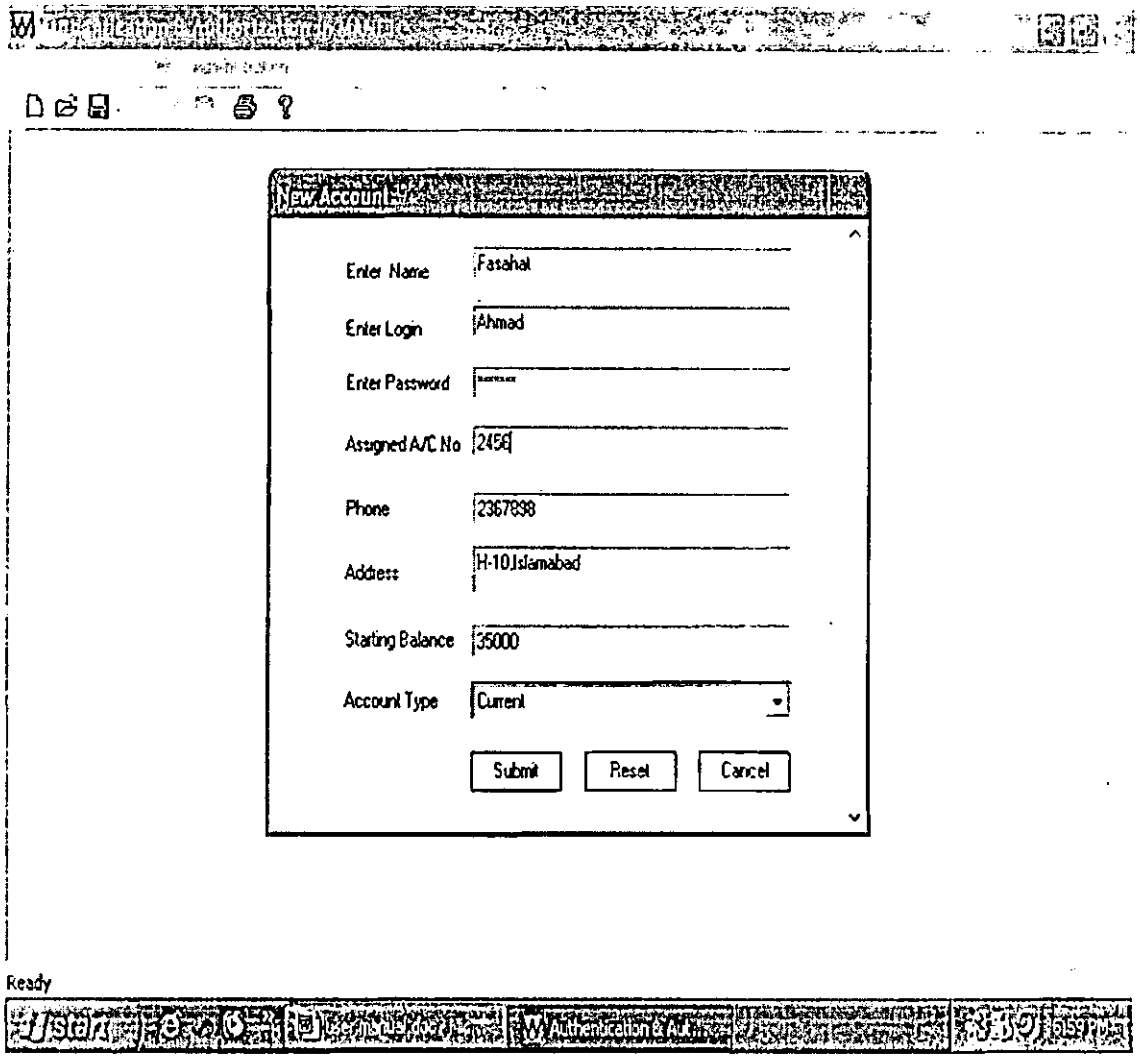


Fig A-5 Dialog showing creating New Bank Customer Account.

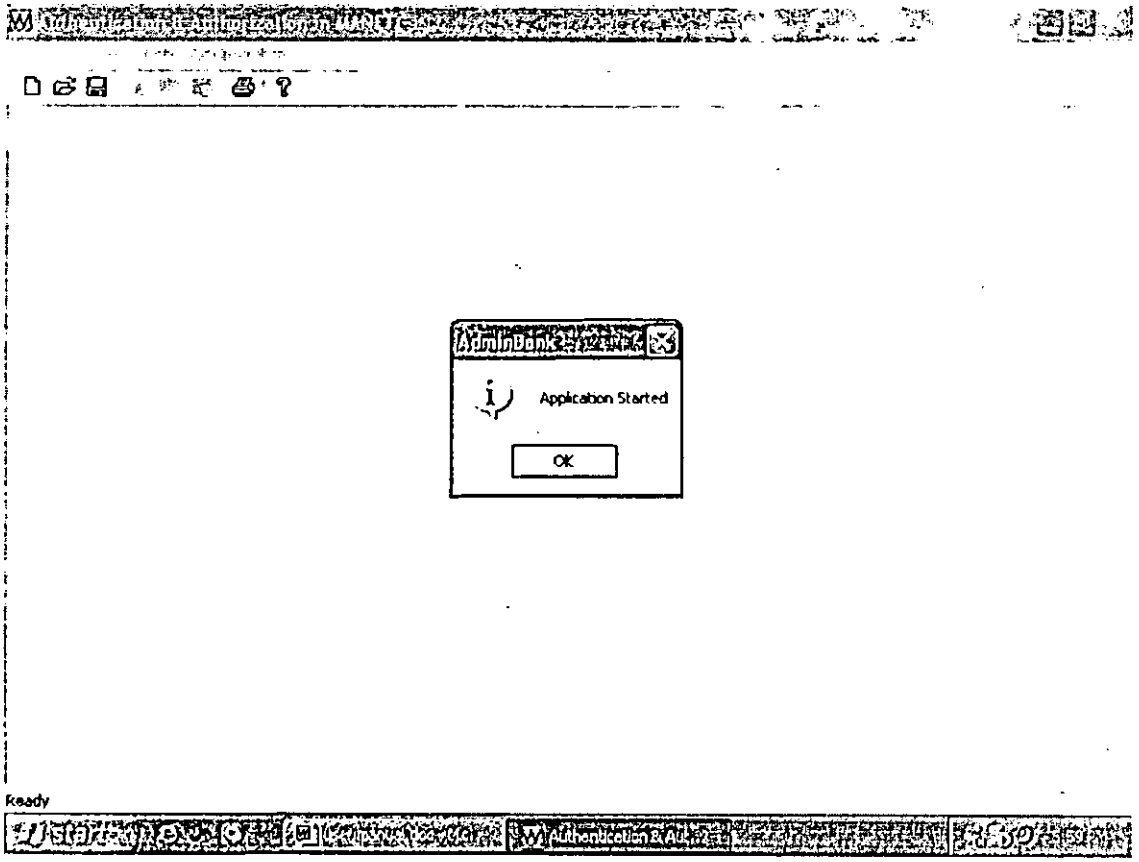


Fig A-6 Service Started by Administrator by clicking from submenu start application and now dialog is showing starting of application.

Appendix B

Proposed Research Modal and Published Work

Published in Pakistan Journal of Applied Sciences 3(5):346-359, 2003
ISSN 1607-8926

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

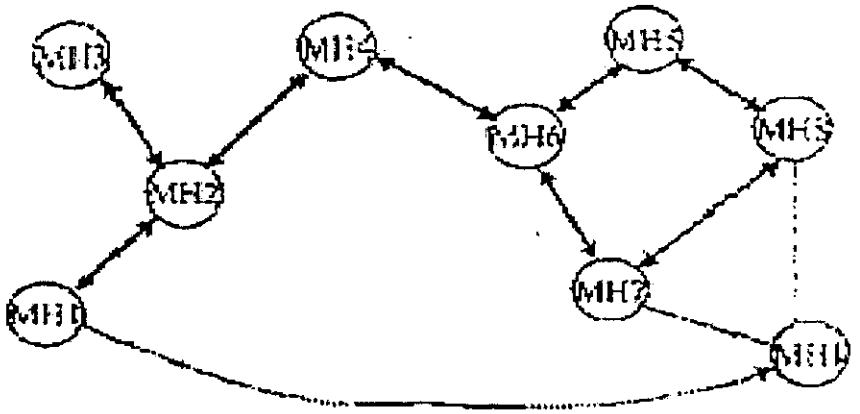


Figure1. A small MANET

Confidentiality, availability, integrity and authentication and non-repudiation are core attributes f communication in secure networks. These security requirements are identical for mobile ad h networks. Encryption and key exchange mechanism can be used to solve the security problems in hoc networks.

Most important characteristics in mobile ad hoc network include dynamic topologies, pow constrained operations, and bandwidth-constrained variable capacity link.

In wireless ad hoc networks security to be achieved is difficult [2] because:

- Wireless links are susceptible to passive as well as active attacks mainly due to open media.
- Nodes with inadequate physical protection can be compromised.
- Sporadic nature of connectivity.
- Dynamically changing topology.
- Absence of Certification authority.
- Lack of centralized monitoring or management point.
- Cooperative nature of algorithms.
- Sleep deprivation torture.

1.1 Threats

The most important threats that mobile ad hoc network have to face are [3]

i) Attack on basic mechanism of the ad hoc network, such as routing. Prevention of these attac requires security mechanisms that are often based on cryptographic algorithms. Routi mechanisms are more vulnerable in ad hoc networks than in conventional networks since in hoc network each device acts as a relay. This means, that an adversary who hijacks an ad h node could paralyze the entire network by disseminating false routing information. A l dramatic but subtler malicious behavior is node selfishness. Moreover, weakness in protoc can be exploited to perform malicious **neighbor discovery.**

ii) Attack on security mechanisms and especially on the key management mechanism. K management is certainly not a problem limited to ad hoc networks; however, because of t peculiarities of ad hoc networks, its solution requires specific attention. Examples of attacks security mechanism are: Public keys can be maliciously replaced; some keys can compromised; if there is a distributed trusted server, it can fall under the control of a malicio party.

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

1.2 Security services

To secure an ad hoc network, we consider following services: confidentiality, Authenticatic Integrity, Access control, Availability and Non-repudiation [4].

1.2.1 Confidentiality: confidentiality is protection of transmitted data from passive attacks. It ensur us that our information can't be disclosed to unauthorized entities. The other aspect of confidentiality the protection of traffic flow from analysis. This requires that an attacker is unable to observe the sour and destination, frequency, length or other characteristics of the traffic on a communications facili The challenge in confidentiality is not only protecting data transported by network but also data stor on device.

1.2.2 Authentication: It enables a node to ensure the identity of the peer node it is communicating wi Without authentication, an adversary could masquerade a node, thus gaining unauthorized access resource and sensitive information and interfering with the operation of other nodes.

1.2.3 Integrity: It assures that messages are received as sent with no duplication, modificatic reordering or replays.

1.2.4 Access Control: It is the ability to limit and control the access to host systems and applicatio via communication links. To achieve this control, each entity trying to gain access must first identified, or authenticated, so that access rights can be tailored to the individual.

1.2.5 Non-repudiation: It prevents either sender or receiver from denying a transmitted message. No repudiation is useful for detection and isolation of compromised nodes.

1.2.6 Availability: It ensures the survivability of network services despite denial of service attacks. denial of service attack can occur at any layer of ad hoc network. For example, on network layer adversary could disrupt the routing protocol. On higher layer, an opponent could corrupt high lev services too. One such target is the key management service, which is essential for any secur framework.

1.3 Roadmap

The rest of this paper is organized as follows: In Section 2 we will discuss the existing secur models for ad hoc networks and their scope as well. In Section 3, we will present our sect authentication model and its different characteristics. Section 4 will discuss the implementation banking scenario. Section 5 will briefly discuss future work plans and section 6 includes conclusion.

2. Existing Security Models in MANET

This section presents various security models for ad hoc networks. Each security model and scope will be discussed one by one.

2.1 Resurrecting Duckling Policy

Authenticity can be achieved in mobile ad hoc networks (which involve only two devices or le computing devices) by Resurrecting Duckling policy, which was introduced by Frank Stajno and Ro Anderson in [5] and extended in [6]. It is particularly suited to devices without display and embedding processor too weak for public-key operations. The fundamental authentication problem is a sect transient association between two devices establishing a master-slave relationship. It is secure in t sense that master and slave share a common secret and transient because the master can solve t association only. Also a master can always identify the slave in a set of devices.

The proposed solution is called the Resurrecting Duckling model. The duckling is the slave devi while the mother duck is the master controller. The duckling may be in one of the two states, *imprint* or *imprintable*, depending on whether it contains a soul or not; it starts (pre-birth) as imprintab becomes imprinted at birth when a mother duck gives it a soul, and it becomes imprintable again death, when the soul dissolves. The soul is a shared secret that binds the duckling to its mother: as lo

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

as the soul is in the body, the duckling will stay faithful to the mother and obey no one Resurrection is allowed,

the name of the policy suggests, but the duckling's ***metempsychosis*** works in reverse: instead of one soul inhabiting successive bodies, here we have one body hosting a succession of souls. The soul is originally transferred from mother to duckling over a non-wireless channel (e.g. electrical contact) in order to bootstrap the rest of the protocol. Death, which makes the duckling imprintable by a new mother, may be triggered by the conclusion of the current transaction or by a deliberate order from the mother duck ("commit suicide now!"), but not by one from an outside principal [5].

A metaphor inspired by biology will help us describe the behavior of a device that properly implements secure transient association. As Konrad Lorenz [7] beautifully narrates, a duckling emerging from its egg will recognize as its mother the first moving object it sees that makes a sound, regardless of what it looks like: this phenomenon is called ***imprinting***.

If several entities are present at the device's birth, then the first one that sends it a key becomes the owner:. As long as the soul stays in the body, the duckling remains alive and bound to the same mother to which it was imprinted. But this bond is broken by death: thereupon, the soul dissolves and the body returns in its pre-birth state, with the resurrecting duckling ready for another imprinting that will start a new life with another soul. Death is the only event that returns a live device to the pre-birth state in which it will accept an imprinting. There are also applications in which only part of the duckling's soul should perish. For example thermometer might be calibrated every six months, and the calibration information must not be erased along with the patient data and user key when the device is disinfected, but only when it is plugged into a calibration station.

During the imprinting phase, a shared secret is established between the duckling and the mother. The mother generates a random secret and encrypts it under the public key of the duckling, from which it gets back a signed confirmation.

In many applications there will only be one satisfactory, effective, cheap solution which is simple physical contact. When the device is in the pre-birth state, simply touching it with an electrical contact that transfers the bits of a shared secret constitutes the imprinting. No cryptography is involved, since the secret is transmitted in plaintext, and there is no ambiguity about which two entities are involved in the binding.

The extended work of the Resurrecting Duckling model also covers peer-to-peer cases [6]. The duckling will always obey its mother, who tells it whom to talk to through an access control list. The bond between mother and duckling is broken by death after which the duckling accepts another imprinting. Death may be caused by the mother itself, a timeout or any specific event. The whole security chain corresponds to a tree topology formed of hierarchical master-slave relationships. The root of the tree is a human being controlling all devices and every node controls all devices in its sub tree. However, if one relationship is broken the relationship to the whole sub tree is also broken.

This extended security model can be applied to very large ad-hoc networks, e.g. networks consisting of smart dust devices [8]. A possible scenario is a battlefield of smart dust soldiers (acting as slaves or siblings) and their general (acting as the master). The master allows its slaves to communicate by uploading in each of them a highly flexible policy so that sibling entities become masters and slaves for a very short time, enough to perform one operation. The mother duck gives the ducklings credentials that allow them to authenticate themselves.

2.1.1 Scope

The Resurrecting Duckling scheme is an appropriate model for a well-defined hierarchy of trust relationships. It particularly suits cheap devices that do not need a display or a processor to perform public-key operations. It perfectly works for a set of home devices. However, more flexible ad-hoc networks may not contain explicit trust relationships between each pair of nodes or to a centralized entity like the mother duck. Deploying a comprehensive network consisting of a hierarchy of a global mother duck and multiple subsidiary local mother duck is very similar to a public-key infrastructure, where the mother duck correspond to Certification Authority (CA), with all its advantages and

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

drawbacks. Even the battlefield scenario raises some problems. Here the soldiers are siblings and obe their mother, the general. If one soldier device wants to authenticate to another device it has to prese its credentials. The second device can then check the Credentials by using its policy. But what happe if all soldiers' do not use the same credentials, i.e. the same secret key to prevent it to be stolen by t enemy. If all devices use the same key the other side might invest considerable effort doing so physical attack [9] to recover the key because it would compromise all nodes. Since the devices cann hold a list of all valid credentials it seems that a further authentication method is needed.

2.2 Password-based Key Agreement

The work developed in [10] addresses the scenario of a group of people who wants to set up secure session in a meeting room without any support infrastructure. People physically present in t room know and trust one another personally. However they do not have any prior means of digital identifying and authenticating one another, such as shared secret or mutually verifiable public k certificate chains or access to trusted key distribution centers. An attacker can monitor or modify a traffic on the wireless communication channel and may also attempt to send messages purporting come from those who are inside the room. There is no secure communication channel to connect t computers. Desirable properties of a protocol that solves this problem should be:

- ***Secrecy:*** The basic requirement of secrecy is that only those entities who know an initi password should be able to learn the resulting session key. An observer must not be able to g any information about the session key.
- ***Perfect forward secrecy*** requires that an attacker who succeeds in compromising o member of the group and learns about his permanent secret information will still be unable recover the session keys resulting from previous runs of the protocol.
- ***Contributory key agreement*** The resulting session key is established by the contributi from all entities participating in the meeting. This ensures that if only one entity chooses contribution key randomly all other entities will not be able to make the key space smaller.
- ***Tolerance to disruption attempts*** The strongest attacker can disrupt any protocol jamming the radio channel or modifying the contents of messages among legitimate membe The protocol must not be vulnerable to an attacker who is able to insert messages. It is assum that the possibility of modifying or deleting messages in such an ad-hoc network is ve unlikely.

The work describes and introduces several password-based key-exchange methods that meet the requirements. A weak password is sent to the group members. Each member then contributes to part the key and signs this data by using the weak password. Finally a secure session key to set up a secu channel is derived without any central trust authority or support infrastructure.

Firstly, a well-known two-party protocol for password authenticated key exchange is describe based on the protocol called Encrypted key exchange (EKE) [11] and extends the work from two-par to multiple parties by electing a leader. The main drawback of the multiparty version is that the lead chooses the common session key unilaterally: the key agreement scheme is non-contributory. T protocol is then modified to extend it to a contributory multi-party protocol.

Secondly, Diffie-Hellman exchange (DH), a classic two-party key agreement protocol is extend to support multi-party password authenticated key exchange. An extension of unpublished protocol Steiner et al [12], in which each member shares a different password with an authentication server, used. This new protocol provides perfect forward secrecy to all players. It is also resilient disruptions. The leader can disrupt the protocol completely. Any other member attempting to disrupt t protocol by sending out a random quantity will not be able to compute the session key.

Finally, a fault-tolerant Diffie-Hellman exchange on a d-cube is extended to handle failures whi were based on the idea proposed by Becker and Wille [13].

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

2.2.1 Scope

Password-Based key agreement model perfectly works for small groups. Authentication is don outside the IT system, e.g. the group members authenticate themselves by showing their passport o common knowledge. This model does not suffice anymore for more complicated environments, though Groups of people who do not know each other or number of people who want to have confidentia exchanges without bringing in knowledge of the rest of the group be able to eavesdrop on the channel are two examples. Another problem arises for large groups or groups at different locations. The secur channel to distribute the initial password is not available anymore. It seems that existing suppor infrastructure is required to set up a secure channel.

2.3 Distributed Public-Key Management

Key management for established public-key systems requires a centralized trusted entity calle Certificate Authority (CA). The CA issues certificates by binding a public key to a node's identity. On constraint is that the CA should always be available because certificates might be renewed or revoked Replicating the CA improves availability. However, a central service goes against the distribute structure of ad-hoc networks.

L. Zhou and Z.J. Haas [14] proposes to distribute trust to a set of nodes by letting them share th key management service, in particular the ability to sign certificates. This is done using threshol cryptography [15]. An (n, t + 1) threshold cryptography scheme allows n parties to share the ability t perform a cryptographic operation so that any t + 1 parties can perform this operation jointly whereas i is infeasible for at most t parties to do so. Using this scheme the private key k of the CA is divided int n shares (S1, S2, Sn), each share being assigned to each special node. Using this share a set of t + special node is able to generate a valid certificate. As long as t or less special nodes are compromise and do not participate in generating certificates the service can operate. Even if compromised node deliver incorrect data the service is able to sign certificates. Threshold cryptography can be applied t well use signature schemes like the Digital Signature Standard (DSS) [16].

Another approach introduced in [3] presents a self-organized public-key infrastructure. The systen replaces the centralized CA by certificate chains. Users issue certificates if they are confident about th identity, i.e. if they believe that a given public key belongs to a given user. Each user stores a list o certificates in its own repository. To obtain the certificate of another entity the user builds a certificat chain using his repository list and implicitly trusted entity's lists, until a path to an entity that has th desired certificate in its repository is found.

2.3.1 Scope

The Threshold Key Management system is a way to distribute a public-key system. For high-valu transactions public-key systems are certainly the only way to provide a satisfactory and legal securit framework. Trust should, as much as possible, be based on knowledge. Since two entities that neve met before cannot have common knowledge – a shared secret – they both have to trust a central entity e.g. a CA to substitute this knowledge. When a user wants to prove his identity to CA, he goes ther with his public key and shows his passport. The CA proves his identity and then binds his identity to hi public key and signs the certificate. How it can be done when the CA is distributed? The user mus prove his identity to all special nodes to prevent that a compromised node passes on faulty information In this case friendship or just knowing each other is considered as common knowledge. The CA sign certificates without proving the identity, that's why; this model cannot be used for high-valu transactions.

The self-organized public-key infrastructure [3] shows similar problems. To make the systen bullet-proven, the entity's identity had to be checked in real world before users issue certificates Furthermore it is assumed that the certificate requester trusts each node in the recommendation chain Finally a significant computing power and time is consumed to obtain a certificate going through th certificate chain. Each node in the chain has to perform public-key operations, first to check th

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

received certificate for authentication (signature verification) and then to sign it before forwarding it (signature generation). This cannot be done in parallel but only one after the other.

Despite its centralized nature a central CA is preferable for applications with high-security demand. To ensure high availability the CA can be replicated. The replicated CA's are as secure as the original CA as long as the replication process is not vulnerable to attacks. The private key of the CA does not get weaker after replication. Much research has been done about efficient public-key systems – for example, a public-key system for mobile systems is presented in [17].

3. Proposed System Model

Imagine a situation in which bank customers are standing in a queue and two or three bank employees are dealing them in making transactions. The customers have to spend a lot of time and also have to wait for their turn in the queue. The bank needs more employees for giving better service to its customers. This scenario, both for the bank and the customers, can be improved by deploying ad hoc networking using Resurrecting Duckling Policy[5,6], with some extensions in our idea. Here, we have also taken inspiration from work of D. Balfanz [22].

Our idea is elaborated as: A user (Bank customer) enters the bank, with his laptop or PDA, to perform some transactions. The bank has mounted the Infrared Bar Code at one side in the bank and also owns a wireless radio link network (e.g. Bluetooth [18] or 802.11). The user or customer enters the bank premises, would walk up to the Infrared bar code and briefly establish physical contact between infrared bar code and his laptop or PDA. This process is termed as ***Pre-authentication***[1], which is done using a ***link-constrained channel***[2] [22]. During Pre-authentication, their public keys will be exchanged. Now the user can sit in the bank premises and his laptop or PDA can then perform standard SSL/TLS[19] key exchange with the bank server over the wireless link (e.g. Bluetooth[18] or 802.11), since he owns a secret key to perform secure transactions and can establish an authenticated and secret communication channel.

Such an exchange of pre-authenticated data ensures the user that he wants to communicate with the device (infrared bar code) by using a special, link-constrained side channel to exchange a small amount of cryptographic information. That information can be used to authenticate standard key exchange protocols performed over wireless link.

Once the pre-authentication is completed, the devices proceed to establish a secure connection between them over the main wireless link. To this end, they can use any established public-key-based key exchange protocol which requires them to prove possession of a particular private key (e.g., SSL/TLS, SKEME IKE, etc.), which will correspond to the public key committed to in the pre-authentication step.

After secret key exchange, the customer or user will send his/her account number along with the password or secret code (which can be issued at the time of opening an account in the bank and can be changed at any time through bank website) to the bank server. The A/C No. plus code or password will be encrypted with secret key (Session key) exchanged during pre-authentication. After this phase, the user can perform his transactions securely over the wireless link (e.g. Bluetooth[18] or 802.11).

We can summarize the basic scheme for pre-authentication as follows.

A) Pre-authentication over location-limited channel.

1- X $\longrightarrow$ Y: H (KU_x)
2- Y $\longrightarrow$ X: H (KU_Y)

B) Authentication over wireless channel with SSL/TLS.

[1] First phase in which duckling and mother exchange some secret information over location-limited channel.
[2] This notion of location-limited channel was introduced by Stajno and Anderson [3, 4], as a part of "Resurrecting Duckling" model for ad hoc networks.

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

X ⟶ Y: TLS_CLIENT_HELLO
..... And so on.

The various symbols denote:

X: Customer's Laptop
Y: Bar code device
KU_X, KU_Y: Public key belonging to X and Y respectively.
$H(KU_X)$, $H(KU_Y)$: one-way hash of encoding of corresponding keys.

3.1 Characteristics of Location-limited Channel

Communication technologies that have inherent physical limitations in their transmissions are good candidates. For example, audio (both the audible and ultrasonic range), which has limited transmission range and broadcast characteristics, can be used by a group of PDA's in a room to demonstratively identify each other. For situations that require a single communication endpoint, channels with directionality such as infrared are natural candidates.

The channel be impervious (or resistant) to eavesdropping. For example, Anderson and Stajano use secret data, such as a symmetric key, exchanged across the location-limited channel to allow participants to authenticate each other. As a result, that authentication protocol is vulnerable to a passive attacker capable of eavesdropping on the location-limited channel, thereby obtaining the secrets necessary to impersonate one of the legitimate participants. A location-limited channel used to exchange such secret pre-authentication data must therefore be very resistant to eavesdropping.

We therefore propose that any physically limited channel suitable for demonstrative identification, on which it is difficult to transmit without being detected by at least one legitimate participant (human or device), is a candidate for use as a pre-authentication channel. Such candidates include: contact, infrared, near-field signaling across the body, and sound (both audible and ultrasound). The amount of data exchanged across the pre-authentication channel is only a small fraction of that sent across the main wireless link, and so we can use channel media capable only of low data rates.

3.2 Advantages of our Approach

This idea of pre-authentication has been generalized to secure arbitrary peer-to-peer ad hoc interactions using a wide variety of key exchange protocols. We have introduced the use of Elliptic Curve Cryptography and can remove the secrecy requirements on link-constrained channels used to authenticate key exchange protocols. More importantly, it allows us to expand the range of key-exchange protocols, which can be authenticated in this manner to include almost any standard public-key-based protocol. As a result, this approach can also be used with an enormous range of devices, protocols and applications (as one of them is discussed).

This approach is significantly more secure than previous approaches; as we force an adversary to mount an active attack on the location-constrained channel itself in order to successfully subvert an ad-hoc exchange. Previous approaches (e.g., use of unauthenticated Diffie-Hellman key exchange) are either vulnerable to either active attacks in the main wireless channel, or, in the case of Anderson and Stajano [5, 6], to passive (eavesdropping) attacks in the location-limited side channel.

3.1.2 Advantages of Pre-authentication

Because legitimate participants would only communicate with entities from which they had received pre-authentication data, we would now require an attacker to perform an active attack – to be able to transmit –not only in the main wireless medium, but also in the location-limited channel.

Pakistan Journal of Applied Sciences 3 (5): 340-359,2003
ISSN 1607-8926

Because of the physical limitations of transmission on location-limited channels, it is significantly harder for an attacker to passively eavesdrop on them, not to mention to actively transmit.

For such an active attack to succeed, the attacker must not only transmit on the location-limited channel, but must do so without being detected by any legitimate participant.

The difficulty of monitoring a pre-authentication for such unwanted participation depends on the type of channel used and the number of legitimate parties involved. The more directed the channel and the smaller the number of parties, the easier it is to monitor. Note that, because of the physical limitations of the channels used and this monitoring requirement, it is only possible to use our techniques to pre-authenticate devices that are physically co-located at the time of first introduction.

3.3 Usage of Elliptic curve cryptography

Elliptic curve cryptography was proposed by Victor Miller[23] and Neal Koblitz [24] in the mid 1980's.Now Elliptic Curve Cryptography (ECC) has evolved into a mature public-key cryptosystem. Extensive research has been done on the underlying math, its security strength, and efficient implementations.

Elliptic curves are curves that are formed as a solution to the following equation:

$$y^2 + a_1 * x * y + a_3 * y = x^3 + a_2 * x^2 + a_4 * x + a_6$$

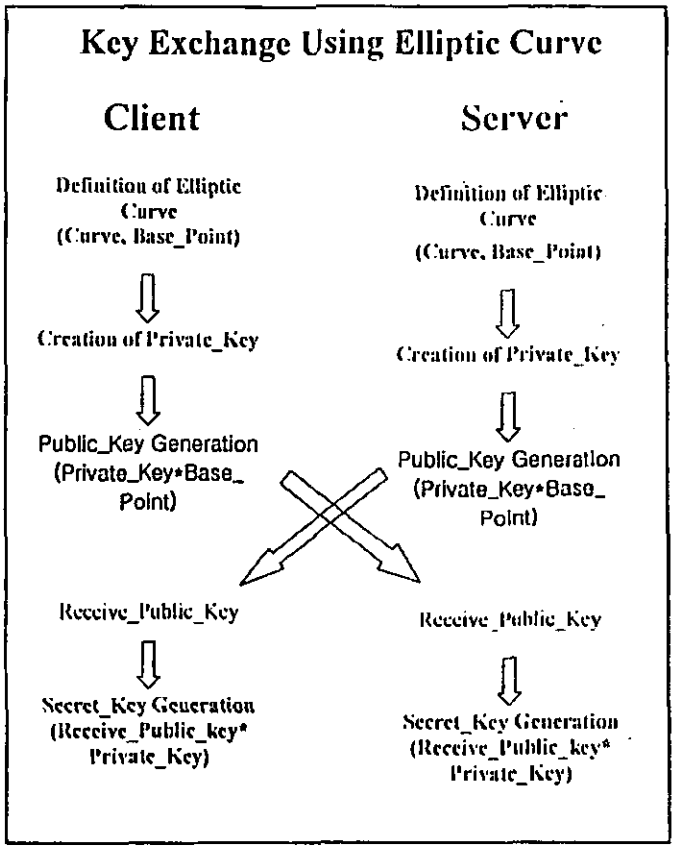
Where x and y form field elements, which can be complex, real, integers, polynomials, or normal basis. The field composed of points $P=(x_i, y_i)$ which solve the above equation form the foundation of elliptic curve cryptography.

At the foundation of every public key cryptosystem is a hard mathematical problem that is computationally infeasible to solve. For instance, RSA and Diffie- Hellman rely on the hardness of integer factorization and the discrete logarithm problem respectively. Unlike these cryptosystem that operate over integer fields, the Elliptic Curve Cryptosystems (ECC) operates over points on an elliptic curve.

Elliptic curve cryptography [23,24, 25, 26] provides a methodology for obtaining high-speed, efficient, and scalable implementations of network security protocols. The security of these protocols depends on the difficulty of computing *elliptic curve discrete logarithm* in the elliptic curve group. The group operations utilize the arithmetic of *points* that are elements of the set of solutions of an elliptic curve equation defined over a finite field. The arithmetic of elliptic curve operations depends on the arithmetic on the underlying finite field. The standards suggest the use of $GF(p)$ and $GF(2^k)$.

The security of ECC relies on the hardness of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP), which states that given P and Q = KP, it is hard to find k. While a brute-force approach is to compute all multiples of P until Q is found, k would be so large in a real cryptographic application that it would be infeasible to determine k in this way.

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926



3.4 Using ECC in SSL and its implementation in our Banking Scenario

In our approach, we have integrated ECC into the Secure Socket Layer Protocol. We chose SSL because it is the most popular and trusted security protocol on the Web. In the form of HTTPS (HTTP secured using SSL), SSL is single handedly responsible for the widespread adoption of e-commerce and many emerging wireless devices too now have SSL capabilities.

We have added ECC support to OpenSSL[27], the most widely used open source implementation of SSL. We have used the OpenSSL0.9.6b cryptographic library for our implementation. We selected ECC in place of RSA because ECC offers the highest strength per bit of any known public-key cryptosystem. ECC not only uses small keys for equivalent strength compared to traditional public key cryptosystems like RSA, the key size disparity grows a security needs increase. This makes it especially attractive for constrained wireless devices because smaller keys result in power, bandwidth and computational savings.

In this paper, we have proposed a secure, efficient and user-friendly solution to our bank transaction problem (and to the problem of authentication in local ad hoc wireless networks in general, for which our bank scenario merely serves as example.)

4. Implementation

We have divided our work in to three major areas. The first step is to implement exchange of pre-authentication data. We have exchanged hash of public keys. Hash code is generated by using SHA. Public/Private key pair is generated by using RSA but later we have implemented ECC [21] for this purpose, as it is more suitable for small devices.

Pakistan Journal of Applied Sciences 3 (5): 346-359,2003
ISSN 1607-8926

In second step, we have implemented SSL for complete exchange of keys and these keys are authenticated by using pre-authentication data.

For these two steps we have used socket programming. The server sockets listens for a connection on both location-limited channel and the primary link, but only admits primary-links connection from clients, who have performed pre-authentication on location-limited channel. Currently we have used serial cable for location-limited channel.

The third step is to transfer data from client to server. For this purpose we have utilized Microsoft Access for database and this whole framework is implemented in VC++6.

5. Future Work

The scheme presented has distinct advantages over traditional authentication and security models. We are focusing our future work on different scenarios relating to security and authentication problems, keeping in view the bandwidth-constrained media and computation limited devices.

Now we are going to implement our work by utilizing Bluetooth devices. To show actual transactions we are using PHP and MySql. Our work will mainly focus on upper layers (e.g. RFCOMM layer, SDP, L2CAP etc.) of Bluetooth by using open source code of Bluetooth device driver in Linux.

6. Conclusion

There is no any single authentication model which ensures authentication in every environment. While for a group meeting in small conference room the Password-based key exchange will work perfectly, for a network defined by hierarchy of trust relationships, the Resurrecting-Duckling policy is an ideal alternative. For guaranteed secure transaction certainly the choice is to use public key system involving the hassle of group certificates.

In this paper we have presented new approaches for peer-to-peer authentication in mobile ad hoc networks. We have constructed our schema on previous work by Anderson, Stajano and others, and presented to perform pre-authentication over location-limited channels. Our schema is public key infrastructure less and resolves naming problem that plagues traditional authentication systems.

Unique location-limited channels: Instead of limiting the concept of imprinting a duckling device with its mother's secret key, we propose to use location-limited channel to bootstrap a wide range of key-exchange protocols. This approach can be experimented to audio, infrared, and contact-based channels, but other media are certainly imaginable.

Composed pre-authentication protocols: we provide a composed recipe for enhancing existing key exchange protocols with a pre-authentication step. And we explained how passive as well as active attacks can be detected by human user or by the system.

No dependency on Public key infrastructure: Key exchange and key agreement protocols depend on authentication step to verify the user. We have suggested a way to solve this problem. A reliance on pre-existing third party naming and trust infrastructures is unnecessary if one can briefly bring communicating parties within close physical proximity. In such a case, our pre-authentication protocols can be used in place of a PKI.

References

[1] Charles E. Perkins, Ad hoc Networking, 2001, Addison-Wesley, London: ISBN: 0-201-30976-9.
[2]Yongguang Zhang, Wenke Lee, Intrusion Detection in ad hoc Networks.
[3] J.-P. Hubaux, L. Butty'an, and S. ¡ Capkun. The quest for security in mobile ad hoc networks. In Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobil hoc), October 2001.
[4] William Stallings, Network and Internet work Security Principles and practice.
[5] F. Stajano and R. Anderson, The Resurrecting Duckling: Security Issues for Ad hoc Wireless Networks. The 7th International Workshop on Security Protocols, LNCS 1796, Springer-Verlag, 1999.
[6] F. Stajano, The Resurrecting Duckling–what next? The 8th International Workshop on Security Protocols, LNCS 2133, Springler-Verlag, 2000.

[7] Konrad Lorenz. Er redete mit dem Vieh, den V"ogeln und den Fischen (King Solomon's ring). Borotha-Schoeler, Wien, 1949.
[8] B.Warneke, M. Last, B. Leibowitz, and K.S.J. Pister. Smart Dust: Communicating with a Cubic-Millimeter Computer. Computer Magazine, IEEE, Jan. 2001.
[9] R. Anderson and M. Kuhn. Tamper resistance – a cautionary note. 2nd USENIX Workshop on Electronic Commerce, 1996.
[10] N. Asokan and P. Ginzboorg. Key Agreement in Ad-hoc Networks. Computer Communications 23, 2000.
[11] Steven M. Bellovin and Michael Merrit. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In proceedings of the IEEE Symposium on Research in Security and Privacy, May 1992.
[12] Michael Steiner, Gene Tsudik, and Michael Waidner. Private communication. Unpublished work described in slides of presentation made at the ACM CCS conference, March 1996.
[13] Klaus Becker and Uta Wille. Communication complexity of group key distribution. In 5th ACM Conference on Computer and Communications Security, pages 1-6.
[14] L. Zhou and Z.J. Haas. Securing Ad Hoc Networks. IEEE Network Magazine, vol. 13, no. 6, 1999.
[15] Y. Desmedt and Y. Frankel. Threshold cryptosystems. Advances in Cryptology –Crypto '89, LNCS 435, Springer-Verlag, 1990.
[16] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. Advances in Cryptology – Eurocrypt '96, LNCS 1070, Springer-Verlag, 1996.
[17] G. Horn and B. Preneel. Authentication and Payment in Future Mobile Systems. Computer Security - ESORICS '98, LNCS 1485, Springer-Verlag, 1998.
[18] The official Bluetooth SIG website. www.bluetooth.com.
[19] T. Dierks and C. Allen. *The TLS Protocol Version 1.0.* IETF Network Working Group, the Internet Society, January 1999. RFC 2246.
[20] J. Lopez and R. Dahab, "An Overview of Elliptic Curve Cryptography," Institute of Computing, State University of Campinas (Brazil: 2000) Available from http://citeseer.nj.nec.com/
[21] M. Rosing, Implementing Elliptic Curve Cryptography, (Greenwich: Manning Publications Co., 1999).
[22] D. Balfanz, D. K. Smetters, P. Stewart and H. Chi Wong, "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks", Internet Society, Conference Proceeding of NDSS Conference 2002
[23] V. Miller. Uses of elliptic curves in cryptography. In H. C. Williams, editor, Advances in Cryptology — CRYPTO 85, Proceedings, Lecture Notes in Computer Science, No. 218, pages 417–426. New York, NY: Springer-Verlag, 1985.
[24] N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48(177):203–209, January 1987.
[25] A. J. Menezes. Elliptic Curve Public Key Cryptosystems. Boston, MA: Kluwer Academic Publishers, 1993.
[26] N. Koblitz. A Course in Number Theory and Cryptography. New York, NY: Springer-Verlag, Second edition, 1994.
[27] The OpenSSL Project, see http://www.openssl.org/.