A QOS-AWARE, SECURE ARCHITECTURE FOR SUPPORTING USER IDENTIFICATION AND USER MOBILITY

BY

VANISH TALWAR

B.Tech, Institute of Technology, Banaras Hindu University, Varanasi 1999

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2001

Urbana, Illinois

To Goddess Saraswati

Abstract

Pervasive Computing and Smart Spaces promise a future wherein the user experiences personalized service and ubiquity in computation. Among the many problems facing current pervasive computing research are the issues of secure user identification and secure user mobility support for services. Specifically, we desire an infrastructure that can support mutual authentication between the user and the computing device, and also provide seamless support of computation and service as the user moves from location to location. We would also like the solution to be location aware so as to automatically discover computing devices and service providers in the vicinity of the user. In this thesis, we have proposed an architecture to support User Identification and User Mobility using heterogeneous handheld wireless devices. This architecture is secure, scalable, QoS-aware and is implemented as a layered protocol stack. Each user has a personal handheld device which stores the identification certificate of the user used for mutual authentication with the service providers and computing devices. The handheld device also stores the state of the service or computation which is used for user mobility support. The Quality of Service is provided by varying the level of security in the system for better performance. Our solution includes a location system for automatically detecting new computing devices based on the location of the user. Such an architecture can be used by any application desiring seamless mobility support. We have specifically demonstrated the system for a video on demand application wherein the user desires to have seamless mobility support for watching a video as he/she moves from location to location. Overall, the thesis provides a novel architecture and mechanism for secure user identification and secure user mobility support to applications and thus is a contribution to the pervasive computing research.

Acknowledgments

I sincerely thank my advisor, Professor Klara Nahrstedt, for her constant guidance and support, encouragement and technical direction for completing this work.

I thank the National Science Foundation for the financial support under contract number NSF CCR 96-23867, DARPA Air Force grant under contract number F30602-97-2-0121, and NASA under grant number NASA NAG 2-1250.

I thank my colleagues at the MONET Group for a great working environment. Thanks are due to David for the discussions and help in the implementation of the cryptography libraries.

Thanks to my family for their constant love, encouragement, support and blessings.

Finally, I thank GOD for HIS divine grace.

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Notations

CA Certificate Authority

AS Authentication Server

HU Handheld Unit

SP Service Provider

VC Video Client

API Application Programming Interface

AKA Authentication and Key Agreement

JVM Java Virtual Machine

$(K_{pub})_X$ Public Key of X

$(K_{priv})_X$ Private Key of X

$Cert_X$ Certificate of X

$E_K(M)$ Encryption of message M using Key K

$D_K(M')$ Decryption of message M' using Key K

$H(M)$ One-way Hash of message M

$Sign_X(M)$ Digital Signature of message M by X

Chapter 1

Introduction

1.1 Fundamentals

The last several years have seen a tremendous growth in high speed networks, distributed systems, processor capability, memory capacity, embedded devices, pocket PCs, wireless technologies. The Internet has also enabled us to network all the computing devices together. We are now moving towards a new era of a world without wires. Current technology advancements have reached a stage where pervasive and ubiquitous computing can become a reality [34][35]. To make this happen, we are faced with challenges at all levels of computation. As efforts are being made to standardize wireless LAN technologies [13], lots of work is being done and needs to be done in ad-hoc wireless routing [29][22], Mobile IP[24] and transport layer issues [15]. There is also a need to build a software middleware infrastructure which could make intelligent use of the current technology advancements to provide the end user with a new experience of pervasive computing. The challenge is to build creative software middleware solutions which are efficient, scalable and extendable.

Researchers all over the world are proposing several solutions for such a ubiquitous computing environment [1][4][12][5][10][3]. Traditional as well as emerging computer manufactures are now rolling out pocketPCs and handheld devices in the post-PC era. Lots of efforts are now being made to build an intelligent software infrastructure for next generation smart appliances. As efforts continue to build a standard middleware software infrastructure which would work ubiquitously like the Internet, simultaneous efforts are being made to provide home and private intranet based middleware solutions using handheld devices. In this thesis, we attempt to provide such a solution to solve some of the problems facing pervasive computing research. Specifically, we are addressing the issues of secure user mobility support to services along with a mechanism to securely identify end users. Our solution solves both these issues using heterogeneous handheld devices along with a provision to vary the quality of security.

1.2 Motivation

With the increasing usage of computers in both industry and home environments, both software and hardware solutions are becoming standardized. The standardization of the software solutions is generating great interest to the computer science community. These standardized software solutions have been coined as 'services'. For example, we are increasingly hearing terms like 'Printing Service', 'Clock Service', 'Audio Service', 'Video Service' etc. These services are provided by service providers and could be invoked across a network. Typically each service has several geographically separated service providers. However, the current implementations of software solutions and services are 'machine-oriented' as against being 'user-oriented'. It is best to understand this with an example : Consider a video on demand service. Current explosion of broadband technology has made video on demand a reality. A video on demand service has a video server and several video clients. The video clients play the desired video and are the service providers to the end user. Suppose a user is sitting in his/her bedroom and requests for a particular video on the video client in the bedroom. However, the user may desire to move to the living room after a while and would like to continue to watch the same video from exactly the same frame where he/she stopped watching. Current implementations require the user to physically stop the video on the video client in the bedroom, remember the stopped frame, physically restart the same video on the video client in the living room, and then physically search for the stopped frame. It would be highly desirable to have a solution which would automatically discover video clients to the user, identify the user to the video clients, track the movement of the user and remember the video state information as the user moves from one video client to the other. In a practical deployment, we would also like all of the above to take place in a secure manner to prevent malicious intruders to masquerade as genuine users or allow them to gain access to secret video state information or modify any of the secret messages being passed on in the system.

There are several other examples where such an infrastructure is desirable. Some of them are audio service, file download service etc. We envision the need for all future services to have user mobility support. The need for user mobility support implies a mechanism to identify users. The challenge facing us is to devise a solution that is secure, feasible, scalable, application independent, cost-effective, QoS aware and works across heterogeneous platforms. We briefly describe these goals

1. Secure : Security is a very important concern in a practical deployment of a proposed solution. The solution should be able to provide sufficient security guarantees.

2. Feasible : The solution proposed should be easy to deploy in a real practical implementation.

3. Scalable : The solution should be scalable so as to work globally. It should also ensure that the existing users and service providers are unaffected by the addition of new users and/or service providers into the system.

4. Application or Service Independent : The solution should provide an interface to be used by any service or application. However, the internal implementation details of the interface should be independent of any application or service.

5. Cost-effective : We would like the solution to be as efficient as possible.

6. QoS aware : The proposed solution should be flexible enough so that the Quality of Service can be negotiated or guaranteed.

7. Heterogeneity : The proposed solution should work across heterogeneous hardware and software platforms. It should also be able to dynamically adapt to changes in operating environment.

In the next section, we describe our approach to the problem and goals.

1.3 Approach

We identify the following problems to be solved

1. Secure User Identification : We need a mechanism to identify users uniquely. We also need to ensure the authenticity of the identification mechanism. This leads us to the need for an authentication mechanism between the users and service providers.

2. Secure User Mobility : We need an infrastructure to track the movement of users in a secure manner. Further, this infrastructure should handle the secure transfer of service state information from one service provider to the other.

3. Location Awareness : We need our solution to be location aware. Hence, we desire an infrastructure to automatically discover service providers in the vicinity of the user.

We solve all the above problems using handheld devices. Each user is assumed to have a personal handheld device. These handheld devices could be heterogeneous with varying processing and memory capability. Our solution has a provision to adjust to such heterogeneity and also provides provisions for the application/user to vary the level of security guarantee for performance. We emphasize and justify the need for this provision in the later chapters of the thesis.

Our approach is to build a layered architecture. This has several advantages. Firstly, the design complexity is reduced and the functionality is very well defined. Each layer offers certain services to the higher layers through access points. Each layer communicates with the corresponding layer on the peer through an end-to-end protocol. The details of these protocols are defined in later chapters.

The entire system is viewed as consisting of several domains. Each domain has an authentication server, a set of service providers and a set of users. The user moves from one service provider to the other either within a domain or across domains.

The security in the system is based on public key infrastructure, certificates and session specific secret keys. The identification is based on certificates. The certificates for the user is stored on his/her personal handheld device. Our attempt is to make the security solution scalable and that the handheld devices and service providers should be as autonomous as possible. This implies that we would like to avoid solutions like Kerberos[26] and Sesame[9] requiring third party intervention for every session. An approximation to the Kerberos or Sesame like solution can be made by issuing certificates of fixed time period.

The tracking of the user is based on the idea of leases. Each user is given a lease to use the service from a particular service provider for a certain time period. The user can continuously renew the lease for continued service. If the user moves away from the current service provider, it's lease renewal with the old service provider fails and the user now obtains lease from a new service provider. The service state information is continuously pushed onto the user's personal handheld device. This information is then passed to the new service provider. We ensure that this entire process takes place in a secure manner. The details of this protocol, termed, Secure Leasing Protocol, are described in later chapters.

The location awareness of the system is based on the range of the wireless link. Currently, all the service providers in the range of the wireless card of the user's handheld device are assumed to be in the physical range of the user. However, future work can replace this mechanism by more sophisticated location systems.

The Quality of Security in the system is varied by varying the lengths of the keys. This is based on the fact that the security level guaranteed by current security protocols like RSA [28] increase with the increase in the key length. However, increasing the length of the key and hence the security level comes at a price of performance. We emphasize this further in later chapters as also our observations of additional tradeoffs between security and memory as also between security and bandwidth.

1.4 Contributions

The key contributions of this thesis are :

- We establish that the use of handheld devices for personal mobility support to services is a feasible solution.
- We develop a Secure User Identification system for securely identifying users to service providers. This identification system is based on sophisticated security protocols running on handheld devices and service providers and involves exchange of message information between the handheld device and service provider over a wireless medium.
- We develop a Secure User Mobility system for securely tracking the movement of the user and managing the secure transfer of state information from one service provider to the other. This system is based on a 'Secure Leasing Protocol' between the handheld device and the service provider.
- We develop a Location System to automatically discover service providers in the vicinity of

the users. This location system is based on the range of the wireless link of the personal handheld device of the user.

- We have provided a mechanism for changing the quality of security in the system. This enables us to support heterogeneous handheld devices of varying processing and memory capability, as also to tradeoff security for performance.

1.5 Preliminaries

It is desirable to discuss certain preliminaries before a detailed discussion of the system . In this section, we briefly present the background necessary for the subsequent chapters.

1.5.1 Introduction to Security Protocols

The security protocols aim to keep the messages and information secure. The following are the security services provided by modern day security protocols [25]:

1. Confidentiality : When Alice (nickname for the sender) sends a message to Bob (nickname for the recipient), both Alice and Bob may want the contents of the message to be kept secret from Eve (nickname for a hypothetical eavesdropper of unknown identity). The security service known as 'confidentiality' strives to ensure that Alice and Bob have the exclusive knowledge of the contents of the message. The confidentiality of messages is achieved using encryption and decryption.

2. Authentication : When Bob receives a message which purports to be sent by Alice, Bob wants to ensure that the message is actually coming from Alice, and not from Imp (nickname for a hypothetical impostor of unknown identity). If Alice is in fact the sender of the message Bob is receiving, she wants to convince Bob that she rather than Imp is the sender. The security service known as 'authentication' answers the question, 'Who sent the message?'

3. Message Integrity : When Alice sends a message to Bob, both Alice and Bob want to ensure that the message has not been modified between the time of sending by Alice and the time of receipt by Bob, neither by innocent computer glitch nor by Imp's malicious tampering.

The security service known as 'message integrity' answers the question, 'What message was sent?'. The integrity of messages is achieved using one way hash functions.

4. Non-Repudiation : When Alice sends a message to Bob and Bob acknowledges receipt of that message, they are potentially adverse to the other. Each will have the incentive to block the other party from a future denial of that party's own actions. Bob wants to ensure non-repudiation of origin, so that Alice cannot later deny the message was sent. Alice wants to ensure non-repudiation of delivery, so that Bob cannot later deny the message was received. The security service known as non-repudiation answers the question, 'Who is stopped from denying origin or receipt of a message?' . The non-repudiation of messages is achieved using digital signatures.

5. Time-Date Stamping : In the future, Alice and Bob also need to prove to third parties or to each other, 'When was Alice's message created by Alice, sent by Alice, or received by Bob?' The security service known as 'time-date stamping' provides this information.

The security protocols fall into two categories : secret key cryptography and public key cryptography. We briefly describe each category below [25]:

1. Secret Key Cryptography : With secret key cryptography, the same key (a stream of bits of a specified key length randomly created by a computer) is used to encrypt a message from plaintext (which can be read by anyone) into ciphertext (which can be read only by those possessing the key), and to decrypt the ciphertext back into plaintext again. The secret key cryptosystem has two inherent security weaknesses. First, the sender and the recipient of the message need to share knowledge of the same secret key, which means that each is required to trust the other not to compromise knowledge which is exclusively known by the two of them. Second, the sender and recipient have a key distribution problem. It is not possible to securely communicate the knowledge of a secret key to both parties who need it, without going offband (using a different channel). However, secret key cryptography algorithms are computationally less expensive than public key cryptography algorithms. Some of the popular secret key cryptosystems are DES [31] and AES[11].

2. Public Key Cryptography : A public-key cryptosystem offers a solution to both the shared key

risk and the key distribution problem. Rather than a single key, a public-key cryptosystem uses a pair of keys which are mathematically linked to each other, so that one key is used for encryption and the other linked key is used for decryption. Each user possesses a key pair, one of which (the private key) is kept secret and is never transmitted, and the other (the public key) is widely publicized. It is computationally infeasible to derive the secret key merely from knowledge of the public key. If Alice the sender wants to encrypt a message for confidentiality, Alice's encryption software will apply the recipient's public key. When Bob the recipient wants to decrypt the message, Bob will use the same encryption software to apply the recipient's private key (i.e., Bob's own private key). The security service of confidentiality (exclusive knowledge) is achieved. By reversing the process, it is possible to affix a digital signature to a message. If Alice the sender wants to digitally sign a message, Alice applies her sender's private key. When Bob the recipient wants to authenticate the origin of the message (i.e., from Alice), Bob applies the sender's public key (Alice's public key). This accomplishes the two security services of authentication (WHO sent the message) and non-repudiation (who is STOPPED from denying the message). In addition, when in digital signature mode, the security service of message integrity is simultaneously accomplished, so that it is certain that Alice's message has not been altered since it was digitally signed by Alice. Every character in Alice's message is run through a one-way hash algorithm. The hash result is then encrypted by applying the public key of Bob, the recipient. When Bob authenticates the origin of the message as coming from Alice, Bob also verifies the integrity of the message by (i) applying his own (Bob the recipient's) private key to decrypt the transmitted hash result; and (ii) comparing the decrypted hash result with a second hash result which Bob the recipient himself prepares from the transmitted message. With reversible public key cryptosystems such as RSA [28] , it is optionally possible to add the security service of confidentiality to digital signature and message integrity. Alice's software first digitally signs her message with Alice's private key, and then super-encrypts the signed message for privacy with Bob's public key. Bob decrypts the signed message with Bob's private key, and then verifies that the message was signed by Alice (and not since altered) with Alice's public key. The potential weak spot in the public key cryptosystem is the possibility that Imp, the impostor, will manage to secretly substitute Imp's own public key (Imp has the matching private key) for the public key of another person, which would nullify or weaken all of the security services offered by the public key cryptosystem. To plug this potential compromise of security, trusted third parties, called Certification Authorities ('CAs'), are being developed to bind a key-pair to the identity of its owner. Some of the popular public key cryptosystems are RSA[28] and Elliptic Curve Cryptosystems[2][31].

In our system, the keys are generated by trusted third party entities, called authentication servers. These authentication servers also act as certificate authorities and create a digitally signed certificate. In normal practice on the internet, the public/private keys are generated by the user and the CAs certify them. However in our case, the user is only equipped with handheld devices of computational power. So we would like a relatively powerful trusted machine to generate keys for the user. Such a trusted machine is the authentication server. The later chapters will explain our system in more detail.

1.6 Thesis Outline

In Chapter 2, we describe the related work done in this area. In Chapter 3, we describe the detailed design of our system. We describe an implementation of the system for video on demand application in Chapter 4 and discuss it's performance in Chapter 5. Finally, we conclude this work and discuss future research directions in work in Chapter 6.

Chapter 2

Related Work

In this chapter, we describe the related work done in the areas this thesis is addressing. As described in the Introduction, this thesis is addressing two main issues : User Mobility and Security in the context of mobile computing. We first describe the related work in these two areas and then describe the Mobile ID protocol [38] which provides user mobility support using Active Badges.

2.1 Related Work on Security in Mobile Environments

Most of the related work for security in mobile environments has been in the context of cellular systems. Of late, security in ad-hoc networks and in home mobile environments are also being studied. However, no existing system matches the exact needs of our system. We describe these needs as we describe the related work in subsequent subsections. The need for security in mobile environments is greater than wired environments due to the easy access of communication channel to the intruders. Some of the important issues which need to be addressed in designing a security scheme for mobile communications has been described in [16] [36]:

1. Autonomy of communicating entities : In mobile environments, the wired workstations, the base stations and mobile nodes could be governed by different organizations, and it is possible for a mobile node from anywhere in the world to walk into the cell of any basestation. So we cannot assume an authenticating authority that can certify all the nodes.

2. Mobility of the users : The mobile users need to be authenticated upon entering each cell. The handoffs should be as efficient as possible.

3. Timestamping : Time synchronization is difficult in presence of mobility. There should be an implicit form of timestamping.

4. Location privacy : The identity of the communicating entities may have to be protected.

5. Power consumption : Mobile nodes typically run on battery and would like to consume as minimum power as possible.

6. Processing and Memory capability : The mobile nodes are much less powerful in terms of processing and memory capability. So the complexity of computations for achieving authenticity and security should be kept lower.

We now briefly describe the related work done to address some of these problems.

2.1.1 Security for Cellular Systems

The general model of Personal Communication Systems (PCS) consists of a user's handset, user's 'home network' and user's 'visited network'. The general model for Authentication and Key Agreement (AKA) process consists of three parts [17] :

1. Provisioning : This is the means by which the handset or the user acquires the bonafide that will enable the network to subsequently recognize him as a legitimate user.

2. Transfer user specific data to service provider : Part Two of the model is the means by which a handset establishes credibility when the user registers with a local service provider. If the local service provider is not in the 'home network', it should only acquires a byproduct or subset of handset credentials.

3. Authentication and Key agreement : The third part of the model is the protocol that is executed to permit network access and establish a key for protection of channel data.

The cellular systems GSM and CDMA which are in vogue today, use secret key systems to achieve the above. [17] gives a good introduction and comparison of the techniques used by GSM and CDMA for the AKA process. The key ideas of the techniques is described below.

In GSM-style systems, the service providers control the security process by issuing a 'Subscriber

Identity Module' (SIM) to the user. The SIM takes the form of a credit card-like device intended for insertion in the handset. The SIM contains a 128-bit number, called K_i, that is unique for each SIM. K_i enables the SIM to authenticate itself to the network. This K_i is the secret key, shared between the service provider and the handset. In CDMA systems, the subscriber-specific information is downloaded into a user's handset by electronic means. For more details, the reader is referred to [17].

When the user moves to a network which is NOT the 'home network', the service provider of the 'visited network' would need to obtain the shared key information from the 'home network' of the user. However, the 'home network' would not like to disclose the shared information directly because doing so could lead to a compromise of security. In GSM systems, this is achieved by sending "triplets" that provide security information without revealing the secret key K_i. In IS-41 CDMA systems, the 'Shared Secret Data' (SSD), which is derived from the security parameter 'A-Key', is sent by the home network to the visited network. The authentication process then takes place through a Challenge/Response mechanism.

Thus, in both GSM and IS-41 systems, the visited network security manager is subordinate to the home security manager. [32] notes that this is an example of centralized security management system, where the home network security manager administers the whole global network security in a concentrated way. In [32], an authentication technique, based on the distributed security management for the Global Mobility Network, is developed. This method assumes that the authentication key K_h is shared between the user and the home network, and that the authentication key K_n is shared between the visited network and the home network. A temporary security manager is generated for a roamer in the visited network in order to provide roaming services. The authentication technique consists of two phases. In the first phase, the authentication control to setup the roaming service environment is negotiated between the visited network, home network and roamer. In the second phase, the authentication control to provide the roaming service is negotiated only between the visited network and the roamer.

In our system, we would like to use public key cryptography for authentication and key agreement process. Also, we would like to devise a mechanism that eliminates the need for the 'visited network' to contact the 'home network' to authenticate a mobile user. We achieve this by designing a two level hierarchy of authentication servers which also act as certificate authorities in our system. We also use temporary certificates in the visited network similar to the idea in [32]. The details of our mechanism is explained in subsequent chapters.

2.1.2 Security for Ad-hoc networks

Ad-hoc networks are a new paradigm of wireless communication for mobile hosts [37]. In an ad-hoc network, there is no fixed infrastructure such as base stations or mobile switching centers. Mobile nodes within each other's radio range communicate directly via wireless links, while those that are far apart rely on other nodes to relay messages as routers. The challenges facing ad-hoc network security are [37] :

1. The wireless links renders an ad-hoc network susceptible to link attacks ranging from passive eavesdropping to active impersonation.

2. Nodes roaming in a hostile environment with relatively poor physical protection, have non-negligible probability of being compromised.

3. An ad-hoc network is dynamic because of frequent changes in both its topology and its membership.

Efforts at securing ad-hoc networks are focussed on authentication and secure routing. [37] presents a security scheme based on threshold cryptography where a set of nodes act as CA. This is because a single CA would be a vulnerable point in the network.
In our system, we assume each domain to be an ad-hoc network governed by an authentication server. This authentication server is assumed to be secure and trusted and will not be compromised. Our system also faces the challenge of devising a solution which should work across different ad-hoc networks.

2.1.3 Security schemes for handhelds and mobile computers

As handhelds and Pockets PCs are becoming popular, efforts need to be done in addressing security solutions for these devices. The handheld devices are less powerful in processing and memory capability than desktop PCs. However, we can still load software programmable solutions onto them. There is currently work being done at the University of Illinois with the development of Tiny Sesame [14], which is a lightweight, component-based security mechanism that is suitable for mobile devices. Tiny Sesame is a subset of UIUC SESAME that supports authentication, integrity checks, simple encryption and RBAC (Role Based Access Control). UIUC SESAME [20] is a portable version of SESAME [9] implemented completely in Java. SESAME (A Secure European System for Applications in a Multi-vendor Environment) [9] is an extension to Kerberos[26] that provides additional services. These services include digital signatures for login, preventing offline dictionary attacks, handling of access control privileges for users and supporting different key management and distribution schemes. Since SESAME is based on Kerberos, it requires a third party Security Server to issue keys for each session. In our system, we would like the handhelds and service providers to be able to authenticate each other independently of a third party server. The Authentication Server is contacted by the handhelds and service providers for registration. The registration process involves generating public/private keys and certificates. These certificates are then used for the authentication process. The session keys are always generated by the service provider. We assume that the service providers can be trusted to generate session keys. In addition, the certificates can be given an expiry date which would require re-registration with the authentication server. Also, we would like to control the Quality of Security in the system by varying the key lengths. However, we would like to use the lightweight and dynamic component loading mechanisms used in Tiny SESAME. Also, like SESAME, we would like to extend the system to support multiple security protocols and the system could adaptively choose one of them.

2.1.4 Quality of Security

There are efforts underway to consider security as a Quality of Service parameter while designing distributed systems. This is due to the tradeoff between security and performance. [23] provides a discussion on this topic. This paper also discusses various design implications regarding security ranges provided in a QoS aware distributed system. Another important point is the need to have application level security while designing systems. We have provided a way to negotiate Quality of Security in the system for performance and feasibility. In the current implementation, the security parameters are specified by the user but the system could be extended to dynamically adjust the level of security based on application and network environment.

2.2 Related Work on User Mobility

As handheld devices are becoming popular, efforts are being made for user mobility support to distributed applications. [18] presents some of the issues and experiences in supporting user mobility. As noted in [38], user mobility implies both terminal mobility and personal mobility. Terminal mobility is defined as the possibility to move the terminal while keeping the service active. It allows the users to travel with their terminals [19]. In Personal mobility, users are not tied to their terminals; the users can travel, find suitable terminals, wherever they are and start services [19]. Teleporting [38] is an example of user mobility support by letting the user interface of an application to follow the user. [19] examines the use of mobile agents for personal mobility support. [38] makes use of a centralized database for storing mobility information. In this thesis, we concentrate on personal mobility support for applications using handheld devices. The mobility information is also stored on the personal handheld devices of the user.

2.3 Related Work on Location Systems

The most popular location system is active badge location system [33]. This system is also used in [38]. In such a system, the badge transmits signals providing information about their location. An antenna detects these signals and can signal the application about the presence of the user. An example of active badges is the AirID system which transmits RF signals. Recently work has been done [27], called Cricket, which has a location support system for in-building, mobile location dependent applications. Mobile and static nodes learn about their physical location by using listeners that hear and analyze information from beacons spread throughout the building. In our implementation, we have used a primitive system which relies on the range of wireless link for obtaining location awareness.

2.4 Mobile ID

[38] describes Mobile ID protocol which is a badge activated application level handoff of a multimedia streaming to support User Mobility. This system uses AirID badges for providing user identification and location awareness. The User Mobility support takes place through a centralized database. Our system differs from Mobile ID in the use of handheld devices instead of AirID badges. The use of AirID badges have the following disadvantages :

1. Poor scalability : Every user has a personal AirID badge. The information about the user is entered in a database at every video client in the system. Thus, adding new users would imply updating the database entry at every video client. Also, a new video client added to the system would have to know the identification information of all the users in the system. Further, if a user moves to a new domain temporarily, his/her badge information would have to be temporarily registered at the video clients. Thus, user identification using AirID badges is poorly scalable if the number of users and video clients is dynamic and the movement of the user is global.

2. Security loopholes : The AirID badges are pre-programmed with low security. Since the badges do not have processing, memory or programming capability, software programmable high security solutions cannot be ported onto AirID badges.

3. Inflexible : Since the AirID badges are not programmable, they cannot be used for any other purpose than identification and location detection.

The centralized database used for User Mobility also suffers from the problem of poor scalability for global movements of the user.

In this thesis, we propose to use handheld devices for secure user identification, secure user mobility and location awareness. Further, the state information is also stored on the handheld devices. We show in later chapters how the use of handheld devices removes the drawbacks of AirID badges.

Chapter 3

System Design

3.1 Introduction

This chapter describes the design of our system. We present a generic design of the components of the architecture which could be used by any application desiring secure user identification and secure user mobility support. In the next chapter, we will describe how these components could be used by a video on demand application.

3.2 System Overview

The system assumes the availability of a personal handheld device for each user in the system. These handheld devices would be used to solve the problem of user identification, user mobility and location aware system. Figure 3.1 shows the overview of the system. The system is viewed as consisting of several domains. Each domain consists of an authentication server, service providers and users with handhelds. In addition, there is a Global Certificate Authority which is visible to the authentication servers of all the domains. The user could move from one service provider to another within the same domain or across domains. The challenge is to provide seamless service to the user as he/she moves and to do so in a secure, efficient and flexible manner. The architecture presented in this chapter achieves the above goals and is able to scale across domains.

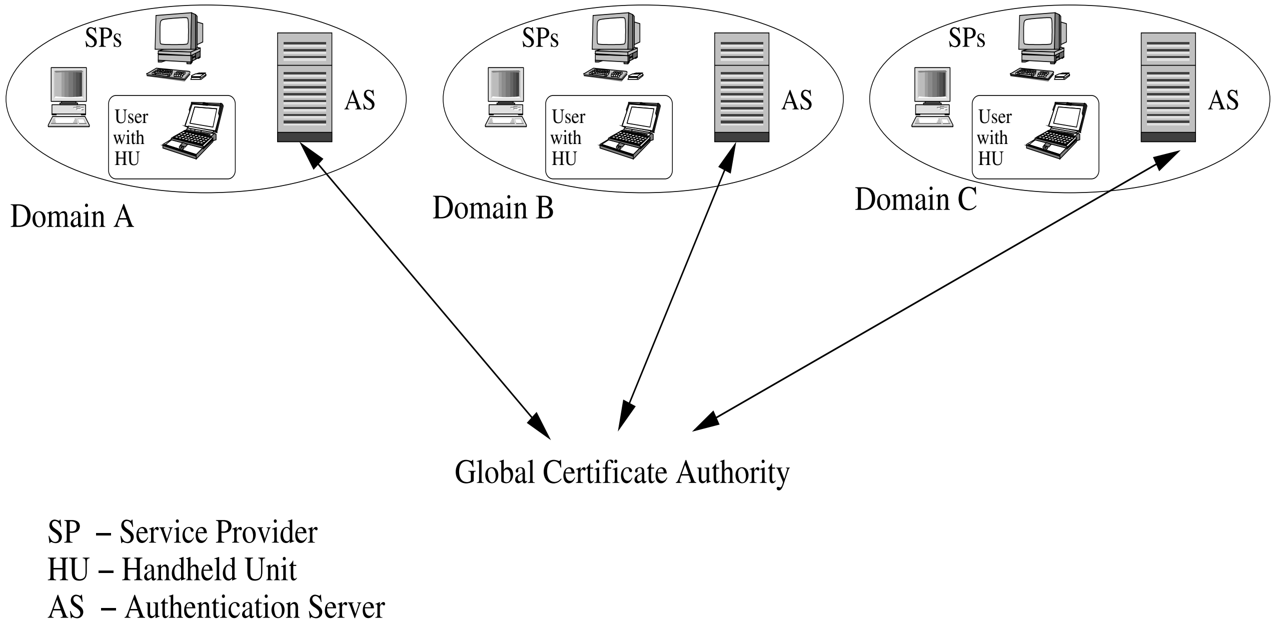


Figure 3.1: System Overview

3.3 System Architecture

Figure 3.2 shows the high level architecture of the system. The architecture is organized into layers. Each layer offers certain services to the higher layers, shielding those layers from the details of how the offered services are actually implemented. The layers on one machine communicate directly with the corresponding layers on another machine. The rules and conversions required during this communication are governed by protocols. Our work focuses on the design of Secure User ID system and Secure User Mobility system. We have also designed and implemented a primitive Location Aware System as also the cryptography libraries. In this chapter, we will describe in detail the Secure User ID System, Secure User Mobility system, Location Aware System and Cryptography layer on the different components of the system.

3.3.1 Secure User ID System

The Secure User ID system is used for a secure and trusted identification of the users to the service providers and vice versa i.e it provides mutual authentication between the users and service providers. The challenges in designing such a system are:

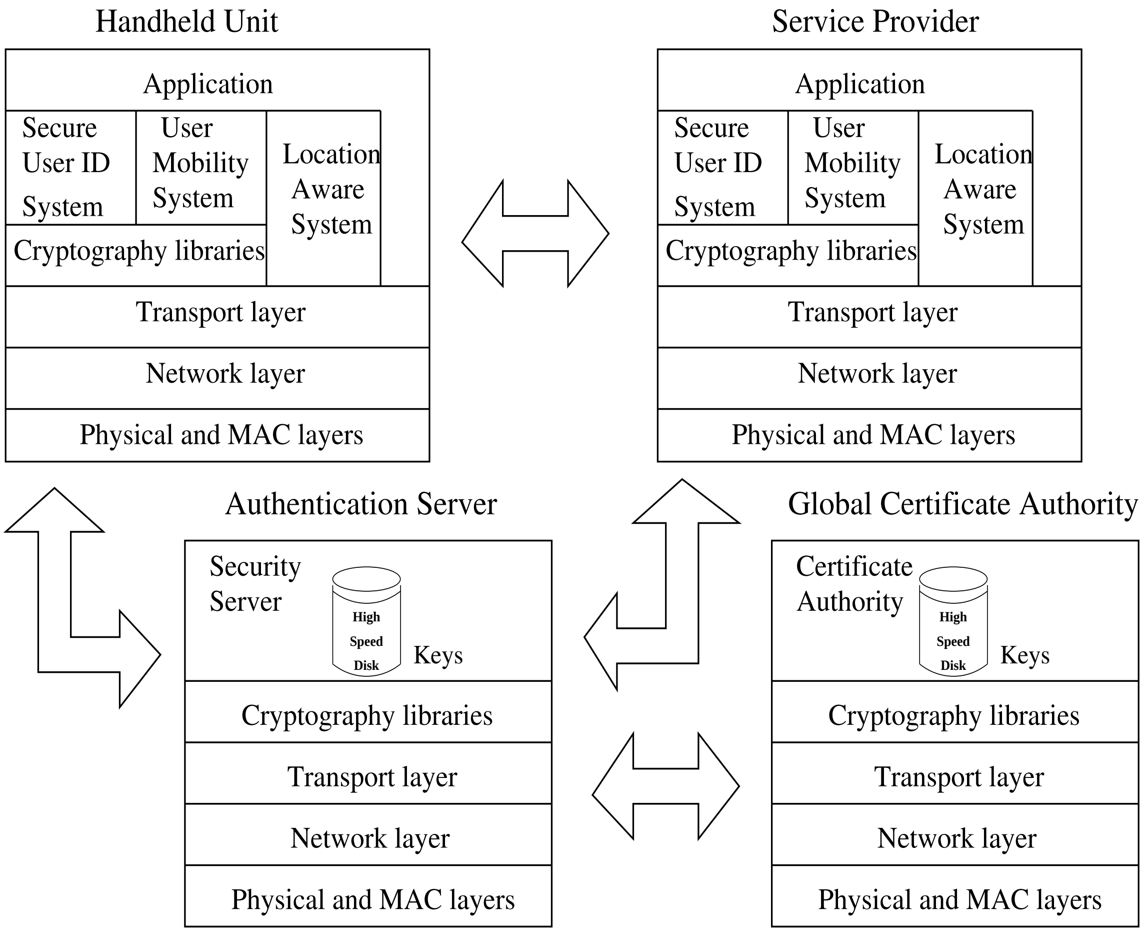


Figure 3.2: System Architecture

1. The system should be scalable so as to work across all domains.

2. It should ensure that the existing users and service providers are unaffected by the addition of new users and/or service providers.

3. The system should provide a mechanism to control the quality of security in the system for better performance.

4. The authentication process should be as efficient as possible with no compromise on security at the chosen security level.

The Secure User ID system meets all the challenges listed above. The security of the system is based on public/private key pairs, certificates and session-specific secret keys. The system works in three phases :

1. Phase 1 : Registration of Authentication Servers
2. Phase 2 : Registration of handhelds and service providers
3. Phase 3 : Mutual authentication of handhelds and service providers

We describe each of these three phases in detail below.

Phase 1 : Registration of Authentication Servers

Each domain of the system has a local authentication server. These authentication servers are assigned certificates by a Global Certificate Authority. The format of the certificates in the system is shown in Figure 3.3. The certificate of AS contains the public key of the authentication server and is signed by the Global Certificate authority. The keys are generated by the Global Certificate Authority in a secure manner. The following is a formal description of the registration process of the authentication servers.

Certificate Issuer
Timestamp
Name of the owner
Public key of owner
Expiry Date
Signature

Figure 3.3: Certificate Format

1. The Global Certificate Authority has been assigned public/private key pair through a secure manner.
2. The Global certificate authority generates public/private key pairs and certificates for the authentication servers.

3. The Global CA sends the private key, $(K_{priv})_{AS}$ of the AS; the certificate $Cert_{AS}$ of the AS; and the public key $(K_{pub})_{CA}$ of the Global Certificate Authority to the authentication server. The authentication servers could also be preloaded with this information.

4. The authentication servers publish their public key information.

Phase 2 : Registration of handhelds and service providers

Each handheld and service provider are associated with a home authentication server in a domain. This home authentication server is responsible for generating public/private key pairs and certificates to the handhelds and service providers. The following is a formal description of the registration process and is shown in Figure 3.4:

1. The authentication servers have been assigned public/private key pairs by the Global CA during Phase 1.

2. Administrator enters AS name and public key on the Handheld Unit/Service Provider.

3. The handhelds and service providers create a registration request message, consisting of :

   (a) Name of the handheld unit/service provider

   (b) AS administrative password

   (c) Session key K for the registration session. This is generated by the HU/service provider

   (d) Maximum security processing capability of the device.

4. The registration request message is encrypted, using the public key of the AS, to produce the message
   $M1' = E_{(K_{pub})_{AS}}(HU/SP\_name, AS\_admin\_password, Session\_Key\_S, Security\_Capability)$

5. AS recovers message $M1 = D_{(K_{priv})_{AS}}(M1')$.

6. AS verifies the AS administrative password. This serves as the authentication for the registration process.

7. Depending on the maximum security processing capability of the device, AS generates an array of public/private keypairs for HU/SP and creates corresponding certificates

$Cert_{HU/SP}[\,]$ of HU/SP. These different keypairs differ in the length of the generated key. All the keys have length less than that corresponding to the maximum security processing capability of the device. The multiple keys correspond to the different Quality of Security levels.

8. AS stores the HU/SP name and certificate in database.

9. AS generates $M2' = E_S((K_{priv})_{HU/SP}[\,], Cert_{HU/SP}[\,], Cert_{AS})$.

10. AS sends $M2'$ to HU/SP.

11. HU/SP recovers $M2 = D_S(M2')$.

12. HU/SP stores the array of private keys and certificates.

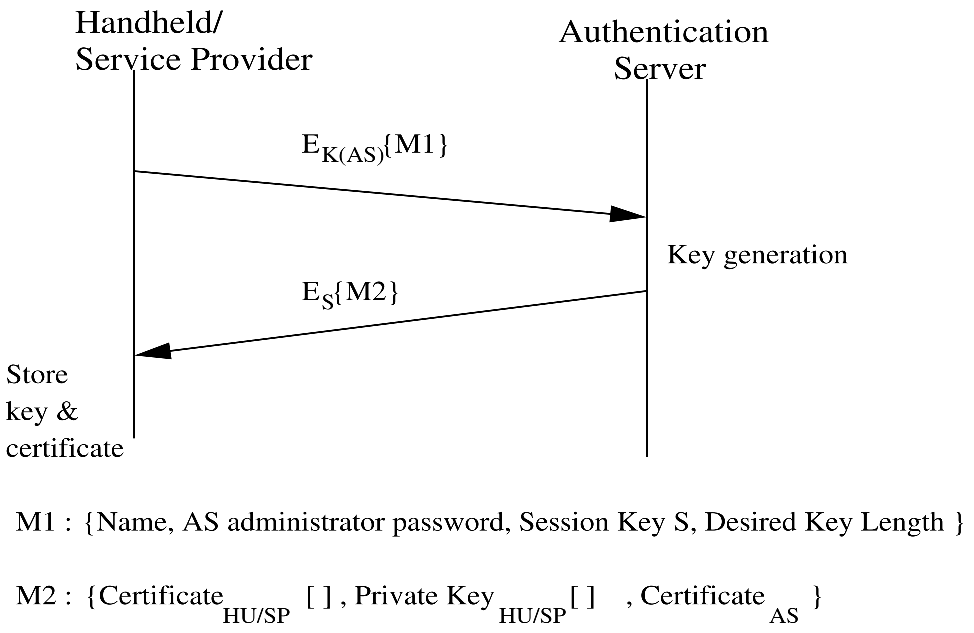


Figure 3.4: Registration of handhelds and service providers

Mutual authentication between handhelds and service providers

The mutual authentication between handhelds and service providers takes place through certificates issued by the authentication servers. Before starting the authentication process, the user can specify the key lengths of the public key and secret key desired for this session. At the end of the authentication process, a session key is established between the handheld unit and service provider. The following is the authentication protocol also shown in Fig 3.5

1. HU sends $Cert_{HU}$ to the service provider.

2. The service provider verifies the certificate $Cert_{HU}$ using the public key $(K_{pub})_{AS}$ and obtains $(K_{pub})_{HU}$.

3. Service Provider, SP, generates a random key S and generates message $M1' = E_{(K_{pub})_{HU}}(Random\_Key\_S, Cert_{SP})$

4. HU recovers $M1 = D_{(K_{priv})_{HU}}(M1')$ and obtains the Random Key S and $Cert_{SP}$.

5. The HU verifies the certificate $Cert_{SP}$ using the public key $(K_{pub})_{AS}$ and obtains $(K_{pub})_{SP}$.

6. HU generates a random key N and generates message $M2' = E_{(K_{pub})_{SP}}(Response\_S', Challenge\_N)$ and sends it to the service provider.

7. SP recovers $M2 = D_{(K_{priv})_{HU}}(M2')$ and obtains the Response Key S' and Challenge key N.

8. The HU verifies the response key S'. If the value of the response key S' is not the value of S sent, authentication fails.

9. SP generates message $M3' = E_{(K_{pub})_{HU}}(Response\_Key\_N')$ and sends it to the service provider.

10. HU recovers $M3 = D_{(K_{priv})_{HU}}(M3')$ and obtains the Response Key N'.

11. The HU verifies the response key N'. If the value of the response key N' is not the value of N sent, authentication fails.

12. HU and SP now communicate using S as a session key.

Authentication outside the home domain : When the user moves to a new domain, the HU registers with the authentication server of the new domain. During this registration process, the handheld unit presents the pair $\{ Cert_{HU}, Cert_{AS} \}$ where $Cert_{HU}$ is the certificate issued to the HU by the authentication server in the home domain. $Cert_{AS}$ is the certificate of the HU's

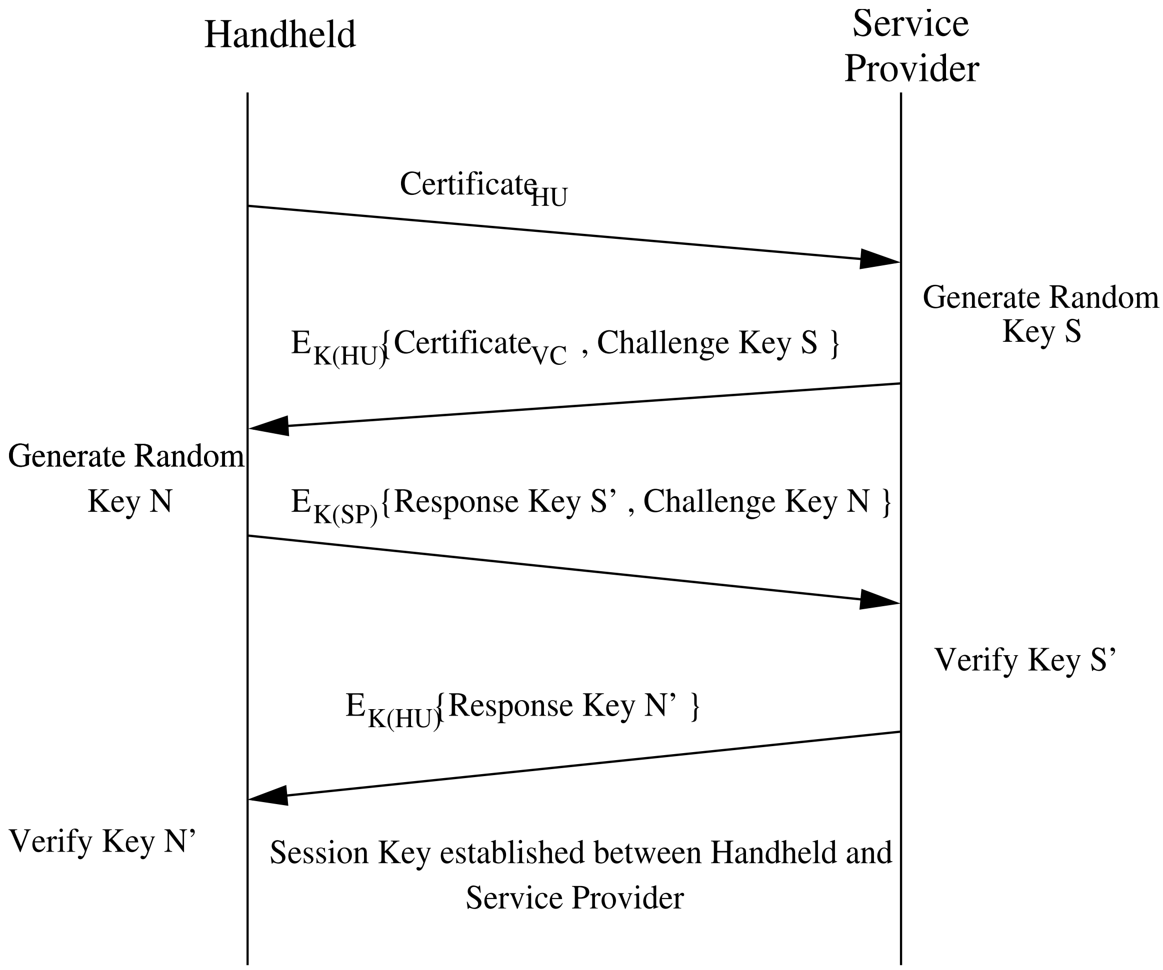


Figure 3.5: Mutual authentication between handhelds and service providers

home authentication server. The authentication server of the new domain verifies $Cert_{AS}$ and using the public key in $Cert_{AS}$, it verifies $Cert_{HU}$. It then issues a new temporary certificate $Cert_{(HU)_{temp}}$, which has the same contents as $Cert_{HU}$ but is signed by the authentication server in the new domain. This temporary certificate would be used in the new domain. In this manner, only the authentication server in the home domain generates the keys for the users and these keys can be used by the user in any domain. Further, the extra overhead involved when moving to a new domain is small. The design is thus both flexible and scalable.

3.3.2 Secure User Mobility System

The Secure User Mobility system assumes that a session key has been established between the handheld unit and service provider. The security is based on this session key and the certificates issued by the authentication server. The User Mobility problem can be stated as follows :

To detect the movement of the user from one service provider to the other and assure seamless mobility support of services to the user by transfer of the service state information among the service providers contacted by the user.

We now describe the Secure Leasing Protocol which solves the above problem. The protocol tracks the movement of the user and transfers the latest state information from the service provider to the handheld unit. When the user moves to a new location, the latest information is transferred from the handheld unit to the new service provider. Thus the state of the service is carried by the user as he/she moves around from one service provider to the other.

As the name suggests, the Secure Leasing Protocol is based on the concept of leasing. It is visualized as follows : The service provider provides the service to the user for a certain lease period. The user can renew the lease with the service provider for continuing the service. In a practical case, this lease renewal would continue as long as the user is in the vicinity of the service provider. When the user moves away from the service provider, the lease renewal should fail leading to the automatic stopping of service by the service provider. Further, the Secure Leasing Protocol requires that the latest service state information be sent back from the service provider to the handheld unit as a response to each lease renewal request. The security of the protocol is achieved as follows:

1. The lease renewal and lease response messages are encrypted using the session key.
2. The service state information is signed by the service provider.
3. In some cases, we would desire to assure the integrity of the state information when it is stored on the handheld and in possession of the user. In such cases, the service state information would be signed by the authentication server before being stored on the handheld unit. The service providers could then verify the state information before using it.

The Secure Leasing protocol is shown in Fig 3.6 and described below :

1. A session key S has been established between Handheld unit and Service Provider using Secure ID system.

2. The service provider starts the service and a service garbage collector daemon.

3. The handheld unit starts a timer.

4. The handheld unit sends a lease renewal message $M1' = E_S(RENEW\_LEASE)$ to the service provider. This message could also be augmented with some application specific information in a practical implementation. If this lease renewal message fails, go to Step 9.

5. The service provider obtains the message $M1$, determines the current state information of the service, renews the lease and continues the service.

6. The service provider creates a message $M2' = E_S(Updated\_State\_information, Sign_{SP}(H(Updated\_State\_information)))$ and sends it to the handheld unit.

7. The handheld unit recovers the message $M2$ and obtains the state information. This state information is stored for later use.

8. Repeat Steps 4-7 till Step 4 fails.

9. The lease renewal message fails if the user is no longer interested in the service or if the user moves away from the service provider. On a lease renewal message failure, the application on the handheld unit is notified with the cause of lease renewal failure.

10. Since the service provider stops receiving the lease renewal messages, the service garbage collector daemon stops the service.

If the lease renewal failure is due to the movement of the user, the application would invoke the location system to find new service providers. The handheld unit now has the latest state information and this information can now be given to the new service provider. The following is the discussion on Secure Leasing Protocol :

Unidirectional message distribution : The Secure Leasing Protocol described above requires a two way message exchange for lease renewal causing a lot of bandwidth usage. This could be

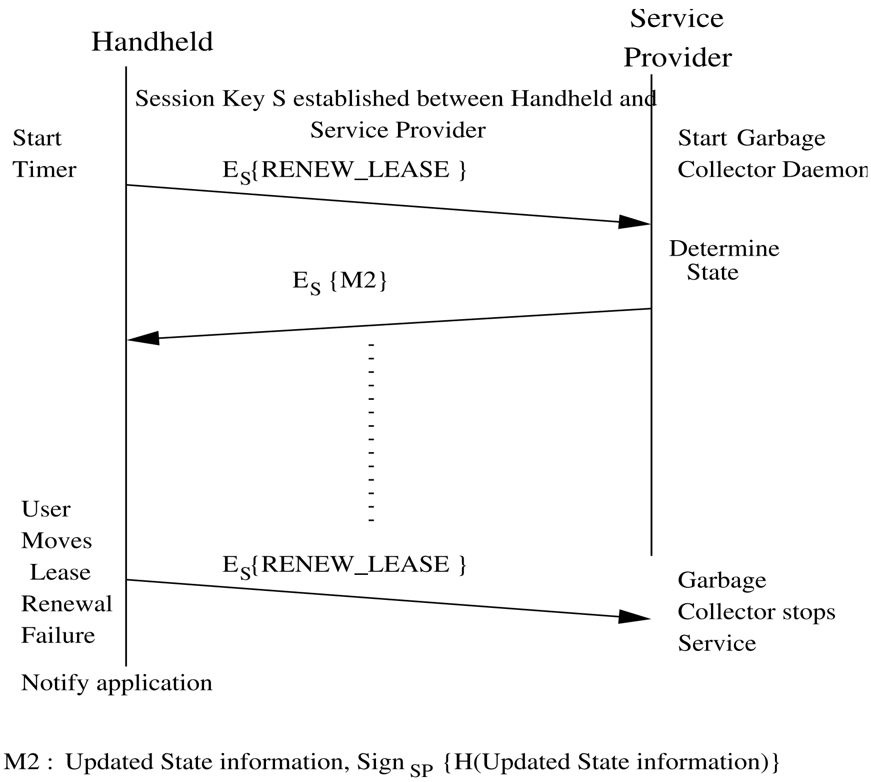


Figure 3.6: Secure Leasing Protocol

unnecessary in some applications. For example, in some applications, we would like the service to continue as long as the user is in the range of the service provider. In such a case, the explicit lease renewal request messages would be redundant. The service provider could periodically send the state information using PUSH technology. If the user goes out of range of the service provider, it can stop sending these messages. On the other hand, some applications may not require any state transfer information. In such cases, a one way lease renewal request message would be sufficient. The protocol described above is a general case in which the application would like to send some extraneous information along with the lease renewal request message and the service provider responds with the state information along with other application specific information. Such an architecture represents a cross layer message passing design to optimize the bandwidth on the wireless network.

Integrity and Non-repudiation of State information : The state information is digitally signed by the service provider for the purpose of non-repudiation i.e the service provider cannot deny later that it sent the state information. Another important security consideration is the integrity of the state information. Since all the messages containing the state information is encrypted/hashed/signed, the integrity of the state information when sent along the communication channel is assured. However, the state information could be maliciously changed by a user/intruder when stored on the handheld unit. In such a case, the state information should be signed by a trusted party. The new service provider would verify this signature before using the state information. One of the assumptions of the system is that each of the entities are only aware of their own certificates and public key of the authentication server. So, the trusted party would be the authentication server of the local domain. Such a scheme would also work when the user moves to a new domain. However, there is an extra overhead introduced in order for the authentication server to sign each state information message. This is an example of the tradeoff between performance and security.

Bandwidth utilization : It can very easily be inferred that the Secure Leasing protocol demands a lot of bandwidth. Most of this bandwidth would be wasted since the only relevant messages are those sent when the user actually moves out of the range of the service provider. In order to have this perfect bandwidth utilization, we need a scheme which can very accurately track the user and notify the system just before the user is about to leave the range of the service provider. Such a scheme though highly desirable is difficult to obtain practically resulting in the need for approximation. This approximate scheme should be able to predict the movement of user through some heuristics (eg. signal strength of wireless link etc.) and require a message exchange only when there is a sufficiently high probability of the user going out of range. Again, building such systems is a tedious task and there are ongoing tasks in building such systems [6]. The Secure Leasing Protocol presented in this section assumes no such optimized user tracking scheme is in place. Hence the only way to detect the presence or absence of the user in the range of the service provider is to periodically ping the service provider (called lease renewal in our case). Since the user can move at any instant, we would like to ping the service provider continuously with low period interval. This obviously has a effect on bandwidth leading to a tradeoff between bandwidth and accuracy of state information. If we increase the time period of lease renewal i.e increase the length of lease period, we could lose on the accuracy of the state information if the user moves at random unpredictable intervals. However, in a practical scenario, the user could be able to fix the length of time it wishes to stay in the range of the service provider. In such a case, the state information message would only be sent at the end of this time period. This would help us save on bandwidth and assure accuracy of the state information but would require the user to define his/her mobility patterns. Thus, we see that there are very interesting tradeoffs and the best system would be dependent on the application being built. The User Mobility system should however have provisions for supporting all these different kinds of systems and allow dynamic switching among them.

3.3.3 Location Aware System

The Location Aware System discovers service providers in the vicinity of the user. Thus the location aware system should be aware of the physical location of the user and the service providers in the range of the user. There is a significant amount of research in progress on location aware systems [6]. Building a location aware system is not the main focus of our research and we have currently used a simple broadcast protocol as a location aware system. This location aware system could be later on replaced by some more sophisticated systems in an actual deployment.

The Location Aware System is shown in Fig 3.7 and is described below

1. Each service provider desiring to provide service has a location system daemon which waits for service requests from users.

2. Whenever a user desires to use a service or whenever a new service provider has to be discovered as a result of user mobility, the handheld unit broadcasts a service request message over an ad-hoc wireless network.

3. The service providers in the range of the handheld unit would receive this service request message and would reply back with a "Service Available" message.

4. The handheld unit receives the replies from the service providers and the user could choose one of them. In our implementation, we use the one that replies first.

This Location aware system thus determines the service providers in the range of the user based on the range of the wireless interfaces, i.e it assumes that the service providers in the range of the wireless interface would be close enough to the users to use their service. This idea is similar to

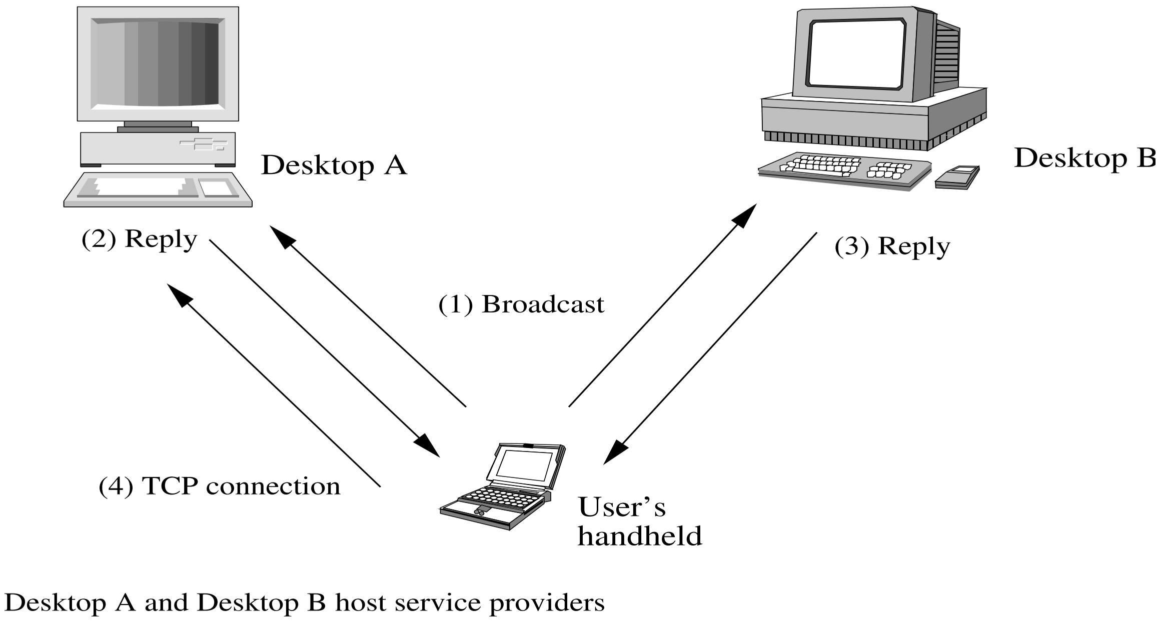


Figure 3.7: Location Aware System

the idea used by security badges wherein the range of RF signals help determine the nearby service providers. We will talk about the details of our experimental setup in a later chapter. Also, the location aware system described above does not have any security guarantees.

3.4 Working of the system

In this section, we describe the steps to be used by an application developer desiring to use the Secure User ID system, Secure User Mobility System and Location Aware system. In the next chapter, we would describe the implementation for a Video on Demand application. The working of the system is also shown in Fig 3.8 and is described below

1. All the service providers and users in a domain go through a Secure Registration process to obtain the certificates.

2. Whenever the user desires a service, it invokes the Location Aware System to discover the service providers in its vicinity.

3. The service providers in the vicinity of the user reply back with a 'Service Available' message.

4. The Secure User ID system is invoked for a secure mutual authentication of the users(a.k.a handheld units) and service providers.
5. The Secure User Mobility system is then started to track the movement of user and managing the service state information.
6. The user moves to a new location.
7. The Secure Leasing Protocol detects the movement of the user and notifies the application. The application again invokes the Location Aware System to discover new service providers.
8. The new service providers in the vicinity of the user reply back with a 'Service Available' message.
9. The Secure User ID system is again invoked for authenticating the user to the new service provider.
10. On successful authentication, the latest state information recorded by the Secure Leasing Protocol is transferred to the new service provider.
11. The Secure User Mobility system is again started to track the further movement of user and to manage the service state information.
12. If the user moves again, steps 7-11 are repeated.

3.5 Quality of Security Negotiation

In this section, we describe a very distinctive feature of our design - negotiation of the quality of security. The motivation for this feature is the significant computational time required by current security protocols. This demand for expensive computation leads to the following two situations

1. The handheld devices have lower processing capability compared to the desktop PCs and workstations. As a result, it becomes infeasible for some handheld devices to handle security protocols requiring expensive computation.

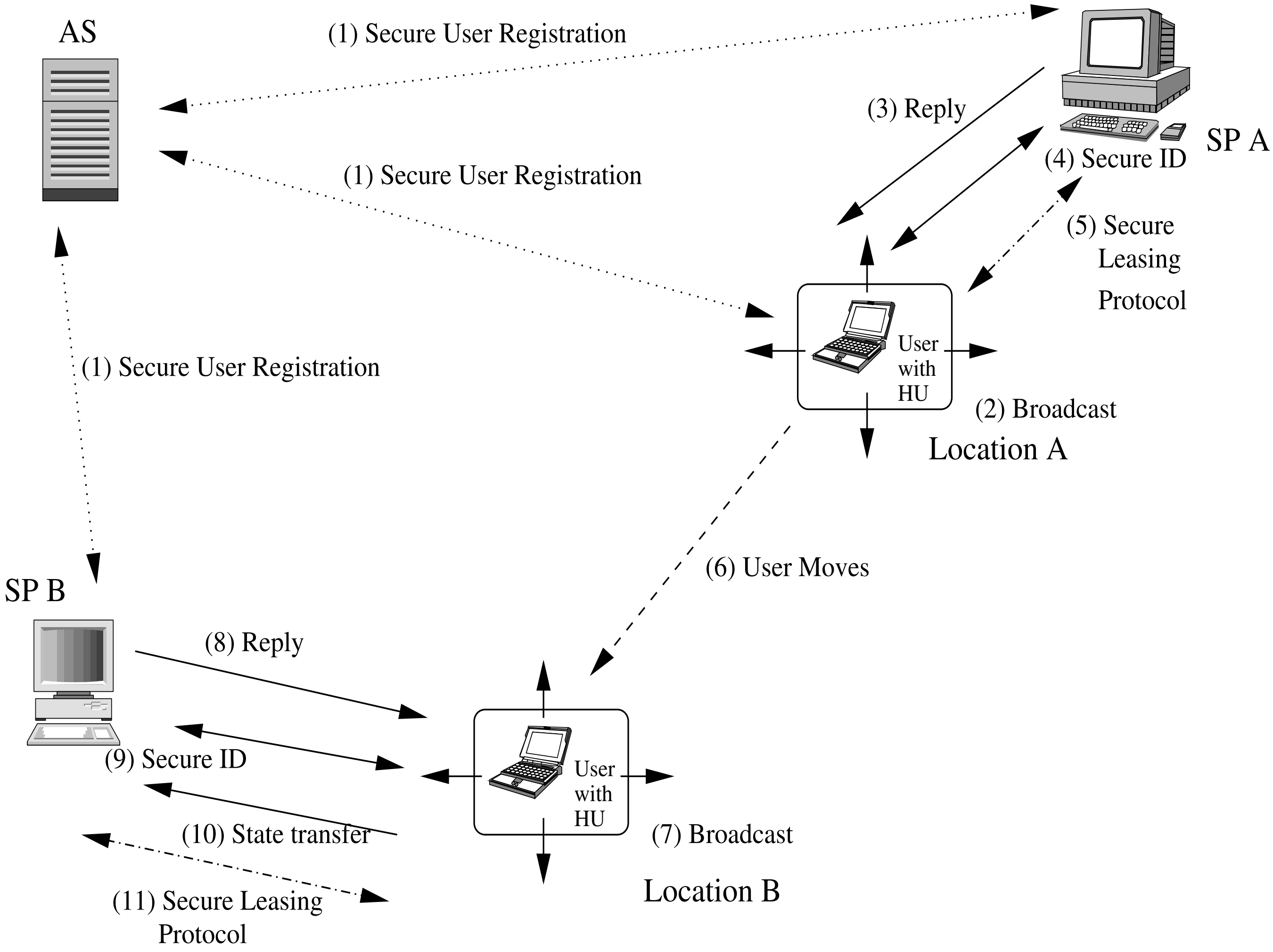


Figure 3.8: Working of the system

2. The extra overhead due to the processing of security protocols may lead to an undesirable degradation in performance.

These two situations demand an architecture flexible enough to adaptively change to the needs of security, performance and feasibility. We achieve this by negotiating the length of the keys (both public/private keys and secret keys) used by the security protocols. This choice is based on the fact that the complexity of the security protocols is proportional to the length of the keys used keeping the other parameters fixed. The key length negotiation takes place as described below in the Secure ID system:

1. During the User Registration phase of the Secure ID protocol, the Security Capability information is passed to the authentication server. This Security Capability field represents the maximum capability of the device for security purposes. In our implementation, we pass the maximum RSA public/private key length and the maximum AES secret key length that could be handled by the device for the purposes of encryption, decryption and key generation. The authentication server then generates multiple certificates of different key lengths but less than the Security capability of the device. As a result of this mechanism, heterogeneous handheld and mobile devices can be supported in the system.

2. During the mutual authentication of the handheld devices and service providers, the user can specify the lengths of the public/private key and secret key that is desired for that particular session. Since each device has multiple certificates differing in the key lengths issued to it by the authentication server, both the sides can adaptively choose the certificate with the required key lengths (if the key lengths are within processing capability). In this way, the user can control the quality of security used in any particular session.

Thus our design provides an answer to address the indispensable tradeoff between security and performance.

3.6 Extensions to the system : Authorization

The Secure User ID system, Secure User Mobility system and location aware system can be used by any application desiring secure user identification and secure user mobility. We can have application specific extensions to the system built over these layers. One of the extensions that would be desirable in most cases is the Authorization and Access Control mechanism in the system. The Authorization and Access Control mechanism would control access to the service components. The challenge is to build an architecture that would work well in mobile environments. Also we would like to use some of the infrastructure built in the underlying layers for handling the specific issues of mobility and security information management. In this section, we describe a design for an authorization scheme, based on Role-Based Access Control for a Video on Demand application.

3.6.1 Video on Demand Application

The Video on Demand application consists of Video Clients, Video Server and users with their personal handhelds. The Video Clients act as service providers to the users and play the video requested by the user. In order to play the video, the Video Client contacts the Video Server. Fig 3.9 shows the interaction between the video server, video client, AS and Handheld. More details on the Video on Demand application are presented in the next chapter.

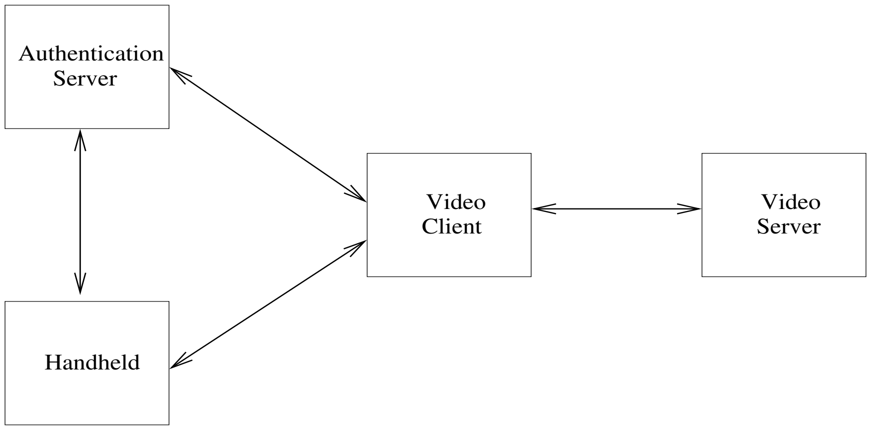


Figure 3.9: Interaction between Video Server, Video Client, AS and Handheld

3.6.2 Role Based Access Control

Role based access control (RBAC) [7] is an alternative to traditional discretionary (DAC) and mandatory access control (MAC) policies that is attracting increasing attention, particularly for commercial applications. The principle motivation behind RBAC is the desire to specify and enforce enterprise-specific security policies in a way that maps naturally to an organization's structure. Traditionally, managing security has required mapping an organization's security policy to a relatively low-level set of controls, typically access control lists.

With RBAC, security is managed at a level that corresponds closely to the organization's structure. Each user is assigned one or more roles, and each role is assigned one or more privileges that are permitted to users in that role. Roles can be hierarchical. For example, some roles in a hospital may be health care provider, nurse, and doctor. The doctor role may include all privileges available to the nurse role, which in turn includes all the privileges available to the health care provider role.

Security administration with RBAC consists of determining the operations that must be executed by persons in particular jobs, and assigning employees to the proper roles. Complexities introduced by mutually exclusive roles or role hierarchies are handled by the RBAC software, making security administration easier.

More information on Role based access control can be got at [30] and also through several papers at [7] and [8].

3.6.3 Description of the system

In order to implement a Role Based Access Control mechanism for a video on demand application, we need to identify the roles, permissions and pre-requisite conditions in the system.

User level Roles

The following user level roles are identified :

1. Registered User - This role signifies a qualification: Registered Users.
2. Video Access - This role signifies the task of accessing the video from the Video Server.
3. Video Play - This role signifies the task of playing the video.
4. Video Stopped - This role signifies the state of the video being stopped.

Administrative Roles

The following are the Administrative Roles:

1. Video Client : This role signifies the qualification of a video client.
2. Video Server : This role signifies the qualification of a video server.
3. Authentication Server : This role signifies the qualification of an AS.

Permissions with User Level Roles

The following permissions are associated with the User Level Roles :

1. Registered User : Request to start a video.
2. Video Access : Request to play the video.

3. Video Play : Request to stop the video.

4. Video Stopped : Request to restart the video.

Permissions with Administrative Roles

The following permissions are associated with the Administrative Roles :

1. Video Client : Assigning users to Video Play and Video Stopped Roles Revoking users from Video Play and Video Stopped Roles.

2. Video Server : Assigning users to Video Access Role Revoking users from the Video Access Role.

3. Authentication Server : Assigning users to Registered Role Revoking users from Registered Role.

There are Video Access, Video Play, Video Stopped Roles for each video. Thus there could be multiple users assigned to any of these roles for a particular video.Similarly, each user could be assigned to these roles for multiple videos.

The Video Client, Video Server and Authentication Server can be many in number. Each of these could be arranged in a hierarchy with different permissions.

The general structure is described in Fig 3.10. The administrators assign users to roles using the can_assign(x,y,R) relation : - A member of administrative role x can assign a user to roles R if the user satisfies the pre-requisite(constraint) relation y.

The administrators revoke users from roles using the can_revoke(x,Y) relation :- A member of administrative role x can revoke a user from role y belonging to set Y.

Pre-requisite conditions

The pre-requisite(constraint) relation represents the policy. The following could be the pre-requisite conditions:

1. Registered User : The AS assigns a user to the Registered Role if it satisfies the policy decision that the registration request is made by a system administrator.

2. Video Access : A Video Server assigns a user to the Video Access Role if the user is a member

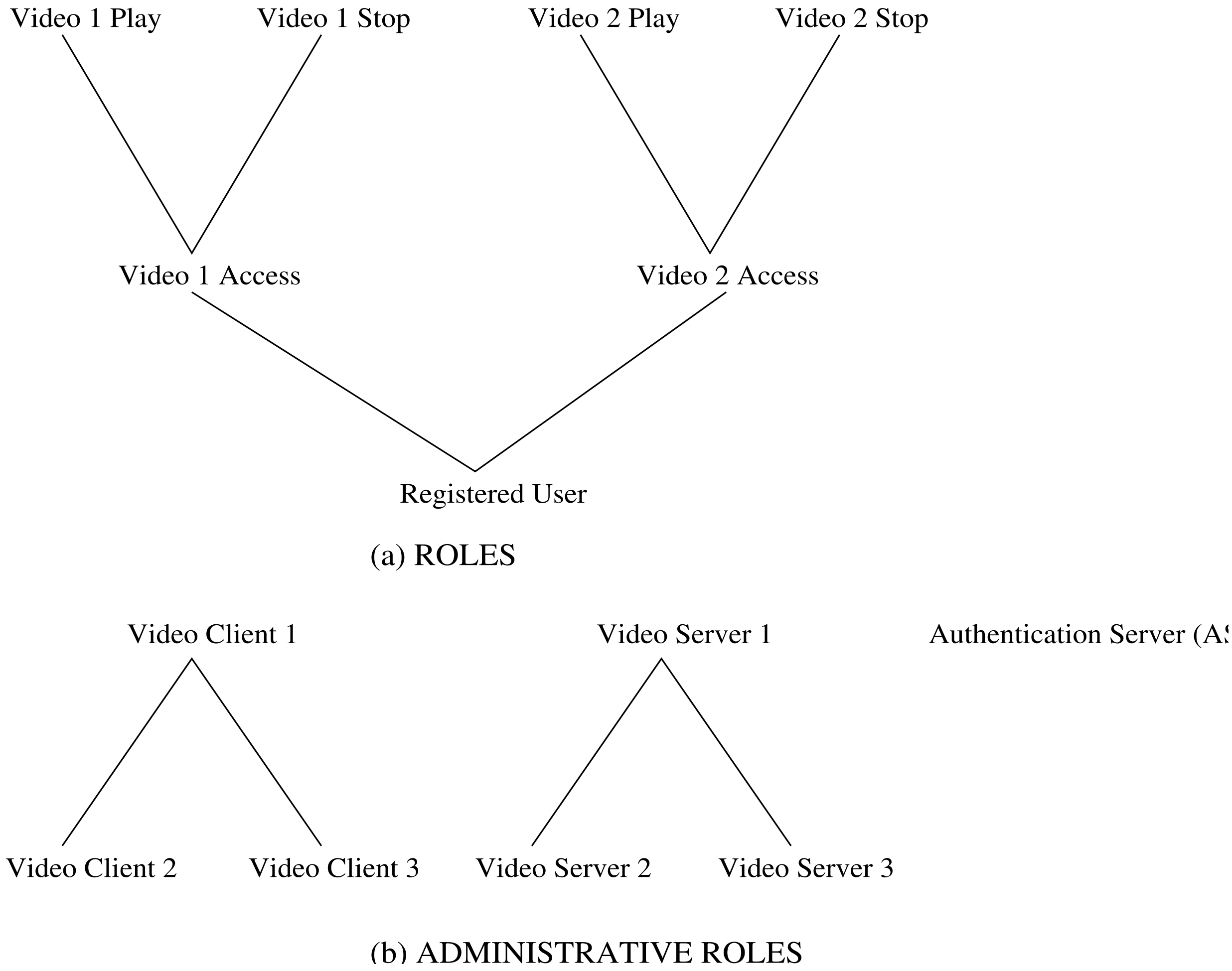


Figure 3.10: Example Role and Administrative Role Hierarchies

of "Registered User" role , the Video Server has access to the video and the policy decision allows the user to access the particular video.

3. Video Play : A Video Client assigns a user to the Video Play Role if the user is a member of the "Video Access" Role and if he privilege attribute of the user is greater than or equal to the user currently executing play session on the video client.

4. Video Stopped : A Video Client may assign a user to the Video Stopped Role if the user is a member of "Video Access" Role and if the user has moved away from the Video client or has explicitly requested a stop.

3.6.4 How it works?

The user in our context is the Handheld Unit(HU). The 'Video Client' in our context is composed of

1. Location Aware System
2. Authentication System
3. User Mobility System

There is a 1-1 mapping between Roles and states. The roles assigned to a user represent the state of the video service for this user.

When the HU registers with the Authentication Server, the AS assigns the user(HU) to the Role 'Registered User'. It also assigns a 'privilege vector' to the user which would be used for implementing policy decisions. The HU stores this state information. Whenever a user wants to use the video service, he/she turns on the service button. The authentication information is passed to the Video Client. After the two way authentication is successful, the 'current state information' along with the 'privilege vector' is given to the Video Client. This information is signed by the AS. The Video Client then forwards the request of the user to the Video Server. If the user is entitled to access the video, the Video Server assigns the user to the state of Video Access. The Video Client applies the can_assign relation to assign the user to Video Play. The Video Client then plays the video on behalf of the user. When the user requests a stop in service, the user is revoked from the Video Play role and assigned to the Video Stopped Role. The current state information for the user is created and passed to the HU. The HU now approaches another Video Client with the new state information.

When the HU moves to a new domain, it must be registered with the AS of the new domain as well. Once registered, it could use the state information of the previous domain.

Chapter 4

Implementation

This chapter provides the detailed description of the implementation of the Secure Identification system, Secure User Mobility system and Location Aware system. We demonstrate the architecture using a video on demand application. In subsequent sections, we describe the details of the video on demand application, the development environment and the implementation details of the various components in the system.

4.1 Video on Demand application

High quality video on demand over the network has become a reality today. It would be highly desirable to have user mobility support for such an application. We would also desire a secure user identification and mutual authentication between the users and video clients. The Secure Identification system, Secure User Mobility system and Location Aware system described in the previous chapter can be used by the video on demand application to have the desired features of user identification and user mobility.

Fig 4.1 and Fig 4.2 show the system overview and system architecture for a video on demand application. The video client/server pair is implemented using the Media Streaming Protocol (MSP) which was developed at the University of Illinois[21]. The MSP is designed to adapt to congestion when it occurs within the network. Techniques to detect network congestion on the transmission path were implemented, and by dropping and adding frames MSP can dynamically adapt to the available network resources. The client and server are both written in Java programming language, while MPEG software decoding engine is written in C programming language.

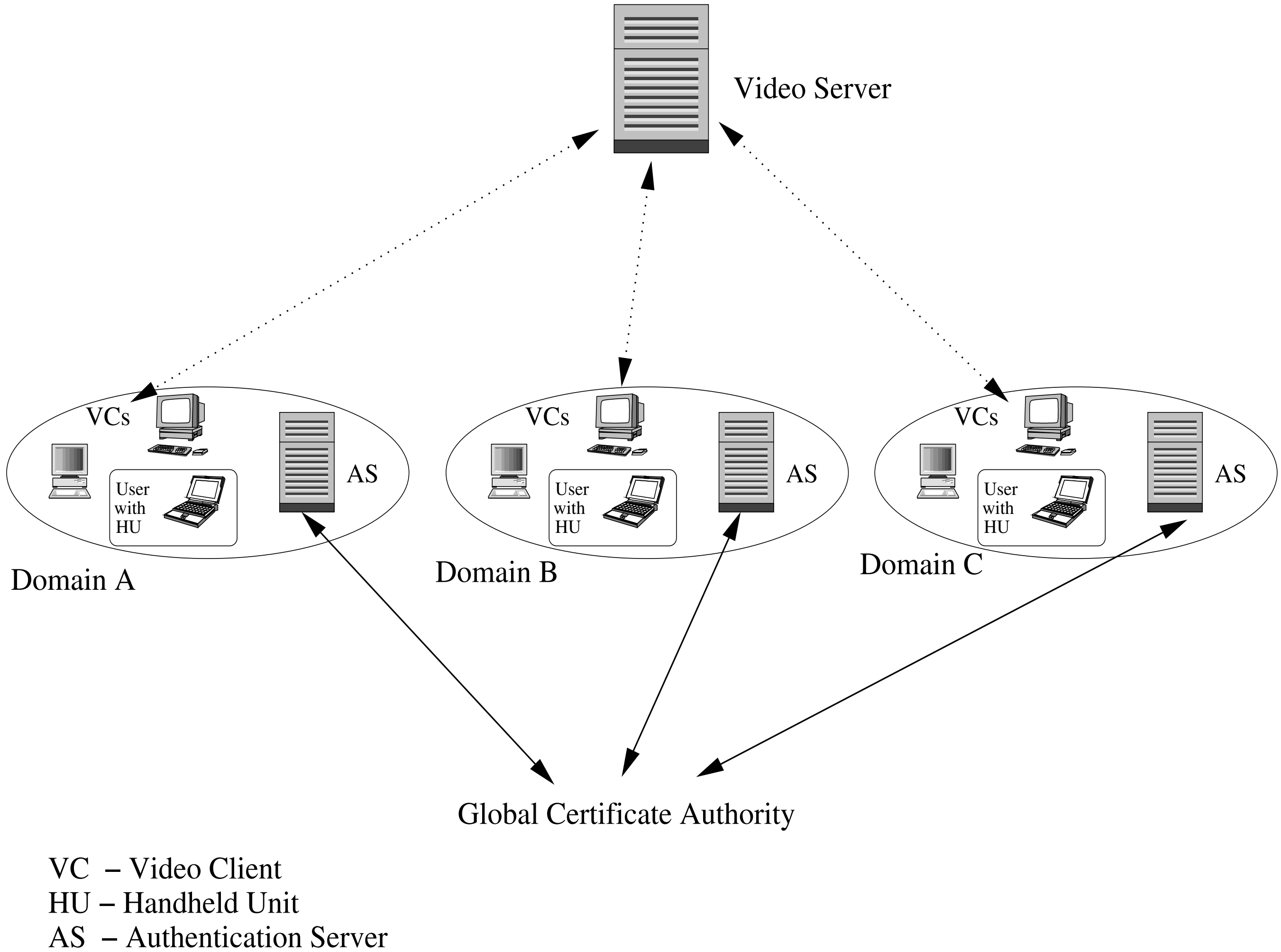


Figure 4.1: System overview for Video on Demand application

4.2 Development Environment

All of the components in the system are written in the Java Programming Language. The byte code for the WinCE environment on the HP Jornada was produced using Sun Microsystem's Personal Java 1.0 and those for Windows 2000 environment was produced using Sun Microsystem's JDK 1.3.

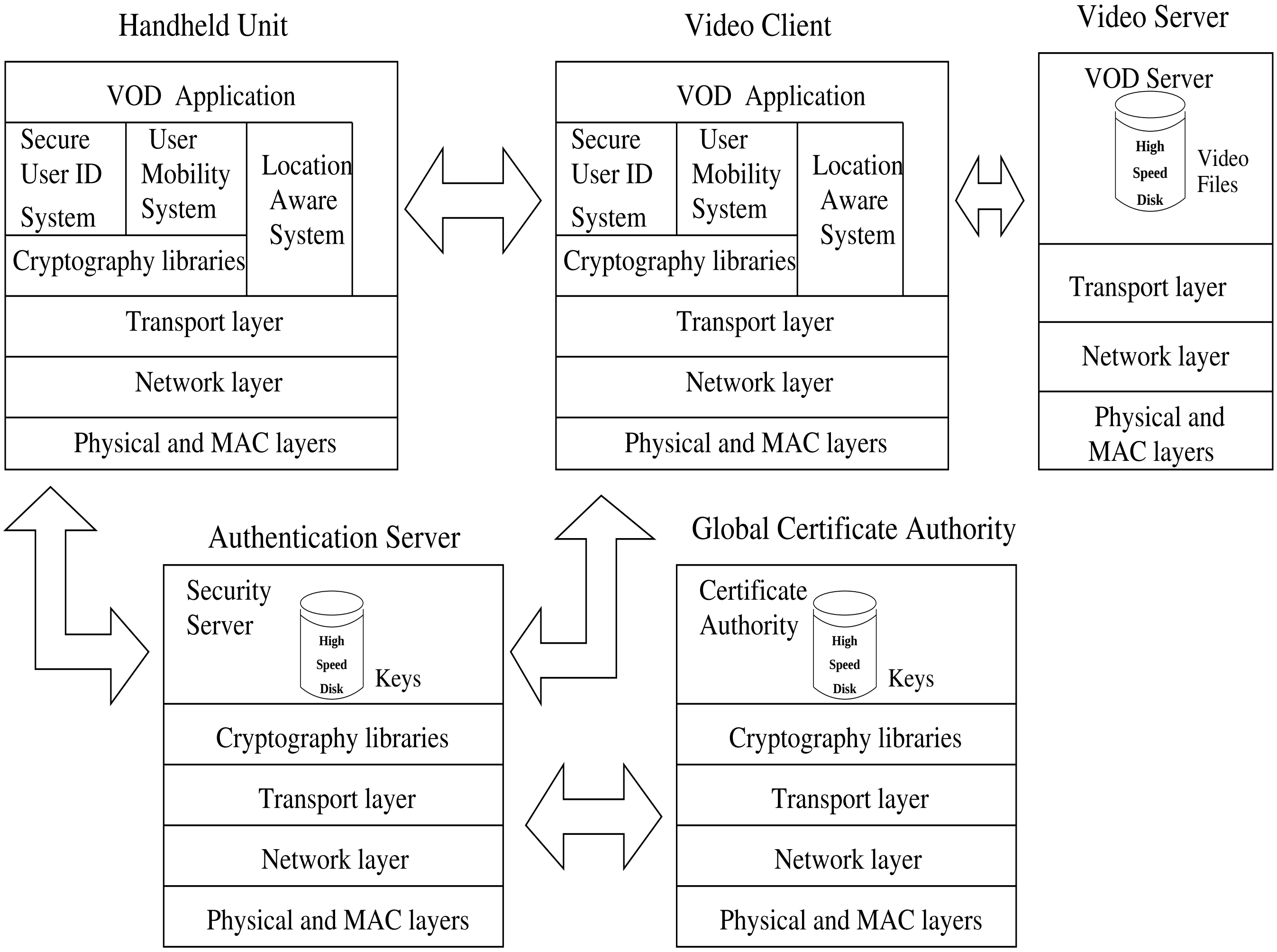


Figure 4.2: System architecture for video on demand application

4.3 Cryptography libraries

The Cryptography libraries are used by the Secure User Identification and Secure User Mobility system for imposing security. The libraries are implemented in Java and provide an API to use the system. We introduce the concept of CryptoStreams - input and output streams which pass data securely. The API for the CryptoStreams allows the user to specify the type of security protocols to be used as well as the security features desired. In the current implementation, the following type of security protocols are supported :

- Public Key Cryptography protocols : This is supported using the RSA algorithm.

- Secret Key Cryptography protocols : This is supported using the AES algorithm.

The following security features are currently supported for CryptoStreams:

- Confidentiality : This is achieved using encryption and decryption modules.
- Integrity : This is achieved using hashing of the message using SHA-1 hash libraries.
- Non-repudiation : This is achieved using digital signatures.

The user of the API can specify more than one security feature at the same time.

The API also has support for generating RSA public/private keys and AES secret keys of desired lengths. There is also an API for creating certificates.

The Cryptography library is an extremely important component in the implementation of the system since it provides all the necessary routines for the purpose of security. The design is object oriented and provides a very easy to use API. Fig 4.3 shows the interface of the public classes Crypto, CryptoInputStream, CryptoOutputStream, RSAKeyPairGenerator and AESKeyGenerator.

4.4 Secure User Identification system

The Secure User Identification system is implemented on the Authentication Server, Handheld and Video Client. It uses the cryptography libraries described in the previous section to achieve security. The components for the implementation of the Secure User Identification system on the authentication server, handheld/video client is shown in Fig 4.4. We briefly describe the salient features.

Authentication Server : At the authentication server, the Secure User Identification system is responsible for the Secure User Registration of the handhelds and video clients. It waits for the registration requests at a well known port. On receiving the request, the Secure User registration protocol is followed and the generated keys and certificates are stored in a main memory resident hash table. The keys and certificates for the authentication server itself are generated by a Global Certificate Authority module.

Handheld/Video Client : At the handheld/video client, the Secure User Identification system is responsible for the Secure User Registration and Mutual Authentication protocol. For the Secure

```
class Crypto {
/* aesEncrypt encrypts a block of data using AES algorithm */
static byte[ ] aesEncrypt(byte[ ] msgData, AESKey aesKey);
/* aesDecrypt decrypts a block of data using AES algorithm */
static byte[ ] aesDecrypt(byte[ ] cryptData, AESKey aesKey);
/* hash calculates the hash for a block of data */
static byte[ ] hash(byte[ ] msgData);
/* rsaEncrypt encrypts a block of data using RSA algorithm */
static byte[ ] rsaEncrypt(byte[ ] msgData, RSAPubKey rsaPubKey);
/* rsaDecrypt decrypts a block of data using RSA algorithm */
static byte[ ] rsaDecrypt(byte[ ] msgData, RSAPrivKey rsaPrivKey);
/* sign applies a digital signature to a block of data */
static byte[ ] sign(byte[ ] msgData, RSAPrivKey rsaPrivKey);
/* verifyHash check to see whether a message hash is valid or not */
static byte[ ] verifyHash(byte[ ] hashedMsg);
/* verifySig check to see whether a digital signature is valid or not */
static byte[ ] verifySig(byte[ ] signedMsg, RSAPubKey pk);
}
class CryptoOutputStream {
/* create a new CryptoOutputStream on top of an underlying outputstream */
CryptoOutputStream(OutputStream in, AESKey aesKey, RSAPubKey rsaKey, RSAPrivKey sigKey);
/* set the state of the CryptoOutputStream */
void setState(byte state);
/* read an array of bytes to the output stream */
void write(byte[] b);
}
class CryptoInputStream {
/* create a new CryptoInputStream on top of an underlying inputstream */
CryptoInputStream(InputStream in, AESKey aesKey, RSAPrivKey rsaKey, RSAPubKey sigKey);
/* returns the number of input bytes which can be returned without blocking */
int available();
/* read an array of bytes from the input stream */
int read(byte[] b);
}
class RSAKeyPairGenerator {
/* returns a new randomly generated RSA keypair, containing a public and private key */
static RSAKeyPair generateRSAKeyPair(int keyLength);
}
class AESKeyGenerator {
/* returns a new randomly generated AES encryption key */
static AESKey generateRSAKeyPair(int keyLength);
}
```

Figure 4.3: Cryptography Library API

User Registration protocol, the user preferences are taken from a GUI and the certificates/keys are obtained from the authentication server. Once the keys and certificates are obtained, the application can invoke the authenticate module which implements the mutual authentication algorithm with application/user specified public and secret key lengths. A session key is returned by the authenticate module. The API for the user of this system is

```
public static boolean setup(); /* registration */
public static AESKey authenticate(Socket connectSocket, int publicKeyLength,
int secretKeyLength); /* mutual authentication */
```

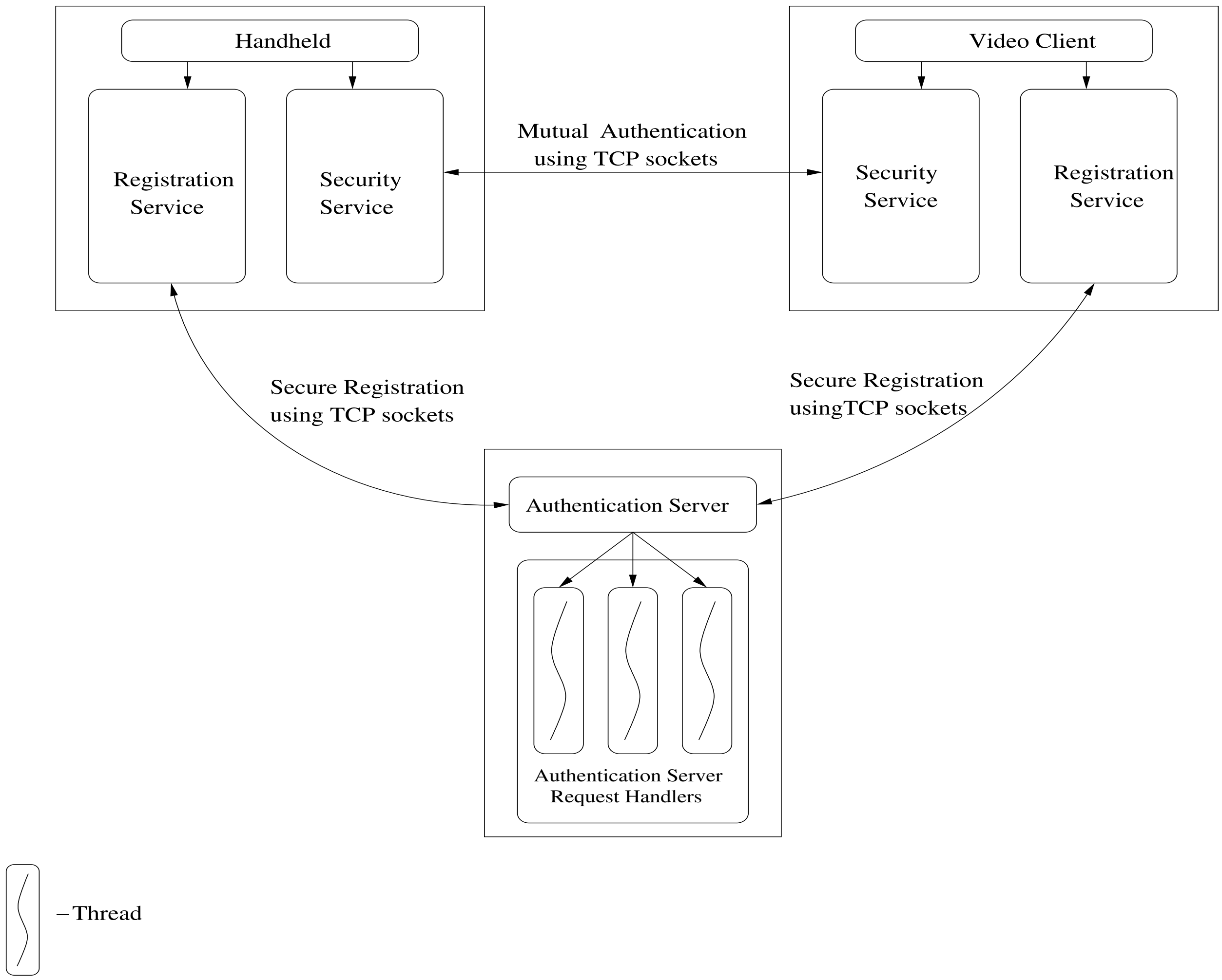


Figure 4.4: The Components for Secure User ID System

4.5 Secure User Mobility system

The User Mobility system is implemented on the Handheld and Video Client. It invokes the cryptography libraries for the purpose of security. The components for the Secure User Mobility system is shown in Fig 4.5. The User Mobility system implements the Secure Leasing Protocol and also manages the video state information. In the current implementation, the video state information is stored in a main memory table. The Mobility service assumes that a socket connection has been made and a session key has been established between the handheld and service provider. The following API is provided by the User Mobility system on the handheld :

```
HUMobilityService(Socket TCPConnection, AESKey secretKey, MobilityState state);
/* constructor */
public void renewLease() throws HUMobilityServiceException; /* lease renewal */
```

On the video client side, a Lease Renewal manager is started whose responsibility is to renew the leases. A Garbage Collector is also started to stop playing the video as soon as the lease renewal fails. Also, if the lease is renewed successfully, an updated state information is passed to the handheld.

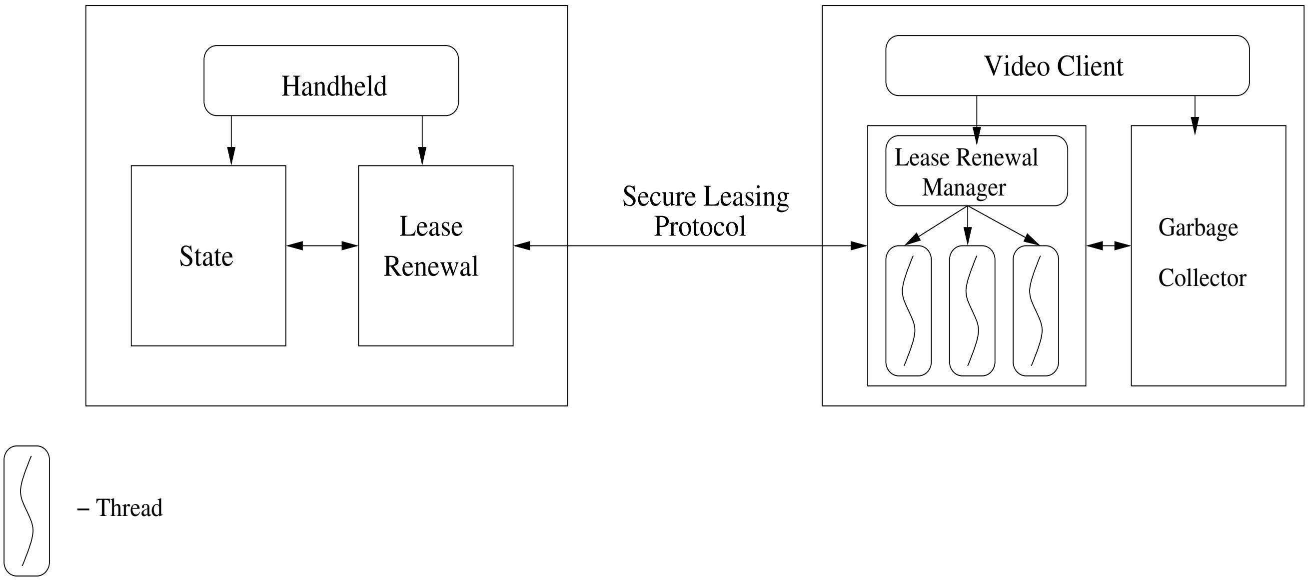


Figure 4.5: The Components for Secure Mobility System

4.6 Location Aware System

The location aware system is implemented as a simple broadcast protocol over the wireless link. The assumption is that the handhelds and service providers(video clients) are connected in a wireless ad-hoc network. The handheld broadcasts a Service Request message at a well known port over a UDP socket. On the service provider side, a daemon is waiting for such broadcast requests on a UDP socket at the well known port. The components for the Location Aware System are shown in Fig 4.6.

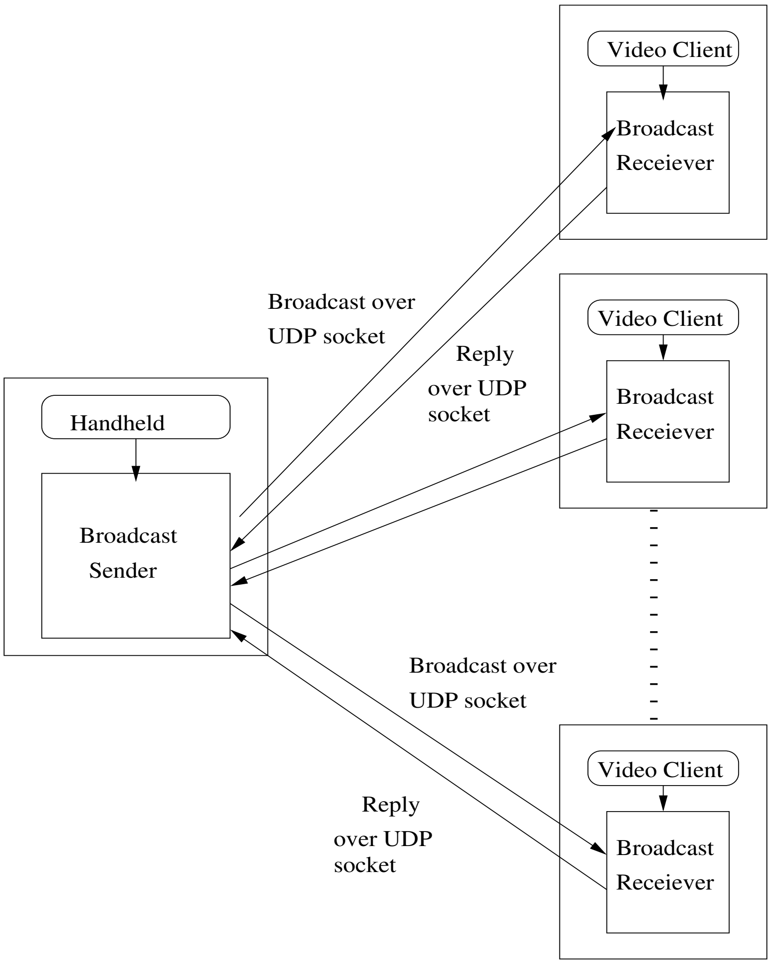


Figure 4.6: The Components for Location Aware System

4.7 Video Client and Server

As mentioned before, the video client/server pair is implemented using MSP [21]. The Video Client and Server provide API for starting the video, stopping the video and obtaining the state of the video.

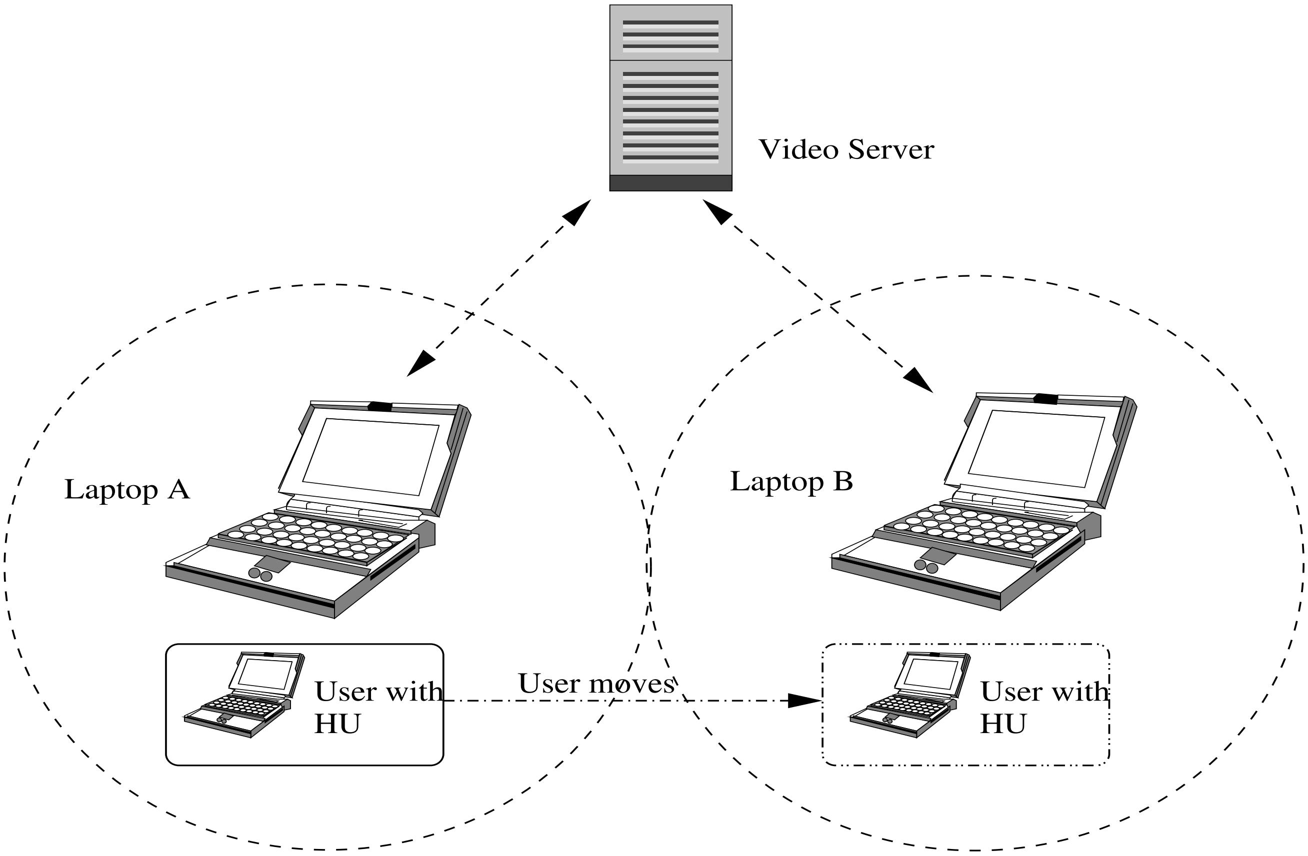
Chapter 5

Experimental Results

In this chapter, we describe the experiments conducted to measure the performance of the system. In the subsequent sections, we describe the experimental setup and the measurement values, followed by discussion on the values.

5.1 Experimental Setup

Fig 5.1 describes the experimental setup. The laptops and handhelds belong to a single domain. Each user has a personal handheld device and is free to move around anywhere in this domain with the handheld device. The experimental measurements were taken for the case when the user moves within a single domain. In the Discussion section of this chapter, we describe the extra overhead that would take place if the user moves to another domain. par Laptop A and Laptop B act as video clients. Laptop A also acts as the Authentication Server for the domain. The handheld devices are the personal devices of the users in the system. The authentication server, video clients and handheld devices are setup in a wireless ad-hoc network using 802.11 Wireless Ethernet. All of the software components of our system run on JVM. The video clients and authentication server use Sun JDK 1.3 and the handheld devices use Sun Personal Java 1.0. The machine specifications, used in the experiments, are given in Table 5.1, Table 5.2, Table 5.3 and Table 5.4. Table 5.5 gives details about the wireless card used for the experiments. The specifications for the authentication server are the same as that for Video Client 1.



Laptop A : Video Client and Authentication Server

Laptop B : Video Client

Figure 5.1: Experimental Setup

Table 5.1: Video Server specifications

Type	Dell PC Dimension XPS B733r
Operating System	Microsoft Windows NT 4.0 Professional
RAM	256 MB
CPU Type	Intel Pentium III
CPU Clock Rate	733 MHz

Table 5.2: Video Client 1 specifications

Type	IBM ThinkPad T20
Operating System	Microsoft Windows 2000
RAM	128 MB
CPU Type	Intel Mobile Pentium III
CPU Clock Rate	850 MHz

Table 5.3: Video Client 2 specifications

Type	IBM ThinkPad T20
Operating System	Microsoft Windows 98
RAM	128 MB
CPU Type	Intel Mobile Pentium III
CPU Clock Rate	850 MHz

Table 5.4: Handheld specifications

Type	HP Jornada 680
Operating System	Microsoft Windows CE
RAM	16 MB
CPU Type	Hitatchi SuperH SH-3
CPU Clock Rate	150 MHz

Table 5.5: Wireless card specifications

Type	ORINOCO WaveLAN Card
Protocol	802.11 Wireless Ethernet
Bandwidth Limit	11 Mbps
Setup-mode	Peer-to-peer Ad-hoc mode

5.2 Measurement metrics

The experimental measurements are taken to measure the extra overhead introduced by the system. Our aim is to provide seamless mobility support to users with a guarantee of the specified level of security. The relevant measurement metrics are

1. The time taken for the Registration process
2. The time lost during the handoff process from one service provider to the other
3. The accuracy of the state information
4. The memory utilization of the handheld

The time taken for the Registration process is the extra time taken for registering a new service provider or new handheld into a domain. This time is the extra overhead required by our system before the service provider/handheld can be used. Our aim is to reduce this time to the minimum.

The time lost during the handoff process is the most relevance measurement metric. It directly affects our aim to provide total seamless mobility support. In our system, this time can be divided into the following contributors

1. Discovery of a new video client
2. Mutual authentication between the handheld and video client
3. Time to transfer the state
4. Time to start the video

The accuracy of the state information is dependent on the lease renewal interval used by the Secure Leasing protocol. If we decrease the lease period, the accuracy would improve at the cost of bandwidth. We have already discussed this issue along with the possible variations in an earlier chapter. Another important consideration specific to the video on demand application is the frame loss. Theoretically, there should not be any frame loss due to the handoff process. But due to the way the video file is encoded, the stopping frame could be any type of frames in the video file. Because only I frame contains the full image information, if stopping frame is B or P frame, then when the session is resumed , the first few frames will be scramble images, until the next I frame is shown. For example, in case the video encoding pattern is (IBBPBBIBBPBB...), the worst-case frame loss caused by session handoff will be 5. We can improve on this by always sending the last I frame before the stopping frame. In that case, no frame will be lost during the handoff.

The memory utilization on the handheld becomes an important measurement metric considering the limited memory capacity of currently available handhelds. We would like to decrease the code sizes of our components and also the memory allocation to the system.

5.3 Results and Observations

In this section, we describe our experimental results and their explanations. The following scenarios are tested

1. A user with a handheld registers in a new domain.

2. Time lost during the handoff process as the user moved from one service provider to the other.

All of the experiments were conducted during periods of low load so that all the relevant wireless bandwidth was available.

5.3.1 Time taken for User Registration

The purpose of this experiment is to measure the time taken to perform user registration. This time is measured from the time a user request is submitted till the time the keys and certificates are obtained and stored. Hence, this time includes the time to create the messages, send them over the wireless link to the authentication server, verification of the message, generation of keys and then the reverse process of sending back the keys and certificates to the user over the wireless link. The user with the handheld approaches the authentication server and the measured times are shown in Fig 5.2.

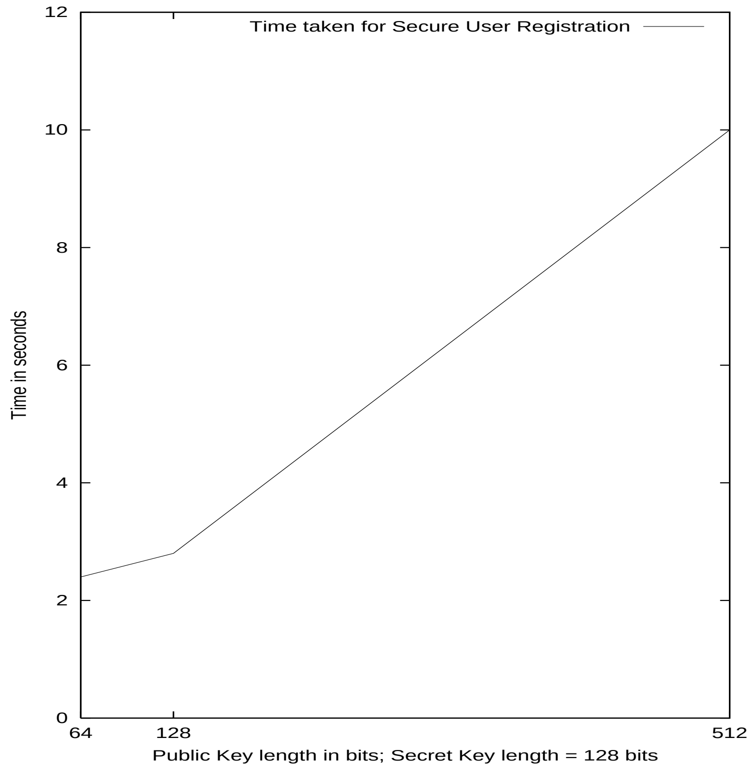


Figure 5.2: Time taken for Secure User Registration

We took measurements for different values of the desired public key length. The desired secret key length has been kept constant at 128 bits. The measured times did not change significantly with the increase or decrease with the length of the secret keys. As can be seen from the graph, as we increase the length of the desired RSA public key length, the time taken for secure user registration increases exponentially from 64 bits to 128 bits to 512 bits. As the length of the key length increases, the time to generate the keys and also encrypt/decrypt the messages increases in the RSA algorithm. This is due to the modular exponentiation involved in the encryption and decryption algorithm of RSA. Increasing the length of the keys corresponds to increasing the level of security of the system [28]. However, as observed in the graph, greater security comes with a penalty in the performance. There is an important tradeoff between security and performance.

5.3.2 Time lost due to handoff process when the user moves within the same domain

The purpose of this experiment is to measure the time lost to restart the video as the user moves from one video client to the other. This is also the time taken to start the video for the first time. The scenario is that the user either gives a request to start the video at the current location or that the user has moved away from a video client and would like to continue to watch the video at a new location. The time measured is the time required for the video to start playing on the video client from the correct desired frame. The relevant contributors to this time are described below

Discovery of a new video client

The time required to discover a new video client is dependent on the location aware system. The time required to process the broadcast request is almost negligible. However, the time required to successfully transmit the broadcast message and reply over the wireless link is dependent on the load on the wireless network and the error rate. Since these factors are unpredictable and varying, we cannot give a consistent measured time. However, in most periods of low load the time taken was found to be close to be negligible (approaching 0 seconds).

Mutual authentication between the handheld and video client

The time required for mutual authentication is measured from the time the request for authentication is submitted till the authentication is successful and the session key is established. Thus, this time includes the time required for the creation and transfer of all the messages required for the authentication as well as the time required to generate the random challenges and session key. The measured values are shown in Fig 5.3

We took measurements for different values of the desired public key length. The desired secret key length has been kept constant at 128 bits. The measured times did not change significantly with the increase or decrease with the length of the secret keys. This graph is very similar to that for the user registration. As can be seen from the graph, as we increase the length of the desired RSA public key length, the time taken for mutual authentication increases exponentially from 64 bits to 128 bits to 512 bits. As the length of the key length increases, the time encrypt/decrypt

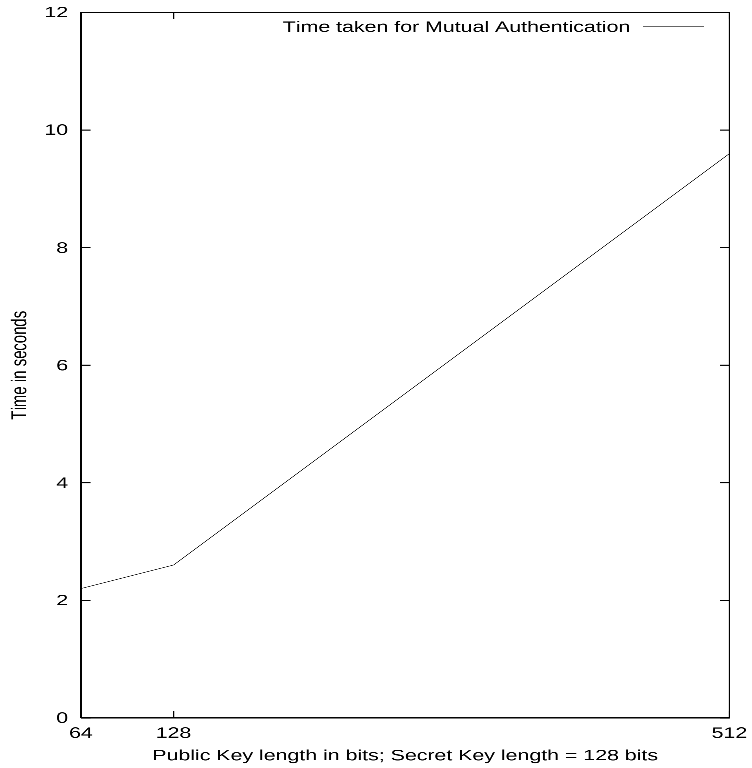


Figure 5.3: Time taken for Mutual Authentication

the messages and verifying the certificates increases in the RSA algorithm. This is due to the modular exponentiation involved in the encryption and decryption algorithm of RSA. Increasing the length of the keys corresponds to increasing the level of security of the system [28]. However, as observed in the graph, greater security comes with a penalty in the performance. We again observe an important tradeoff between security and performance.

Time to transfer the state

This time is mainly governed by the time required to send the state information over the wireless link. The state information in our case is very short and so we do not estimate this time to be significant. As observed in the subsection describing the time to discover a new client, this time is unpredictable but not significant.

Time to start the video

The time to start the video is dependent on the video on demand application used. In our implementation, the time for the first video frame to be displayed on the video player on Video Client1 was found to be 4 seconds in average using 100 Mbps Wired Ethernet connection between the Video Client1 and Video Server. However, this time is purely dependent on the video on demand application used and is beyond the scope of our system.

5.3.3 Time lost due to handoff process when the user moves across domains

When the user moves across domains, he/she would have to register the handheld in a new domain to obtain a temporary certificate. This certificate would contain the same keys as the permanent certificate but would be signed by the new authentication server. This time for registration has a graph similar to Fig 5.2. After the new temporary certificate is obtained, all the measurements of the previous section hold true. Thus when the user goes to a new domain, there will be an extra overhead of registering in the new domain. Once registered, there would be no more extra overhead by our system due to the new domain.

5.4 Memory utilization on the handheld

We measured the byte code sizes of the various components of our system on the handheld. Table 5.6 shows the details.

Table 5.6: Java byte code sizes on the handheld

Component	Code size
Security (including Cryptography libraries)	118 KB
User Mobility	6KB
Location System	3 KB
VOD application	24 KB

As can be observed, the security components have the largest byte code size. This is because of the more complicated code and extra messages for the security protocols compared to the other components. We also found that the total physical memory allocated on Jornada did not exceed 5 MB at any instance of the deployment of the system. This includes the memory, allocated for Personal Java JVM. The 5 MB memory size is quite significant, compared to the total memory size (combined ROM size and RAM size) of 16MB on the HP Jornada 680.

Another observation is the memory size of the certificates used in the system. With the increase in the length of the security keys, the memory size of the certificates also increases.

5.5 Discussion

The experimental results and observations described in this chapter bring about in sharp focus the various tradeoffs that result in the system with the introduction of security. We identify the following tradeoffs

1. Tradeoff between security and performance
2. Tradeoff between security and memory size
3. Tradeoff between security and bandwidth

The tradeoff between security and performance is well observed in the graphs presented in this chapter. The tradeoff between security and memory size is observed by the code sizes and the certificate sizes. The tradeoff between security and bandwidth is due to the increase in the size of the certificates with the increase in the lengths of the keys. This results in a significant tradeoff in the bandwidth during the authentication and registration process. Considering these tradeoffs, we would like to emphasize the need to flexibly change the level of security to improve performance, memory size or bandwidth. We achieve this by a provision to change the quality of security at the application level.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This work makes several important contributions to the pervasive computing research. We have established through the work in the thesis that the use of handheld devices for personal mobility support is a feasible solution. We have also established that handheld devices can be used for secure user identification, using sophisticated security protocols. Further, we have established several important security tradeoffs while designing a system with handheld devices of limited processing and memory capability, and wireless links of limited bandwidth. These tradeoffs are tradeoff between security and performance, tradeoff between security and memory and tradeoff between security and bandwidth.

We have approached the design of the system as a layered and modular architecture. This enables third part developers to plug in a different implementation of the components, if required. The system addresses three problems - secure user identification, secure user mobility and location awareness. We have built separate subsystems for solving each of these three problems using handheld devices. The system is viewed as consisting of several domains. Each domain has an authentication server, one or more service providers and users. Each user has a personal handheld device. The user moves along with the handheld device within the domain or across domains. The challenge is to build a scalable, secure, modular and flexible infrastructure to have seamless mobility support of the desired service. The security of the system is based on public/private key pairs, certificates and session-specific secret keys. The Secure User Identification system consists of two phases : Secure User Registration and Mutual Authentication between handhelds and service providers. The User Mobility system is based on Secure Leasing Protocol which keeps track of the movement of the user as also manages the service state information for mobility support. We have noted an important tradeoff between accuracy of the state information and bandwidth utilization using the Secure Leasing Protocol. The Location aware system is based on the range of the wireless link and it uses a broadcast based protocol to discover service providers in the vicinity of the user.

One of the key highlights of the system is the provision for changing the quality of security of the system. This is achieved by having multiple certificates of different key lengths and using one of these certificates based on the specification by the user/application. The experimental results confirm the need for such a provision and we show the affect on performance by using different key lengths.

Overall the system is a step in the direction of building smart infrastructures using heterogeneous handheld devices. With the rapid interest in building smart, feasible, secure and easily deployable solutions for user mobility and user identification, this thesis endeavors to be a stepping stone towards building a truly pervasive world of intelligent software and hardware solutions.

6.2 Future Work

The attempt in designing and building intelligent secure information spaces using heterogeneous handheld devices is yet incomplete, there are many open areas of research. Future work along this direction lie in improving and enhancing the existing system as also in extending the existing system. We discuss some of the potentially rewarding areas of future work below.

Improving existing system The following are some of the improvements possible in the system

1. Security protocols : The RSA cryptography library could be replaced by elliptic curve cryptography library[2] to improve performance and reduce certificate size for the corresponding same security guarantees.

2. Location Aware System : The current implementation of location aware system is based on the range of the wireless link. This implementation can be made more sophisticated and also should be made more secure and preserve the privacy of the users, whenever necessary.

3. Leasing Protocol : As observed in the discussions, the leasing protocol consumes a lot of bandwidth. Efforts should be made to build or include systems that can predict the movement of the user with more sophisticated techniques.

Extending existing system : The following are some potential extensions to the system.

1. Access Control : The authorization and access control mechanism proposed in the thesis for video on demand application should be implemented. This will require defining policy rules for assigning privilege levels to the users as also deciding pre-requisite conditions. The support for access control is a very desirable logical extension to the existing system.

2. Service State management : It will be interesting to look into the various possibilities and issues of state management. One of the possible issues is replicating the state management data into a mobility database for the purposes of fault tolerance and security. Another issue is to exploring a good structure for storing state data obtained from multiple services or from various configurations of the same service.

3. Key expiry : The public key certificates currently have a time expiry field. The certificates become invalid after the expiry time. However, to implement such a feature requires all the clocks to be synchronized. It is worthwhile to look into a solution which does not require such restrictions.

4. Handheld as service provider : With the increasing improvement in the processing, memory and storage capability of the handhelds, it would be possible that the handhelds themselves become service providers. It would then become interesting to look at the new problems arising due to this. For example,for a video on demand application, the handhelds would become video clients and the video would be streamed directly to it. In such a case, the user mobility and identification could shift to the secure identification of the handheld with the gateway and the user mobility problem would be to manage the state information as the handheld switches among the gateways while moving from location to location.

References

[1] Active Spaces project at UIUC. Available from http://choices.cs.uiuc.edu/ActiveSpaces.

[2] Elliptic curve cryptography whitepaper. http://www.certicom.com/research/wecc3.html.

[3] HP Cooltown. Available from http://cooltown.hp.com.

[4] IBM T Spaces. Available from http://www.almaden.ibm.com/cs/TSpaces.

[5] Jini. Available from http://www.sun.com/jini.

[6] Location based services. http://www.mobilein.com/location_based_services.htm.

[7] NIST Resources For Role Based Access Control. Available from http://csrc.nist.gov/rbac/ and http://hissa.nist.gov/rbac/.

[8] Role Based Access Control Papers. available from http://www.list.gmu.edu/publications.htm.

[9] SESAME. Available from https://www.cosic.esat.kuleuven.ac.be/sesame/.

[10] The Ninja Project. Available from http://ninja.cs.berkeley.edu.

[11] The Official AES Page at NIST. http://csrc.nist.gov/encryption/aes/ .

[12] UPnP. Available from http://www.upnp.org.

[13] Wireless LAN Association. available from http://www.wlana.org/index.htm.

[14] Jalal Al-Muhtadi, Dennis Mickunas, and Roy Campbell. *Tiny SESAME on the HP Jornada*. Technical Report UIUCDCS-R-2001-2198, Department of Computer Science, University of Illinois at Urbana-Champaign, January 2001.

[15] H. Balakrishnan, S.Seshan, and R.Katz. Improving Reliable Transport and Handoff Performance over Wireless Networks. In *ACM Wireless Networks*, December 1995.

[16] V. Bharghavan and C.V Ramamoorthy. Security Issues in Mobile Communications. In *International Symposium on Autonomous Decentralized Systems*, April 1995.

[17] Dan Brown. Techniques for Privacy and Authentication in Personal Communication Systems. In *IEEE Personal Communications*, August 1995.

[18] Martin G. Brown. Supporting User Mobility. In *IFIP World Conference on Mobile Communications*, 1996.

[19] Stefano Campadello and Kimmo Raatikainen. Agents in Personal Mobility. In *Proceedings of the First International Workshop on Mobile Agents for Telecommunication Application*, October 1999.

[20] M. Chandak. *U-SESAME : Achieving a Portable Authentication Access Control, and Delegation Protocol*. Masters Thesis, Department of Computer Science,University of Illinois at Urbana-Champaign, 1999.

[21] C.Hess. *Media Streaming Protocol: An Adaptive Protocol For the Delivery of Audio and Video over the Internet*. Masters Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1998.

[22] C.Perkins and E. Royer. Ad-hoc On-Demand Distance Vector Routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, February 1999. also available from `http://alpha.ece.ucsb.edu/ eroyer/txt/aodv.ps`.

[23] Cynthia Irvine and Timothy Levin. Quality of Security Service. In *Proceedings of the New Security Paradigms Workshop*, September 2000.

[24] D. Johnson. Scalable Support for Transparent Mobile Host Internetworking. In *Mobile Computing*. Kluwer Academic Publishers, 1996.

[25] Charles R. Merrill. A Cryptography Primer. `http://www.cla.org/RuhBook/chp2.htm` .

[26] B. Clifford Neuman and Theodore Tso. Kerberos: An Authentication Service for Computer Networks. In *IEEE Communications*, September 1994. also available from `http://web.mit.edu/kerberos/www/papers.html`.

[27] Nissanka B. Priyantha, Anil Chakraborty, and Hari Balakrishnan. The Cricket Location-Support System. In *Mobicom*, 2000.

[28] Ron Rivest, Adi Shamir, and Len Adleman. A Method for obtaining Digital Signatures and Public Key Cryptosystems. In *Communications of the ACM*, February 1978. pages 120 - 126.

[29] E. Royer and C.K Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. In *IEEE Personal Communications Magazine*, April 1999.

[30] Ravi Sandhu, Venkata Bhamidipati, and Qamar Munawer. The ARBAC97 Model for Role-Based Administration of Roles. In *ACM Transactions on Information and Systems Security*, February 1999.

[31] Douglas R. Stinson. *Cryptography: Theory and Practice*. CRC Press, 1995.

[32] Shigefusa Suzuki and Kazuhiko Nakada. An Authentication Technique Based on Distributed Security Management for the Global Mobility Network. In *IEEE Journal on Selected Areas in Communications*, October 1997.

[33] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The Active Badge Location System. In *ACM Transactions on Information Systems*, January 1992.

[34] Mark Weiser. The computer for the twenty-first century. In *Scientific American*, September 1991.

[35] Mark Weiser and John Seely Brown. The Coming Age Of Calm Technology. available from `http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm`.

[36] Yuliang Zheng. An Authentication and Security Protocol for Mobile Computing. In *Proceedings of IFIP World Conference on Mobile Communications*. Edited by J. L. Encarnacao and J. M. Rabaey, pp.249-257, Chapman & Hall, 1996.

[37] Lidong Zhou and Zygmunt J. Haas. Securing Ad Hoc Networks. In *IEEE Network*, 1999.

[38] Bo Zou. *Mobile ID Protocol : A Badge Activated Application Level Handoff of a Multimedia Streaming To Support User Mobility.* Masters Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2000.