A secure authentication infrastructure for mobile users

Gregor v. Bochmann and Eric Zhang
School of Information Technology and Engineering, University of Ottawa
800 King Edwards, Ottawa, Ontario K1N 6N5 P.O. Box 450 Stn.A
{bochmann, ezhang}@site.uottawa.ca

1. Introduction

With the introduction of the World Wide Web, electronic commerce has begun to enhance the traditional commerce practice in the exchange of merchandize and information. Recently, the emergence of wireless networks and mobile devices has introduced further commodities for using telecommunication services and electronic commerce transactions on the go. Mobile commerce may be defined as the exchange or buying and selling of commodities, services or information on the Internet through the use of mobile handheld devices. However, in this chapter we take a little larger view of mobile commerce by including the notion of "mobile users" which means that the user may be in a foreign country, in an unusual environment and may use, for the electronic commerce session, any device that happens to be available, for instance a workstation in a hotel business lounge or the handheld device belonging to a friend.

While many aspects of **mobile** commerce are identical to the same aspects of normal electronic commerce, in general, there are certain aspects that are specific to mobile commerce. These aspects are either related to the limitations of handheld devices, such as (a) the limited computation power of most handheld devices related to CPU power and battery life and (b) certain limitations of the communication bandwidth which depends on the particular wireless networking technology in use, or related to the notion of "mobile users", such as (c) the security implications of using unknown ad hoc devices that are locally available and (d) the fact that the user may need to be authenticated by a foreign organization that provides network access facilities and other services within the foreign domain where the user temporarily resides.

In this chapter, we principally deal with the problem of user authentication and the establishment of trust relationships between the different parties involved in an electronic commerce transaction. In this context, we consider specifically the aspects (c) and (d) above which are specific to mobile commerce. To a lesser degree we are also concerned with aspect (a) and (b).

In Section 2, we explain the requirements for an authentication infrastructure for electronic commerce by identifying the partners that are typically involved in transactions and the trust relationships that are required. We also describe the security requirements, such as authentication, access rights, payment credentials, anonymity (in certain cases), as well as the traditional requirements such as privacy and integrity of message exchange. Then we review in Section 3 first the three general schemes for authentication, namely authentication based on a shared key, on public/private key pair, and on biometric information. After this introduction, we review certain authentication protocols that are currently in use or proposed, and discuss their applicability to electronic commerce applications and in particular to the requirements of mobile users as identified by points (c) and (d) above.

In Section 4, we then propose a secure authentication protocol for mobile user that (1) combines ease of password-based authentication with the power of public key technology, (2) can be executed on an ad hoc device that happens to be available in the environment of the mobile user, and (3) provides authentication support for (i) the normal electronic commerce transactions, (ii) for obtaining the necessary transmission resources from the local Internet service provider (ISP) (e.g. to view a high-quality video from some given video-on-demand server), and (iii) for authentication to arbitrary third parties (e.g. for a secure IP-telephone conversation). The protocol is based on a password-based user identification procedure performed by the authentication authority where the user is registered, and also involves an agent of the foreign domain where the mobile user is visiting. The use of public key technology is limited in order to satisfy the limitations of handheld devices concerning computing power and battery life.

We believe that the authentication protocol described and analyzed in Section 4 contains a number of interesting features that make it suitable as an alternative to the other authentication protocols that can be used for mobile commerce, as explained in the Conclusions.

2 Requirements for authentication infrastructure

In order to discuss the requirements for authentication in mobile e-commerce applications, we start with the presentation of a typical application scenario. We then identify specific roles played by the different parties involved and discuss the trust relationships between the parties and other security requirements.

2.1 Example scenario

We consider the following scenario of a mobile user of e-commerce facilities: Bob has a subscription to an e-learning course with company Teach-Inc. Now Bob is on a business trip in a hotel in Paris and uses a rented portable computer in his hotel room to study another chapter of the subscribed course. Then he checks the balance of his personal account at his Bank in Canada and buys some food for delivery from the near-by Paris-Bistro restaurant. The next day, he travels through Paris. After an IP-telephone conversation with his friend Alice using his hand-held PDA/phone through a wireless Internet connection available in a shopping center, he decides to do some money transfer from his Montreal account using the same PDA device. Then he uses the PDA to watch an adult movie from an Internet video store.

2.2 Generic roles in e-commerce

In order to clarify the discussion of security requirements, we first try in the following to identify the major parties and their roles within the e-commerce environment from a generic point of view. We identify the following basic roles:

- User: This is the person (or agent) that takes initiatives for e-commerce transactions. In the context of mobile e-commerce, it is typically a person on the move, using a mobile terminal, such as a PDA or mobile phone, or a fixed terminal which is publicly available or belongs to third parties (e.g. a visited friend) not involved in the transaction. In our scenario, Bob is the user.

- Service provider: This is an organization or a person that provides a service that the User is interested in. It includes the computer through which the service is effectively provided. In many cases, the service transaction also involves real goods, such as the delivered food in our example. The service may be involve a fee to be paid by the user, or may be freely available. Examples of service provides in our example are: the Teach-Inc company, the restaurant, the bank, the video store, and the long-distance telephone company used for the telephone call with Alice.

- Network access provider: This is the organization that provides network access to the mobile user. Although this may be considered a service provider, we distinguish this role because of the special role of the network access service and the related security requirements (to be discussed below). Unlike other service providers, the network service provider either provides free service for all users, as for instance the wireless Internet service provider in the shopping center, or will provide at least initial free access to any new user to allow his/her identification and/or establishment of payment procedure.

- Third parties: These are other persons or organizations that participate in the transaction initiated by the user. For instance, if we consider the telephone conversion of Bob with Alice as a transaction, Alice plays the role of a third party.

In addition, there are certain parties that play the role of providing appropriate references about the user. We can identify the following reference roles:

- Credit reference: This is a role typically played by a credit or debit card organization. For example, Bob may use a credit card or some equivalent electronic version as payment instrument for his transactions with the restaurant or the video store.
- Authentication authority: This could be an authority that attests that the person in our scenario is Bob XYZ that lives in Ottawa at 300 Stewart Street, or an authority that attests Bob's age to allow him the viewing of an adult movie. This role is also played by the government of Canada when it emits Bob's passport which is required for the visit to Paris.

2.3 Various trust relationships

Depending on the particular e-commerce application, different trust relationships are required between the different parties involved. Based on our example scenario, we identify the following most important relationships between the generic roles:

- Authentication

Applications that involve personal data of the user require the authentication of the user by the service provider. Inversely, the user usually also wants to authenticate the service provider so that he/she could be assured that he/she is dealing with a trustworthy party. Furthermore the transaction may involve the exposure of additional personal information. This is the case when Bob accesses his banking service. Mutual authentication is usually also required between the user and any third party, especially in the case of a communication service. An example is the telephone call between Bob and Alice.

- Access rights

Many e-commerce services could in principle be provided to anonymous users, that is, the service provider does not need to authenticate the user. For instance, Teach-Inc does not really care whether it is Bob that accesses the e-learning course, as long it is assured that the user has obtained the access rights to the course (through some previous transaction in which some access permit would have been established, probably against payment). Another example is Bob's viewing of a video; here the service provider must satisfy the policy that adult movies can only be seen by users of a certain age. In Canada, the user's driver's license is typically used as a reference for checking the age of a person. For e-commerce purposes, a public key authentication certificate may also include such information.

- Payment credentials

Payment is an essential part of the e-commerce framework. Payment methods can be classified into cash-based methods and methods based on payment credentials, such as credit and debit cards. The latter payment methods involve a credit institution as a third party that asserts that the service provider will be paid the amount due as long as this amount is within the user's credit limit. All transactions in our example scenario involve payment, except for the viewing of the on-line course for which the access rights were obtained through an earlier transaction during which Bob subscribed to the particular course. Payment may also be involved for the use of communication services, including network access, unless this service is provided free of charge.

2.4 Other security requirements

In addition to authentication, access control and payment credentials discussed above, e-commerce applications often have other security requirements, such as the following:

- Privacy of communication

The communication between the user and the service provider, and possibly the other parties participating in the transaction, should remain private, that is, should be protected from leaking out to other parties not involved in the transaction. Sometimes, certain information should only be available to specific parties in the transaction, as for instance in the SET protocol for electronic credit/debit card payment, where the store will to see the details of the goods purchased by the user, but not the credit institution.

- Integrity of message exchanges

Message integrity ensures that messages exchanged between the parties involved in a transaction are not changed during transmission neither through transmission errors nor intruders.

- Verifiable signatures

Signed messages or documents are required in case of important transactions. The signature by user A of a given message becomes significant if the signature is verifiable in the following sense: The receiver of the message can verify that the message was signed by user A, and the user cannot repudiate the signing of the message, that is, a third party playing the role of an arbiter may be able to determine whether it was user A that signed the message or some other person.

- Anonymity

As mentioned above, many services could in principle be provided to anonymous users. In certain situations, anonymity becomes a user preference or requirement. For instance, in many situations the user does not want any other person to know that he/she is buying certain goods. In other situations, the user may not want to be recognized, or the user wants that his presence in the particular geographical area remains secret. In order to allow an anonymous user to participate in e-commerce applications, it is nevertheless required to verify access rights or payment credentials. It is therefore important that these references can be provided without interfering with the user's anonymity.

3. Review of authentication methods

In this section we discuss authentication methods and protocols, and how they could be used for mobile applications. Before reviewing existing authentication protocols, we briefly present the major generic approaches to authentication. Finally, we discuss some common issues, such as the need for an authentication authority for mobile users getting involved in new relationships, and the need for trusting the software in the devices that the mobile user may happen to use.

3.1. Generic approaches to authentication

Generally, authentication is accomplished through a sharing secret between user and authentication server. The server could be a stand-along workstation which is in charge of authentication or a module integrated into a multi-functional server. In terms of type of shared secrete, the authentication methods can be cataloged to three sub-catalogs: symmetric authentication, asymmetric authentication and biometrics authentication.

3.1.1 Authentication based on a shared secret

Also called symmetric authentication, this approach to authentication is based on a secret key that is shared among two parties or more. Typically, these parties are the user and a service provider. Basically, mutual authentication is realized between the two parties by the exchange of messages that are encrypted by a symmetric encryption algorithm using the shared secret as the key. By decrypting the message with the same key, the other party can verify that the sender is in possession of the secret key. If the key is not exposed, correct authentication is assured. The common password authentication schemes currently used by most servers are based on this principle.

The major challenge of this approach is key management, especially key distribution and the strength of the key. The approach is suitable for centralized systems where a central server each potential users. Key distribution is accomplished when the user first registers him/herself at the central server. Although applying the same key for message encryption/decryption repeatedly increases the possibility of breaking the key, the strength of the key could be improved by changing the password periodically based on pre-built agreements between the user and the server.

3.1.2 Authentication based on public keys

Also called asymmetric authentication, this approach is based on a public/private key pair. Authentication is based on the possession of the private key, and the other parties in the transaction would use the public key for encrypting or decrypting messages. A public/private key pair provides for the authentication of the party having the private key; for the authentication of the other party, another private/public key pair is required. For instance, a server could authenticate a user by sending some random number encrypted by the public key of the user, which could only be decrypted using the private key; the user should then return the decrypted random number to the server as proof of his identity.

Public/private key technology also provides for verifiable signatures. Normally, a the message to be signed is hashed and the hash value is encrypted with the private key which results in the signature which is sent together with the original message. By decrypting the signature with the public key and comparing the result with the hash value of the received message, the recipient of the message verifies the signature. This

verification can be performed by any party having received the message and the signature. Since only the sender has a copy of the private key, he cannot repudiate the signing of the message.

In order to provide reliable information about the public keys of various users and organizations, a Public Key Infrastructure (PKI) is provided which consists of a collection of authentication authorities that give out signed authentication certificates which include the public key of the user or service provider together with certain attributes, such as the name and possibly the address, employment, age, etc.

3.1.3 Authentication based on biometric information

Instead of creating big random numbers that serve as shared or private/public keys, this approach is based on biometric information that is characteristic of the user. Examples are of such information are fingerprints, eyeball scans and DNA recognition. This authentication approach cannot be used for authenticating organizations. Like the shared key in the case of symmetric authentication, the biometric information of the registered users is stored in the database of a central server which represents the authentication authority. Authentication is performed by reading again the biometric information on the individual and comparing the result with the value stored in the database.

3.2 Discussion

The public/private key approach to authentication is basically much more suitable for e-commerce applications because, once a user is registered with an authentication authority based on PKI, he/she can be authenticated by any other party without any pre-established relationship. In contrast, shared key and biometric authentication requires a pre-established relationship with the party by whom the user wants to be recognized. In addition, the public/private key approach provides at the same time for verifiable signatures which are very important for many e-commerce application.

Unfortunately, the algorithms performing public/private key encryption are much less efficient that shared-key encryption algorithms. This is of concern for mobile devices that usually have lower CPU power and battery limitations. Therefore one usually tries to limit the use of public/private key technology for mobile devices as much as possible.

Another issue is the secure storage of the private key. The public/private keys are much longer than password and cannot be remembered by the human user. Therefore they must be stored in computer-readable form and only be accessible and useable by the user that owns it. In the case of mobile commerce, the key may either be stored in a personal mobile device (PDA or mobile phone) belonging to the user, or in a small card (e.g. smart card, SIM card or SD memory card) readable by the device used by the user.

3.3 Existing security protocols

We mention in this section a number of security protocols that could be applied for mobile commerce applications and shortly discuss their benefits and limitations.

3.3.1 Radius

The Radius mechanism using CHAP (challenge handshake authentication protocol) [1] is widely used by Internet Service Providers to give point-to-point protocol access with mobility [2]. The example shown below indicates that this kind of protocol is not compatible with our mobile commerce requirements. The Radius-CHAP message exchanges are presented in Figure 3. The protocol uses a challenge value CV. K is a key shared by the network access server (NAS) and the authentication authority, called Radius Server.

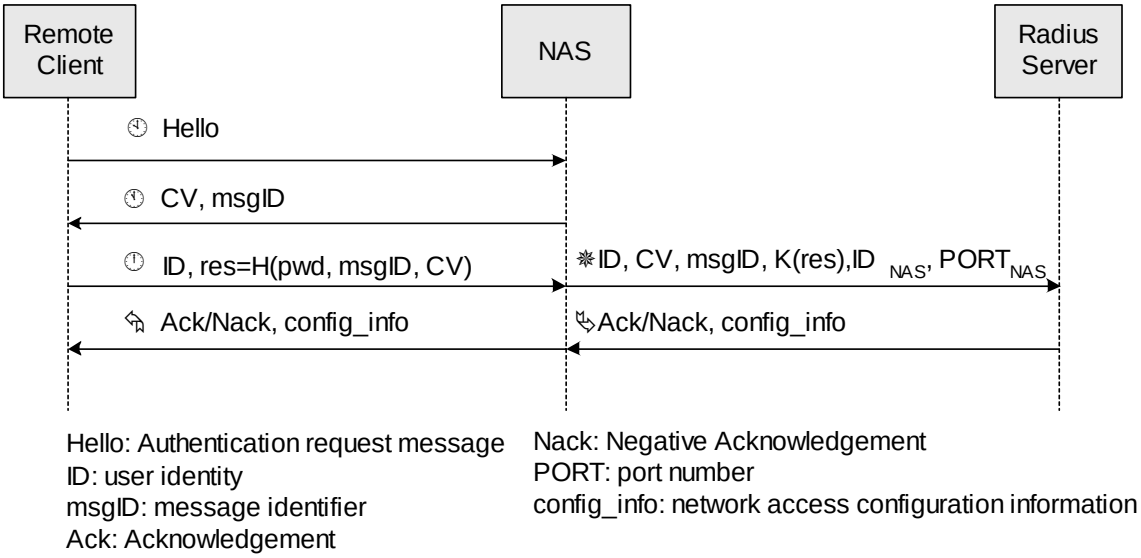

Figure 3: RADIUS-CHAP message exchange.

The user first communicates with the NAS to be given a challenge value. The user gives the answer (*res*) that is forwarded by the NAS to the Radius server. The latter checks the validity of *res.* The authentication answer is included in the reply.

The NAS and the Radius server are supposed to know and to trust each other. And the link between them is supposed to be secure. Anonymity cannot be provided with this scheme. Moreover, the NAS generates the random challenge value *CV* and sends it to the user in plaintext along with a CHAP identifier (called 'msgID' in the figure) which allows attackers to perform a chosen plaintext attack by guessing the password to calculate *H(pwd, msgID, CV*) and comparing the result with the value 'res' included in

the message. Radius was designed for centralized network infrastructure and fails to meet the requirement mobile users.

3.3.2 Kerberos

Like in the case of Radius, Kerberos uses a centralized authentication server where the shared password of the user is stored. This server plays the role of a centralized key distribution center (KDC) to assist in key management [3]. A ticket or authenticator is issued by the authentication server to the user for service access control as shown in figure 4 (a copy from [4]).

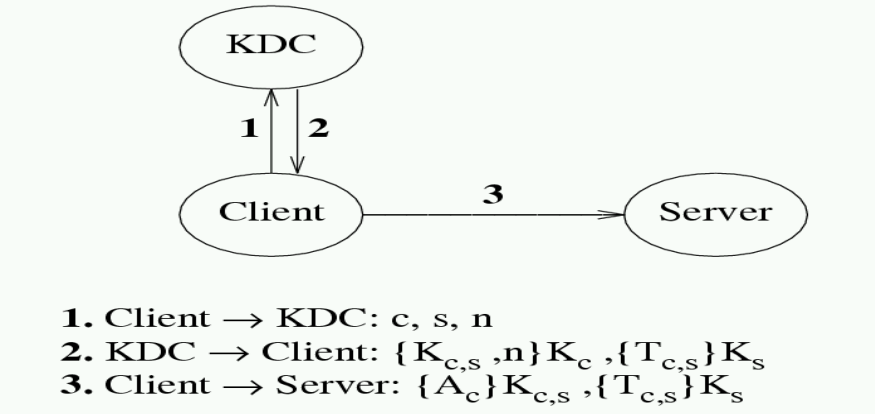

Figure 4: getting and using Initial Ticket

The ticket will be used to authenticate the user at the server providing the service and to generate a sub-session key. Anonymity cannot be provided since the client has to send out his/his identity as well as required services in clear to the KDC. This information is sent unencrypted and could be listened to by any third party sitting on the communication path.

The major challenge Kerberos faces is the first message exchange between clients and the KDC. In the scenario above, when Bob comes to the shopping center, he has no knowledge about the KDC. How could he make sure that the KDC he talks to is a real trustable KDC instead of a fake one sitting in the middle and trying to damage? Moreover, since Bob is a foreign user for the KDC in Paris, how could a secret key be distributed between them prior to authentication? While Kerberos has the function of providing a ticket for services in a foreign domain, this mechanism is impractical especially when the user's visit is unpredictable.

3.3.3 SSL

SSL stands for Secure Sockets Layer and is renamed by IETF as TLS [5] (Transport Layer Security). Originally developed by Netscape, SSL is especially used by Web browsers to provide authentication and privacy for sensitive Web applications. SSL contains various options for authentication including several versions of public/private

key authentication. The protocol also provides for a fresh shared session key that can be used for encrypting the messages exchanged over the session.

3.3.4 XML security extensions

Security Assertion Markup Language (SAML), is the first industry standard for enabling secure e-commerce transactions through the eXtensible Markup Language (XML) [6]. Independent of any particular platform, SAML enables companies to securely exchange authentication and authorization information with customers, vendors and suppliers, while the XML Key Management Specification (XKMS) [7] efficiently manages digital signatures and encryption. A supplement, XMLPay [8] provides further facilities for payment transactions to build trust-supported B2B and B2C e-commerce.

3.3.5 Smart cards and SIM card

Many types of smart cards and the SIM-card used with mobile phones contain a authentication certificate including the public key of the user (owner of the card) and some attributes (e.g. user name) and the associated private key. For security reasons, the private key will never be communicated through the card reader interface. Instead, any message to be encrypted or decrypted with the private key is transferred to the card and the result of the operation is returned to the card reader. Thus, any device that can interface with the card could perform an authentication handshake with a remote party through which the owner of the card would be identified as the user.

3.3.6 Other protocols

SSH [9] is a protocol that provides secure access over insecure channels to remote server computers, including file transfer and a command line interpreter. Two version of the protocol are available. SSH1 provides both server and user authentication, while SSH2 only provides user authentication, but it is more secure. The Diffie Hellman Algorithm [10] is used to negotiate a shared secret key.

SHTTP was designed to secure only HTML (Hypertext Markup Language) web pages. Server and client preferences and security constraint are negotiated for each web page or set of pages. The client-side public key certificates are optional, “as it supports symmetric key-only operation mode” [11].

There are also extensions of the IP protocol for mobility [12] and security [13], however, the security framework at the IP level is not very useful for mobile commerce applications.

3.4 Discussion of the requirements for mobile commerce

Comparing the authentication and other security requirements for mobile commerce discussed in Section 2 with the authentication methods described above, we come to the following conclusions:

- The public/private key technology is the preferred method for authentication since it only requires the registration of the user with a single authentication authority and allows authentication to third parties without any pre-established relationship. It also provides a simple scheme for signatures.

- The public/private key technology utilizes some form of PKI which consist of a collection of registration authorities that provide signed public key certificates which contain the public key of a user together with certain user attributes.

- In addition, commerce applications require other forms of references, such as payment credentials and other kinds of certification, such as proof of age, proof of competence, etc. Similar to public key certificates, such references could also be provided in the form of signed documents that contain just the necessary information, signed by an appropriate certification agency. For instance, a credit credential would be signed by a bank. In an extreme case, when the user wants to remain anonymous, the credit credential destined for a network access provider in a foreign domain may contain the following information: "Communication charges up to an amount of 10$ will be covered for the current user." (See Section 4 for a more detailed example).

- Among the existing authentication protocols, SSL and smart cards appear to be most interesting for mobile commerce, however, they do not provide support for payment credentials and other references for users that want to remain anonymous.

3.5 The concept of a home directory

We have seen in the earlier subsections that, whatever the authentication scheme chosen, each user has to register in at least one authentication authority. In our work on quality of service management for distributed multimedia applications and mobile users [14], we identified the need for what we called a "home directory" where the user profile and preferences are stored. In the case of IP telephony, the home directory would also play the role of the user's proxy agent, that is, it would be the place to where incoming communication requests would be sent, since the user profile would contain information about the device through which the user (who may be on the move) would accept such a request at the given time.

We note that such a home directory may also include user preferences concerning commerce applications. It may also be sensible to combine such a home directory with the function of the authentication authority mentioned above.

3.6. The need for trusted software

One of the difficulties with mobile commerce is the fact that the user may use a device that is locally available, like for example the portable computer Bob rented from the hotel. In such a case, there is the problem of trusting the software running on that device. Trusting software, in general, is problematic. As early as 1984, Ken Thompson wrote that "You can't trust code that you did not totally create yourself." [15]. In the case of the software residing in a device locally available, we could normally assume that it contains standard software, however, it is not excluded that, for instance, the previous user inserted a piece of code performing some extra tasks, such as recording all activities of the subsequent users and sending a log of these activities to a given destination for espionage, for instance. If the device contains a smart card interface and the smart card is used by the user, the malicious software may also send additional encoding and decoding commands to the smart card as part of a fake transaction with some third party without the knowledge of the legitimate user.

It is difficult to systematically exclude these possibilities of fraud. One way to reduce these risks is to download certified software from trusted service providers. However, the fraudulent software operating system that performs the download and verification of the certification may download a fraudulent software version from some other source and present to the user a window which (falsely) attests the successful checking of the certification. It appears that we can only hope that such things would occur only very infrequently.

4. Password-based authentication for mobile users with support for public key technology

In the following, we describe a new authentication protocol for mobile users which is based on a secret password shared between the user and the authentication authority and supports the creation of a new public/private key pair for which the authority provides an authentication certificate and the private key is stored in the device the user happens to use at that time. After providing an architectural overview and describing how the protocol would be used, we provide a detailed description of the protocol, discuss its properties, analyze its robustness against security attacks and discuss possible design choices for the detailed definition of the protocol.

4.1 Architecture Overview and design objectives

Let us consider part of the usage scenario described in Section 2.1: While Bob is on a business trip in Paris, he makes an IP-telephone conversation with his friend Alice using his hand-held PDA/phone through a wireless Internet connection which is available in a shopping mall provided by a third party, say *France Telecom.* We may identify the following security concerns in this context: (a) *France Telecom* wants to see payment credentials for the cost of providing the telecommunications facilities to Bob. However, Bob may want that his presence in Paris remains unknown and therefore requires anonymity. (b) Bob may want to authenticate *France Telecom* to be

sure that he uses a trustworthy carrier, although he should use end-to-end encryption to ensure the privacy of the telephone conversation. (c). To persuade Alice to accept the incoming call that claims to be from Bob, Bob's PDA must be authenticated to Alice as belonging to Bob, and vice versa. Note that the authentication procedure at Bob's side is symmetrically identical with Alice's side, and the authentication between Bob and Alice is the same as between Bob and *France Telecom;* we could therefore only focus on how Bob and *France Telecom* authenticate each other. The architecture of the authentication protocol between the latter two is shown in Figure 5.

Figure 5 provides an architectural overview including the different parties involved in this scenario. Besides the parties mentioned above, the figure also shows Bob's Home Agent and a Certification Authority. Bob's home agent plays the role of Bob's authentication authority, while the Certification Authority is part of the public key infrastructure (PKI) and allows the Foreign Agent and Bob's Home Agent to authenticate one another based on certificates of their public keys provided by the Certification Authority. The certificate of the Foreign Agent may also be used by Bob to check the authentication of *France Telecom* in our example scenario.

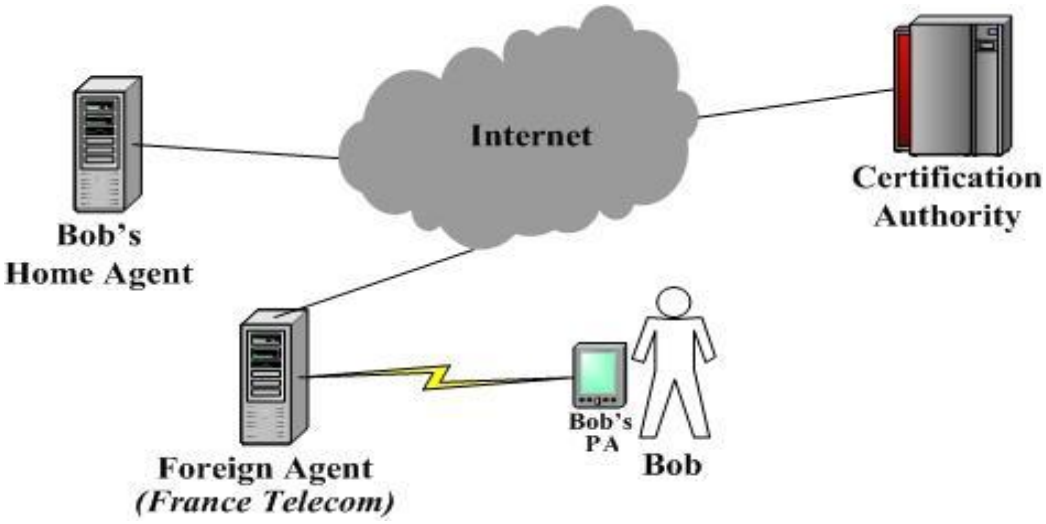

Figure 5: Architectural overview

The main design objectives for the proposed authentication protocol are the following:

1. The user's authentication is based on a secret password that is shared between the user and the Home Agent.

2. The protocol leads to the creation of a new public/private key pair that can be used for the authentication of the user. The private key will reside on the

device that the user is currently using and an authentication certificate signed by the Home Agent is provided for the new public key.

3. A trust relationship is established between the Home Agent and the Foreign Agent based on reciprocal authentication, and payment credentials for the user are transmitted by the Home Agent to the Foreign Agent.

4. The user may remain anonymous for the Foreign Agent.

We note that the use of a secret password for authentication has the advantage that it is easily implemented with a relatively short password (of a length of approximately 6 to 10 characters) that the user can remember. The authentication based on public key technology requires a much longer private key that must be stored in some device or card carried by the user. This makes it difficult for the mobile user to use any device that may be locally available. On the other hand, public key technology is essential for authentication to third parties and for the generation and verification of signatures. This is the reason for the second design objective. The main characteristic of this new authentication protocol is therefore to combine the use of a password with public key authentication. The new public/private key pair generated by the authentication protocol may be used for authentication to third parties, for instance for Bob's telephone conversation with Alice, and allows the user to generate verifiable signatures.

We note that the Radius protocol also uses password-based authentication, but it does not provide the creation of a public key certificate for authentication to third parties. Also, it assumes that the Network Access Server (NAC), which corresponds to the Foreign Agent in our architecture, is associated with a single Radius Server, while our protocol foresees interworking with a variety of different Home Agents throughout the world.

Objective (3) is important. In fact, no initial trust relationship is assumed between the user and the Foreign Agent. However, when the authentication protocol completes successfully, the Home Agent will have authenticated the Foreign Agent, and the resulting trust is indirectly available to the user. On the other hand, the user may remain completely anonymous to the Foreign Agent (as stated in Objective (4)). In fact, the payment credentials, in the form of a ticket T, are directly transmitted by the Home Agent. This ticket may also be used by the user to obtain services from other service provides within the foreign domain.

It is important to note that the protocol is structured in such a way that the user side of the protocol, also called Personal Agent (PA) is realized by software that runs on the device that the user happens to use within the foreign domain. This device may be his/her own PDA, but it may also be any device that happens to be available. The user has to trust the integrity of the software that represents the PA, but it does not have to trust the Foreign Agent.

4.2 Protocol description

4.2.1 Protocol Overview

The message exchanges of the authentication protocol are shown in Figure 6. One can identify the following three steps:

1. A locally broadcast preliminary message (number 1) provides information about the FA, e.g. the FA's IP address and its public key. This information allows the user to start the following authentication exchange.

2. The user (here Bob, or his Personal Agent, PA) sends an authentication request to the FA (message 2). The request is encrypted by a randomly generated session key Ks_1, which is protected by the FA's public key. The FA uses its private key to get the session key and the information about Bob's Home Agent, including its address.

3. The FA then forwards the authentication request to the HA after having removed the encryption with the session key (messages 3). Depending on the outcome of the authentication, the HA either replies a positive authentication response (messages 3.1 and 3.1.1) or a negative response (messages 3.2 and 3.2.1). These messages include information about the reasons for either success or failure. The Foreign Agent recognizes the message and also forwards the information to Bob. In the message from the FA to the user, this information is encrypted with a session key Ks_2, while the message exchanges between the FA and the HA pass through an encrypted connection.
4. The authentication between Bob and the FA is successfully achieved if the message 3.1.1 is received, otherwise the NACK value in the message 3.2.1 will indicate the reason of the refusal. The NACK value is determined by the HA; Bob can have confidence that the FA did not change the value in the message 3.2.1 by calculating $H(HV_1, N_2, NACK, pwd)$ and comparing it with the value of HV_2 received from the FA.

4.2.2 Detailed protocol description

The sequence of message exchanges of the protocol are shown in Figure 6, and the various information fields of the messages are indicated. We give in the following an explanation of the abbreviations used.

- ID_X denotes the unique identifier of the user X (for instance: *Bob@domain.net*)
- KU_X: public-key of user *X*
- KR_X: private-key of user *X*
- *Ks*: a session key (symmetric key)
- *K(M)* means *M* is encrypted using key *K*
- *N*: a nonce

- *SecCx*: a secure connection, e.g. realized through TLS
- *CSR*: Certificate Signing Request (defined in PKCS#10 standard)
- $CERT_X$: Certificate of user *X* (defined in X.509)
- *pwd*: password of the user

Some particular values are defined as follows:

- $HV_1 = H(ID_B, CSR_B, N_1, pwd)$ is calculated by Bob and can be used by the FA as a session identifier
- $HV'_1 = H(ID_B, CSR_B, N_1, pwd)$ is calculated by the HA and is compared with HV_1
- $HV_2 = H(HV'_1, N_2, NACK, pwd)$ in negative case and $H(HV'_1, N_2, ACK, pwd)$ in positive case
- Ks_1: session key, is randomly chosen by Bob and only used in message 2
- $Ks_2 = H(N_1,N_2, Ks_1)$ is a session key, in which N_2 is selected by the HA and used between PA and FA after the authentication.
- $Ks_3=H(N_1, N_3, pwd)$ is a session key selected by the HA and only known by the PA and HA.

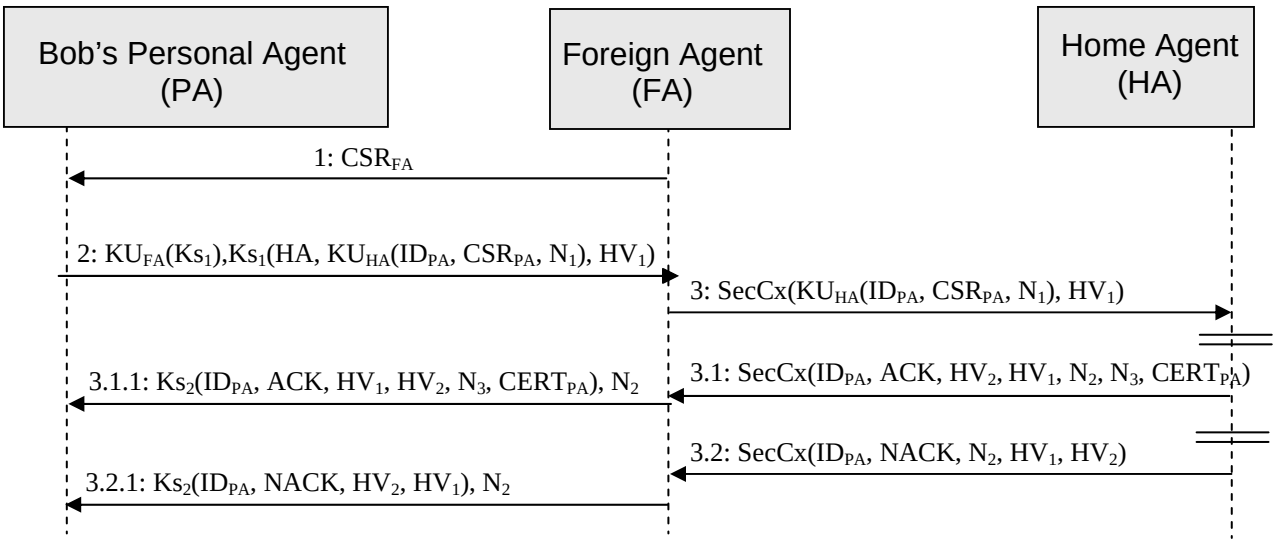

Figure 6: Message exchanges of the authentication protocol

Each of the messages shown in the figure is further explain in the following.

Message 1: Service agent advertisement: A broadcast message in the local domain informs Bob about the location and the digital certificate of the FA. This could be realized through existing protocols, such as the Dynamic Host Configuration Protocol (DHCP) or Jini.

Message 2: Authentication request: Bob's device executes the following steps in order to prepare this message:

(1) Bob generates a random number N_1 and a certificate-signing request (*CSR*) according to the PKCS#10 standard [16]. To perform the *CSR*, Bob generates a pair of public and private keys KU_{PA} and KR_{PA} on his terminal. KR_{PA} is stored on the terminal in a secure key store and is never sent over the network. The *CSR* includes KU_{PA} and a proof of possession of the private key. Bob encrypts his identity information along with N_1 and CSR_{PA}, using a public key of his HA KU_{HA} retrieved from an available standard authentication authority on his terminal.

(2) Bob generates a digest, called HV_1, of all the above information and the password *pwd*.

(3) Bob then selects a random session key Ks_1 that is used to encrypt all the above information HV_1 as well as the information of his HA. This will allow the FA to forward HV_1 to the HA. Ks_1 is then encrypted with FA's public key which was obtained from the FA's certificate included in Message 1.

Message 3: Forwarded Authentication request: The following steps relate to the forwarding of the authentication request to the HA:

(1) The FA receives Message 2, decrypts $KU_{FA}(Ks_1)$ using KR_{FA}, and then decrypts $Ks_1(ID_{HA}, KU_{HA}(ID_{PA}, CSR_{PA}, N_1), HV_1)$.

(2) With the help of ID_{HA}, the FA establishes a secure connection with Bob's HA and sends $KU_{HA}(ID_{PA}, CSR_{PA}, N_1), HV_1$.

Authentication by the HA: The HA receives Message 3, computes its own digest HV_1' and compares it with HV_1. If they are equal, authentication succeeds and an "ACK" (acknowledgment) is returned; otherwise a "NACK" (negative acknowledgement) is returned.

Message 3.1: Authentication reply - Ack: The HA performs the following steps:

(1) HA signs Bob's *CSR* and generates two random numbers N_2, N_3.

(2) Using the current secure connection established with the FA, the HA sends back a message including the answer of the authentication process (ACK), the hash value HV_1 sent by Bob that uniquely identifies the request, and security material for Bob $(ID_{PA}, N_2, N_3, HV_2, CERT_{PA})$.

Message 3.1.1: Forwarded Authentication reply - Ack: The FA performs the following steps:

(1) FA receives Message 3.1, calculates Ks_2, encrypts ID_{PA}, ACK, HV_2, HV_1, N_3 and $CERT_B$ with Ks_2 and transmits it together with the nonce N_2 in clear as Message 3.1.1.

(2) Bob receives this message. He computes $Ks_2=H(N_1, N_2, Ks_1)$ and decrypts Ks_2 $(ID_{PA}, ACK, HV_1, N_3, HV_2, CERT_{PA})$. He also computes $Ks_3=H(N_1, N_3, pwd)$.

Bob now shares a security association with FA based on the shared key Ks_2, and with HA based on the shared key Ks_3 . He can establish a security association with a new party or sign a document using $CERT_{PA}$ and KR_{PA}.

Message 3.2: Negative Authentication Reply - Nack: the HA performs the following steps:

(1) HA generates a random number N_2.

(2) HA prepares a "NACK" answer that includes a rejection reason (e.g. "revoked user" or "password expired") and the hash value HV_1 that identifies Bob's request. It then computes $HV_2=H(HV'_1, N_2, NACK, pwd)$.

(3) Using the current secure connection established with the FA, the HA sends back the answer including N_2, the answer of the authentication process (NACK), HV_1 sent by Bob, and HV_2, which serves as a proof of answer and ID_{PA}.

Message 3.2.1: Negative Authentication Reply Forwarded - Nack: the FA performs the following steps:

(1) FA receives Message 3.2. It computes $Ks_2=H(N_1, N_2, pwd)$ *and* encrypts ID_B, ACK, HV_1 and $HV2$ with Ks_2 and transmits it together with N_2 in clear text as Message 3.2.1.

(2) Bob receives this message, calculates Ks_2 and decrypts Ks_2 $(ID_B, NACK, HV_1, HV_2, N_2)$ and then computes $HV'_2=H(HV_1, N_2, NACK, pwd)$ to check that this authentication answer actually comes from the HA.

Bob now knows that his request has been rejected and he has received the reason.

4.2.3 Discussion

This protocol was inspired by a similar protocol described in [17]; however, it contains the following improvements compared to the protocol of [17]:

- Minimal usage of public key technology at the PA side to satisfy the limitation of computing capability and battery power of mobile devices. Through the authentication protocol, public key encryption is used only twice in Message 2. After the initial authentication, there is a session key shared between Bob' PA

and the FA (Ks_2), as well as between the PA and the HA (Ks_3). Further negotiation will be based on these session keys using symmetric key operation.

- A hash value is included to prevent that a misbehaving third party may introduce itself between two nodes, such as HV_1, HV_1', HV_2;

- The mobile user relies on his/her HA to authenticate the FA. Since Bob does not have a root certificate, his PA could not verify the FA' certificate sent in Message 1. Instead, the PA will send an encrypted request to the FA which should then be forwarded to the HA. If the FA could not be authenticated by the HA, the secure connection between these two parties could not be established. Without the secure connection, the request would not be sent. Therefore the PA would time-out after waiting for a reply message from the FA. Such a time-out indicates that the FA may have failed to get authentication.

- Anonymity option: The user's anonymity can be guaranteed by hiding the user information from the FA and using tickets provided by the FA to gain access to services within the domain of the FA. The anonymity option implies the following modifications to the protocol. In the case of a positive acknowledgement, Message 3.1 now becomes $SecCx\ (ACK, HV_1, N_2, N_3, Ks_3\ (CERT_{PA}))$ and Message 3.1.1 becomes $Ks_2\ (ACK, HV_1, N_3, T, Ks_3\ (CERT_{PA})), N_2$. We note that ID_{PA} is removed from these two messages. Instead of sending $CERT_{PA}$ in clear, it is now encrypted with a shared session key Ks_3 which is only known by the PA and the HA. A ticket T is created by the FA and sent to the PA to be used for local service access. The service server would validate the ticket and provide service upon validation regardless who presents the ticket. In the case of a negative reply, Message 3.2 becomes $SecCx\ (NACK, N_2, HV_1, HV_2)$, and Message 3.2.1 becomes $Ks_2\ (NACK, HV_2, HV_1), N_2$. As in the previous case, ID_{PA} is removed from these two messages.

4.3 Verification of the authentication requirements

In case of positive authentication, all six possible cross-authentications between the three parties take place:

- Bob authenticates the FA: Bob trusts the HA to authenticate FA. After decrypting Message 3.1.1 or 3.2.1, he knows that FA received HV_2 from HA because that value could only be computed knowing the password. That means HA authenticated FA previously when establishing the secure connection *SecCx*.

- Bob authenticates the HA: After decrypting Message 3.1.1 or 3.2.1, Bob knows that the HA computed HV_2, because the HA is the only agent that knows the password.

- The FA authenticates the HA: The FA checks the certificate of the HA before sending Message 3 when establishing the secure connection.

- The FA authenticates Bob: After receiving Message 3.1 or 3.2, the FA knows the authentication answer of the HA and trusts the authentication done by the HA. In addition Bob can decrypt Message 3.1.1 if and only if he recovers Ks_2 from N_2. If he does so and uses Ks_2 to communicate later with the FA, the latter knows he shares some information with the HA.

- The HA authenticates Bob: After receiving Message 3, the HA compares HV_1 with HV_1' to check Bob's password.

- The HA authenticates the FA: The HA checks the certificate of the FA when FA tries to establish a secure connection.

In the case of a negative response, Bob is sure that the answer was prepared by HA because of the following reasoning. After receiving Message 3.2.1, Bob checks that the negative answer was made by the HA by computing HV_2. The latter value could have been computed only by HA and is related to Bob's initial request because of the presence of HV_1. This check is useful to verify that no third party is misbehaving in the middle between Bob and the HA. The presence of ACK/NACK in the HV_2 computation is useful to check that the middle party did not change the accept/refusal reason.

4.4 Consideration of typical security attacks

We discuss in the following a few typical security attacks and how the protocol copes with them.

- Spoofing attack of a malicious user: A malicious user, says Eve, may try to usurp Bob's identity. Authentication information is included in HV_1 sent by Bob to the FA in Message 2. Since Eve does not know Bob's password, the HA while calculating HV_1' finds a different value and does not authenticate Eve as Bob. In message 3.2, the HA sends the authentication result to the FA so that the FA knows that Bob (actually Eve) is not authenticated.

- Spoofing attack of servers (the FA, the HA) is denied by the systematic use of digital certificates. Bob relies on the HA to authenticate the FA (see Section 4.2).

- Replay attacks of an authentication request are impossible owing to the nonce. If an attacker tries to replay messages or a rogue FA tries to replay messages, this will be detected by the HA that keeps all successful login nonces for a given time period (e.g. a few days). Even if the attack is not detected by the HA, a malicious user replaying the request message could not decrypt Message 3.1.1 because he/she would require the knowledge of the key K_{S2} which can only be calculated with the knowledge of K_{S1} which is generated by Bob.

- Denial of service (*DoS*) attacks would consists of sending rogue authentication request that would consume both bandwidth and processing time at the FA and the HA. Such an attack can be realized more easily by simultaneously mass replay attacks.

It would make the HA compute all the key material for each request. Denial of service is a general and open issue for any service on the Internet.

4.5 Comments on the detailed design of the protocol

The description of the authentication protocol given in Section 4.2 represents, in some sense, an "abstract protocol", that is, only the logical meaning of the message parameters is described, while the coding of these parameters is left undefined. It is important to note, however, that a complete protocol specification (describing all requirements for an implementation) should also include the definition of the parameter encoding and the description of the cryptographic functions that are used. It is clear that the choice of these cryptographic functions has a strong impact on the level of security that can be obtained by the given "abstract protocol". In the following, we give some comments on the possible choices.

The cryptographic hash function used in the authentication protocol may be MD5, SHA-1 or possibly SHA-2 with a key size of 128 bits, 160 bits and up to 512 bits, respectively. These algorithms are known to be secure against typical attacks (e.g. birthday paradox attack).

Private-key algorithms should be chosen such that the length of the key can be adapted to the computational power of the mobile terminal. Triple-DES, Blowfish and AES (Advanced Encryption Standard) are such algorithms. Elliptic Curve Cryptography (ECC) should preferably be used for public-key encryption, rather than RSA, to make use of its shorter key length at equal security level.

Secure connections could be set up in several ways since both FA and HA own a digital certificate. TLS (Transport Layer Security), IPsec (IP security), IKE (Internet Key Exchange) or any secure link establishment protocol could be used between the two agents.

The nonces are random numbers used for multiple purposes: N_1 acts as a salt for HV_1 to prevent message replay and attacks on the hashing algorithm. It also makes HV_1 a unique identifier of Bob's request. N_2 links Ks_2 and Bob's session key K_{S1} and is used as a challenge for Bob. Only the user who has the key K_{S1} can decrypt Message 3.1.1 to get the certificate, or Message 3.2.1 to understand the reason of failure. N_3 links Ks_3 and Bob's password and is used by the HA to securely transmit the session key Ks_3 to Bob without disclosing it to the FA.

Note that the protocol satisfies all the requirements when executed on a user-owned mobile device. However, when executed on any device that may be locally available to the mobile user, there are two common problems (which are not related to this particular protocol): (a) The user has to trust the integrity of the software (as explained in Section 3.6), and (b) the private key generated by the protocol may be left on the device and used by other people, if the user does not properly terminate the application.

5. Concluding discussion

We gave an introduction to the authentication requirements for electronic commerce by identifying the commerce partners and required trust relationships, and by describing the security requirements including authentication, access rights, payment credentials, anonymity (in certain cases). We also reviewed existing paradigms for authentication and corresponding protocols, and discussed their suitability for electronic commerce applications. We considered in particular the requirements stemming from user mobility, which include the security implications of using unknown ad hoc devices that are locally available and the fact that the user may need to be authenticated by a foreign organization that provides network access facilities and other services within the foreign domain where the user temporarily resides.

We then proposed a secure authentication protocol for mobile users that combines the ease of password-based authentication with the power of public key technology, and can be executed on an ad hoc devices. It provides authentication support (i) for electronic commerce transactions, (ii) for obtaining the necessary transmission resources from the local Internet service provider (ISP) and (iii) for authentication to arbitrary third parties. We believe that this authentication protocol contains a number of interesting features that make it suitable as an alternative to the other authentication protocols that are currently in use. In fact, this authentication protocol is not limited to electronic commerce applications, but could be used as well for other distributed applications, such as IP telephony and multimedia teleconferencing.

We note that in the context of electronic commerce and other applications, there is not only the need for authentication of users and services, but also a need for obtaining other kinds of references, such as payment credentials, age certifications, or competence certificates. Such references may be provided in the form of signed certificates, similar to authentication certificates, but containing different information attributes. It is also important to allow the user of commerce applications to remain anonymous; for this case one has to foresee certificates that do not contain the name of the user, nor other identifying information. An example is a payment credential for an anonymous user who is only identified to the commerce server by a random number without any other significance.

We finally note that the use of ad hoc devices that may be available in the local environment of the mobile user poses certain security threads, since it is very difficult to ensure the security of the software that runs on such a computer. For the present purposes, we assume that this risk can be kept sufficiently small by using certified software down-loaded over the Internet. However, future research may identify methods for closing the remaining loop-holes.

Reference

[1] W.Simpson, PPP *Challenge Handshake Authentication Protocol* (CHAP), Request for Comments, RFC 1994, The Internet Engineering Task Force, August 1996.

[2] B. AbdelAziz, *Using IKE and Radius with MobInTel*, Technical report, University of Ottawa, Ottawa, Canada, December 2000.
[3] J. G. Steiner, B. Clifford Neuman, and J.I. Schiller. *Kerberos: An Authentication Service for Open Network Systems*. *In* Proceedings of the Winter 1988 Usenix Conference. *February, 1988*
[4] John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so, *The Evolution of the Kerberos Authentication System*. *In* Distributed Open Systems, *pag*es 78-94. IEEE Computer Society Press, 1994.
[5] *SSL/TLS, RFC,* ftp://ftp.isi.edu/in-notes/rfc2246.txt
[6] *Draft Security Assertion Markup Language Specification* http://www.xmltrustcenter.org/saml/docs/draft-sstc-core-12-final.pdf
[7] *XKML Specification* http://www.verisign.com/resources/gd/xml/xkms/xkmsv1-1.pdf
[8] *XMLPay Specification* http://www.verisign.com/resources/gd/xml/xmlpay/xmlpay.pdf
[9] *SSH Overview* http://www.ietf.org/ids.by.wg/secsh.html
[10] *Introduction of Diffie-Hellman* http://www.rsasecurity.com/rsalabs/faq/3-6-1.html
[11] *IETF draft of SHTTP* iew http://www.ietf.org/proceedings/99jul/I-D/draft-ietf-wts-shttp-06.txt
[12] S. Glass, T. Hiller, S. Jacobs,and C. Perkins, *Mobile IP Authentication, Authorization, and Accounting Requirements*, Request for Comments, RFC 2977, The Internet Engineering Task Force, October 2000
[13] *Secure IP Overview* http://www.ietf.org/html.charters/ipsec-charter.html
[14] K. El-Khatib, N. Hadibi, and G.v.Bochmann, *Support for Personal and Service Mobility in Ubiquitous Computing Environments*, EuroPar 2003
[15] Thompson, *Reflections on Trusting Trust* (Comm. ACM, Aug.1984).
[16] RSA Laboratories, *PKCS #10: Certification Request Syntax Standard*, Technical Report, November 1993
[17] I. Dupré-la-Tour, G. v. Bochmann and J. Y. Chouinard, *A secure authentication infrastructure for mobile communication services over the Internet*, in Communications and Multimedia Security Issues of the New Century (Proc. IFIP Working Conf. CMS'01, Darmstadt), R. Steinmetz et al. (Eds.), Kluwer Academic Publ. 2001, pp. 405 – 416