Certificate Management in Mobile Devices

Vesa Hämetvaara

University of Tampere
Department of Computer
and Information Sciences
Master's Thesis
14.5.2002

University of Tampere
Department of Computer and Information Sciences
Vesa Hämetvaara: Certificate Management in Mobile Devices
Master's Thesis, 58 pages
May 2002

Abstract

Digital certificates are one of the building blocks in delivering security environment for wireless devices. This thesis describes how certificates are obtained, managed and used in the mobile environment. The main focus of the thesis is the security technology suited for mobile phones. However, the technology introduced is not targeted only for mobile phones, since these devices are currently in the middle of an evolution from phones to personal multipurpose devices. In the near future the PDA and the phone will converge into one.

ACKNOWLEDGEMENTS

I thank for my lovely wife for being so supportive and encouraging during my thesis work. I thank for support my employer Nokia, and especially Tapio Välimäki for the title, Heikki Lahtinen and Timo Heikkinen for advising, Antti Huupponen for valuable comments, and Ari Laaja and others for support. Also thanks to Erkki Mäkinen and Juhani Paavilainen from the Department of Computer and Information Sciences. One more person, who had significant influence on my thesis work, is worth mentioning: our six-month-old daughter, Anni.

CONTENTS

FIGURES

1 INTRODUCTION

The purpose of this thesis work is to study and evaluate existing and ongoing standards and draw a big picture over the area of certificate based security in mobile devices. It is very easy to find comprehensive books and papers about the information security in Internet and about the usage of certificates in that context. But applying certificates in mobile environment is not very well covered in scientific texts. I have been fortunate for being able to get my hands on studies from some of the major universities and private research centres that handle this topic. Unfortunately these studies are not public so it has not been possible to use them as references.

In the eve of much anticipated mobile revolution, it is important to establish an open and vendor independent platform for mobile Internet services. The standardising work for global mobile Internet has already begun. Mobile device manufacturers have band together in order to specify these standards. New high speed, packet switched wireless network technologies together with more capable client devices enable more feature rich service applications: shopping, banking and new "to be invented" services.

Without proper security, mobile Internet services cannot take-off. Certificates have an essential role in the technology enabling mobile commerce. With the help of certificates, a point-to-point security will be possible ensuring secure transactions between parties.

WAP Forum specifies the infrastructure for delivering certificates over the air connections (WPKI, WTLS [WTLS00], etc). Whereas MET [METWP01] specifies how to use certificates in practise (CUE, PTD requirements, etc.).

This thesis covers a wide area of managing digital certificates in mobile environment. In chapter two we will go through the basic concept of information security and cryptography. In chapter three the digital certificates are introduced in more detail. In chapters four and five we get familiar with the wireless world. Common characteristics of the wireless devices and couple of important standards are introduced. Chapter six covers the major players in the field. Chapter seven summarises the results and present some scenarios for the future.

2 INFORMATION SECURITY AND PAIN

A PAIN model describes a conceptual framework for a theory of the information security. The framework defines the requirements for implementing secure applications for electronic information exchange. Tools for applying security are based on cryptographic algorithms for encryption/decryption, digital signatures and protocols for key management and data exchange.

To be able to implement inter-operative systems these algorithms must be standardised. Two cryptographic systems exploited in the real life standards are symmetric key and public key cryptosystems. The main difference between those two is in a way the cryptographic keys are distributed. While in the symmetric key system a common encryption and decryption key is distributed secretly, in the public key systems the keys for these operations are separate and does not require a secure key exchange. In fact, the private part of the key pair of the public key cryptosystem never have to leave the originator's domain.

Digital signature offer means for ensuring that the message is intact. By signing the message the receiver can verify who the sender really is and the signer cannot deny sending the message.

2.1 PAIN Model

The PAIN model in information security means that the security can be divided into four categories [MEN96, 4][TAN96, 578]:

- Privacy,
- Authentication,
- Integrity and
- Non-repudiation.

Privacy (Also secrecy or confidentiality.) simply means that nobody, outside a communicating group, is able to read the data delivered between other parties. In electronic information exchange the privacy is usually achieved by means of cryptography. The sender encrypts all messages into such format that without a decryption key it is impossible to open the message.

Authentication is needed for insuring that all parties of the communication are the ones they are claiming to be, that the data really is originated from certain party or for controlling access to services (authorisation). Authentication of parties can be achieved for example by traditional password mechanism or by using other shared secret like finger print. Certificates provide somewhat stronger shield against password guessing or similar attacks. In this case, a certificate is like a passport that acts as a proof of identity. In case of client-server communication, it is feasible to separate authentication of the server to the user and authentication of the users to the server. The detailed mechanism for entity authentication will be described later.

Sometimes the term authorisation is separated from the term authentication as one of the five elements of information security. Consider for example a big corporate information system. All the employees may have an access to the system, but not all will be granted rights to change the information. All employees can authenticate to the system, but they still have different levels of authorisation. Because the means for ensuring both authentication and authorisation can be implemented by certificates, I made a decision to combine both terms into one category.

Data integrity means that there must be a way to ensure that nobody has been able to tamper with the data received or sent by other party. To assure data integrity, one must have the ability to detect data manipulation by unauthorised parties. Data manipulation includes such things as insertion, deletion, and substitution [MEN96, 4]. Integrity can be achieved for example by digitally signing the content to be sent. When the receiver verifies the signature he can be sure that nobody has been able to alter the message. In addition to digital signature there is also other means for ensuring data integrity. For example by calculating and verifying checksums or hash values.

Non-repudiation can be defined as an attribute of a communication, which protects against a party to the communication denying that it occurred [WAR97, 98].When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary. For example, one entity may authorise the purchase of property by another entity and later deny such authorisation was granted. A procedure involving a trusted third party is needed to resolve the dispute [MEN96, 4].

2.2 Attacks Against PAIN

An attacker (adversary) is an entity that uses unauthorised means to threaten communication. Passive and active attacks are the two main categories. A passive attacker threatens privacy by listening and analysing for example network traffic in order to be able to conceive pieces of confidential information. An active attacker uses active means to disturb, alter, destroy or to otherwise produce harm against authentication and integrity of information systems.

Some of the most common threads to information systems are [WAR97, 95]:

- System penetration where unauthorised person gains access to a system. An attacker may use for example brute force technique to guess passwords or exploit a security weakness in the system.

- Authorisation violation where authorised person missuses the system. For example an user account is used to gain administrator rights.

- Planting where an intruder leaves behind a planted capability to perpetrate future attacks. For example a piece of software that appears to be legitimate (Trojan) leaves a back door to the system for the attacker.

- Communication monitoring (eavesdropping) that can be used for example to listen passwords and other pieces of confidential information. Even if the communication is encrypted a clever attacker may be able to deduce important hints about the data merely by for example collecting statistical information.

- Communication tampering where an attacker modifies or disturbs communication for example by changing data records. Sometimes an attacker may cheats others by claiming to be some other legitimate party of communication and persuading hand over confidential information.

- Denial of Service where an attacker for example floods a server system with bogus requests so that not even authorised users can get service.

- Repudiation where a party of communication denies that the communication ever occurred.

2.3 Cryptographic Algorithms

A cryptosystem defines a pair of data transformations called encryption and decryption. Encryption is a process, which is applied to data, known as plaintext. Encryption transforms the plaintext data into ciphertext. The result of applying decryption transformation to the ciphertext is again plaintext [WAR97, 101].

The transformation process for encryption and decryption needs a key, which defines the result. Usually the algorithm remains the same independent from the session. Only the key changes between sessions.

Cryptographic algorithms can be divided into two main categories: symmetric key algorithms and public key algorithms. With symmetric key algorithms, the same key is used for both encryption and decryption. For the public key algorithms, two keys are needed. The cryptoalgorithms themselves are not very interesting in this study's point of view since the techniques used later are relatively independent from the algorithms. For comprehensive introduction of cryptography refer for example to [MEN96].

2.3.1 Symmetric Key Cryptography

In symmetric key algorithms, the same key is used for encryption and decryption has to be shared with two communication parties. No one outside the communicating parties can gain access to the key. The symmetric key encryption ensures the confidentiality and the authentication of information. The confidentiality, of course, is originated from the encryption of data. The authentication is caused by the fact that only trusted parties have knowledge of the secret key.

A symmetric cryptosystem operates well as either a block ciphering or a stream ciphering. In a block cipher, the encryption functions operate on a fixed-size block of plaintext and generate a fixed-size block of ciphertext with the same length. The decryption function operates to the opposite direction resulting plaintext from fixed-size ciphertext block. Stream cipher can operate over a plaintext message or stream of data of arbitrary size, generating ciphertext of the same size; a stream cipher typically processes the data as a sequence of characters, where a character can be considered to be one bit or a small number of bits [MEN96, 103].

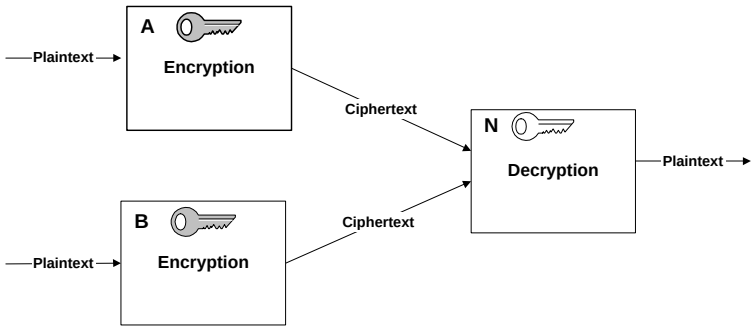


Figure 2.1 Public key used for encryption.

The most commonly used and known symmetric key cryptographic algorithm is called DES. DES is a symmetric key cipher, which operates on 64-bit blocks of data and employs a 56-bit key [WAR97, 103]. Because of the relatively short key length various extending algorithms have been developed to meet today's challenges set by the ever-increasing processing power. Consult for example [MEN96] for more information about DES and its variations, or other algorithms like IDEA and Blowfish.

2.3.2 Public Key Cryptography

As opposed to symmetric key algorithms, the public key algorithms use two keys: one for encryption and another for decryption. The keys are called private and public, respectively. Only the owner of the key pair knows the private key. That is why it is sometimes called a secret key. Inherited from the algorithm used for generating the key pair, there is no way to deduce the private key from the public key. Hence, the public key can be delivered to anyone interested across an unsecured medium, for example Internet.

The public key cryptography can be used in two modes of operation. With the first mode, the public key is used for encryption. A receiver, that is, the owner of the private key can decrypt the message with its private key. Nobody else is able to decrypt the message, but anyone who holds the public key can use it for encryption. Consider Figure 2.1 where N receives a message encrypted by A. Even when both A and B have access to N's public key B is not able to encrypt the message sent by A. Privacy is guaranteed in this case. However, there is no guarantees for integrity or authentication. The message could have been sent by anyone having an access to N's public key. N cannot

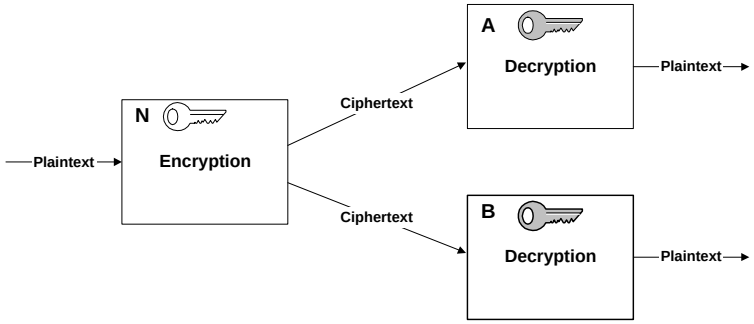


Figure 2.2 Public key used for decryption

say for sure who was the originator of the message and that the contents is intact.

In the second mode the private key is used for encryption and public key for decryption. Consider Figure 2.2 where N sends an encrypted message to A who is can read the it after decryption using N's public key.

By using the private key as an encryption key, public key cryptography can be used for data origin authentication and for ensuring the integrity of the message [MEN96, 109]. Some public key cryptosystems provide only authentication mode, but not the encryption mode. These are called irreversible public key cryptosystems. Systems that provide both authentication and data integrity are called reversible public key cryptosystems.

2.3.3 Digital signatures

The digital signature provides means for ensuring integrity and non-repudiation of electronic messages. A digital signature is a number dependent on some secret known only to the signer, and, additionally, on the content of the message being signed. If the message is changed the signature calculated again would not be the same as the original. In theory it may be possible form two messages that produce the same signature but it is highly improbable that the other message makes any sense to the receiving party.

Signature must be verifiable: if a dispute arises as to whether a party signed a document, an unbiased third party should be able to resolve the matter equitably, without requiring access to the signer's secret private key [MEN96, 425]. In public key systems the signature can be verified by using the public key corresponding to the secret key that was used to sign the message. Power-

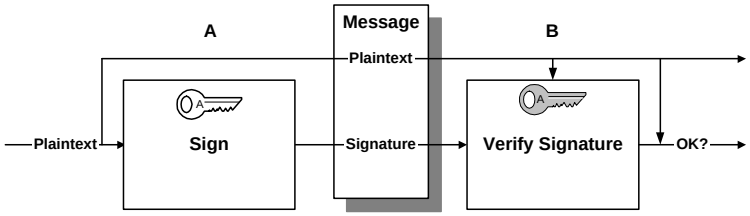


Figure 2.3 Digital signatures

ful digital signature capabilities, which do not require that the verification key be kept in secret from the recipient, can be built using public key technology [WAR97, 112].

Figure 2.3 illustrates the overall process of signing and verifying a message. The originator A signs the plaintext with his private key and attaches the signature into the message. The recipient B then verifies the message with the public key of A.

RSA Digital Signature

In one of the standard digital signature mechanisms used, RSA, the encrypted version of the message is sent attached to a copy of the plaintext message. The verifier must decrypt the signature with the originators public key. If the plaintext and the decrypted signature are the same, the message is intact and originated from the sender. Figure 2.3 illustrates the process of the simplified RSA digital signature scheme.

The above method of signing messages has one big problem. The signature doubles the size of the message. With long messages, the signature is obviously a waste of recourses. The answer for the problem is to use a encrypted hash value (digest) as a signature [WAR97, 114]. The hash value is calculated from the plain text messages using a hash function, for example DSA, MD5 or SHA-1. This digest is always fixed length and usually much shorter than the message itself. Typically digests are from 56 to 128 bytes long. The digest is the encrypted with the senders private key and attached as a signature with the plain text message. The receiver can be sure that the message is intact if the decrypted signature and a digest the receiver calculated from the message are the same.

2.4 Key Management

Key management includes ensuring that key values generated have the necessary properties, making keys known in advance to the particular systems that need them, and ensuring that keys are protected as necessary against disclosure and/or substitution [WAR97, 117].

When the literature deals with key management issues at least the following operations are usually covered:

Generation Generation of a key involves an algorithm and satisfactory random data. The key's tolerance against attackers depends on the quality of the key generation operation. None of the key generation algorithms has been proved unbreakable, but choosing a publicly known strong algorithm ensures a satisfactory security for several years.

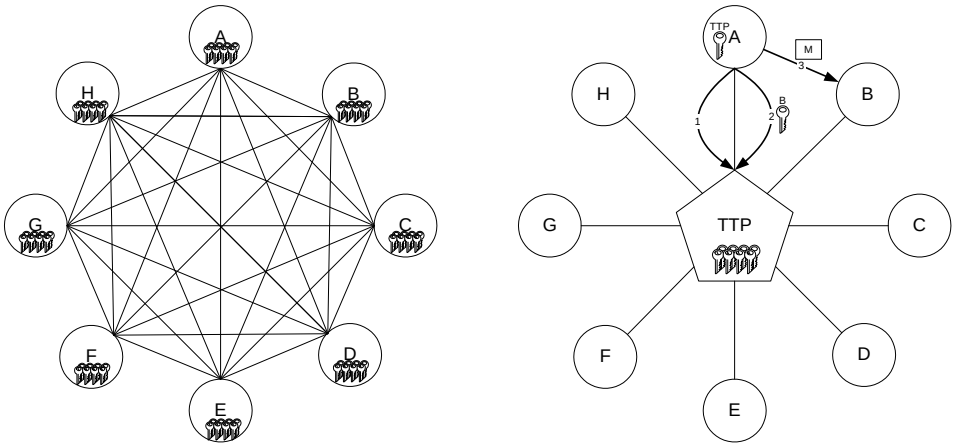
Registration Registration involves linking a generated key with its particular use.

Distribution There is a fundamental difference on distributing the keys depending on whether the cryptosystem used is based on symmetric key or public key technique. In symmetric key cryptosystems the secret must be set in danger when the key is distributed to other parties. In the public key cryptosystems no secret data have to be shared with other parties. The difference between the two methods are described in chapter 2.4.1.

Replacement Normally the lifetime of a key is limited and after the validity period the key must be replaced with a new key.

Revocation Key revocation may be necessary in exceptional circumstances. Reasons for key revocation include the decommissioning of a system with which a key was associated, suspicion that a particular key may have been compromised, or changes in the purpose for which the key was used [WAR97, 119].

In addition to the operations above, the literature often handles issues like registration of keys, backup/recovery, key escrow and termination.



(a) n^2 keys (b) n keys

Figure 2.4 Key management

2.4.1 Distribution

Symmetric Key Systems

Because the same key is used for both encryption and decryption in symmetric key cryptography, the consequence is that the key has to be delivered secretly to all parties to communication. If the aim is to be able to communicate with several groups having different keys the key distribution and management can become a burden. If the communication consists of n parties, the overall number of keys in the system is x^2. Figure 2.4 shows (a) almost all solutions proposed incorporate a Trusted Third Party (TTP), who is responsible for maintaining a repository of keys.

In Figure 2.4 (b) A wants to send a message to B. First, A obtains the secret key of B from the TTP using TTP's secret key, which has been delivered to all parties beforehand. A sends the message to B. The overall number of keys is $n + 1$. Hence, the workload of managing keys is reduced significantly.

Public Key Systems

In public key systems only the public part of the key-pair have to be delivered to other parties. The private key is always kept in hand of the owner and can be stored into a safe place, such as a smart card. On the other hand, delivery

of the public key does not require being secured, because there is no way of misusing it. Public keys can even be distributed through a public server like one of those dozens set up for PGP keys in Internet.

Means of public key delivery is not limited to just using TTP. The following five different public key distribution models are introduced [MEN96, 555]:

1. Point-to-point delivery over trusted channel. For example delivery physically from hand to hand. This kind of delivery is suitable if a high level security is required since the authentication and integrity can be guaranteed personally.

2. Direct access to a trusted public file. An example of this kind of delivery could be a LDAP directory specified in [RFC 2559] where the keys are stored and the clients can make direct requests for keys. Authentication and integrity of the keys can be achieved by for example certification chain described in 3.1.

3. Use of an on-line trusted server. This method allows a similar access to the public keys as the previous method but the requests are not directed to the files it self. Instead, the requests may consists of a name for the subject or some other attribute. The server processes the requests according to its internal rules and responds with a key information..

4. Use of an off-line server and certificates. In this method the TTP plays more active role that in previous two. In this case the TTP is often called Certification Authority (CA) who's role is explained in Chapter 3. An example for this kind method is described in 5.5.2.

5. Use of systems implicitly guaranteeing authentication of public parameters. In some systems the authentication of the users is implemented by using for example electronic identity cards or finger prints. In this kind of systems the public keys can be expected to be authenticated already as a service of the system.

Even if the confidentiality when distributing the public key is not required, the authentication is. Otherwise, a risk of an impostor falsely claiming to be someone else increases. In open networks, like Internet, the authentication

(and also integrity) of public key is often maintained using certificates, which are essentially public keys that are digitally signed by a trusted third party. Certificates are covered in more detail in Chapter 3.

3 DIGITAL CERTIFICATES

A digital certificate is an electronic counterpart for passport. It is an assurance of an identity of the subject and issued by a trusted third party. Oversimplifying, in public key infrastructures, certificates consist of the subject's public key and a digital signature of a trusted third party, who is responsible for reasonably strict checking of the subject's real identity. This TTP is often called Certification Authority, CA. A reliable signature assures the integrity of the certificate.

For verification of the signature included in the certificate a public key of the CA is needed. Fortunately, the CA's public key can be delivered also in a certificate. The only problem is that how to make sure we have an untouched CA certificate. The answer to the problem is a special kind of certificate called root certificate. Root certificates are guaranteed to be trusted and usually they must be distributed via off-line route. For example, in WWW browsers, root certificates are included already in the software package and the user cannot alter them.

In addition to further introduced X.509 and WTLS certificates there are other commonly used certificate types as well. For example, PGP (Pretty Good Privacy) certificates, which are well suited for messaging applications like email. However these certificate types are not used to offer seamless, protocol level security. PGP certificates are distributed from a friend to another or collecting certificates into public on-line repositories forming networks of trusted parties. The "Net of Trust" formed by PGP communities cannot really offer a common framework for generic security services because the PGP is too specialised in one application, email. The same applies to other certificate based systems as well. They are too specialised, proprietary or otherwise not widely used. That is why they are not interesting in the context of this work.

3.1 Certificate Chain

The recursive paradigm of obtaining and verifying certificates leads us to a general model, called the certificate chain. Figure 3.1 illustrates the situation. The topmost certificate, the root CA certificate, must be self-signed. Under the root certificates there are other CA certificates that are signed by the root. The user certificates are at the end of the branches. The verification hierarchy under

one root CA certification system is called certification tree. Several certification trees can be cross certified by other root CAs so that they form a forest of hierarchies. In real life there might be number of cross certified root CA's used even inside of one organisation. In commercial products there can be dozens of root certificates preinstalled from different commercial certification authorities. These root certificates are usually not the ones from the top of the tree, but some certificates under the root.

Figure 3.1 illustrates a simplified representation of an ideal case where there is only one trusted root CA certificate in the system and the direction of the certification always goes from top to bottom. If the subject of certificate A needs to verify that the certificate C's public key can be trusted, the verification path will go through six nodes. It is mandatory that A have access to all certificates along the path.

In addition to root CA certificates also non-root CA certificates can be cross certified as illustrated in Figure 3.1 where CA certificate "Local CA 1.2" is cross certified by "Sub CA 3". If the B's subject needs to verify C's public key the certification path without the "shortcut" would have to go through seven nodes but with the cross certification only five nodes are involved.

From the software point of view, verifying any arbitrary certificate using the chain looks easy. However, in reality, it is the most difficult part of the whole security system based on certificates. Obtaining and verifying missing nodes between the root and the leafs is not a trivial task, as will be proved later in Section 5.2.6.

3.2 Employment of Certificates

Certificates, public key certificates especially, provide a fundamental building block for secured electronic communication. Underlying cryptographic algorithms can easily be changed to meet respective needs of parties. The encryption algorithm or the digest algorithm used in any particular certification type is not fixed and often the certificate itself can contain information of the used algorithms.

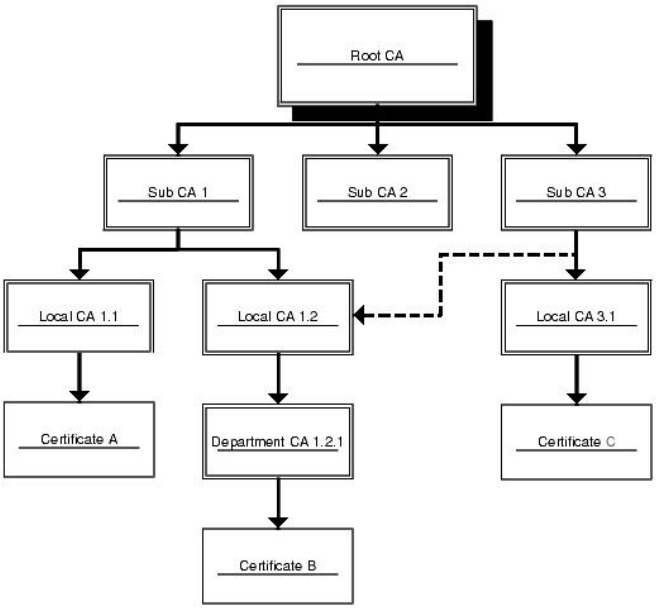


Figure 3.1 Certificate Chain

3.2.1 Encryption

In theory, one can use a public key included in a certificate to encrypt messages, but in practise, public key encryption algorithms are too slow for this purpose. Thus, often the public key in certificates are used only for establishing a onetime session key, which is then used for encryption of messages. The session key must be suitable for symmetric ciphering algorithm that provides better performance. This kind of method is called Hybrid Key Transport Protocol or Digital Envelope.

Diffie-Hellman Key Agreement

Diffie-Hellman (DH) [RFC 2875] is maybe the most famous key agreement protocol for establishing a shared secret key between two parties. DH be used to generate a session key when establishing an encrypted channel between two communicating parties. The basic version of DH key agreement protocol is presented as follows [MEN96, 516]:

A and B each send the other one message over an open channel. Shared secret K known to both parties A and B. An appropriate prime p and generator of $\mathbb{Z}_p^*(2 \leq \alpha \leq p-2)$ are selected and published. Protocol messages:

$$A \rightarrow B: \quad \alpha^x \, mod \, p \quad (1)$$
$$A \leftarrow B \quad \alpha^y \, mod \, p \quad (2)$$

Perform the following steps each time a shared key is required.

1. A chooses a random secret x, $1 \leq x \leq p-2$, and sends B message (1).

2. B chooses a random secret y, $1 \leq y \leq p-2$, and sends A message (2).

3. B receives α^x and computes the shared key as $K = (\alpha^x)^y mod \, p$.

4. A receives α^y and computes the shared key as $K = (\alpha^y)^x mod \, p$.

Diffie-Hellman key agreement has been criticised as being vulnerable to "Man-in-the-middle" attack where an attacker acts between parties so that A and B both believe being in direct contact with each other but in reality the attacker is able to eavesdrop the communication.

3.2.2 Entity Authentication

Public key certificates are widely used for authenticating parties to communication. Authentication based on public key cryptography has an advantage over many other authentication schemes because no secret information has to be shared by the entities involved in the exchange. There are two main categories of the public key authentication; digital signature based and non-signature based [MEN96, 507].

One method of certificate based authentication is based on usage of digital signature combined with a public key certificate. I this method a user (claimant) attempting to authenticate itself must use a private key to digitally sign a random number challenge issued by the verifying entity. This random number is a time variant parameter, which is unique to the authentication exchange. If the verifier can successfully verify the signed response using the claimant's using the claimant's public key, then the claimant has been successfully authenticated [USG97].

Needham-Schroeder Public Key Protocol

Needham-Schoeroeder public key protocol represents one of the key establishing protocol suited for authentication of peers that have access to each others public key certificates. The protocol is described as follows [MEN96, 508]:

$P_x(Y)$ denotes public key encryption (for example, RSA) of data Y using party X's public key; $P_x(Y_1, Y_2)$ denotes the encryption of the concatenation of Y_1 and Y_2 . k_1 , k_2 are secret symmetric session keys chosen by A, B, respectively.

Assume A, B possess each other's authentic public key. (That is, both of them have access to other parties verified public key certificate.)

$$\begin{aligned} A \rightarrow B: \quad & P_B(k_1, A) \quad (1) \\ A \rightarrow B \quad & P_A(k_1, k_2) \quad (2) \\ A \rightarrow B \quad & P_B(k_2) \quad (3) \end{aligned}$$

1. A sends B message (1).

2. B recovers k_1 upon receiving message (1), and returns to A message (2).

3. Upon decrypting message (2), A checks the key k_1 recovered agrees with that sent in message (1). (Provided k_1 has never been previously used, this gives A both entity authentication of B and assurance that B knows this key

4. A sends B message (3).

5. Upon decrypting message (3), B checks the key k_2 recovered agrees with that sent in message (2). The session key may be computed as $f(k_1 : k_2)$ using an appropriate publicly known non-reversible function f.

3.2.3 Authorisation

Public key certificates bind a public key and an identity, and include additional data fields necessary to clarify this binding, but are not intended for certifying additional information. Sometimes additional information for authorisation is needed to be included into certificates directly. These kinds of certificates

are called attribute certificates. Attribute certificates are similar to public key certificates but specifically intended to allow specification of information (attributes) other than public keys, such that it may also be conveyed in a trusted manner.

Attribute certificates provide a way to represent authorisation. For example a system administrator grants user a certificate that allows him to access specific database information but does not allow him to make modifications to this database. Traditionally, the authorisation is handled by maintaining access rights in the system where the users are required to authenticate themselves. When the user logs into the system his access rights are updated according to the access list. These access right lists may be a burden for the system administrators. In attribute certificates the authorisation is carried in the certificate itself.

Every system has at least implicitly defined policy dictating what is allowed and to whom. These policies can be collected together and grant users attribute certificates that for example define their level of rights to the system. This level of rights (authorisation) may consist of, for example, the following levels: basic, exclusive and administrator. For database systems these levels could mean, respectively, read only access for certain tables, access to additional tables and full control. When a basic level user needs access to exclusive level tables, the old certificate is discarded and a new with updated level granted.

3.2.4 Data Integrity

Data integrity is guaranteed by calculating check sums or hash values from the original data. By checking the calculated value against the sender's announced value the receiver can be sure that the data is not modified. If the plaintext is combined with a secret key common to both parties, the originator of the message can be positively identified. The secret key can be delivered by means of encryption and authentication introduced above.

3.2.5 Non-repudiation

In cryptography, the term non-repudiation means a service for providing a proof of data integrity and origin so that non of the parties of communication can deny it occurred [WAR97, 315]. In other words, neither parties of the

communication can repudiate being in contact with each other, nor can they falsify the data sent during communication. The system has to support third party verification at any given time during or after data exchange. It is then necessary to collect evidences during the communication for later verification. Evidences include information about the parties, their authorisation and the data that is exchanged.

There is, however, no need to disclose the data itself to third party. That would not be in line with communication security. Third parties must be trusted, but not to such extent that the secret information should be disclosed to them. To collect evidences meta-data of the communication is enough; signatures, timestamps and other pieces of information. For comprehensive introduction to non-repudiation services see [WAR97, Chapter 8].

3.2.6 Secure Transactions

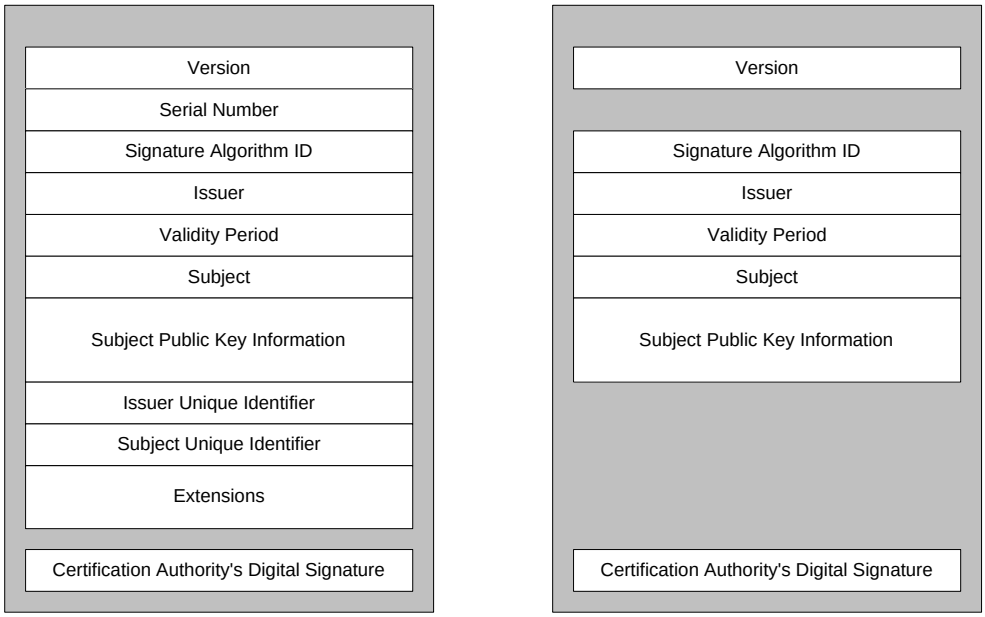
Secure transaction is actually a combination of all the parameters in characteristics above. Certificates are considered as a key part of securing transactions. There are implementations of seamless security protocol build into user level applications. Probably the most famous of them is SSL (Secure Socket Layer) developed by Netscape Corporation [NET98]. WTLS protocol for wireless devices is introduced in Chapter 5.2.

3.3 Certificate Life Cycle Issues

Many standards for managing and delivering certificates have been proposed. The most common of them, X.509 PKI, was originally presented by RSA. Later the standard was adopted and developed by ITU.

3.3.1 Certificate Issuance

Before a certification authority can issue a certificate for a subscriber, the subscriber needs to register the certification authority, typically by completing and submitting a certificate application. Registration involves the establishment of a relationship between the subscriber and CA, and the lodging of certain subscriber information with the CA [WAR97, 207]. When the CA is assured of the identity of the applicant, the certificate can be generated and delivered to the subject.



(a) X.509 Certificate (b) WTLS Certificate

Figure 3.2 Digital Certificates

3.3.2 Certificate Update

Normally certificates have a limited validity time, which can vary from few years to few minutes. After the expiration time the certificate must be updated. The way in which this is accomplished depends upon practises of the CA. In case of compromised key, either CA's or subjects, the certification must also be updated.

3.3.3 Revocation

At the time of a suspected key compromise or other reasons during the validity period of a certificate the CA by itself can issue a revocation or another authorised party can request revocation from CA. CAs frequently publishes certificate revocation lists (CRL), from which the certificate verifier can check if the certificate is revoked.

3.4 Public Key Certificate types

The most widely recognised standard public key certificate format for communication protocol level security is that defined in the X.509 standard [MEN96, 214]. Another certificate format, WTLS certificate, is based on X.509 but designed for wireless communication. In this section those two certificate formats are introduced. There are also plenty of other certificate types, but they are not really used in wireless devices.

3.4.1 X.509 Certificate

X.509 is de facto standard format for the certificates in Internet. In Figure 3.2 on the preceding page a simplified X.509 public key certificate is illustrated. The X.509 certificate consists of the fields shown in Figure 3.2 (a).

Version field describes the version of the encoded certificate. When extensions are used, as expected in this profile, use X.509 version 3 (value is 2). If no extensions are present, but a Unique Identifier is present, use version 2 (value is 1). If only basic fields are present, use version 1 (the value is omitted from the certificate as the default value) [RFC 2459].

The entity that created the certificate is responsible for assigning it a **serial number** to distinguish it from other certificates it issues. This information is used in numerous ways, for example, when a certificate is revoked its serial number is placed in a Certificate Revocation List (CRL).

Signature Algorithm ID identifies the algorithm used by the CA to sign the certificate.

Issuer Name follows the X.500 format name of the entity that signed the certificate. This is normally a CA. Using this certificate implies trusting the entity that signed this certificate. (Note that in some cases, such as root or top-level CA certificates, the issuer signs its own certificate.)

Each certificate is valid only for a limited amount of time. This **Validity Period** (Valid Not Before, Valid Not After) is described by a start date and time and an end date and time, and can be as short as a few seconds or almost as long as a century. The validity period chosen depends on a number of factors, such as the strength of the private key used to sign the certificate or the amount one is willing to pay for a certificate. This is the expected period that entities can rely on the public value, if the associated private key has not been

compromised.

Subject Name is a name of the entity whose public key the certificate identifies. This name uses the X.500 standard, so it is intended to be unique across Internet. This is the Distinguished Name (DN) of the entity, for example, CN=Vesa Hametvaara, OU=MSW, O=Nokia, C=FI (These refer to the subject's Common Name, Organisational Unit, Organisation, and Country.)

Subject Public Key Information is the public key of the entity being named, together with an algorithm identifier which specifies which public key cryptosystem this key belongs to and any associated key parameters.

Fields **Issuer unique identifier** and **Subject unique identifier** may only appear if the version is 2 or 3. The subject and issuer unique identifiers are present in the certificate to handle the possibility of reuse of subject and/or issuer names over time. [RFC 2459]

X5.09 certificates in real life usually contain some **Extensions** that are only additions in version 3. Some of the extensions are proprietary serving some specific function. Standard extensions are listed in [RFC 2459].

Signature of the above fields using the algorithm identified in Signature Algorithm ID field.

3.4.2 WTLS Certificate

The WTLS certificate is specified in [WTLS00] by WAP Forum. Compared to X.509 it is smaller and, thus, optimised for wireless communication. The WTLS certificate consists of the fields shown in Figure 3.2 on page 20 (b) [WTLS00].

Version of the certificate for the current specification is always 1.

Signature Algorithm used to sign the certificate may be any of the supported in WTLS specification.

Issuer of the certificate defines who signed the certificate. Certificates are usually signed by Certification Authorities.

Validity Period (**Valid Not Before** and **Valid Not After**) defines the beginning and the end of the validity period of the certificate, expressed in standard UNIX 32-bit format (seconds since the midnight starting JAN 1, 1970).

Subject is the owner of the key, associated with the public key being certified.

Public_Key_Information consists of **Public key type** that is the algorithm of the public key and **Parameter Specified** that define parameter relevant to

the public key.

Signature of the above fields using the algorithm identified in Signature Algorithm ID field.

4 CERTIFICATES IN THE WIRELESS ENVIRONMENT

The wireless communication introduces a number of interesting new challenges for certificate handling compared to wired communication. First, wireless network protocols vary greatly in the way they handle security issues. In Internet world we have already used to exercise well-established public key security protocols and applications like SSL/TLS and PGP. But in the wireless world, the development has been going on only for few years. Second, the network itself sets some challenges. Wireless network is often slow and unreliable, and always subject to eavesdropping because of its broadcasting characteristic. Third issue is mobile devices. Their low computational power should be considered, which means a security protocol requiring heavy computation is not adequate [CHA97, 1]. Also limited memory and user interface set requirements for security implementation.

4.1 The Network Infrastructure

Unlike in Internet world, the wireless networks are not based on any globally accepted universal standard, that is, Internet Protocol (IP). Currently there are many competing network standards in the world. The way they handle security varies significantly. The only things that are common to all wireless networks is that they are inherently slow, unreliable and the physical medium is always broadcasting type.

Wireless networks can be divided into three main categories:

- Public Land Mobile Networks (PLMN)
- Wireless Local Area Networks (WLAN)
- Wireless Personal Area Networks (WPAN).

The first category comprises of well-known standards like GSM or CDMA and their packet based enhancements GPRS and CDMA2000. The security level of these mobile networks varies from almost non-existence to relatively secure communication. For example in GSM networks the user authentication and the data security is remarkably high compared to older analogical networks like NMT. Although somewhat secure, these protocols are not by themselves secure enough for ensuring security in the application level.

The WLAN family consists of such protocols where the network coverage is not as extensive as with PLMN. These protocols correspond to the local area network (LAN) in the wired world. The most well known standard for WLAN is IEEE 802.11. The security in IEEE 802.11 is at least very questionable. A group of researchers was able to crack the encrypted passwords used in the protocol within few hours [WEP01]. Fortunately, some well established security protocols known in the IP networks can be used.

Wireless personal area networks comprise of short distance networks that are meant to be used in peer to peer like situations, for example information synchronisation with a personal computer and a hand-held device. Two examples of this kind of protocols are IrDa and Bluetooth. IrDa, which stands for Infrared Data Association, is a replacement for serial line communication. The most recent WPAN standard, Bluetooth, is a short distance (<10 meters) radio network, which is based on a small, very cheap, embeddable radio chip that can be used for establishing a communication link between two or more parties in very limited area within few meters.

Because of the wide variety of existing bearer technologies and communication protocols, the wireless communication needs a common security standard in order to inter-operate well. There has been some dispute whether the network bearers in the wireless communication should be able to provide security adequate for all purposes. Researches have suggested PKI based security mechanism for GSM [CY99]. Other studies and standards have also been published. But the main issue remains: Considering the vast variety of wired and wireless networks, how to both enable fully secure communication and interoperability between all other networks?

The question has already been answered: Adapting the same or at least fully compatible techniques as in Internet world will be the key to success. That means that PKIX standards should be the starting point also in the wireless world.

4.2 Client Devices

By far the most common portable device is a mobile phone regardless of the used protocol. The primary function of the mobile phones used to be transmitting of voice. In order to transport voice via communication line only 9600 bits/sec is enough. Because of the portability, phones are usually very lim-

ited by recourses: processing power, memory and the capability of the user interface. The first mobile networks used analogical technique so the focus of research was the quality of voice rather than data transfer and its applications. However, when most mobile networks changed into digital technology feasible data applications emerged.

Another type of portable devices is PDA (Personal Data Assistant). Many PDAs are capable of connecting to a network using a mobile phone either externally via serial cable or infra red (two box solution) or embedded into the device itself. PDAs usually have much more powerful processor and more memory. Larger display and perhaps an adjusted keyboard or a stylus makes the user experience friendlier compared to phone devices with additional PDA functionality.

The history of above two devices is very different, but the future will probably converge. Mobile phones were meant to connect to networks but not process any data. On the other hand, the purpose of the PDAs is to enable processing of data away from the users desktop PC. In the market, there already exist devices that have features of both combining data processing with networkability.

These so called third generation devices are the main concern in this thesis, because they provide the most interesting challenges and possibilities to study.

4.2.1 Memory

Typical amount of RAM in today's personal computer is around 128 megabytes, which is more than enough for computing cryptographic algorithms needed for public key systems. With enormous hard disks, the storage space will never be a problem when storing private keys and certificates.

Typical amount of overall memory in wireless devices varies from just a few megabytes to maximum a of 30 megabytes. Because the hard disks used in the PCs are not suitable for portable devices, the memory type used for permanent storage has to be very expensive flash memory. Moreover, that same scarce space has to carry also the whole operating system and user applications. Of course, the manufacturing technique of flash memory will advance and the price of the flash memory will decrease. In the mean time we just have to cope with shortage of memory recourses.

For example in current Nokia devices preinstalled certificates, cryptoalgo-

rithms and protocol modules requires around 100 kilobytes of disk space. This does not sound like it would be any problem but one has to remember that some other features always have to be dropped in order to include the security features.

According to WAP Forum's specifications, mobile entities (ME) must be able to process certificates of size up to at least 700 bytes. MEs that support X.509-based server authentication must be able to process server certificates of size up to at least 1000 bytes and CA certificates of size up to at least 2000 bytes [WCERT01, 10]. With the recent phones' memory configuration, this should not cause big problems.

4.2.2 Processing Capacity

Until recently, the processing power in embedded devices has been a problem for a pleasant user experience. It has not been possible to use high performance microprocessors in these devices because of the power consumption and heating problems. According to [FHW00], the most well known PDA, Palm Pilot, is not a suitable device for some cryptographic primitives. The RSA 512 bit key generation takes approximately 4 minutes on its 16MHz Motorola 68000 processor. Singing with this key takes about 7 seconds. The issues are much worse with the 1024 bit RSA where the key generation takes 30 minutes.

Fortunately, the battery technique and processor development has enabled the usage of more powerful processors. Some of the devices nowadays have more processing capacity than a typical microcomputer ten years back. In the near future, devices must be able process live video stream. So there should not be any big problems in processing power concerning certificate handling.

However, if some of the algorithms are meant to be executed on smart cards, as the current security specifications suggest (for example, [WIM00]), the processing power may still be an issue, since the processor in smart cards cannot handle heavy computations.

4.2.3 SIM Cards

Smart cards are single-chip computers that have non-volatile memory and are able to perform a limited number of well-defined operations. User can store his key on a smart card that usually has some physical security features. Signing

process also takes place inside the smart card, which means that the user's key is never seen outside of the card [FHW00].

SIM (Subscriber Identification Module) is a special smart card used by GSM phones to carry the owners identifying information. The user can have access to subscribed services irrespective of a specific terminal. By inserting the SIM card into another GSM terminal, the user is able to receive and make calls from that terminal, and to receive other subscribed services.

SIM card would be an ideal storage for personal certificates. The operator who delivers the SIM card to the user could generate a private key for the user and attach a signed X.509 personal certificate. Unfortunately, this is not yet true. More about a special type of smart card, WIM card, in chapter 5.3 on page 40.

4.3 Mobile Services

Mobile services are usually browser-based and the user interface is tailored for small displays. The logic is mostly in the server side running on ordinary WWW servers. The markup language used can be plain HTML, cHTML, WAP WML or other specialised language. From the security point of view WML (Wireless Markup Language) specified by WAP Forum is the most advanced because of its WIM interface.

4.3.1 Banking

In the mid 90's, a study about the user expectations concluded that the most useful application for mobile devices is banking [NOK98]. It is easy to predict that banking will stay as the number one application also in the future, especially now when the great mobile boom is over without leaving any new truly innovative service applications behind. As in the 90's the mobile banking was about tailored text messaging, in early 2000 it has developed into a browser based service.

The mobile browser has one important difference when compared to the banking services of the wired world; devices, especially mobile phones, are closely tied to one person, which makes it feasible to involve user certificates for authentication. The bank certainly wants to offer to its customers a secure and relatively easy way of using banking services. Banks in general are the

most interested party developing new security concepts. A threat of an unauthorised person reaching financial information is a very strong motivation.

4.3.2 Electronic Mail

According to the same study, electronic mail is the second mobile application. People want to read their e-mails also when they are "on the move". To put more generalised terms the users have a need to communicate with each other. Just like the mobile phone itself has met the needs for intimate person-to-person communication, the mobile e-mail will meet the needs for more formal communication.

In today's business environment e-mail is the most important way of communication. So it's not a surprise that some high-end mobile devices come with a native IMAP and SMTP support and some of them can use SSL when communicating to a mail server. In that case the devices must support usage of certificates.

4.3.3 Shopping and Auction

Another high demand application for mobiles seems to be shopping. Although, it looks like it' s going to stay as a high demand application only in service providers side who want to sell people goods anywhere and anytime, with very low costs. At least for now there is very little evidence that users really want to buy goods using their mobile devices. In the WWW world, some goods are successfully traded electronically. For example, books, computer hardware and such, but the mobile shopping is still non-existent.

4.3.4 Custom Niche Services

Unlike the three major applications above, some companies have developed tailored mobile services for their own needs. For example, United Parcel Service offers a wireless service for its customers who can check the itinerary of their packets [UPS01]. These kinds of niche services are probably the most useful of all.

4.4 Mobile Software

In applications and high level communication protocols point of view it does not matter if the underlying bearer is a GSM data call, GPRS packet radio protocol, UMTS wide band CDMA protocol or any other low level data carrier. The only differences would be a speed of communication. Of course, faster transfer rates allow more feature rich applications than in the past. However, the programming API for communication does not change even if the bearer changes. It is expected that the mobile software will be using either IP protocol stack or compatible since that is the standard for Internet.

4.4.1 Security Protocols

SSL/TLS

SSL (Secure Socket Layer) developed by Netscape Corporation from versions 1.0 to 3.0 is widely used in Internet . Later the standard has been adopted by IETF and is now known as TLS 1.0 (Transaction Layer Security). It is a security layer above TCP and consists of two separate sub-protocols, namely Record Protocol and Handshake Protocol. The SSL Record Protocol defines the basic format for all data items sent during the session. It provides compressing of data, generating an integrity check-value on the data, encrypting the data, and ensuring that the data receiver can determine the correct data length. [WAR97, 169]

The SSL Handshake Protocol is used to negotiate which protection algorithms will be used to authenticate the client and server to each other, to transmit required public key certificates and to establish the session keys. Different key establishment algorithms can be supported, including RSA key transport, Diffie-Hellman key agreement and the KEA. [WAR97, 170]

WAP - WTLS

WAP (Wireless Application Protocol) is a whole range of protocol specifications designed for wireless communication. The WAP counterpart for TLS is called WTLS (WAP Transaction Layer Security). The WTLS is explained in more detail in chapter 5.2.

4.4.2 User Applications

One could say that there is not such thing as mobile application software. There is only desktop software that is fitted to run on wireless hardware. People want to use the same applications in their mobile devices as when they sit in front of the personal computer. It's the quality of the fitting and the user experience that differs from a vendor to another and from one product generation to another. When the digital convergence really happens, probably some truly mobile applications start appearing into the market.

Browsers

The ultimate goal for all mobile browsers is to enable access to World Wide Web, where all the services are. Some mobile browsers and standards have succeeded better than others, but usually the user expectations are too high to met. The Japanese I-Mode with its micro-browser is the most successful so far. Other contenders like WAP WML and its predecessors are maybe technically superior, but lag behind in the user experience and number of services. Some devices, especially on PDA side, have real WWW-browsers, but the screen size is too small for most of the today's WWW services. Moreover, the network speed has not been acceptable.

Certificates have been widely used with mobile browsing for few years now. To be more precise, the usage of certificates has been enabled, but very few people have used mobile browsing in general. Nevertheless, the technology exists. Most WAP and mobile WWW browsers can establish secure connections with servers. Enabling secure transactions in mobile browsing has been a primary goal for many standards and organisations.

Electronic Mail

E-mail has been perhaps the most useful piece of software for mobile devices. Usually the interface of the devices fits rather well for reading E-mail. Although, compared to PC, composing messages requires much more effort because the user interface for typing is a compromise between size and usability.

Most of the E-mail applications are capable of connecting to mail servers using SSL or TLS, hence using certificates for authentication and secure data communication.

Calendar and Business Card Applications

These so called PIM (Personal Information Management) applications are always included in the wireless devices. They are not interesting from the security point of view. The interesting part is the on-line synchronisation software, which synchronises the information from the server so that the data entered in desktop PC will come also to mobile device. For example, SyncML, which is a new XML based open protocol for on-line synchronising calendar events and contacts. It is specified that SyncML uses HTTP or WSP protocols. Hence it is possible to use secure certificate based connection when security is needed [SMLH02, SMLW02].

Electronic Wallet

Electronic wallet can be defined as a piece of software or hardware that carry the users certificate and credit/debit card information to be used in electronic transactions. Information in the wallet and during transactions is protected by cryptographic means.

People often carry their identification card and credit/debit cards in the traditional wallet. The same information can be easily presented in any electrical form and in any device, for example in mobile phones. Some phones already have the capability to hold wallet information but they are not yet widely used.

5 WPKI

5.1 Introduction to WPKI

As a part of the WAP security standards, WAP Forum has introduced a "Wireless Public Key Interface" definition fitted to the wireless environment. [WPKI01]: *The general model is adaptable to many certificate types including X.509, X.68 and the certificate format defined in WTLS [WCERT01, 10]. The WTLS certificate has the advantage of being very compact, easily implemented in code, and easily parsed which may be important for initial implementations of WAP clients. In addition to the extend possible, the WAP PKI will work interchangeably with existing X.509v3 certificates in existing Internet applications, in order to leverage the existing Internet PKIs. Any new format that requires major changes to the installed vase of certificate-processing products and CA infrastructure is unlikely to be easily adopted in a short time-frame.*

The WPKI specification adopts the following model [WPKI01, 12]:

- WTLS server and Root CA certificates stored in the device are WTLS certificates.
- Client certificates and CA Roots stored in servers are X.509 certificates.
- Client certificates and CA Roots which are to be sent over the air (OTA) and/or stored in WAP client devices will be according to the X.509.
- Storage of the certificate URL in the device, rather that the full client certificate, is the preferred mode, when X.509 format certificates would otherwise be expected to be transferred OTA.
- Storage of X.509 client certificates in the device is expected to be the exception, unless they are provisioned on the device.

WAP Forum has tried to keep a balance between well-established interface (PKI) and scarce resources of the mobile environment. Using X.509 certificates in the servers will maintain the interoperability with existing content servers and adopting URL certificates saves bandwidth for other tasks.

5.2 WTLS

WTLS Wireless Transaction Layer Securely is a protocol layer in WAP stack that is responsible for establishing and maintaining a secure connection between a client and a server. It is a system level component as opposed to its original model TLS, which is partly an application level protocol. WTLS also enables a connection-less secure mode that TLS is lacking. There are several classes of WTLS models. In class 2, CA public key certificates are used for authenticating the server. In class 3, also client side public key certificates are used.

It must be noted that even if the channel is encrypted, the data must be decrypted in the gateway before the data is again encrypted for SSL/TSL connection to the server. Even if the decrypt/encrypt operation takes place in the memory, it can be scanned. This is why the security in WAP has been criticised. The industry is talking about premature encryption termination. By locating the WAP gateway to a secure place behind firewalls, the risk of eavesdropping can be minimised.

Premature encryption termination is not the only feature in WTLS that has raised criticisms. According to a researcher despite their close resemblance, the WTLS protocol appears to be more vulnerable to attacks than TLS [MJSA99].

WAP Forum has specified that for WAP 2.0 the TLS protocol should be used instead of WTLS. This is because the devices have become more capable than before. However, it will take few years before WAP 2.0 capable devices are commonly used. In the mean time, we have to settle for WTLS.

5.2.1 WTLS Class 1

WTLS class 1 offers an encrypted channel between the client and the server. However, it does not incorporate certificate-based authentication. This level of security is enough for the situations, where the information is confidential, but the authentication of the parties is not so essential.

5.2.2 WTLS Class 2

CA public key certificates are used in WTLS class 2 for authenticating the server. This class is suitable, for example, for credit card payment, where the

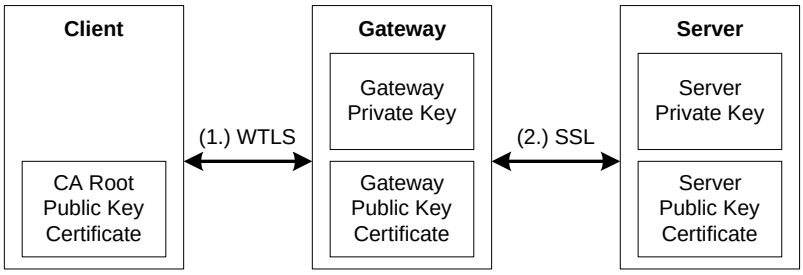


Figure 5.1 WTLS Class 2

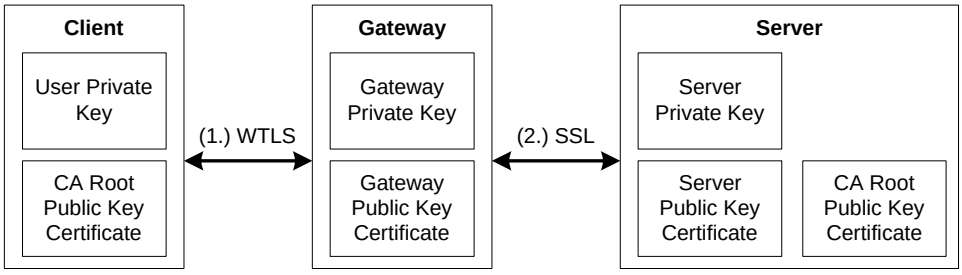


Figure 5.2 WTLS Class 3

client wants to be sure the server is the one it is supposed to be, but the authentication of the client is not so important. For the server side it is enough that the credit card company accepts the payment, even if it's unauthorised. If the server needs to be sure about the client, authentication must be performed by some other means, for example by password query.

In order to authenticate the server, the client must verify the certificate of the server against the CA certificate. The verification is explained in 5.2.6. If the verification succeeds, the client can be sure of the server's identity.

5.2.3 WTLS Class 3

In class 3 the verification of the client's certificate in server side is almost identical to the class 2, where the client verifies the server's certificate. The only exception is that there is probably more computing resources available in the server side and checking the CRL is much easier. The user must have a private key stored in the device and a corresponding user certificate, signed by a CA, either in the device or in external repository from where the server can retrieve it.

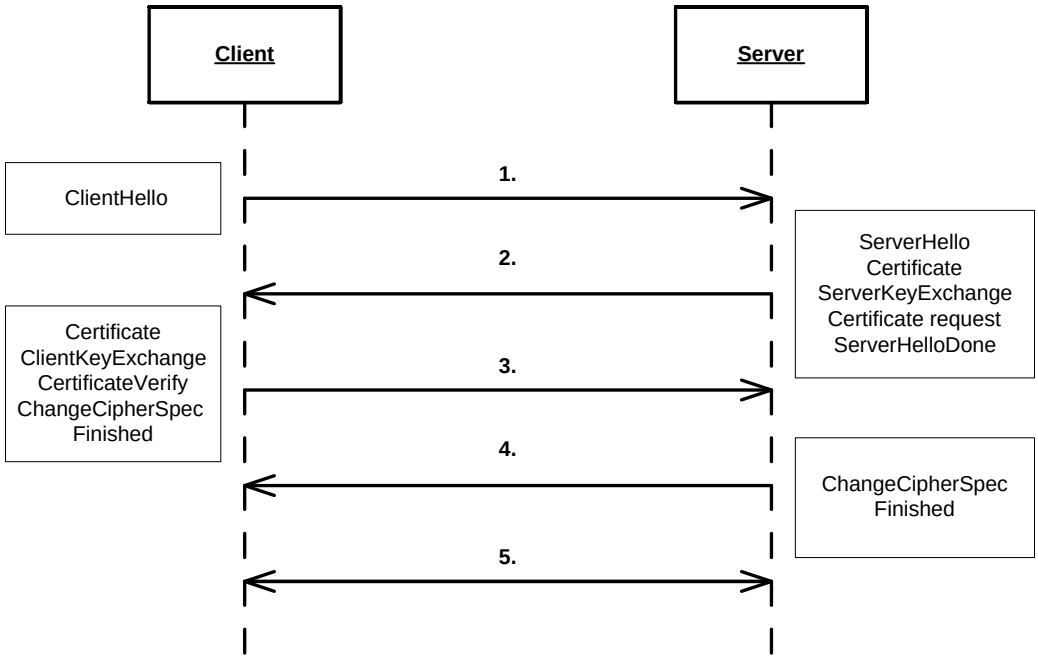


Figure 5.3 Full WTLS Handshake [WTLS00, 49]

5.2.4 End-to-End Security

Application level end-to-end security can be achieved with a SignText operation specified in WAP 1.2 [WSCL00]. This function takes a string input and outputs a signed string. The server can challenge the client by calling the function in a WAP page. The client signs the plain text challenge with its private key and sends it to back to the server, which should then be able to verify the signature with the client's public key. The SignText operation must be protected with a PIN query, so that unauthorised person cannot pretend to be the client.

WAP 1.2 and 2.0 specifications introduce a system level end-to-end security model, where the default WAP gateway re-directs the client to another gateway, which is located in the same security zone as the content server [WETE01].

5.2.5 WTLS Authentication

In Figure 5.3 the full WTLS handshake procedure is illustrated as a sequence diagram. (1.) The handshaking is initiated by the client by sending a Client Hello packet to the server. (2.) The server responses with its public key certificate and other session parameters. For example, the used cipher suite and

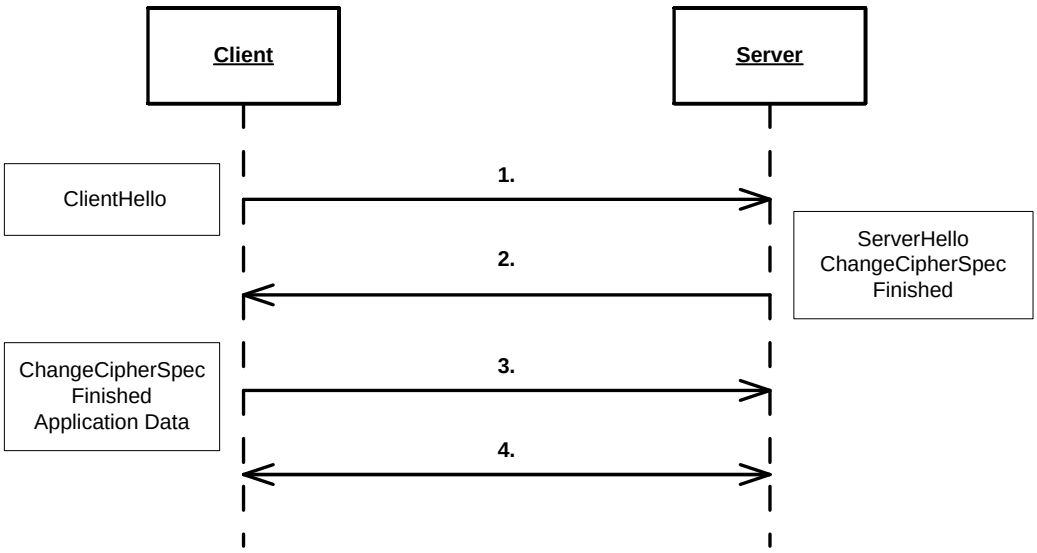


Figure 5.4 Abbreviated WTLS Handshake [WTLS00, 50]

premaster key, which is used to form a master key for this particular session, are sent along with the response from the server. In case of WTLS class 3 or End-to-End protocol version the server requests a client certificate for authenticating the client. In class 2 the client must be authenticated by other means, usually by requesting a user name and password. (3.) The Client verifies the server certificate and sends back its own certificate along with other session parameters. (4.) The server verifies the client's certificate and sends a response to the client. (5.) Both client and server are now authenticated, the session's master secret is calculated and the used cipher suite agreed. The application data pipe is open.

After the full WTLS handshake is done once the session established earlier can be reused for as many times as the parties agree. In Figure 5.4 the previous session parameters are stored in persistent storage in both client and server. There is no need to send certificates or premaster secret because the session id and the shared secret key are available to be used immediately. Although sometimes when the security must be maximised, the full handshake can be executed during the connection.

Sometimes the client does not have to send it's certificate over the air, instead the server can retrieve it from a certificate distribution system using defined exchange protocol, for example [RFC 2510, RFC 2559, RFC 2585]. Figure illustrates the messaging sequence between the client and the server in that

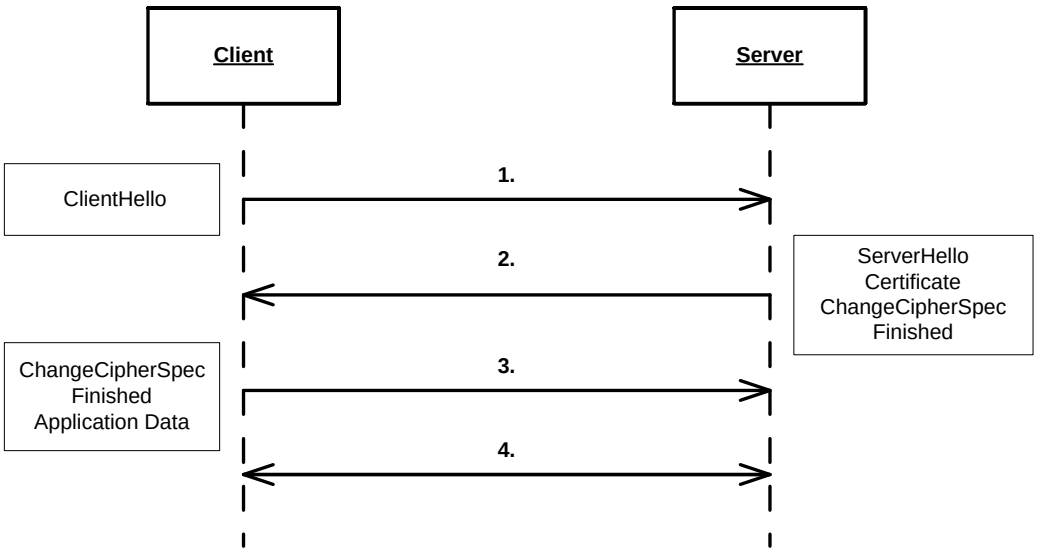


Figure 5.5 Optimised Full WTLS Handshake [WTLS00, 50]

case. (1.) The client sends a Hello message to the server. (2.) The server knows that for this particular client it has to retrieve the certificate from an other service, so it does not send a certificate request to the client. (3.) Client responses as in full handshake, but without its certificate. The data pipe for application is now open.

Optimised full handshake can be used also when the client and the server share a secret, which is securely provisioned into the client device. This shared secret can be used as a premaster key.

5.2.6 Server Certificate Verification during Handshake

Client's possibilities to verify the server certificate using long validation chain described on page 13 are limited. Theoretically, it is possible to implement some kind of an on-line certificate repository for mobile clients. However, every certificate in the chains should be verified for possible CRL entry or, in the worst case, the whole certificate may have to be fetched from a network server. Even to determine if one certificate is on the current CRL takes a lot of effort, as Russell describes [RUS99].

To determine if the certificate in question is in a CRL, the computer needs to:

1. Determine the class of CRL.

2. Determine the version.
3. Extract the individual item of the data (issuer name, signature, certificate list, etc.).
4. Identify the validity period.
5. If expired, reject the CRL, and take security actions.
6. Identify the signature.
7. Identify the signature algorithm used.
8. Reconstruct the original unsigned CRL.
9. Calculate the corresponding hash.
10. Identify the signer.
11. Obtain the signer's public key.
12. From the signer's key and the signature algorithm, calculate the hash signed by the signer.
13. If the hash is different from that of the local calculation, reject the CRL and take security actions.
14. Search through the list of identifiers of revoked certificates, looking for a match with the identifier of the certificate under analysis.

Although Russell does propose some faster ways to determine if the certificate is valid [RUS99], it's because of the complexity of the chain verification, why the industry has not yet implemented any universal chain verification to the mobile devices. Usually the client devices have some preinstalled CA certificates to be used for verification of the server certificates. In some devices, it is even possible to install new root certificates as described in Chapter 5.4.1. Therefore, in practise there can be only two or three level trust chains, a signer's root certificate for and possibly one intermediate CA certificate. Sometimes the device has no signer's certificate installed and the only way in that

case is to ask from the user if he trusts the server certificate in question, or not. Fortunately, the situation is not worse than in WWW world.

On-line Certification Status Protocol

It is expected that in near future the speed of the wireless communication will allow on-line status queries. OCSP is a standard protocol that can be used for checking the status of certificates. The protocol is described as follows [RFC 2560]:

The On-line Certificate Status Protocol (OCSP) enables applications to determine the (revocation) state of an identified certificate. OCSP may be used to satisfy some of the operational requirements of providing more timely revocation information than is possible with CRLs and may also be used to obtain additional status information. An OCSP client issues a status request to an OCSP responder and suspends acceptance of the certificate in question until the responder provides a response.

This protocol specifies the data that needs to be exchanged between an application checking the status of a certificate and the server providing that status.

5.3 Storing Certificates in WIM

WIM is a tamper-resistant device that can store private keys so that a key compromise is made impossible or at least very hard. In addition to private keys WIM can carry other cryptographic objects and algorithm libraries. WIM uses an object model that makes it possible to access keys, certificates, authentication objects and proprietary data objects in a simple device [WIM00, 6]. In WPKI, WIM offers the needed cryptographic operations for WTLS. For example, it is responsible for storing and verifying certificates.

WIM can be, and usually is, implemented on a smart card identical to SIM. Thus, it is possible to combine those two into one SWIM card, or use two separate cards in the device. The smart cards can be protected with a PIN code to prevent unauthorised access. For the users, the SWIM card provides a nice way to maintain their personal security environment even after changing the device. Just by inserting the WIM card into the new device, the user is able to authenticate to all the familiar services as before.

From security point of view, there is no requirement of storing certificates in a tamper resistant place. Storing certificates in WIM may be useful from point

of view of logistics and portability. In WTLS, the server may retrieve client's certificate from its own sources. In addition, it is possible to store a certificate URL instead of the certificate itself in WIM [WIM00, 15].

Although specified for WAP the certificates stored in WIM can also be used by other applications like e-mail or WWW browser to be used with SSL. In that case, the device has to offer an exported API to WIM operations.

5.4 Trusted CA Certificate Handling in WPKI

The WPKI definition currently supports two types of certificates used as a CA certificates:

- WTLS certificates (Mandatory)
- X.509 certificates.

5.4.1 Delivery

WPKI introduces two primary methods for delivering of CA certificates into a client: prepackaging and downloading. Just like in Internet world operators and manufactures, usually include some root certificates to the client. For example, WWW-browser packages usually contain a certain set of well-known root certificates from different Certification Authorities. In case of mobile phones, the root certificates can be stored in tamper-proof read-only memory in the manufacturing phase. The user can be reasonably sure that the certificates stored are trustworthy.

Another way to install new certificates is to download them from the network to the client device. In practise, downloading is initiated by following a link in a hypertext document. In the client device there may be some preconfigured starting page that contain links to some repositories of certificates. It is also possible to fetch certificates via FTP or LDAP protocols.

Technically, it is possible to deliver CA certificates via WAP PUSH messages, but the WPKI specification explicitly forbids the usage of server initiated delivery methods. [WPKI01, 18]. PUSH messages allow delivery of arbitrary content to the client devices that are enabled to listen such messages. We cannot expect users to understand the meaning of such message. It would be

highly probable that the user just accepts a malicious CA certificate when his intention is to get rid of the annoying message.

The WPKI specification warns the implementers about potential dangers of easy installation of CA certificates. "The first installed CA has the opportunity to fairly easily introduce new CAs, and users are reasonably likely not to understand the significance of accepting a new CA using this mechanism." [WPKI01, 23] The specification also suggests that operators should be responsible for provisioning installable CA certificates, but does not represent any suggestions for implementation. One possible solution, which should not be over looked, is to prohibit downloading of certificates entirely. Operators are known to be paranoid when it comes to the question of their possible liability in court.

5.4.2 Verification

Verification of preinstalled certificates is not usually needed. The user should be confident that the operator has already verified that these CA certificates can be trusted. In addition, the CAs usually guard their private keys tightly. Downloading of new self-signed certificates is of course much more complicated than just prepackaging all the needed certificates into the device. For starters, the users have to be able to find the correct URL to even reach the certificate needed. Moreover, simply downloading and installing them could be risky if the user is, as they typically are, ignorant about security issues. The risk comes from the fact that the self-signed certificate is guaranteed to be intact but not trusted in any way. Usually the decision of whether the certificate is trusted or not is delegated to the user.

In order to help the user to be more capable of making the decision about whether to trust the certificate and the signer or not, WPKI suggests two methods for the verification of these certificates. They both sound technically solid, but some may object whether they really guarantee the trust part of verification. But to be fair, these methods are certainly better than the ones used with WWW browser where the verification of downloaded certificates is almost non-existent.

Out of Band Hash Verification

In this method, a hash value is computed from the certificate together with a CA information block. The certificate and the CA information are included into a package with a special mime type: *application/vnd.wap.hashed-certificate*. The precise profile of this mime type is presented in [WPKI01, 19]. The value of the hash is delivered via some out of band channel to the client where it's compared to the hash value computed in the client side. Snail mail letter or an advert in a magazine should be enough "out of band" for delivering the hash. The method goes as follows.

- Server's tasks:
  - Form a self-signed certificate. The format may be either WTLS or X.509.
  - Compute a SHA-1 hash from the certificate.
  - Distribute the certificate and CA information in defined data structure.
  - Deliver the hash via out of band channel, for example printed media, web page.
- Device:
  - Download the certificate from the server.
  - Compute a SHA-1 hash from the certificate.
  - Prompt the user to enter hashed value manually.
  - Verify the hash.
  - Inform the user about the decision.

The content of the mime type application/vnd.wap.hashed-certificate is specified as follows:

```
struct {
    CharecterSet    character_set;
    Opaque          displayName <0^8-1>;
} CertDisplayName;
```

```
struct {
    uint8           version;
    CertDisplayName displayName;
    Certificate     trustedCACert;
    opaque          cainfo_url <0^8-1>;
    HansAlgorithm   hash_alg;
} TBHTrustedCaInfo;
```

Signature Verification Method

- Server
  - Compute a signature from the certificate.
  - Distribute the certificate and the signature with a mime type `application/vnd.wap.signed--certificate.`
- Client
  - Download the content.
  - Verity both outer and inner signatures.
  - Inform the user about the result.

Because the hash value is too long for the users to enter in full length, WPKI describe a method of just entering a part of the hash without lowering a level of security [WPKI01, 21].

In WPKI, the verification of new CA certificates is supposed to be performed during the installation time. In some cases it may be feasible for users to be able to manually initialise the verification of already installed certificates, for example, in the situation when a person has both a used device in which there is unknown CA certificates installed or when CA certificates are stored into WIM or comparable removable medium.

5.4.3 Key Rollover

All certificates have a limited valid time. For this reason, there must be a way to replace the old certificate with the new one. For trusted CA certificates rollover

WPKI introduces a mime type application/vnd.wap.rollover-certificate. The content of such mime type consist of a newly generated certificate signed with the old certificate. The structure of the mime type specified as follows.

```
Struct {
    SignedTrustedCAInfo    signed_trusted_CA_information <0..2^16-1>
} RootCertificateRolloverBlock
```

The clients must be able to validate the new certificate. The verification consists of the following steps:

1. Use the root public key that they currently have to verify the signature on the trusted CA information block.

2. Perform all the checks indicated in section 5.4.2 on the trusted CA information block

3. Accept the root certificate rollover; replace the current root CA certificate with the new root CA certificate.

5.5 User Certificate Handling

Obtaining a public key certificate is not a trivial task for the users. First the users should generate a key pair, save it to secure place in the device and then he should get the public key somehow certified by a CA, which is accepted by the service. This is why we need simple and seamless procedure for obtaining certificates.

5.5.1 Obtaining User Certificates On-line

The WPKI specification introduces a special entity called PKI Portal. The portal is responsible for providing an interface for making on-line requests. The certification authority, for example, can provide such an interface or it can be a completely separate entity. The specification [WPKI01, 17] illustrates one possible way to retrieve the user certificates. The following sequence is suggested: (see Figure 5.6.)

1. The client connects to a service using standard WAP browser.

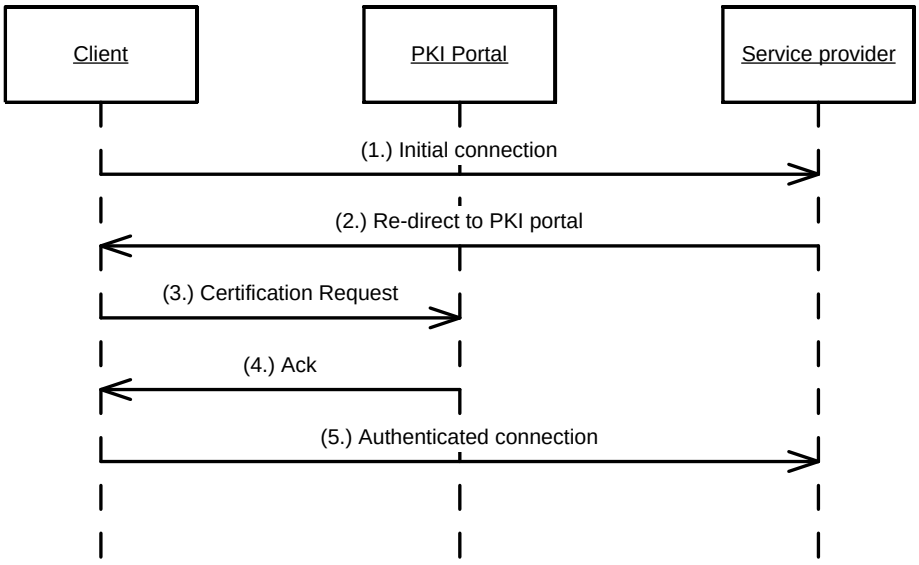


Figure 5.6 User Certificate Request

2. The service wants the user to authenticate and re-directs the user to the PKI Portal to obtain a user certificate.

3. The client makes a certificate request to the PKI Portal.

4. The portal acknowledges the requests and gives the client guidance on how to retrieve the certificate. The Portal either sends the generated certificate to the client or just a reference to the certificate in on-line repository somewhere else in the network. The portal must use some proof of possession procedure to make sure that the user has a private key corresponding to the public key.

5. The client initiates a new connection to the server and this time is able to authenticate with the certificate.

The above scenario looks very simple way of obtaining certificates. However, it must be noted that this scenario is not the only way to obtain user certificates. This scenario seems to involve actions from the users and it is highly unlikely that they understand what they are supposed to do and why. More suitable scenario for obtaining user certificates is described in the following section.

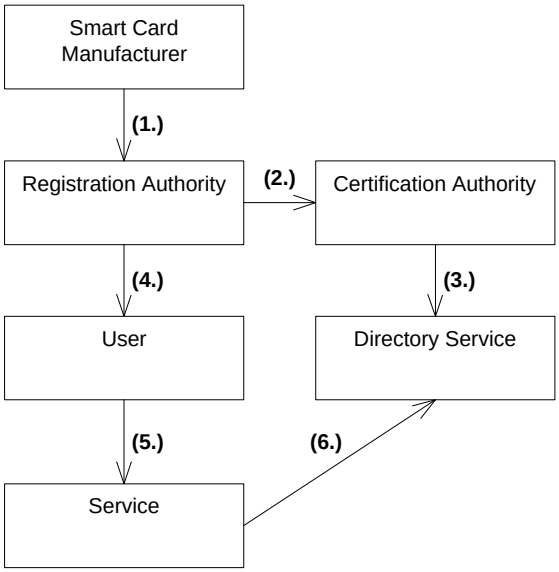


Figure 5.7 Obtaining User Certificates Off-line

5.5.2 Obtaining User Certificates Off-line

In Figure 5.7, the most probable scenario of obtaining user certificates is illustrated. The scenario does not introduce any new parties, except the registration authority. However, from the user's perspective, the procedure is not more difficult than simply obtaining a SIM card from the operator or applying for a credit card.

1. A smart card manufacturer delivers a WIM card to the registration authority. In most cases, the registration authority will be a mobile operator. In the WIM card there is two private keys, one for authentication and one for signing. Public keys and the PIN codes are delivered with the WIM card. Nobody can alter the private keys in the card, so they are safe.

2. After the user has made a request for the WIM card, the registration authority requests the certification authority to deliver certificates for the user and corresponding public keys in the WIM card. Sometimes the CA can be the RA itself. This could be the case with, for example, a bank who wants to make sure that any information of its clients will not be disclosed, not even names.

3. The CA either stores the certificates into the directory service or delivers them to RA. The certification of the private keys will be performed following the policy of the CA.

4. The RA delivers the WIM card and PIN codes to the user.

5. The user connects to the service, which requires a certificate from the CA. During the handshake, the server retrieves the client certificate from the directory service.

5.6 Certificate Storage Management

The users have to be offered some way to manage certificates installed in the device. In practise, this means some kind of a user interface where the user could:

- View certificate information. According to [MCU01, 38], at least certificate label, subject, issuer, validity period and the fingerprint should be displayed.

- Remove unnecessary certificates. This may not be possible if some of the certificates are stored in ROM. In this case the next feature should be implemented.

- Adjust the level of trust per certificate. A suitable set of levels could be: “Trust always”, “Ask when connecting” and “Never trust”.

- Manually initiate verification. The verification must, as a minimum, check the validity period and the verification chain. Checking against CRL should be implemented if the device is capable of doing that.

The MET group lists the first two features above as mandatory in their specification [MCU01, 43].

6 STANDARDS AND ORGANISATIONS

As often in the computing world there are a plenty of standards to choose from. That is the case also in the wireless world. Fortunately most of the standards in the security business are compatible with each other and are based on well-tested principles in Internet.

6.1 Potential PKI Portals and CAs

In many countries the government is responsible for granting identity cards for its citizens. Also, in Finland and Sweden the government grants electronic identity cards that can be used for digitally signing on-line transactions with the government offices and some private service providers. Therefore the government could act as a nation wide certification authority. Another governmental organisation suitable for offering a portal could be, for example, the European Union.

However, the PKI Portal as suggested in the WAP specifications [WPKI01] is not likely to be implemented globally or even nation wide. There is no such entity that could be trusted worldwide. More likely, different interest groups will begin forming several, probably overlapping, PKI Portals. For example, a Finnish mobile operator Radiolinja has recently announced that it will start working as a certification authority for its subscribers and offer WPKI functionality. Several service providers, insurance companies, a major bank and a big retail chain have made an agreement with Radiolinja that they will accept Radiolinja's signed certificates for transactions. [RAL02] One must not be an expert to see that the operators in general are the most suitable for the job.

6.2 Standardising Organisations

There are many organisations that are interested in developing a world-class standard for secure mobile transactions. Most of them are co-founded by big corporates and the main goal is to make sure that the infrastructure allow people to consume their money anywhere, anytime and with ease. In other words, the goal is to establish a common business environment for device manufactures, operators and service providers.

6.2.1 WAP Forum

WAP Forum develops specifications for enabling wireless services in mobile terminals. The standards are available for downloading by anyone. "Wireless Transaction Layer Secure" and "WAP Identity Module" among with other security related specifications are widely considered as reference material for implementers of secure services. WAP Forum has adopted the same technology as in Internet and fitted it to wireless world.

6.2.2 SET

One of the early pioneers in Internet secure transactions area is SET organisation (Secure Electronic Transaction), which defines itself as follows: *SET is an open technical standard for the commerce industry developed by Visa and MasterCard as a way to facilitate secure payment card transactions over Internet. Digital Certificates create a trust chain throughout the transaction, verifying cardholder and merchant validity, a process unparalleled by other Internet security solutions.*

The SET has never been a big success in the PC world, because the users need to buy a relatively expensive piece of equipment, the smart card reader. Otherwise SET could have become a widely spread standard. SET is one potential registration/certification authority for WPKI.

6.2.3 MET

As the WAP Forum has specified the standards from the implementers overview, MET (Mobile Electronic Transaction) has taken an other perspective for secure transactions, the user experience. MET embraces current WAP standards including WTLS and WIM. The driving force for MET specifications is a predictions that wireless devices will transform into something they call as Personal Trusted Device (PTD). The MET specifications describe common usage scenarios for WAP shopping, banking and other commonly known applications in user's perspective.

6.2.4 MOBEY

Mobey Forum also embraces WAP architecture, like MET, but also incorporates the same kind of technology as SET. From their technical specification

[MTE01]: *The objective of the Mobey Forum is to enhance the use of mobile technology in the financial services market, including banking, payments and brokerage. The methods of Mobey to achieve this are by creating business and technical requirements, evaluating potential business models and technical solutions, and then making recommendations to standardisation bodies, handset manufacturers, payment schemes and technology suppliers to implement the required solutions.*

7 CONCLUSIONS AND THE FUTURE

WPKI provides a public key infrastructure for mobile devices and brings the the X.509 certificates used in Internet to the mobile users. Together they basically allow using the same certificate based services in mixed environment where the underlying network bearer and device can change from wired PC to mobile phone. The private key needed for transactions can be carried in a tamper resistant smart card, WIM card. WIM specification could be SET done right if the manufacturers can bring enough applications and devices with reasonable prises to the market.

On of the most questionable security features on WAP is the fact that in WTLS connections the WAP gateway decrypts the communication between the client and the gateway in order to be able to encrypt it for TLS connection established between the gateway and the content server. I would not personally trust national secrets over the current WAP protocol. Fortunately in the future specifications the WAP gateway will be left in history and a pure TLS protocol will be utilised also in mobile services.

As the capacity of the mobile networks and the processing power of the client devices evolve, all the bottle necks from the security point of view will be fixed. There will be no excuses not to incorporate strong authentication and privacy methods for wireless services. However, the political issues, like who can you trust as a CA, will not be solved purely by technical progress. The author's prediction is that the mobile operators will take a leading role in building of WPKI portals. Some of this is already in sight, since the first pilot projects have been gone through and the final announcement for starting real life services have been made.

One thing that bothers when reading papers and specifications about WPKI. The infrastructure and theory is complicated and terms are abstract even for a person who has been playing and developing these mobile devices for several years. How are the manufacturers going to get the ordinary users to understand what is this certificate thing all about? The specifications don't seem to take into account that the users do not even want to know how the security in their devices work. What is the point of, for example, to tell the users that they have to verify the signature of the server certificate they just downloaded from their bank's homepage? Fortunately, MET initiative is trying to address at least some of these questions. The user interface requirements for the mobile

security could be a title for a whole new thesis.

7.1 Future

One of the key benefits that certificate-based authentication brings for ordinary users is the possibility to get rid of the massive number of username-password pairs we have to remember nowadays. Certificate-based authentication should be commonly used within few years.

The most interesting new application for the future is an incorporation of the mobile phone's security element (WIM/SWIM) in the personal computer. One reason mentioned for the partial failure of SET payments was the fact that people were not willing to buy smart card readers for their home PCs. Using the reader in the phone and connecting the two devices with Bluetooth short distance radio enables an easy and secure way to make on-line transactions in Internet.

7.1.1 Hybrid Payment Scenario

Let a term hybrid payment denote a mixed usage of different security elements, communication devices and protocols for executing a payment transaction. Consider a scenario where a user wants to buy a book from some Internet bookstore using his credit card information. A WIM card, which is issued by his credit card company, is inserted into his phone. The phone itself is in a pocket of his jacket hanging on the wall somewhere in the apartment. When he is about to execute the transactions the PC software searches the phone for accessing the credit card information. The user accepts the transaction with his personal PIN code and the transaction is completed. Of course, the user could have performed the transaction using his phone while on a train heading for work on Monday morning. In that case he would have to use the browser in his device and the interface would be a bit different. However, the WIM card would still be in his phone and the PIN code would be the same that he enters to the PC.

8 SUMMARY

In the beginning of this thesis the area of the information security in cryptographic point of view was described, public key cryptography being in the centre role of the theory. Digital certificates and how they are connected to the public key cryptography was introduced in chapter three. Issuance, update and revocation were the main issues concerning the certificate life cycle in the wireless world. The two most used public key certificate types X.509 and WTLS was introduced as an example.

In Chapter four the wireless environment was described. The environment consisted of the following elements: network infrastructure, client devices, services and application software. The affect of using certificates with these key elements was discussed. In the following chapter a WAP security, where WIM and WTLS are the main points of the model, was introduced. Certificate issuance, update and revocation in WAP environment was introduced in detail, as was also the WIM usage for storing private keys, certificates and certificate URLs.

In chapter six most of the key organisations and the standards that they specify were introduced, the top player being WAP Forum.

REFERENCES

[AVS98] Antti Vähä-Sipilä, *"Salausavainten ja luottamuksen hallinnasta avoimissa tietojärjestelmissä"*, M.Sc. Thesis, Tampere University of Technology, 1998

[CHA97] Chang-Seop Park, *"On Certificate-Based Security Protocols for Wireless Mobile Communication Systems"*, IEEE Network, September/October, 1997, Volume: 11 Issue: 5, pages 50-55

[CY99] Chi-Chun Lo, Yu-Jen Chen, *"Secure Communication Mechanism for GSM networks"*, Consumer Electronics, IEEE Transactions on , Volume: 45 Issue: 4 , November 1999 Pages: 1074 -1080

[FHW00] Margus Freudenthal, sven Heiberg, Jan Willemson, "Personal Security Environment on Palm PDA", IEEE Proceedings, 2000, also http://citeseer.nj.nec.com/freudenthal00personal.html

[MCU01] "MET Consistent User Experience", Version 1.1, Nov 2001, http://www.mobiletransactions.org

[MJSA99] Makku-Juhani O. Saarinen, *"Attacks Against the WAP WTLS Protocol"*, Bart Preneel (ed.) Proc. Communications and Multimedia Security '99, Kluwer Academic Publishers, 1999.

[MEN96] A. Menezes, P. van Oorschot, S. Vanstone. *"Handbook of Applied Cryptography"*, CRC Press, 1996

[MTE01] *"The Preferred Payment Architecture - Technical Documentation"*, Version 1.0, Jun 2001, Mobey Forum, http://www.mobeyforum.org/PPATechnical.pdf

[METWP01] *"MET Overview White Paper"*, MET, 2001, http://www.mobiletransaction.org

[NET98] *"Introduction to SSL"*, Netscape Communications Corporation, 1998, http://developer.netscape.com/docs/manuals/security/sslin/contents.htm

[NOK98] *"The demand for mobile value-added services"*, Nokia, 1998, http://www.nokia.com/press/nps_white_papers.html

[RAL02] *"Radiolinja WPKI Langaton Julkisen Avaimen Järjestelmä"*, Radiolinja, 2002, http://www.radiolinja.fi

[RFC 2459] R. Housley, W. Polk, W. Ford, D. Solo, *"Internet X.509 Public Key Infrastructure, Certificate and CRL Profile"*, January 1999, http://-www.ietf.org/rfc/rfc2459.txt

[RFC 2510] C. Adams, S. Farrell, *"Internet X.509 Public Key Infrastructure Certificate Management Protocols"*, March 1999, http://-www.ietf.org/rfc/rfc2510.txt

[RFC 2559] S. Boeyen, T. Howes, P. Richard, *"Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2"*, April 1999, http://-www.ietf.org/rfc/rfc2559.txt

[RFC 2560] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams, *"X.509 Internet Public key Infrastructure. Online Certificate Status Protocol - OCSP"*, Feb 2002, http://www.ietf.org/internet-drafts/draft-ietf-pkix-rfc2560bis-01.txt

[RFC 2875] H. Prafullchandra, J. Schaad, *"Diffie-Hellman Proof-of-Possession Algorithms"*, July 2000, http://www.ietf.org/rfc/rfc2875.txt

[RFC 2585] R. Housley, P. Hoffman, *"Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP"*, May 1999, http://www.ietf.org/rfc/rfc2585.txt

[RUS99] Russell S., *"Fast Checking of Individual Certificate Revocation on Small Systems"*, IEEE Proceedings, 6-10 December, 1999, pages 249-255

[SMLH02] *"SyncML HTTP Binding"*, Version 1.1, Feb 2002, http://www.syncml.org/docs/syncml_http_v11_20020215.pdf

[SMLW02] *"SyncML WSP Binding"*, Version 1.1, Feb 2002, http://-www.syncml.org/docs/syncml_wsp_v11_20020215.pdf

[TAN96] Andrew S. Tanenbaum, *"Computer Networks"*, Third Edition, Prentice Hall, 1996

[UPS01] *"UPS Wireless Phone Solutions"*, United Parcel Service, 2001, http://www.ups.com/bussol/solutions/wireless/phones.html

[USG97] *"Entity Authentication Using Public Key Cryptography"*, U.S. Department of Commerce / National Institute of Standards and Technology, FIPS PUB 196, 1997

[WAR97] Warwick Ford, Michael S. Baum, *"Secure Electronic Commerce"*, Prentice Hall PTR, 1997

[WCERT01] *"WAP Certificate and CRL Profiles Specification"*, Version 211, May 2001, http://www.wapforum.org

[WEP01] Adam Stubblefield, John Ioannidis, Aviel D. Rubin, *"Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, AT&T Labs Technical Report TD-4ZCPZZ"*, 2001, http://www.cs.rice.edu/-~astubble/wep/

[WETE01] *"WAP Transport Layer End-to-end Security"*, Version 28, June 2001, http://www.wapforum.org

[WIM00] *"Wap Wireless Identification Module"*, Version 18, February 2000, http://www.wapforum.org

[WTLS00] *"WAP Wireless Transport Layer Securely"*, Version 18, February 2000, http://www.wapforum.org

[WPCO01] *"WAP Provisioning Content"* Version 24, July 2001, http://www.-wapforum.org

[WPKI01] *"WAP Wireless Public Key Infrastructure"* Version 24, April 2001, http://www.wapforum.org

[WSCL00] *"Wap Wireless Crypto Library"*, Version 05, November 1999, http://www.wapforum.org