Public key infrastructure in mobile systems

J. Dankers, T. Garefalakis, R. Schaffelhofer and T. Wright

In current mobile systems, some applications to some extent already use public key techniques and an underlying public key infrastructure (PKI) to provide end-to-end security, and such use is widely expected to grow. This paper provides an overview of the basic techniques and the entities that are involved in a PKI and describes how they are used in current mobile systems. The paper also highlights the envisaged use of PKI in future mobile systems and the challenges that brings, drawing on recent results of the European Union's SHAMAN project*.

1 Introduction

Mobile systems for communication have spread all over the world at a rapid pace, so that today most people are equipped with mobile devices. In the past, mobile phones were used mainly for telephony services; nowadays, however, other services, such as the delivery of information, are gaining increasing importance. With these different services other kinds of devices are being introduced; the latest mobile terminals provide an extended range of features and are able to access networks in alternative ways.

In this evolving situation, a public key infrastructure (PKI), such as that used in the fixed network to provide end-to-end security, has been seen as the enabling technology for providing the security required for the information, services and access means offered by mobile systems. This paper provides an overview of PKI, highlights its use in current mobile systems and discusses its possible use in future mobile systems.

2 PKI overview

All security mechanisms deployed today are based on either symmetric/secret key or asymmetric/public key cryptography, or sometimes a combination of the two. Here we will introduce the basic aspects of the secret key and public key techniques and compare their main characteristics; a detailed description of cryptographic mechanisms and their application can be found in Reference 1. The most important elements and procedures that constitute the public key infrastructure on which public key techniques rely will then be briefly explained. A comprehensive description of a public key infrastructure can be found in Reference 2.

Secret key techniques

Secret key techniques are based on the fact that the sender and recipient share a secret, which is used for various cryptographic operations, such as encryption and decryption of messages and the creation and verification of message authentication data. This secret key must be exchanged in a separate out-of-band procedure prior to the intended communication. For example, in the GSM (Global System for Mobile) cellular radio system the secret key that is shared between the mobile subscriber and their home operator is installed on a subscriber identity module (SIM) that is owned by the mobile subscriber and administered in the database of the subscriber's home operator. The need to exchange a secret key prior to the intended communication complicates the provision of security for transactions between entities that do not have a pre-established relationship.

Authentication is done by proving possession of the preshared secret key to each other. A widely used method for doing this is the challenge-and-response method. A challenge is sent to the challenged node, which then calculates a response using the challenge and the secret key as input for an algorithm. This response is sent to the challenger, which performs the same operation and compares the result with the received response.

The administration and management of secret keys, including their generation, distribution, renewal and tamper-resistant storage, can become very complicated, as the number of keys grows as the square of the number of entities: for each pair of entities a secret key has to be created and distributed, so that for a group of n entities communicating with each other $n(n-1)/2$ keys are required.

Because of the need for preshared secret keys, secret

*SHAMAN—Security for Heterogeneous Access in Mobile Applications and Networks—is project 2000-25350 of the European Commission's IST (Information Society Technologies) programme. It aims to provide security solutions for heterogeneous access and distributed dynamically-configurable terminals, and is investigating the use of PKI to support the selected security solutions.

key based solutions have low scalability. A major advantage of secret key techniques is that they are computationally very fast compared to public key techniques. This is the main reason why many protocols today still use secret key mechanisms for authentication.

Asymmetric key pairs

Unlike a front-door key, which allows its holder to lock or unlock the door with equal facility, the public key used in cryptography is asymmetric—knowing just the public key one can encrypt a message with relative ease but decrypt it, if at all, only with considerable difficulty. As a very simple example, most people can determine the square of 2, fewer people can calculate the inverse function, the square root of 2.

Besides being one-way functions, cryptographic public keys are also 'trapdoor' functions—the inverse function can be computed easily if a private key is known. Thus, if *Pub* is the recipient's public key (known also to the sender and used by him to encrypt the message), *Priv* the recipient's private key (unknown to the sender) and *M* the message:

$$\mathit{Priv}(\mathit{Pub}(M)) = M$$

Priv and *Pub* are easy to compute, but revealing *Pub* does not reveal an easy way of determining *Priv*.

If *Priv* and *Pub* are such that their operation can be reversed, i.e. if

$$\mathit{Pub}(\mathit{Priv}(M)) = M$$

they can be used to implement digital signatures for authentication purposes.

A commonly used asymmetric algorithm is the RSA (Rivest, Shamir, Adleman) algorithm. This relies on the difficulty of factorising a modulus into its two large prime-number factors (see the article on 'Modern data encryption' by Colin Boyd in the October 1993 issue of *Electronics & Communication Engineering Journal*, pp.271–278).

Public key techniques

Public key techniques are based on the use of asymmetric key pairs (see the Panel). Usually each user is in possession of just one key pair. One of the keys of the pair is made publicly available, while the other key of the pair is kept private. Because one of the keys is available publicly there is no need for a secure out-of-band key exchange, however there is a need for an infrastructure to distribute the public key authentically. Because there is no need for preshared secrets prior to a communication, public key techniques are ideal for supporting security between previously unknown parties.

Authentication is achieved by proving possession of the *private key*. One mechanism for doing this is a digital signature, which is generated with the private key and verified using the corresponding public key, i.e. by the public key bound to the entity generating the signature.

Public key techniques make it possible to establish secret *session keys* dynamically. A simplified procedure is for one end-entity to calculate a secret session key and send it encrypted with the public key of the entity with which it wants to initiate a session. That entity then obtains the secret key by decrypting the received information with its private key.

As the public key of a key pair is usually published in a directory, the overhead associated with distributing key material to communicating parties is reduced significantly in comparison with solutions based solely on secret key techniques. For a group of n entities communicating with each other, only n key pairs are required.

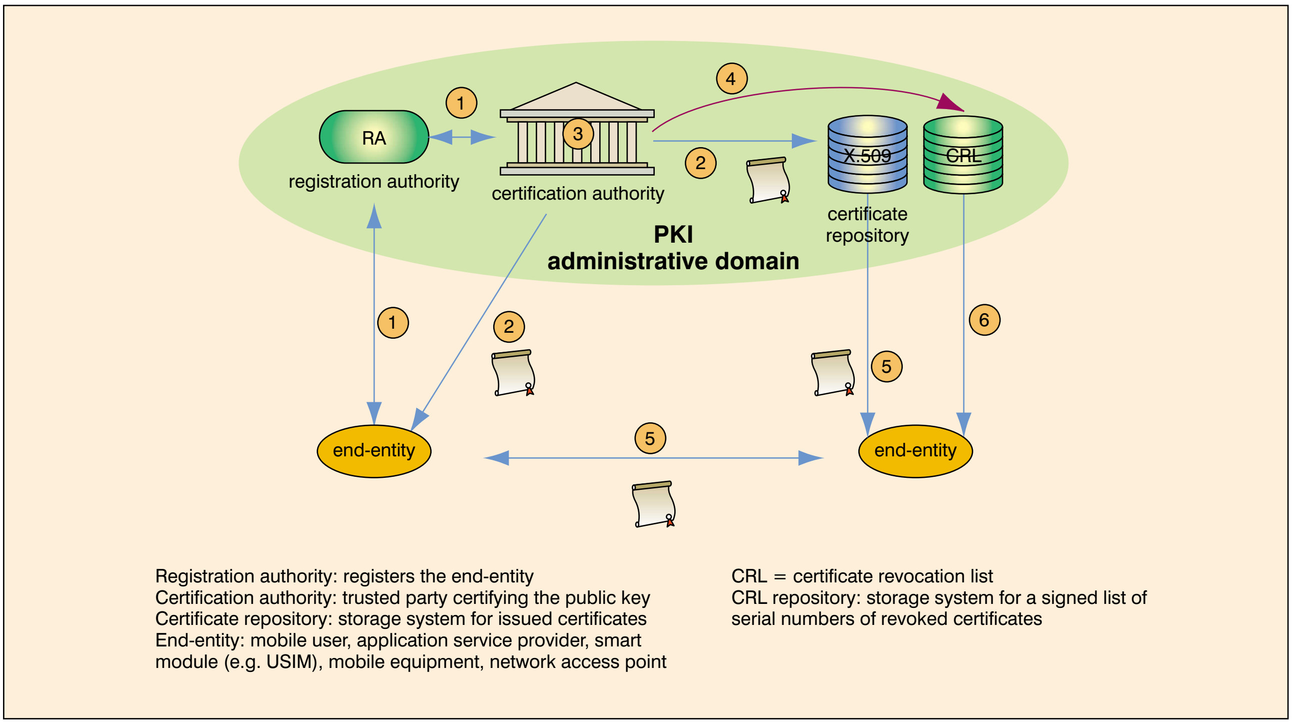
A drawback of public key techniques is that they are computationally very intensive, which makes them less suitable for devices of limited size and processing power, such as mobile phones.

The advantages of the public key techniques described above do not come free. They must be paid for by additional organisational measures and more sophisticated client logic. Furthermore, this additional overhead leads to extended user-interaction, for instance in handling certificates.

Certificates

A key element in the use of public key techniques is the certificate, a data structure that binds a public key to an entity in an authentic way. The data structure is signed by an independent third party, which has to be trusted by the entities using the certificates issued by this trusted party. The certificate guarantees that the public key is bound to the entity that is stated in the certificate. This assurance of correct binding and hence assurance that the certified party has been identified is the crucial requirement for public key techniques. Among other information the certificate contains:

- a certificate number, which is a unique number relative to the certificate issuer
- the name of the issuer of the certificate
- the name of the certificate owner
- the public key of the owner
- the algorithm used to calculate the signature
- a validity period, which specifies the period during which the certificate is valid. Limiting the validation period increases the security.



Registration authority: registers the end-entity
Certification authority: trusted party certifying the public key
Certificate repository: storage system for issued certificates
End-entity: mobile user, application service provider, smart module (e.g. USIM), mobile equipment, network access point

CRL = certificate revocation list
CRL repository: storage system for a signed list of serial numbers of revoked certificates

Fig. 1 Basic concept of a PKI

- extensions, which are optional. One of the extensions may, for example, refer to the policy or rules governing the issue of the certificate in an administrative domain. It is largely because of the use of extensions that different certificate formats exist profiled for specific uses. There may be different versions of each format. This can cause a lot of interoperability problems between different administrative domains that have their own policy and use of extensions.

PKI

The management of certificates during their lifecycle in an administrative domain requires an infrastructure—the public key infrastructure (PKI). This will now be explained with reference to Fig. 1, which represents an elementary PKI.

The core component of a PKI is the certification authority (CA). This authority is trusted by the end-entities in its administrative domain and is responsible for the status of the certificates it issues.

The main steps of certificate life cycle management are as follows:

(1) *Registration and key pair generation:* Before end-entities can use the services supported by the PKI they must register with it. During registration the identity of the end-entity is established and verified according to the policy of the administrative domain. The registration procedure is dependent on which entity generates the key pair:
  - If the certification authority generates the key pair then the private key is securely passed, via out-of-band mechanisms, to the registering end-entity.
  - If the end-entity generates the key pair, then the public key is passed to the certification authority, which checks whether the registering end-entity really possesses the corresponding private key (by means of proof of possession mechanisms).

  The certification authority may offload certain registration functions to a registration authority to enhance scalability and decrease operational cost. The issue of certificates and certificate revocation lists (CRLs) rests solely with the certification authority, however.

(2) *Certificate generation and distribution:* Once the end-entity has been verified and the key pair generated a certificate is issued and distributed to the end-entity and to a certificate repository.

(3) *Certificate expiration:* The certification authority has to renew certificates when they expire. It is informed of expired certificates by the end-entity.

(4) *Certificate revocation:* Another task of the certification authority is to revoke certificates, for example when the corresponding private key has been compromised.

(5) *Certificate retrieval:* End-entities retrieve certificates from the certificate repository or may exchange certificates, depending on the security protocol.

(6) *Certificate validation:* To validate certificates end-entities need to retrieve the certificate revocation lists from the CRL repository or may make use of on-line certificate status check protocols (OCSP).

Standardisation activities

Standardisation activities relevant to PKI are the definition and formalisation of PKI concepts, and the specification of certificate formats and processing rules. With regard to the specification of certificate formats a distinction has to be made between groups that define certificate formats and groups that profile defined certificate formats for specific environments and uses. Groups defining certificate formats include the ITU (the X.509 certificate), the Internet Engineering Taskforce's (IETF's) SPKI (Simple Public Key Infrastructure) and Open PGP (Pretty Good Privacy) working groups and the United Nations' EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport) group. Those *profiling* certificate formats, primarily X.509v3 certificates, include the ISO's TC68 committee, the IETF's PKIX (Public Key Infrastructure X.509), S/MIME (Secure Multi-purpose Internet Mail Exchange), IPsec (Internet Protocol Security) and TLS (Transport Layer Security) working groups, and the WAP (Wireless Application Protocol) Forum (now the Open Mobile Alliance). Certificate and CRL repository issues are connected to the X.500 and LDAP (Lightweight Directory Access Protocol) efforts.

The most important and widely accepted certificate format is the ITU-T's X.509 recommendation[3], which has also been published as ISO/IEC International Standard 9594-8[4].

One of the most important standards activities related to PKI today takes place in the IETF's PKIX working group[5]. PKIX deals with the definition of so-called 'X.509-profiles'. Since the original ITU-X.509 certificate standard leaves too many options for the contents of a certificate, an X.509 profile[6] defines the fields for Internet certificates exactly, in order to allow for better interoperability. PKIX also standardises a variety of other formats and protocols required to manage and operate a PKI.

Another series of widely used specifications, known as 'public key cryptography standards' (PKCS), has been issued by RSA Security Inc.[7] They deal with data structures and algorithm usage for basic applications of asymmetric cryptography.

PKI in fixed networks

Nowadays public key techniques and their supporting PKI are used in the fixed network by a number of security protocols* to support the establishment of the session keys required by the protocol to provide confidentiality and integrity, as well as the parties involved in initiating the session. Public key techniques are also used to support the provision of secure execution environments by signing downloadable code.

The killer application for public key techniques was the ability to provide end-to-end security between two unknown parties, first in closed environments and later on in the Internet.

3 PKI in current mobile systems

Characteristics of current mobile systems

In today's mobile communication systems access to services is granted in accordance with the subscription the user has with the service provider. Even when access is granted based on a prepaid method, there is a long-term contractual subscription between the service provider and the subscriber. The concept of pre-established security relations offers the possibility of integrating security with mechanisms based on symmetric cryptography. The providers of second or third generation mobile networks deliver smartcards with pre-installed symmetric keys, which are used to authenticate the mobile device and, in case of third generation networks, also to authenticate the access network. The authentication method is based on the trust relationship that exists between the access network provider and the service provider via a roaming agreement, and between the user and the service provider via the service subscription. The symmetric session key for confidentiality and protecting the integrity of data sent over the air is derived during the authentication phase.

Confidentiality and integrity over the whole path between two parties, i.e. end-to-end security, is not provided by these systems and therefore has to be provided at application level.

Public key mechanisms and their supporting PKI are not used in current mobile environments to provide network access security, because:

- a secret key that is preshared between the mobile node (e.g. a smart card) and the service provider can be installed relatively easily as part of the subscriber subscription procedure
- non-repudiation is not a stringent requirement for network access, and
- symmetric cryptography provides a much better performance than public key cryptography.

Recent usage of PKI in mobile environments

PKI is used in mobile environments by a number of security protocols and/or security schemes in the same way as it is used in the fixed network, as explained before. However the PKI is adapted to cope with the limitations of mobile environments. These protocols and/or security schemes, which are described in the next subsection, may be used by applications to provide end-to-end security. One such application is the Wireless Application Protocol (WAP), specified by the WAP Forum[8], which defines the standards by which Internet data moves to and from wireless devices.

WAP1.2 uses the Wireless Transport Layer Security (WTLS) protocol (see below) to protect the messages in the wireless network part, i.e. between the wireless device and the WAP gateway. The WAP gateway transforms the WAP1.2 stack to/from the IP stack†, relays the data between the wireless and the wired network and communicates with the Web server. Because WTLS and TLS are incompatible, content must be decrypted and re-

*Internet Key Exchange (IKE), Transport Layer Security (TLS), e-mail security using Secure Multi-purpose Internet Mail Exchange (S/MIME), and e-commerce applications like Secure Electronic Transactions (SET).

†WDP to/from TCP, WTLS to/from SSL/TLS, WTP/WSP to/from HTTP and binary WML to/from WML (XML)

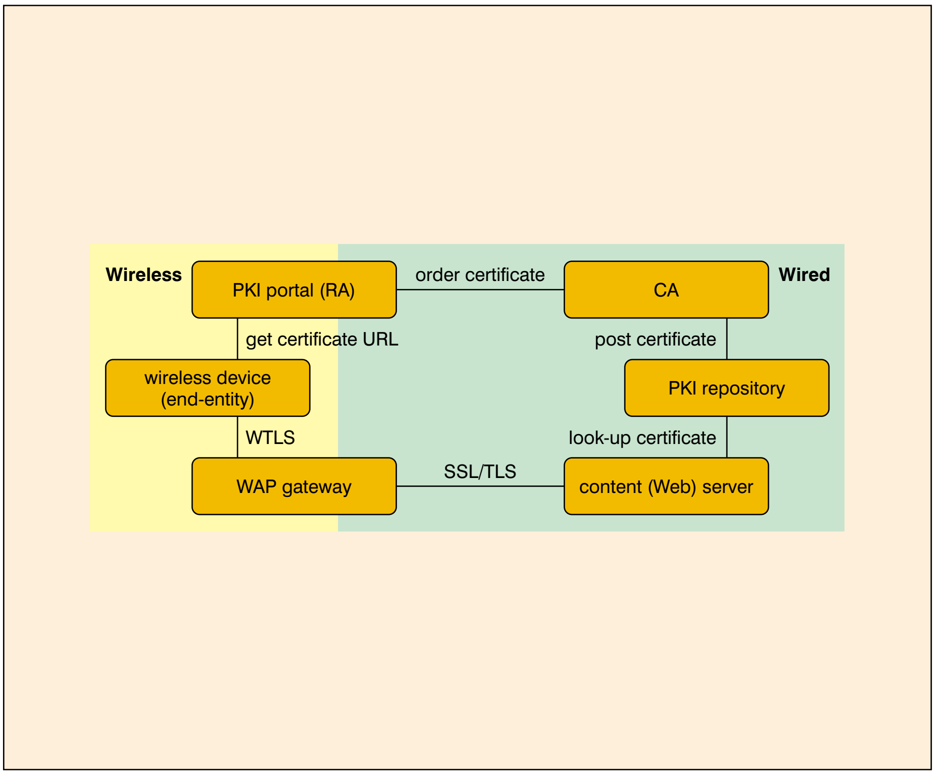


Fig. 2 Major components and operational flow of WPKI

encrypted as it passes through the WAP gateway. This means that both client and server rely on the 'chain of trust' built from the client to the WAP gateway and from the WAP gateway to the originating server. As decryption and encryption have to be done in the WAP gateway, WAP1.2 does not provide a mechanism for performing true end-to-end security between the client and the Web server.

WAP2.0 no longer requires the WAP gateway as such, as the mobile WAP 2.0 browser supports standard Internet network and transport protocols, i.e. TCP/IP. This means that TLS can be supported by the handset and can run end-to-end from the handset to a Web server supporting TLS without the need for decryption at the gateway.

The WAP Forum has defined a profile of TLS for the mobile environment. This simply specifies the cipher suites within the large number specified in TLS that must be supported by the handset and server in order to guarantee interoperability. In addition, as TLS uses the X.509v3 certificate format as profiled in RFC (Request for Comments) 2459 (X.509-PKIX), the WAP Forum has also specified a profile of the X.509v3 certificate format, the X.509-WAPcert. The X.509-WAPcert profile is optimised for handsets but should still allow most X.509-PKIX certificates in existing use for TLS to be processed by the handset.

As the standards by which Internet data can be moved to and from wireless devices are defined by WAP, it may be considered as the enabler of m-commerce. The PKI enables secure m-commerce transactions via wireless devices and the provision of non-repudiation, which is often a requirement for m-commerce. Security for m-commerce is described in Reference 9.

Protocols and security schemes

The following subsections provide an overview of the current and emerging use of public key cryptography and the underlying PKI in mobile telecommunications, as identified by working party WP3* of the SHAMAN project. A more detailed description is available in Reference 10.

WTLS

The WTLS (Wireless Transport Layer Security) protocol[11] is a PKI-enabled security protocol, designed for securing communications and transactions over wireless networks. It is used with the WAP transport protocols to provide security on the transport layer between the WAP client in the mobile device and the WAP server in the WAP gateway. The security services provided by the WTLS protocol are authentication, data confidentiality and data integrity. Applications are able selectively to enable or disable WTLS services depending on their security requirements and the characteristics of the underlying network (e.g. an application may disable the WTLS confidentiality service on networks that already provide this service at a lower layer). WTLS provides functionality similar to the Internet transport layer security systems TLS and SSL (Secure Sockets Layer), but it has been optimised for use over narrow-band communication channels and incorporates datagram support. WTLS is being implemented in all major microbrowsers and WAP servers.

The WTLS protocol consists of a record layer protocol and a handshake protocol. The handshake protocol is used to initiate a secure session. It allows the WAP client in the mobile equipment and the WAP server in the WAP gateway to agree on a protocol version, to select cryptographic algorithms, optionally to authenticate each other, and to generate a shared secret. The shared secret is used by the record layer protocol to provide data integrity and confidentiality. To provide the authentication service and to generate the shared secret the WTLS handshake protocol may use public key cryptographic techniques and a PKI designed for wireless environments (WPKI), as described below. Provision is also made for the use of uncertified public keys for the generation of the shared secret. However, this implies that neither the client nor the server is authenticated, which makes the system vulnerable to man-in-the-middle attacks.

*WP3 is the working party within SHAMAN investigating PKI issues to support the security mechanisms for future mobile systems specified by the working parties WP1 and WP2.

To guarantee optimum security, a tamper-resistant device stores the sensitive data (permanent private keys) and performs the security functionality (e.g. cryptographic operations) using this sensitive data. Certificates are integrity protected by the signature of the issuing party, so they can be exposed without danger.

WPKI

Just as WML (Wireless Markup Language) and WTLS are optimised versions of HTML (HyperText Markup Language) and TLS with respect to mobile environments, so WPKI (Wireless Public Key Infrastructure) is an optimised extension of a traditional public key infrastructure for the wireless environment. It is concerned with the security requirements imposed by WTLS on a PKI. Just as the most commonly used PKI standards for the wired networks are those of the IETF, so WPKI standards[12] are the most commonly used for wireless networks.

WPKI architecture for WAP1.2: As shown in Fig. 2 WPKI requires the same components as a traditional public key infrastructure: an end-entity application (EE), a registration authority (RA), a certification authority (CA) and a PKI repository. However, in WPKI, the end entities and the registration authority are implemented differently, and a new notion, referred to as the PKI-portal, is introduced. The end-entity in WPKI runs on the WAP device. It is responsible for the same functions as the end-entity in a traditional PKI.

The PKI portal can be a dual-networked system, like a WAP gateway. It typically functions as the registration authority and is responsible for translating requests from WAP clients to the registration and certifcation authorities. The PKI portal interoperates with WAP devices on the wireless network and with the certification authority on the wired network.

An extensive description of the WPKI is contained in Reference 12.

Wireless-specific adaptations in WPKI: The WPKI has optimised the PKI protocols, the certificate format and the cryptographic algorithms and keys with respect to mobile environments as follows:

- Traditional PKI service request encoding is optimised by standardising the message formats and thus avoiding the use of BER (Basic Encoding Rules) and DER (Distinguished Encoding Rules).
- New certificate formats, which are significantly reduced in size compared to a standard X.509 certificate, are defined for certificates that need to be sent over the air: the WTLS certificate format for server certificates and the X.509-WAPcert format (a X.509v3 format profiled for WAP) for client certificates. The sending of client certificates over the air is avoided as far as possible. Instead X.509-PKIX certificate formats, X.509v3 formats as profiled in RFC2459, are stored in a PKI directory from which the server can retrieve them. Another possibility is that the WAP client presents to the server the location (URL—Uniform Resource Locator) of its X.509-PKIX certificate, to reduce storage and transmission bandwidth. X.509-PKIX format certificates will not be transmitted over the air.
- Traditional RSA (Rivest, Shamir and Adleman) and DSA (Digital Signature Algorithm) based signature schemes are supported. ECC (Elliptic Curve Cryptography) based schemes are also recognised to be beneficial for their shorter key lengths and more efficient signature computation.

Status validation mechanisms, like certificate revocation lists and OCSP (Online Certificate Status Protocol) for the wireless client, have not yet been specified for WPKI. To provide a work-around for the lack of client-side status validation facilities, short-lived server certificates were introduced to obviate the need for a separate revocation check. The certification authority authenticates a server typically for one year and issues a new short-lived certificate, with a lifetime of typically 48 hours, every day of that year. For revocation the authority simply ceases issuing further short-lived certificates. The client requires sufficiently accurate time awareness.

SIM Application Toolkit

The SIM Application Toolkit (SAT), or simply 'SIM Toolkit', is a specification of subscriber identity module and terminal functionality that allows the SIM to take control of the terminal (the SIM is usually a 'slave' to the 'master' terminal) for certain functions. There is a specific SAT specification describing how encrypted SMS (Short Message Service) messages can be exchanged between the SIM and an external server, transparently to the terminal.

Symmetric methods are presently used to secure these messages, but these methods could be secured using public key methods as well. For example, a set of SMS messages to be sent from the server to the SIM could be encrypted using a symmetric key that was itself encrypted using a public key corresponding to a private key on the SIM. The SIM would then be the only entity that could decrypt the SMS message set. The SMS message set could also, or alternatively, contain a signature, generated with the private key of the server, that could be verified using the corresponding public key present on the SIM. Similarly, in the reverse direction the SIM could send out encrypted and signed SMS message sets.

However, it should be noted that the increased length of signatures using public key methods as compared to 'signatures' (e.g. message authentication codes) using symmetric methods is very significant in the SMS environment, as SMS messages are only 160 bytes long. A single SMS message would be required for the signature alone if RSA keys with 1024 bit moduli were being used.

In defence of the use of public key methods for SAT secure messaging it might be said that the use of public key techniques would allow easier key management than if only secret key methods were used and multiple parties wanted to communicate securely with the SIM. Against this though, it could be argued that the operator would not want an open model for secure communication with the SIM (such communication could have access to privileged functions of the SIM and the terminal) but would want to be fully in control of which parties could communicate with the SIM in this way.

The successor of the SIM-Application Toolkit for UMTS is the USIM-Application Toolkit (USAT). Apart from SMS, USSD (Unstructured Supplementary Service Data) may also be used as a bearer[9].

The mSign approach

The Mobile Electronic Signature Consortium[13] of companies related to mobile security has published the mSign protocol specification, which is intended to be a standard for interoperable mobile signatures. Message formats, message types as well as security levels are specified within this publication.

The major goal of the specification was to provide a framework for the introduction of mobile signatures for devices having different cryptographic capabilities. The new idea introduced in the mSign-protocol is for the signature to be generated in accordance with the crypto-capabilities of the client device. This means that a signature may be generated on the client device or by a trusted party on behalf of the original user. A powerful device may perform the cryptographic computations itself; alternatively a device may generate a signature that does not conform to a standard and which is then 'translated' by the mobile operator. A third possibility that is proposed is to delegate the generation of the signature completely. In summary, the mSign protocol may be used to provide digital signatures by means of PKI but it also offers alternative ways for signature generation at the client side.

IKE

IKE (Internet Key Exchange) is an authentication and key management protocol[14] used by IPsec to establish security associations (SAs) between parties communicating over the Internet Protocol. Security associations (essentially a shared secret key) can be established either manually or dynamically. IKE is a protocol for authenticated key exchange, thus providing dynamic key management.

The operation of the IKE protocol is divided into two phases, mainly for performance reasons. In phase 1, the communicating parties establish a common secret by means of an authenticated Diffie–Hellman key exchange mechanism. The authentication of the communicating peers can be done by means of RSA signed nonces, requiring a PKI; by RSA encrypted nonces, requiring only a public key, which may be established manually or dynamically (e.g. derived from a previous SA established by means of RSA signed nonces); and by preshared secret keys. After the successful completion of phase 1, a so-called IKE-SA is established. The IPsec security association, which is needed by IPsec protocols such as AH (Authentication Header) and ESP (Encapsulating Security Payload), is created in phase 2, and is based on the IKE-SA established in phase 1.

As explained in the previous paragraph, with the RSA-signed-nonces method certified public keys are used (during phase 1 of IKE). There is, therefore, a need for a public key infrastructure. IKE specifies the use of the PKIX profile of X.509v3 certificates described in References 6 and 15. As many implementations do not adhere completely to the specified format, an Internet draft[16] has been produced to achieve interoperability in the Internet market.

XML-related protocols

XML, the eXtensible Mark-up Language specified by the World Wide Web Consortium (W3C)[17], provides a universal format for structured documents and data on the Web. It has been extended recently to provide PKI-based functionality. A key management framework has been introduced and standards for encryption and generation of signatures have been written.

The philosophy of key/certificate management in XML is to source out some of the functionality that is usually done by the client to a trusted party.

The XML key management specification (XKMS) comprises two parts: the XML key information service specification (X-KISS) and the XML key registration service specification (X-KRSS).

X-KISS allows a client to delegate some or all of the tasks required to process XML signature elements to a trusted party. A key objective of the protocol design is to minimise the complexity of applications using an XML signature. So the application is relieved of the complexity and syntax of the underlying PKI and is independent of the specification this PKI is based upon. A good example for the concept of delegating functionality to a trusted party is the use of OCSP (Online Certificate Status Protocol) for requesting the current status of a certain certificate without the client having explicit OCSP functionality.

The X-KRSS specification defines a protocol for a Web service for the registration of public key information. The protocol provides authentication of the applicant and proof of possession (POP) of the private key in the case that the client has generated the key pair.

Besides the key management system an XML compliant syntax for signatures (XML Signature) as well as for encrypted content and information for the recipient to decrypt this content (XML Encryption) was specified. The latter specification defines, in addition, processes to encrypt/decrypt digital content. As the general XML syntax allows different parts of information to be put together and looked at as one document, these different parts may have to be handled in different ways. Therefore it must be possible to apply the security functions mentioned above to these parts separately, thereby making the syntax more complex. An example is a customer record where different people are allowed to view different parts of the encrypted document.

MExE

MExE (Mobile Execution Environment) provides a standardised execution environment for second and third generation mobile devices. The specification[18] was initially done by the European Telecommunications Standards Institute (ETSI) and then handed over to the 3rd

Generation Partnership Project (3GPP). A major goal was to provide a secure execution environment that is independent of the hardware device. MExE makes use of PKI to verify the origin and the integrity of a downloaded application and to grant/deny certain rights after checking.

This is done using a mechanism whereby the MExE-environment receives an executable signed by the providing party. The signature is verified and the application is matched to one of the following security domains: 'Manufacturer', 'Operator' or 'Third party'. Executables belonging to one of these domains are implicitly regarded as 'trusted'. MExE executables that are not signed or are signed with a signature key that cannot be verified by the mobile device are designated 'untrusted'. Untrusted executables operate within a 'sandbox', an environment with very limited access to the device's functionality and services. The owner of the device has the possibility to specify a policy on the rights that are granted to an application belonging to a certain security domain.

4 PKI in future mobile systems

Characteristics of future mobile systems

It is commonly believed that mobile systems beyond the third generation will be characterised by a variety of wireless access networks connected to an all-IP based core network, enabling global roaming using the radio technique best suited to support the requested service, and by terminals that may consist of several different components forming personal-area networks (PANs).

Security provision for these future mobile systems will require extensions to the symmetric-cryptography-based security methods available for third generation mobile systems. For various areas, discussed in the next subsection, public key based security methods and the underlying PKI offer the means of providing the required security.

Although PKI is a mature concept for providing security in homogeneous environments, such as enterprise networks, a number of challenges have to be overcome to use it in these future mobile systems, which are characterised by heterogeneity. These challenges are discussed below..

Areas to use PKI in future mobile systems

Access to mobile networks

Access to future mobile networks may be subscription based as in today's public mobile communications, but access may also be granted on the basis of online payment methods, such as credit cards.

If access is granted by online payment the user does not have a subscription with the provider of the requested service, which means that there is no pre-established relationship between the two. A public key based mechanism for authenticating the access network must be used. Although the public key mechanism is not used for network access security in current mobile systems, it is a candidate for future mobile systems as it avoids involvement of the home network. Public key based authentication and key agreement protocols use a certificate-based trust infrastructure. The certificates are issued by an entity trusted by the mobile node and the access network. The common trust entity may, for example, be a third party or the service provider of the mobile node. The SHAMAN project is investigating the suitability of the following public key based authentication and key agreement protocols for network access security: IKEv2[19] and JFK[20] (both successors of IKE), and SRP[21].

Communication between mobile core networks

To enable global mobile communication for the user there is a need for communication between the mobile core networks of different operators. The use of different networks to support global roaming is based on roaming agreements between operators. Security for communication between mobile core networks is not included in the present specifications of public mobile systems such as GSM. When the GSM architecture was introduced in the early nineties only a few providers were able to offer services at all and these providers trusted each other. Nowadays it is far easier for companies to provide mobile services, so that implicit trust between all the providers can no longer be assumed. PKI may be the appropriate instrument to secure communications between these core networks.

Intra-PAN communication

It is becoming quite common for users to possess several mobile devices (e.g. mobile phone, personal digital assistant, notebook computer, earpiece) that are able to communicate over a wireless interface. Roughly speaking, all the devices of a single user that are physically close define a personal-area network (PAN). The SHAMAN project document D03 provides a rigorous definition of this notion. As radio communications are, in general, easy to intercept and/or change, communications between devices in a PAN should be secured, i.e. source authentication, data integrity and data privacy should be provided. As the characteristics of the PAN are quite different from those of a large-scale communications network (for example there are only a few devices and these have limited capabilities and are owned by a single person), some interesting problems arise. Some of these problems, which are being discussed in the SHAMAN project, are considered further below.

Inter-PAN communication

Communication between PAN networks is quite different to interactions between core networks. As PANs are local structures and are managed internally, so that the role of the certifying party is probably within the PAN, new requirements arise. The structure of communicating PANs can be different, too. On the one hand there might be a master component in each PAN acting as a gateway; alternatively every device of one PAN might be able to communicate with every device in the other PAN. Of course, the first alternative is more complex to handle internally, whereas the second one has to deal with trust hierarchies and is therefore more complex between the

PAN networks. The most important issue to solve in this context is how the two PANs can establish an authentic initial connection. This means that some sort of cross-certification process has to be run beforehand. This mechanism is certainly easier for the first architecture described above, as a trust relationship only has to be established between two dedicated devices; in the second approach trust hierarchies have to be 'linked'.

Challenging issues

Managing the complexity of a PKI for limited devices

Mobile devices have limited processor capacity and memory storage: a trade-off between power, battery life and computing ability must usually be made. Therefore, we should start from the assumption that resources are scarce. Public key techniques and a PKI, however, demand high processor capacity and memory storage.

A high processing capacity is required for performing public key operations and for the construction and validation of certificate chains. The processing of certificate chains may be a very complicated and time-demanding operation, dependent on the length of the certificate chain and the possible inclusion of relations using cross-certification. Cross-certification is required when users from different PKIs are to be able to trust each other's certificates. Also the generation of a shared secret session key consumes processor resources.

There is also a high demand for storage capacity to store the certificates and certificate revocation lists. These are stored/cached for performance reasons and to save bandwidth on the wireless link.

One approach to overcoming the problem of limited devices is to outsource some activities to a server. Examples of protocols that do this are the Delegated Path Validation (DPV) protocol, which allows delegating all path validation to an OCSP server, and the Delegated Path Discovery (DPD) protocol, which allows delegating path construction to an OCSP server. Another example is the Simple Certificate Validation Protocol (SCVP), which allows outsourcing certificate handling to a server.

Managing the complexity of a PKI for limited bandwidth

Another challenge is managing the complexity of a PKI for limited bandwidth. Bandwidth constraints may arise due to the limited channel capacity between the security module (SM) and the module device hosting the security module. The radio link will probably not be a problem in the future: fourth generation mobile networks will offer broadband capacities that will allow fast and effective processing of PKI processes. The only problem that could exceptionally arise is with the download of large certificate revocation lists, especially if no caching mechanism is available on the mobile device and the operation is time-critical.

There is a high demand for bandwidth, mainly due to the construction and validation of certificate chains. This is due to the distribution of certificates and certificate revocation lists. Possible solutions may be the use of delta revocation lists and the delegation of activities to a network server.

Interoperability issues

There are several interoperability issues if a PKI is to be used in future mobile systems. These need to be investigated in detail and finding appropriate solutions is a major challenge.

One of the interoperability issues is the existence of different certificate formats, which are tailored to the environments in which they are used. WTLS, for example, which is used in a wireless environment, uses certificates that have a limited number of parameters. The most widely used format is the ITU format X.509, which is profiled by the IETF for use on the Internet. Several proprietary certificate formats also exist.

Even if the same format is used, interoperability problems may arise due to the certificate extensions that are defined. These extensions may have standardised and proprietary values. When an entity receives a certificate with an extension marked as critical and does not understand the extension, the certificate is rejected, and this complicates the security functions.

Interoperability can be increased by:

- specifying certificates with as few parameters as possible
- restricting the use of extensions and the use of the criticality flag.

Even when the certificate format conforms to the standard, the applications using the certificates may be incompatible, e.g. due to implementation errors.

Another source of interoperability problems may be the use of directory services based on different protocols. Different PKI authorities may apply different policies when issuing certificates. This means that certificates issued in one security domain may not be acceptable in another security domain having, for example, more restrictive policies.

The problem becomes even worse if some major suppliers of a certification authority use non-standard certificate features and/or do not follow standardised profiles.

Organisational issues

There are a number of different scenarios for the generation of the key pairs to be used in a public key infrastructure. Generation of the key pair may be distributed among the mobile devices or smart cards or may be centralised on the manufacturer of the devices or the owner of the smart cards. Another possibility is the central generation of a key pair at the request of a mobile user. In each case it must be considered whether a cryptographically good key can be assured, whether the public key can be certified in a secure way, and whether lawful interception has to be supported.

Another issue for consideration is the storage of the private key. Possible locations are the security module issued by the network operator (e.g. the SIM in GSM), an additional security module issued by a third-party service provider, or a file on a device (a software-certificate).

To develop a PKI for future mobile systems, a trust model needs to be defined. This may be done by assigning functions or roles to the parties involved in the PKI and by identifying the required trust relations between them. The role of a certification authority, the basic element of a PKI, could be taken by the home network provider, the access network provider or by a trusted third party.

A PKI may be more complex when the access network provider takes the role of certification authority, especially when the access network providers do not co-operate. The issue of certificates by the home network provider gives the subscriber greater identity privacy from the access network provider.

In defining the trust model, the trust relations should be minimised, as this will increase the level of security.

Another challenge is to minimise the amount of user interaction, as this is an inconvenience for users who will not usually be acquainted with the security procedures.

Dedicated PKI solutions

As described earlier, future mobile terminals may consist of components in physical proximity to each other and forming a personal area network. The components are interconnected with local communication links such as short-range wireless connections, e.g. Bluetooth. To provide secure communication between the components a new concept of PKI will be introduced for user-friendly and fast session-key agreement between the components. This concept is called the 'personal PKI', the basic idea of which is that a master component functions as the certification authority for the components within the PAN. This shows that dedicated PKI solutions are being developed for future mobile networks.

Personal PKI

Symmetric cryptographic techniques can in principle provide security for the internal communications of a PAN. Public key techniques, however, offer certain advantages that make them preferable. For instance, they provide the devices with digital signature capabilities. Furthermore, if symmetric techniques are employed, then any two devices have to agree on a common secret key before they can communicate securely. This would imply several initialisations (imprinting devices with secret keys) for each device. In this subsection, we consider some challenging problems regarding what we call the Personal PKI, i.e. the public key infrastructure employed in the PAN. The assumption is that the PAN contains at least one device that is capable of functioning as a certification authority and is known to every device in the PAN.

Initialisation: Before a device can communicate securely, it needs to create a key pair and communicate the public part authentically to the certification authority. In return, it receives a public key certificate signed by the authority. One possible way to initialise the devices that is being considered within the SHAMAN project is by imprinting. This is based on weak password-authenticated data exchange. Roughly speaking, the device and the certification authority exchange information over the insecure channel. Subsequently the integrity of the received data is checked by means of a short message authentication code (MAC) that is computed using the weak password and is readily verified by the user. The MAC is never transmitted, but rather displayed, so it cannot be intercepted, and thus the password is secure against offline dictionary attacks. Challenging issues arise when the device under initialisation is very limited, e.g. has no display or no numerical keypad.

Multiple certification authorities: Personal-area networks are necessarily of a more *ad hoc* nature than fixed networks, as devices may enter and leave the network at any time. This causes complications when devices with certain functionalities are not present in the network. With respect to security, the device with the most important functionality is the certification authority. In order to avoid a single point of failure, when the CA is unavailable, one may consider the possibility of having several devices in the PAN that can potentially function as CAs. In this 'multiple CA' environment, several interesting problems arise. For instance, if the acting CA leaves the PAN, another 'secondary' CA takes over. This transition should be as transparent to the other PAN devices as possible. Therefore, the secondary CA should keep a state practically indistinguishable from the primary CA, raising synchronisation issues. Furthermore, an issue usually not present in conventional PKIs is that of the security of the CA. Namely, a CA may become unavailable because it has been compromised (e.g. stolen). In this case, more drastic measures have to be taken, as the signature of the compromised CA is no longer to be trusted. Those issues are being considered within the SHAMAN project.

5 Conclusions

Nowadays PKI, tailored for wireless environments, is used by a number of security protocols to enable end-to-end security for services and applications, such as WAP and m-commerce in current mobile systems. These systems do not use PKI for securing network access, because there always exists a pre-established relation between the service provider and the mobile subscriber, allowing the use of secret-based methods, which are more efficient.

Future concepts for mobile devices accessing networks may be based on spontaneous networking, i.e. gaining access to a network without a prior relationship to the network provider. This means that the communicating parties have to provide credentials for authentication without knowing each other from prior sessions. In this case authentication must be based on certificates and a common trusted third party. A PKI is needed for certificate management through their lifecycle. For the same reason, securing the communication between personal-area network components requires a local PKI. However, enhancements to current PKI procedures, being defined by the SHAMAN project, are needed before PKI can be used in future mobile networks.

Acknowledgment

The work described in this paper has been supported by the European Commission through the IST Programme under Contract IST-2000-25350. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at their sole risk and liability.

References

1 MENEZES, A., VAN OORSCHOT, P., and VANSTONE, S.: 'Handbook of applied cryptography' (CRC Press, 1996). See also http://www.cacr.math.uwaterloo.ca/hac

2 ADAMS, C., and LLOYD, S.: 'Understanding public-key infrastructure: concepts, standards, and deployment considerations' (Macmillan Technical Publishing, 1999)

3 'Information technology—Open Systems Interconnection—The Directory: Public-key and attribute certificate frameworks'. ITU-T Recommendation X.509 (03/2000)

4 'Information technology—Open Systems Interconnection—The Directory: Public-key and attribute certificate frameworks'. ISO/IEC 9594-8: 2001 (4th edn.)

5 PKIX: see http://www.ietf.org/html.charters/pkix-charter.html

6 HOUSLEY, R., FORD, W., POLK, W., and SOLO, D.: 'Internet X.509 public key infrastructure—Certificate and certificate revocation list CRL profile'. Internet RFC 3280, April 2002

7 PKCS standards and PKI related information: see http://www.rsasecurity.com/rsalabs/pkcs/index.html

8 Wireless Application Protocol (WAP) Forum: see http://www.wapforum.com

9 SCHWIDERSKI-GROSCHE, S., and KNOSPE, H.: 'Secure m-commerce', *Electron. Commun. Eng. J.*, October 2002, **14,** (5), pp.228–238

10 SHAMAN, D04: 'Initial report on PKI requirements for heterogeneous roaming and distributed terminals'. See http://www.ist-shaman.org/

11 'Wireless Transport Layer Security (Version 6-apr-2001)'. See http://www1.wapforum.org/tech/documents/WAP-261-WTLS- 20010406-a.pdf

12 'WPKI WAP-217-WPKI (Version 24-apr-2001)'. See http://www1.wapforum.org/tech/documents/WAP-217-WPKI-20010424-a.pdf

13 mSign: see http://www.msign.org

14 'The Internet Key Exchange'. Internet RFC 2409, November 1998

15 'Internet X.509 public key infrastructure—Certificate and CRL profile, eighth draft, July 2001'. Internet RFC 2459 (revision)

16 Internet Draft: 'A PKIX profile for IKE'. See http://community.roxen.com/developers/idocs/drafts/draft-ietf-ipsec-pki-req-05.html

17 World Wide Web Consortium: see http://www.w3c.org

18 'Mobile Execution Environment (MExE); Functional Description Stage 2 (Release 4)'. 3GPP TS 23.057 v4.1.0 (2001-03)

19 'Proposal for the IKEv2 Protocol'. See http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ikeve2-02.txt (April 2002)

20 'Just Fast Keying (JFK)'. See http://www.ietf.org/internet-drafts/draft-ietf-ipsec-jfk-04.txt

21 'The SRP authentication and key exchange system' See http://www.ietf.org/rfc/rfc2945.txt

©IEE: 2002

Received 26th June 2002

Jozef Dankers obtained an Engineering Degree in Electronics at the Industrial Highschool Geel (Belgium) in 1978 and received a certificate for the post-academic program in Telecommunications and Telematics at the University of Antwerp (Belgium) in 1994. He commenced working for Siemens Atea in 1979 and has mainly been involved with the development of the signalling system #7 (SS7) protocols, for which he represented Siemens Atea in the ITU-T and ETSI standardisation bodies. Since 1999 he has been involved with research on security for mobile systems.

Address: ICDMNG, Siemens Atea, Atealaan 34, B-2200 Herentals, Belgium.
E-mail: jozef.dankers@siemens.atea.be

Ralf Schaffelhofer graduated in mathematics at Justus Liebig University Giessen (Germany) in 1998, where he specialised in cryptography. He started working as a security consultant for T-Systems-Nova in 1998, his main areas of work being public key infrastructures, authentication mechanisms, security concepts and application-security. Currently Mr. Schaffelhofer is involved in projects dealing with the corporate PKI of Deutsche Telecom, e-market-platforms, security for Internet providers and the SHAMAN project.

Address: T-Systems Nova GmbH, Technologiezentrum ES21d, D-64307 Darmstadt, Germany.
E-mail: ralf.schaffelhofer@t-systems.com

Tim Wright co-leads the Security Technologies Team within Vodafone Global R&D. This team is the centre of security expertise for the Vodafone group. Tim joined the Vodafone Fraud Control Team in 1995 and worked on fraud issues for two years. He has been working on security standards and research for the last four years. He has been working on 3G, WAP, Java and Internet security and e-commerce strategy. Tim has been an active member of ETSI SMG10, 3GPP S3 (the security group), 3GPP T2 sub-group 1 (MExE), the WAP Security Group, and lately, security-related aspects of the Java Community Process. He is the technical and standardisation lead for DRM within Vodafone.

Address: Vodafone Ltd., 2–4 London Road, Newbury, Berkshire, RG14 1JX, UK.
E-mail: timothy.wright@vodafone.com

Theo Garefalakis was until recently with the Information Security Group, Royal Holloway University of London, Egham, Surrey, TW20 0EX, UK.
E-mail: tgaref@netscape.net