3GPP TSG SA WG3 Security — S3#21 **S3-010622**

27 - 30 November, 2001

Sophia Antipolis, France

Source: Telenor, Nokia

Title: Using PKI to provide network domain security

Document for: Discussion

Agenda Item:

Basic PKI concepts

Introduction to PKI

The advantages of public key security compared to secret key are:

- Out-of-band distribution of keys can be avoided
- Better suited for large scale deployment
- Supports establishment of secure communication between entities that are previously unknown to each other

The problem related to security between strangers is unfortunately not completely solved migrating to public key systems. Also in a public key setting it is far from obvious that a public key claimed to belong to a certain entity really does so. There is a need for an "introducer" that vouches for the binding between a public key and the identity of its owner. Such a guarantee is provided by a digital certificate. The management of digital certificates through its whole lifecycle, from initialisation through utilisation to cancellation, is what public key infrastructure – PKI, is all about.

PKI services

There is no such thing as a comprehensive or authorized list of PKI services. In literature one can find almost all kinds of security services named as PKI-services. For our purpose it will be more fruitful to narrow the list. It could provide a good start to distinguish them from the security services that is ultimate from the users perspective, namely *authentication*, *integrity* and *confidentiality*. In this context we would also prefer to regard *authorization/access control* and *non-repudiation* as belonging to this category. We suggest to regard PKI services as services *supporting* these primary security services mentioned above in a context of public key cryptography. The following table provides a suggestion for some useful PKI services (although by no means exhaustive):

Certificate issuing	Certificate validation	Certificate revocation
Key generation	Key backup	Key recovery
Secure time stamping	Cross-certification	Privilege management

Table 1 Some important PKI services

The granularity of the service definitions can always be questioned. As an example we here include several distinct steps in the handling of certificate requests in the term *certificate issuing*. It will greatly vary from application to application how comprehensive a set of services that is needed. (E.g. in applications where big transactions of money takes place, services supporting confidentiality and non-repudiation would be requisite and where sensitive medical data are transferred, services to support integrity and authorization would be desirable.) The subset of services needed in UMTS network domain security might be less than the services in the table above. Key pairs can be generated outside the PKI. In that case key backup and key recovery are neither relevant. Time stamping service might have some justification in an inter-operator scenario. Depending of the chosen PKI architecture, cross-certification might be relevant. A minimum subset of services needed in UMTS network domain security would encompass

- key generation
- key distribution
- certificate issuance
- certificate validation

- certificate revocation.

PKI architecture

In order to provide the services some entities conducting certain roles has to be in place. A Certification Authority is an entity offering the basic certification services. Among the services are issuance, validation and revocation of certificates and possibly key

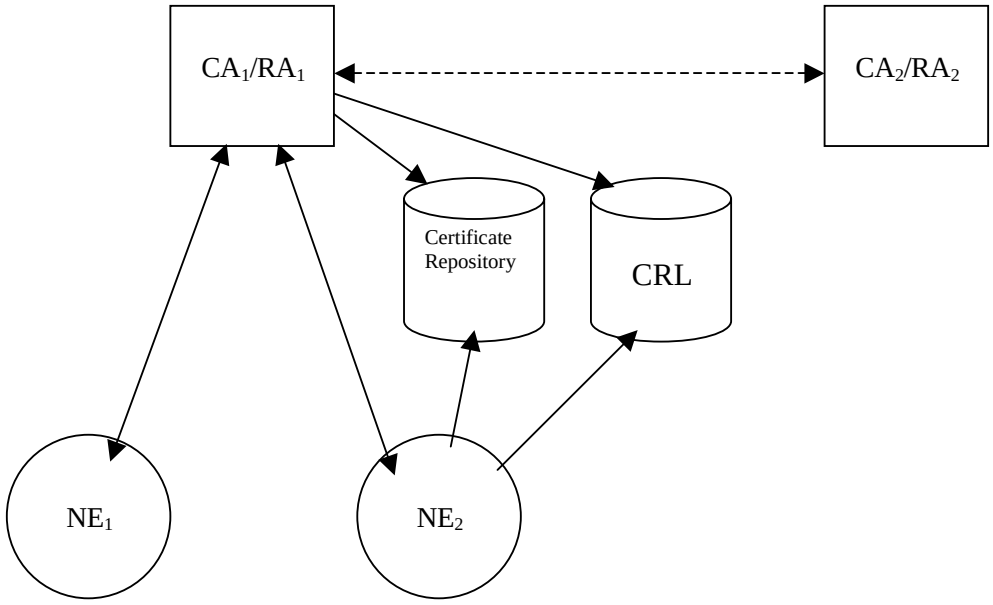

Figure 1: Elements of a PKI

generation. A Registration Authority can offload the CA with certain functions like

- establishing and confirming the identity of a new network element
- initiate the certification process on behalf of a network element
- generate keying material on behalf of a network element
- perform certain key/certificate life cycle management functions, such as to inititate a revocation request or a key recovery operation on behalf of a network element

Furthermore, there have to be publishing entities where certificates can be fetched and revocation lists can be inspected.
A simple PKI is illustrated is illustrated in figure above.

The roles of the PKI elements are:

Abbreviation	**Full name**	**Role**
NE	Network Element	Part of UMTS core network – not part of PKI
CA	Certification Authority	Responsible for issuing and revoking certificates. Possibly responsible for inter-CA

		relations
RA	Registration Authority	Responsible on behalf of CA for authenticating the NE on initial request for certification.
CRL	Certification Revocation List	Database maintained by CA where list of revoked certificates is published
	Certificate Repository	Database maintained by CA from which the digital certificates can be retrieved

Table 2 Different roles of PKI elements

Digital certificate

A digital certificate constitutes the means by which the relying user is assured that

- the integrity of the public key (and any other associated information) is sound
- the public key (and any other associated information) has been bound to the claimed owner in a trusted manner

Although several types of certificates exist, the X.509 is the most widely accepted standard. It has proven applicable in a wide variety of applications largely due to the flexibility in the current version 3. In X.509v3 just a smaller number of fields are always present, but it is possible to define extensions that is relevant for the application in question. These extension fields can be set as mandatory or optional. The set of fields used in a particular application of X.509v3 certificates and the mandatory/optional status of these fields constitutes a *profile*. While the X.509v3 standard is very open, a profile defines the limiting rules suitable for a particular use.

Trust relations

Two communication parties relying on a common CA can communicate securely. CAs can be organised in hierarchies, meaning that two communication parties can communicate securely also if the two CAs on which they trust is not the same but have a common root CA on top of the hierarchy. Two CAs can also be cross-certified, meaning that a digital certificate issued by one of them is acknowledged by the other and/or the other way around.

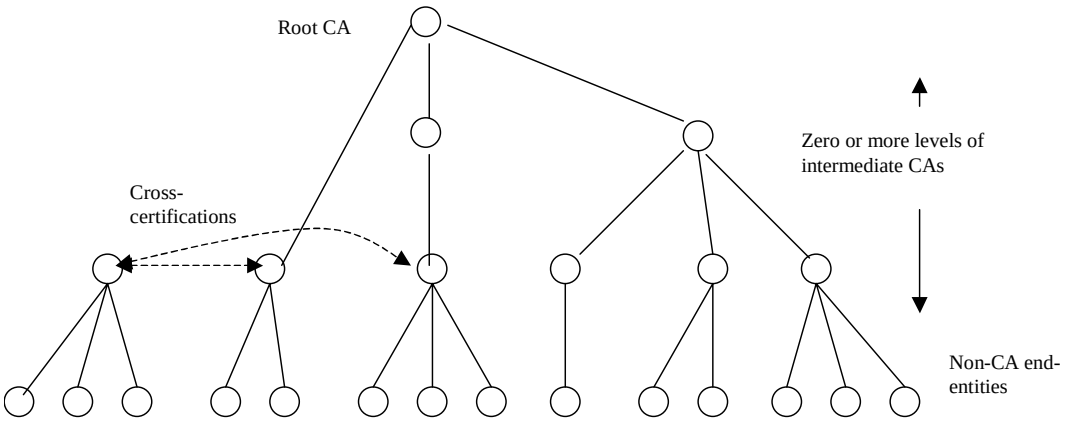

Figure 2: Example of trust hierarchy

Use of PKI to provide network-internal security

Scope of this document

A public key infrastructure in UMTS could be deployed for two main purposes. One is to support *end user applications* and the other is to support the need for what one could call *UMTS network internal trust management*. In the latter case three areas are of interest:

1. Network access security
2. Intra-operator network domain security
3. Inter-operator network domain security

At an early stage in the development of UTMS it was considered to use public keys in the process of authenticating the users (network access security). This idea was abandoned, mainly due to performance considerations and because it was a design goal to make it somewhat similar to the authentication in GSM. The UMTS AKA (Authentication and Key Agreement) thus ended up being quite similar to the one in GSM. The authentication in both systems are based on a secret key shared between the USIM and the home location register of the user.

With network domain security we here primarily mean *secure communications between network elements*. These network elements can belong to a single operator (intra-operator NDS) or they can belong to different operators (inter-operator NDS). In a broader definition the inter-operator scenario could be further extended to comprise *business relationships* concerning economic responsibilities (e.g. billing). The business aspects are considered to be out of the scope for this contribution, and so are the network access security as well as end user security.

Thus, in this document we will focus on the use of PKI to support

- intra-operator NDS
- inter-operator NDS

Motivation for PKI in UMTS core network

Scalability and key distribution

So far, in proposals for core network security in UMTS based on IPsec, agreements on keys and security associations are carried out on a bilateral basis between operators. As the number of operators and network elements increases, it would constitute a more *scalable* solution to replace individual bilateral relationships with an infrastructure supporting the use of public keys (PKI). In this way, secure communication can be achieved without having to distribute secret keys.

To illustrate the benefit of using such an asymmetric key system compared to a symmetric system: The number of keys needed in a symmetric system with *n* network elements communicating with each other is n*(n-1)/2, i.e. when n grows, the number of keys increases exponentially. In the public key case, the corresponding need for keys amounts to 2*n. So when n becomes large, the costs in

terms of key generation and distribution associated with the introduction of network element *n+1* are very dissimilar in the two
cases.

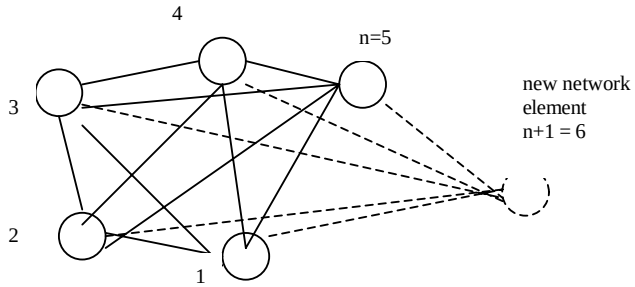

Figure 3: Symmetric case - adding a new network element

Symmetric case: n network elements requires n*(n-1)/2 different secret keys. Introduction of a new network element (n+1) requires establishing n new secret keys (dashed lines).

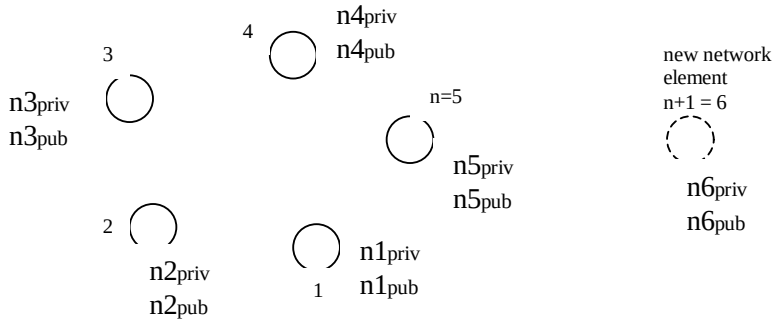

Figure 4: Asymmetric case - adding a new network element

Asymmetric case: each element has a private/public key pair. Thus n network elements requires 2*n keys. Introduction of a new network element (n+1) requires establishing only 2 new keys, i.e. a new private/public key pair for the new network element.

Dynamic key management

Authentication between network elements in UMTS Release 5 is so far planned to be based on pre-shared secrets. This is a somewhat rigid way to provide authentication. A properly designed PKI (based on digital certificates) will have more dynamic mechanisms to issue certificates for new network elements and to exclude certificates that are no longer valid. A certificate should for example be revoked if the corresponding private key is compromised or if a network element of some other reason should no longer be trusted.

More manageable trust

In TS 21.133 two requirements related to network domain security are:

- It shall be possible to secure the infrastructure between operators.
- There shall be a secure infrastructure between network operators, designed such that the need for HE trust in the SN for security functionality is minimized.

Both requirements address inter-operator security. The first requirement just states that in one way or other it should be possible to provide a secure infrastructure between operators. The second one deals with *trust relationships*.

In the first releases of UMTS the HE trust in the SN is fundamental. The AKA procedure heavily relies on the assumption that the HE can trust the SN and delegate the execution of the authentication to the SN. By introducing a commonly trusted third party the prerequisite for bilateral trust is reduced. The HE will then at least be able to authenticate the SN in a secure way. One could further consider whether a certificate for a SN network element should include information about its AKA implementation. In that case the certificate could provide the HE with confidence that the SN is trustworthy. But this would make the certificate rather application specific.

Public key shortcomings

It should be noted that secret key cryptography has its clear advantages when it comes to key lengths and computational load. Therefore public keys should not necessarily replace secret keys in all applications. The secret key regime is well suited for providing confidentiality and the public key system should primarily be used for authentication and secure transport of (symmetric) session keys.

Description

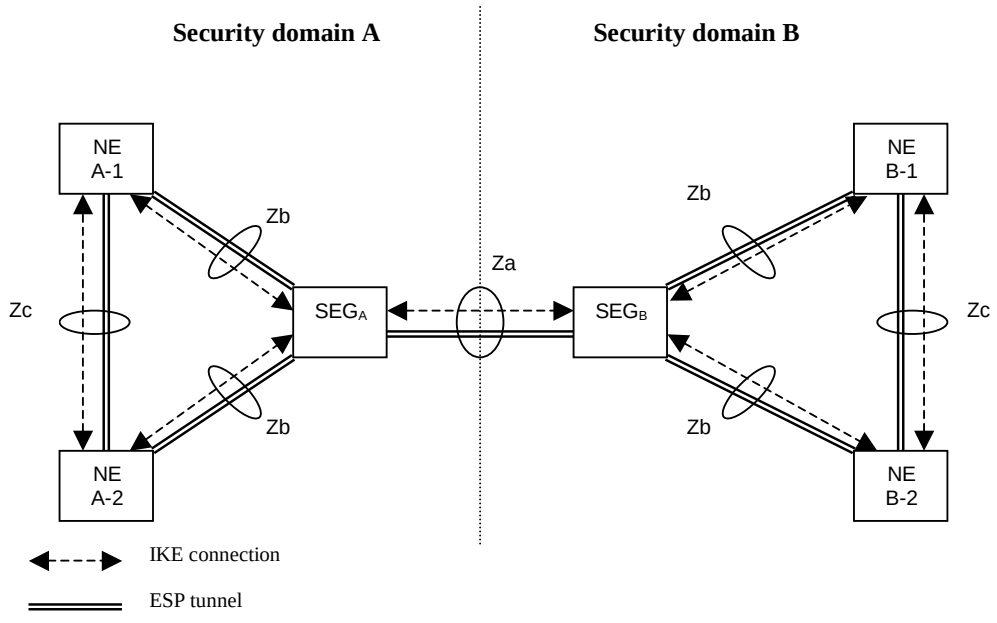

Figure 5: Secure communication between two security domains

Note that the current proposals in UMTS applies to securing the control plane, i.e. the signaling. In the current proposals, all secure communications are planned to be routed through a security gateway (SEG_A and SEG_B in the figure above).

A public key infrastructure could be introduced stepwise in the UMTS core network. The order of the steps would be decided by the needs.

Proposed phases

In each of the cases described below (Phase 1.a, 1.b and 1.c) the SEG_A needs the following certificates when negotiating IKE Phase 1 with SEG_B:

- $Cert(CA_A)_{CA_A}$, the trusted root certificate [$Cert(X)_Y$ = Public-key certificate of X, issued by Y]
- $Cert(CA_B)_{CA_A}$, the reverse cross-certificate
- $Cert(SEG_A)_{CA_A}$, the local device certificate

Naturally, SEG_B needs the similar certificates to be able to negotiate with SEG_A.

During IKE Phase 1, SEG_A receives $Cert(SEG_B)CA_B$ from SEG_B and then goes through the following validation path:

CA_A-> CA_B -> SEG_B

The corresponding validation path in SEG_B is:

CA_B-> CA_A -> SEG_A

Phase 1.a: One CA per operator, exchanging cross-certificates

- There is one CA per operator, exchanging cross-certificates with CAs of other operators. This would solve the inter-operator trust management.
- Every network element can get its exclusive certificate from the local CA. This would solve the intra-operator trust management.
- The public-key certificate of the local CA is stored in every network element in a secure manner.
- IKE is used for key exchange between SEGs (i.e. Za-interface), and the authentication is based on public keys instead of pre-shared secrets. In IKE Phase 1, SEG elements exchange their device certificates, signed by local CAs.
- IKE is used for key exchange between internal network elements and SEGs (i.e. Zb- and Zc-interfaces), and the authentication is based on public keys instead of pre-shared secrets. In IKE Phase 1, the elements exchange their device certificates, signed by local CA.
- The certificates needed in network elements are manually preloaded from the local CA and then cached locally.
- The certificates are revoked by manual methods.

Pros: Simple to implement and to start with.

Cons: 1) Certificate revocation is not scalable, and very slow. 2) The SEG has to preload all the needed certificates (especially cross-certificates) in advance, although they may never be needed. 3) The number of needed cross-certificates grows as in symmetric-key distribution problem; introducing a new CA potentially requires establishing N new cross-certificates, where N = number of existing CAs. Thus, the number of needed cross-certificates in a system of N distinct CAs is approximately N^2. However, introducing a new SEG inside one CA requires establishing only one (device) certificate.

Phase 1.b: as in Phase 1.a, except:

- CAs publish the certificates in repository, and SEGs retrieve the needed certificates from the repository and then cache them locally. The publishing and retrieving may be based on FTP or HTTP, as described in [RFC2585].
- The certificates are revoked by manual methods

Pros: The SEG can retrieve the needed certificates (especially the cross-certificates) from the repository on demand.

Cons: same as items 1) and 3) in Phase 1.a.

Phase 1.c: as in Phase 1.a, except:

- CAs publish the certificates in repository, and SEGs retrieve the needed certificates from the repository and then cache them locally. The publishing and retrieving may be based on FTP or HTTP, as described in [RFC2585].
- CAs publish the CRLs in repository, and SEGs retrieve the CRLs from the repository and then cache them locally. The publishing and retrieving may be based on FTP or HTTP, as described in [RFC2585]. Optionally, OCSP may be used for on-line query of certificate status.

Pros: Certificate revocation handling improved

Cons: Using CRLs may be a faster method than using a manual one, but it is still critized from a performance, scalability, and timeliness perspective (see [Adams99]).

Phase 2: Several levels of CAs

- The structure of CAs could migrate towards a hierarchy (i.e. one ore more levels of CAs above the operator's CA), possibly having one common inter-operator CA as the root.

The basic operations (of Phase 1.a,1.b and 1.c) are illustrated in the following figures.

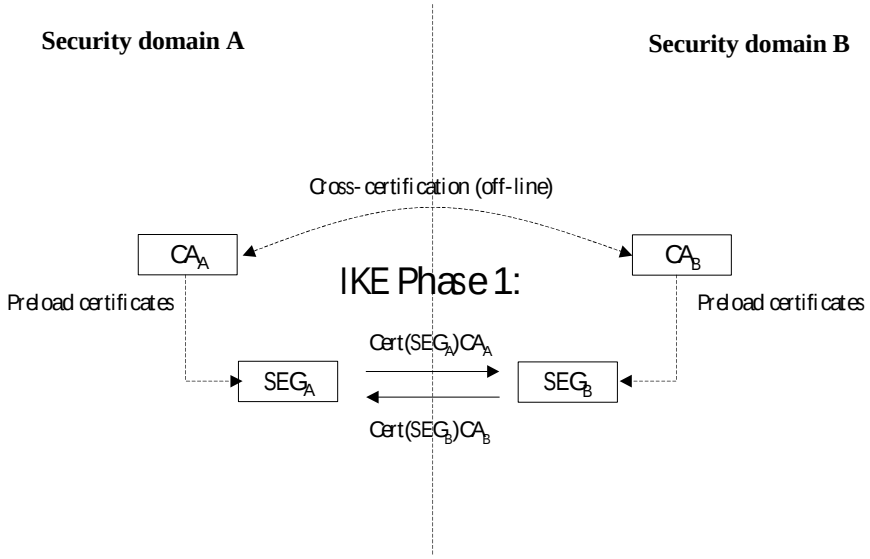

Figure 6 Certificate handling according to Phase 1.a

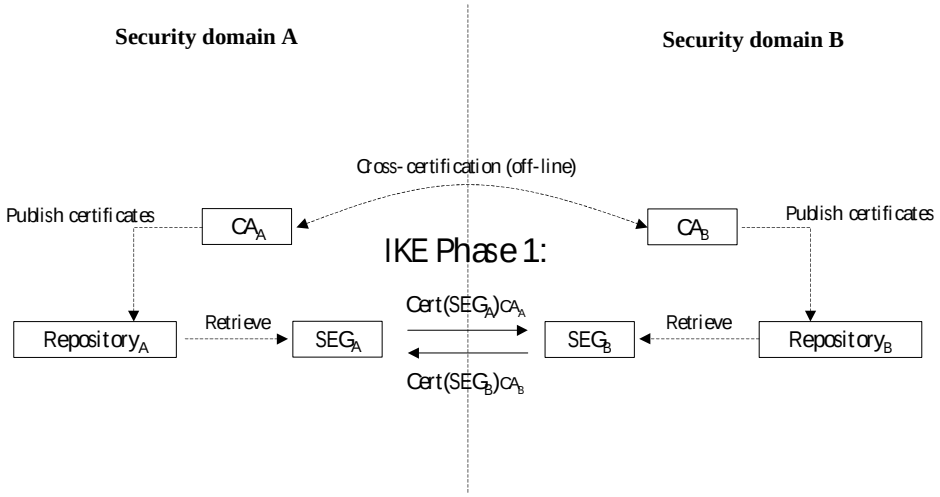

Figure 7 Certificate handling according to Phase 1.b

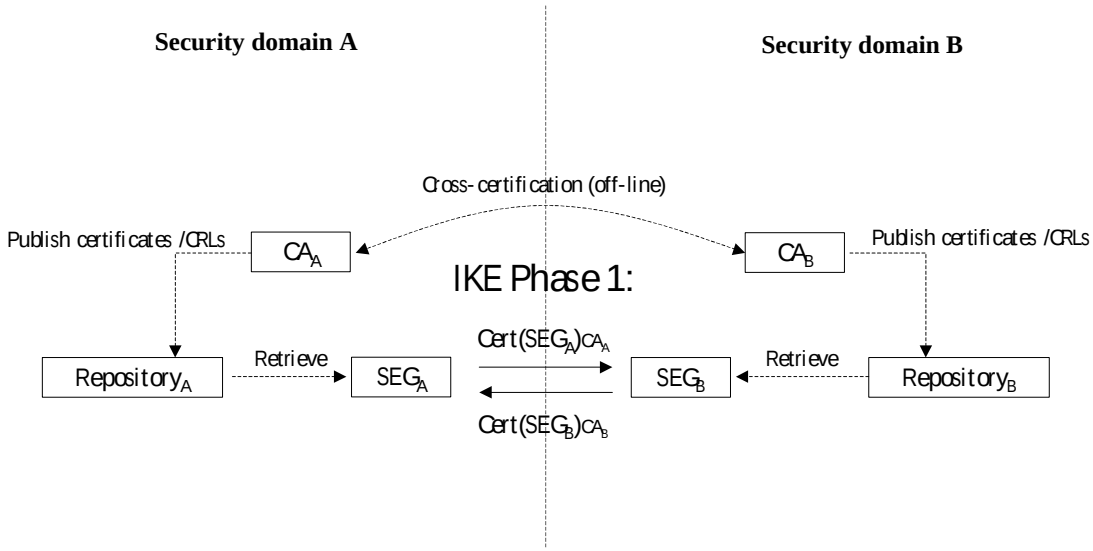

Figure 8 Certificate handling according to Phase 1.c

Deployment issues

Before we can propose any kind of public key infrastructure for UMTS, some important deployment issues must be considered first [Adams99]. These issues are listed in Table 3. The speculation is done from the security domain administrators' point of view.

PKI deployment issue	**Options**	**Speculation & recommended option (if any)**
Trust model	Two fundamental trust models may be applied in the enterprise (i.e. operator)	The distributed model (with cross-certification) is recommended. It is more

	context: Strict hierarchy of CAs based on superior/subordinate relationships OR Distributed mesh based on cross-certification.	flexible because it allows CAs to come and go with minimal disruption to the other CA domains.
Sourcing	In-sourcing OR Out-sourcing	It should be possible to either in-source or out-source the whole PKI or some parts of it. For example, outsourcing the CA from a third party, but retain the rest of the infrastructure in house.
Build vs. Buy	Build OR Buy	The suggested technology should be such that buying the technology is easier and faster than building it from the scratch. This aims at more faster deployment of the whole PKI concept.
Closed vs. Open environment	Closed environment OR Open environment	For the suggested infrastucture the intra-domain communications (i.e. closed environment) are of primary concern at the initial phase. However, the infrastructure should not prevent evolution towards inter-domain communications (i.e. open environment).
Certificate format	X.509, SPKI, PGP,…	The suggested solution should support at least X.509 certificates because of their vast support by the commercial PKI products. Other certificate types are for further study. For X.509 certificate requests and encoding, the appropriate standards (such as [RFC2511]) should be followed.
Standard vs. proprietary solutions	Standard solutions OR Proprietary solutions	The solution must be based on appropriate standards (found in other items of this table).
Interoperability	<The list of certificate and CRL profiles to be used> <The list of industry accepted standards to be met> <The list of PKI-enabled applications> <The list of policy issues (certificate policies) to be considered>	The X.509 certificate and CRL profiles need to be agreed starting from the initial phases (at least for cross-certificates). See [RFC2459].
On-line vs. Off-line operation	On-line operation OR Off-line operation	Off-line operation with cacheing should be preferred at the initial phases. At later phases, On-line operations may be considered.
Peripheral support	<The list of cryptographic hardware to be used in PKI-elements> <The list of APIs to be supported, like	Not relevant at initial phases

	PKCS#11>	
Facility requirements	<The list of supported physical and procedural safeguards for PKI components>	Not relevant at initial phases
Personnel Requirements	<The list of user groups to be defined: Security officers, operators, admins,…>	Not relevant at initial phases
Certificate revocation requirements	<The list of certificate revocation mechanisms to be supported> <The list of protocols to be supported> <The list of requirements concerning performance, timeliness, and scalability>	Initially, as simple as possible (manual revocation). At later phases, periodic checking of CRLs (Certificate Revocation List) may be used. Optionally, OCSP (Online Certificate Status Protocol) may replace or supplement the process of CRL checking. See [RFC2560].
End-entity roaming requirements	<The list of requirements for PKI to support a roaming user>	Not relevat at initial phases.
Key recovery requirements	Implement a key recovery facility: - as part of CA ? - as a separate component ? - not at all ?	Not needed at initial phases.
Repository requirements	<List of repository requirements related to certificate delivery, revocation, and policy issues>	For faster deployment of the whole PKI, it is recommended that Web servers or ftp-based servers are used as repositories at the initial phases. See [RFC2585]. The information stored in the certificate/CRL repository must be protected against unauthorized modification.
Disaster planning and recovery	<List of needed safeguard practises>	Not relevat at initial phases.
Security assurance	<List of criteria for evaluating the PKI security>	Not relevat at initial phases.
Risk mitigation	<List of requirements to be considered when selecting specific technology vendors>	Not relevat at initial phases.

Table 3 PKI deployment considerations

To be investigated further

The ultimate argument for introducing PKI in UMTS core network security will be its scalability properties. Therefore one has to consider thoroughly how fast the number of network elements that is sharing a security association is likely to grow. At first sight it seems probable to us that the number will be large enough to justify the public key approach.

Introduction of a PKI will probably slow down security procedures. Getting access to frequently updated certificate information (e.g. from CRLs) has the price of more latency. Therefore it has to be investigated whether PKI-introduced latency will be significant for UMTS network performance.

Control plane versus user plane: The current proposals in UMTS concerning NDS applies to the control plane.

References

[Adams99] Adams, C., Lloyd, S. *Understanding Public-Key Infrastructure.* Macmillan Technical Publishing, 1999.

[RFC2459] Housley, R. et al. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*. Internet Request for Comments (RFC) 2459. January, 1999.

[RFC2511] Myers, M. et al. *Internet X.509 Certificate Request Message Format*. Internet Request for Comments (RFC) 2511. March, 1999.

[RFC2560] Myers, M. et al. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. Internet Request for Comments (RFC) 2560. June, 1999.

[RFC2585] Housley, R. et al. *Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP*. Internet Request for Comments (RFC) 2585. May, 1999.